Quadratic Secret Sharing and Conditional Disclosure of Secrets*

Amos Beimel[1], Hussien Othman[2], and Naty Peter[3]

[1,2]Ben-Gurion University of the Negev, Be'er-Sheva, Israel
[3]Tel-Aviv University, Tel-Aviv, Israel
[1,2]{amos.beimel,hussien.othman}@gmail.com, [3]natypeter@mail.tau.ac.il

July 5, 2021

Abstract

There is a huge gap between the upper and lower bounds on the share size of secret-sharing schemes for arbitrary n-party access structures, and consistent with our current knowledge the optimal share size can be anywhere between polynomial in n and exponential in n. For linear secret-sharing schemes, we know that the share size for almost all n-party access structures must be exponential in n. Furthermore, most constructions of efficient secret-sharing schemes are linear. We would like to study larger classes of secret-sharing schemes with two goals. On one hand, we want to prove lower bounds for larger classes of secret-sharing schemes, possibly shedding some light on the share size of general secret-sharing schemes. On the other hand, we want to construct efficient secret-sharing schemes for access structures that do not have efficient linear secret-sharing schemes. Given this motivation, Paskin-Cherniavsky and Radune (ITC'20) defined and studied a new class of secret-sharing schemes in which the shares are generated by applying degree-d polynomials to the secret and some random field elements. The special case $d = 1$ corresponds to linear and multi-linear secret-sharing schemes.

We define and study two additional classes of polynomial secret-sharing schemes: (1) schemes in which for every authorized set the reconstruction of the secret is done using polynomials and (2) schemes in which both sharing and reconstruction are done by polynomials. For linear secret-sharing schemes, schemes with linear sharing and schemes with linear reconstruction are equivalent. We give evidence that for polynomial secret-sharing schemes, schemes with polynomial sharing are probably stronger than schemes with polynomial reconstruction. We also prove lower bounds on the share size for schemes with polynomial reconstruction. On the positive side, we provide constructions of secret-sharing schemes and conditional disclosure of secrets (CDS) protocols with quadratic sharing and reconstruction. We extend a construction of Liu et al. (CRYPTO'17) and construct optimal quadratic k-server CDS protocols for functions $f : [N]^k \to \{0, 1\}$ with message size $O(N^{(k-1)/3})$. We show how to transform our quadratic k-server CDS protocol to a robust CDS protocol, and use the robust CDS protocol to construct quadratic secret-sharing schemes for arbitrary access structures with share size $O(2^{0.705n})$; this is better than the best known share size of $O(2^{0.7576n})$ for linear secret-sharing schemes and worse than the best known share size of $O(2^{0.585n})$ for general secret-sharing schemes.

*The work of the authors was partially supported by Israel Science Foundation grant no. 152/17 and a grant from the Cyber Security Research Center at Ben-Gurion University. Part of this work was done while the first author was visiting Georgetwon University, supported by NSF grant no. 1565387, TWC: Large: Collaborative: Computing Over Distributed Sensitvie Data. The first author was also supported by ERC grant 742754 (project NTSC). The second author was also supported by a scholarship from the Israeli Council For Higher Education. The third author was also supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, and by the Rector's Office at Tel-Aviv University.

1 Introduction

A secret-sharing scheme is a cryptographic tool that enables a dealer holding a secret to share it among a set of parties such that only some predefined subsets of the parties (called authorized sets) can learn the secret and all the other subsets cannot get any information about the secret. The collection of authorized sets is called an access structure. These schemes were presented by Shamir [44], Blakley [20], and Ito, Saito, and Nishizeky [32] for secure storage. Nowadays, secret-sharing schemes are used in many cryptographic tasks, see, e.g., [13] for a list of applications. There are many constructions of secret-sharing schemes for specific families of access structures that have short shares, e.g., [32, 18, 22, 33, 19, 16, 45]. However, in the best known secret-sharing schemes for general n-party access structures, the share size is exponential in n [36, 5, 8], resulting in impractical secret-sharing schemes. In contrast, the best known lower bound on the share size of a party for some n-party access structure is $\Omega(n/\log n)$ [24, 23]. There is a huge gap between the upper bounds and lower bounds, and in spite of active research for more than 30 years, we lack understanding of the share size.

One of the directions to gain some understanding on the share size is to study sub-classes of secret-sharing schemes. Specifically, the class of *linear* secret-sharing schemes was studied in many papers, e.g., [22, 33, 15, 12, 11, 27, 28, 42]. In these schemes the sharing algorithm applies a linear mapping on the secret and some random field elements to generate the shares. For linear secret-sharing schemes there are strong lower bounds, i.e., in linear secret-sharing schemes almost all n-party access structures require shares of size at least $2^{0.5n-o(n)}$ [11] and there exists explicit n-party access structures require shares of size at least $2^{\Omega(n)}$ [43, 41, 42]. It is an important question to extend these lower bounds to other classes of secret-sharing schemes. Furthermore, we would like to construct efficient secret-sharing schemes (i.e., schemes with small share size) for a richer class of access structures than the access structures that have efficient linear secret-sharing schemes (which by [33] coincide with the access structures that have a small monotone span program). Currently, only few such constructions are known [16, 45].[1] Studying broader classes of secret-sharing schemes will hopefully result in efficient schemes for more access structures and will develop new techniques for constructing non-linear secret-sharing schemes. In a recent work, Paskin-Cherniavsky and Radune [39] perused these directions – they defined and studied a new class of secret-sharing schemes, called polynomial secret-sharing schemes, in which the sharing algorithm applies (low-degree) polynomials on the secret and some random field elements to generate the shares.

In this paper, we broaden the study of polynomial secret-sharing schemes and define and study two additional classes of polynomial secret-sharing schemes – (1) schemes in which the reconstruction algorithm, which computes the secret from the shares of parties of an authorized set, is done by polynomials, and (2) schemes in which both sharing and reconstruction algorithms are done by applying polynomials. We prove lower bounds for schemes of the first type (hence also for schemes of the second type). We then focus on *quadratic* secret-sharing schemes – schemes in which the sharing and/or reconstruction are done by polynomials of *degree-2*, and provide constructions of such schemes that are more efficient than linear secret-sharing schemes. Thus, we show that considering the wider class of polynomial secret-sharing schemes gives rise to better schemes than linear schemes.

As part of our results, we construct conditional disclosure of secrets (CDS) protocols, a primitive that was introduced in [30]. In a k-server CDS protocol for a Boolean function $f : [N]^k \to \{0,1\}$, there is a set of k servers that hold a secret s and have a common random string. In addition, each server Q_i holds a private input $x_i \in [N]$. Each server sends one message such that a referee, who knows the private inputs of

[1] In [45] they construct efficient secret-sharing schemes for access structures that correspond to languages that have statistical zero-knowledge proofs with log-space verifiers and simulators.

the servers but nothing more, learns the secret s if $f(x_1,\dots,x_k)=1$ and learns nothing otherwise. CDS protocols have been used recently in [36, 4, 5, 8] to construct the best known secret-sharing schemes for arbitrary access structures. Continuing this line of research, we construct quadratic k-server CDS protocols that are provably more efficient than linear CDS protocols. We use them to construct quadratic secret-sharing schemes for arbitrary access structures; these schemes are more efficient than the best known linear secret-sharing schemes.

1.1 Our Contributions and Techniques

Polynomial Sharing vs. Polynomial Reconstruction. Our conceptional contribution is the distinction between three types of polynomial secret-sharing schemes: schemes with polynomial sharing (defined in [39]), schemes with polynomial reconstruction, and schemes in which both sharing and reconstruction are done by polynomials. For linear secret-sharing schemes (in which the secret contains one field element) these notions are equivalent [33, 12]. In Section 9, we extend this equivalence to multi-linear secret-sharing schemes (i.e., schemes in which the secret can contain more than one filed element). In Section 3.1, we give evidence that such equivalence does not hold for polynomial secret-sharing schemes. We show that a small variation of a secret-sharing scheme of [16] for the quadratic non-residuosity modulo a prime access structure has an efficient secret-sharing scheme with degree-3 sharing.[2] Following [16], we conjecture that the quadratic non-residuosity modulo a prime is not in NC (the class of problems that have a sequence of circuits of polynomial size and poly-logarithmic depth). By our discussion in Remark 4.6, every sequence of access structures that has efficient secret-sharing schemes with polynomial reconstruction is in NC. Thus, under the conjecture about quadratic non-residuosity modulo a prime problem, we get the desired separation.

Lower bounds for Secret-Sharing Schemes with Degree-d Reconstruction. In Section 4, we show lower bounds for secret-sharing schemes with degree-d reconstruction. Using a result of [35], we show a lower bound of $\Omega(2^{n/(d+1)})$ for sharing one-bit secrets. We also show that every secret-sharing scheme with degree-d reconstruction and share size c can be converted to a multi-linear secret-sharing scheme with share size $O(c^d)$ (with the same domain of secrets). Using a lower bound on the share size of linear secret-sharing schemes over any finite field from [42], we obtain that there exists an explicit access structure such that for every finite field $\mathbb{F}$ it requires shares of size $2^{\Omega(n/d)}\log|\mathbb{F}|$ in every secret-sharing schemes over $\mathbb{F}$ with degree-d reconstruction. Furthermore, this transformation implies that every sequence of access structures that have efficient secret-sharing schemes with degree-d reconstruction for a constant d is in NC.

Quadratic Multi-Server Conditional Disclosure of Secrets Protocols. Liu et al. [37] constructed a quadratic two-server CDS protocol for any function $f:[N]^2\rightarrow\{0,1\}$ with message size $O(N^{1/3})$. In Section 5, we construct quadratic k-server CDS protocols with message size $O(N^{(k-1)/3})$. By our lower bounds from Section 4, this is the optimal message size for quadratic CDS protocols. Our construction uses the two-server CDS protocol of [37] (denoted $\mathcal{P}_{\mathrm{LVW}}$) to construct the k-server CDS protocol. Specifically, the k servers $Q_1,\dots,Q_k$ simulate the two servers in the CDS protocol $\mathcal{P}_{\mathrm{LVW}}$, where Q_1 simulates the first server in $\mathcal{P}_{\mathrm{LVW}}$ and servers $Q_2,\dots,Q_k$ simulate the second server in $\mathcal{P}_{\mathrm{LVW}}$.

Quadratic Multi-Server Robust Conditional Disclosure of Secrets Protocols. In a *t-robust* CDS protocol (denoted t-RCDS protocol), each server can send up to t messages for different inputs using the same

[2] We present it as a CDS protocol for the quadratic non-residuosity function. Using known equivalence, this implies a secret-sharing scheme, as in [16].

shared randomness such that the security is not violated if the value of the function f is 0 for all combinations of inputs. RCDS protocols were defined in [5] and were used to construct secret-sharing schemes for arbitrary access structures. Furthermore, Applebaum et al. [5] showed a general transformation from CDS protocol to RCDS protocol. Using their transformation as is, we get a quadratic RCDS protocol with message size $\tilde{O}(N^{(k-1)/3}t^{k-1})$, which is not useful for constructing improved secret-sharing schemes (compared to the best known linear secret-sharing schemes). In Section 6, we show that with a careful analysis that exploits the structure of our quadratic k-server CDS protocol, we can get an improved message size of $\tilde{O}(N^{(k-1)/3}t^{2(k-1)/3+1})$.

Quadratic Secret-Sharing Schemes for Arbitrary Access Structures and Almost All Access Structures. Applebaum et al. [5] and Applebaum and Nir [8] showed transformations from k-server RCDS protocols to secret-sharing schemes for arbitrary access structures. In [8], they achieved a general secret-sharing scheme for arbitrary access structures with share size $2^{0.585n+o(n)}$. In Section 7, we plug our quadratic k-server RCDS protocol in the transformation of [8] and get a quadratic secret-sharing scheme for arbitrary access structures with share size $2^{0.705+o(n)}$. This should be compared to the best known linear secret-sharing scheme for arbitrary access structures, given in [8], that has share size $2^{0.7576n+o(n)}$.

Beimel and Farràs [14] proved that for almost all access structures, there is a secret-sharing scheme for one-bit secrets with shares of size $2^{\tilde{O}(\sqrt{n})}$ and a linear secret-sharing scheme with shares of size $2^{n/2+o(n)}$. By a lower bound of [11] this share size is tight for linear secret-sharing schemes. In Section 7, we construct quadratic secret-sharing schemes for almost all access structures. Plugging our quadratic k-server CDS protocol in the construction of [14], we get that for almost all access structures there is a quadratic secret-sharing scheme for sharing one-bit secrets with shares of size $2^{n/3+o(n)}$. This proves a separation between quadratic secret-sharing schemes and linear secret-sharing schemes for almost all access structures.

Quadratic Two-Server Robust CDS Protocols. Motivated by the interesting application of robust CDS (RCDS) protocols for constructing secret-sharing schemes, we further investigate quadratic two-server RCDS protocols. In Section 8, we show how to transform the quadratic two-server CDS protocol of [37] to an RCDS protocol that is $N^{1/3}$-robust for one server while maintaining the $\tilde{O}(N^{1/3})$ message size. In comparison, the quadratic two-server $N^{1/3}$-RCDS protocol of Section 6 has message size $\tilde{O}(N^{8/9})$, however, it is robust for both servers. This transformation is non-blackbox, and uses polynomials of degree t to mask messages, where the masks of every messages of t inputs are uniformly distributed. Non-blackbox constructions of RCDS protocols may avoid limitations of constructing using CDS protocols as a blackbox.

1.2 Open Questions

Next, we mention a few open problems arising from this paper. We show non-trivial lower bounds for secret-sharing schemes with degree-d reconstruction. In [39], they ask the analogous question:

Question 1.1. *Prove lower bounds on the share size of secret-sharing schemes with degree-d sharing.*

We show a construction with degree-3 sharing that under a plausible conjecture does not have degree-3 reconstruction. We would like to prove such a separation without any assumptions.

Question 1.2. *Prove (unconditionally) that there is some access structure that has an efficient secret-sharing scheme with polynomial sharing but does not have an efficient secret-sharing scheme with polynomial reconstruction.*

Question 1.3. *Are there access structures that have an efficient secret-sharing scheme with polynomial reconstruction (of non-constant degree) but do not have an efficient secret-sharing scheme with polynomial sharing?*

We construct quadratic CDS protocols and secret-sharing schemes for arbitrary access structures. For quadratic CDS protocols we prove a matching lower bound on the message size. However, for larger values of d, the lower bound on the message size of degree-d CDS protocols is smaller.

Question 1.4. *Are there degree-d CDS protocols with smaller message size than the message size of quadratic CDS protocols? Are there degree-d secret-sharing schemes that are more efficient than quadratic secret-sharing schemes?*

Perhaps the most important question is to construct efficient secret-sharing schemes for a wide class of access structures.

Question 1.5. *Construct efficient degree-d secret-sharing schemes for a larger class of access structures than the access structures that have efficient linear secret-sharing schemes.*

1.3 Additional Related Works

Conditional Disclosure of Secrets (CDS) Protocols. Conditional disclosure of secrets (CDS) protocols were first defined by Gertner et al. [30]. The motivation for this definition was to construct symmetric private information retrieval protocols. CDS protocols were used in many cryptographic applications, such as attribute based encryption [29, 10, 46], priced oblivious transfer [1], and secret-sharing schemes [36, 17, 4, 5, 14, 8].

Liu et al. [37] showed two constructions of two-server CDS protocols. In their first construction, which is most relevant to our work, they constructed a quadratic two-server CDS protocol for any Boolean function $f : [N]^2 \to \{0, 1\}$ with message size $O(N^{1/3})$. In their second construction, which is non-polynomial, they constructed a two-server CDS protocol with message size $2^{O(\sqrt{\log N \log\log N})}$. Applebaum and Arkis [2] (improving on [3]) have shown that for long secrets, i.e., secrets of size $\Theta(2^{N^2})$, there is a two-server CDS protocol in which the message size is 3 times the size of the secret. There are also several constructions of multi-server CDS protocols. Liu et al. [38] constructed a k-server CDS protocol (for one-bit secrets) with message size $2^{\tilde{O}(\sqrt{k \log N})}$. Beimel and Peter [17] and Liu et al. [38] constructed a linear k-server CDS protocol (for one-bit secrets) with message size $O(N^{(k-1)/2})$; by [17], this bound is optimal (up to a factor of k). When we have long secrets, i.e., secrets of size $\Theta(2^{N^k})$, Applebaum and Arkis [2] showed that there is a k-server CDS protocol in which the message size is 4 times the size of the secret. Gay et al. [29] proved a lower bound of $\Omega(\log\log N)$ on the message size of two-server CDS protocols for some function and a lower bound of $\Omega(\sqrt{\log N})$ on the message size of linear two-server CDS protocols. Later, Applebaum et al. [3], Applebaum et al. [7], and Applebaum and Vasudevan [9] proved a lower bound of $\Omega(\log N)$ on the message size of two-server CDS protocols.

Polynomial Secret-Sharing Schemes. Paskin-Cherniavsky and Radune [39] presented the model of secret-sharing schemes with polynomial sharing, in which the sharing is a polynomial of low (constant) degree and the reconstruction can be any function. They showed limitations of various sub-classes of secret-sharing schemes with polynomial sharing. Specifically, they showed that the subclass of schemes for which the sharing is linear in the randomness (and the secret can be with any degree) is equivalent to multi-linear schemes up to a multiplicate factor of $O(n)$ in the share size. This implies that schemes in this subclass cannot significantly reduce the known share size of multi-linear schemes. In addition, they showed that the subclass of schemes over finite fields with odd characteristic such that the degree of the randomness in the sharing function is exactly 2 or 0 in any monomial of the polynomial can efficiently realize only access structures whose all minimal authorized sets are singletons. They also studied the randomness complexity of schemes with polynomial sharing. They showed an exponential upper bound on the randomness complexity (as a function of the share size). For linear and multi-linear schemes, we have a tight linear upper bound on the randomness complexity.

2 Preliminaries

In this section we define secret-sharing schemes, conditional disclosure of secrets protocols, and robust conditional disclosure of secrets protocols.

Notations. We say that two probability distributions $\mathcal{Y}_1, \mathcal{Y}_2$ over domain $\mathcal{X}$ are identical, and denote $\mathcal{Y}_1 \equiv \mathcal{Y}_2$, if $\mathcal{Y}_1(x) = \mathcal{Y}_2(x)$ for every $x \in \mathcal{X}$. We denote by $\binom{N}{[m]}$ the set of all subsets of N of size m. We say that $g(n) = \tilde{O}(f(n))$ if $g(n) = O(f(n) \log^c n)$ for some constant c, i.e., the $\tilde{O}$ notation ignores poly-logarithmic factors.

Operations in $\mathbb{F}_{2^d}$. Let d be an integer and consider addition and multiplication in $\mathbb{F}_{2^d}$. These operations can be implemented as operations in $\mathbb{F}_2$ with the same degree. Recall that an element in $\mathbb{F}_{2^d}$ can be represented as a polynomial of degree $d-1$ over $\mathbb{F}_2$. We represent it as d elements in $\mathbb{F}_2$. Let $R(\sigma) = \sum_{k=0}^{d-1} e_k \sigma^k + \sigma^d$ (where $e_0, \dots, e_{d-1} \in \{0,1\}$) be an irreducible polynomial over $\mathbb{F}_2$ that generates $\mathbb{F}_{2^d}$. Let $A(\sigma) = \sum_{k=0}^{d-1} a_k \sigma^k$ and $B(\sigma) = \sum_{k=0}^{d-1} b_k \sigma^k$ (where $a_0, \dots, a_{d-1}, b_0, \dots, b_{d-1} \in \{0,1\}$) be two elements in $\mathbb{F}_{2^d}$. Then, the sum of the two elements is represented by summing their coefficients in $\mathbb{F}_2$. Furthermore, the multiplication of two elements is done by multiplying the two polynomials and then reducing the result modulo $R(\sigma)$. Let $\sum_{k=0}^{2d-2} c_k \sigma^k$ (where $c_0, \dots, c_{2d-2} \in \{0,1\}$) be the product the two polynomials $A(\sigma)$ and $B(\sigma)$. Notice that c_k is a polynomial of degree 2 in $a_0, \dots, a_{d-1}, b_0, \dots, b_{d-1}$. Let $P_1, \dots, P_{2d-2}$ be the polynomials such that $P_k(\sigma) = \sigma^k \bmod R(\sigma)$, where $P_k(\sigma) = \sum_{j=0}^{d-1} P_{k,j} \cdot \sigma^j$ for every $1 \le k \le 2d-2$. Then,

$$(\sum_{k=0}^{2d-2} c_k \sigma^k) \bmod R(\sigma) = \sum_{k=0}^{2d-2} c_k P_k(\sigma) = \sum_{j=0}^{d-1} \sum_{k=0}^{2d-2} c_k P_{k,j} \sigma^j.$$

Therefore, the degree of the multiplication is the same as the degree of computing $c_0, \dots, c_{2d-2}$, i.e., the degree of multiplying the two elements in $\mathbb{F}_{2^d}$.

Secret-Sharing. We start by presenting the definition of secret-sharing schemes.

Definition 2.1 (Access Structures). *Let $P = \{P_1, \dots, P_n\}$ 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*.*

Definition 2.2 (Secret-Sharing Schemes). *A* secret-sharing scheme Π *with domain of secrets S is a mapping from $S \times R$, where R is some finite set called the set of random strings, to a set of n-tuples $S_1 \times S_2 \times \cdots \times S_n$, where S_j is called the* domain of shares *of party P_j. A dealer distributes a secret $s \in S$ according to Π by first sampling a random string $r \in R$ with uniform distribution, computing a vector of* shares $\Pi(s, r) =$

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

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

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

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

SECURITY. *Every unauthorized set cannot learn anything about the secret from its shares. Formally, for any set* $T = \{P_{i_1}, \ldots, P_{i_{|T|}}\} \notin \Gamma$*, every pair of secrets* $s, s' \in S$*, and every vector of shares* $(s_{i_1}, \ldots, s_{i_{|T|}}) \in S_{i_1} \times \cdots \times S_{i_{|T|}}$*, it holds that* $\Pi_T(s, r) \equiv \Pi_T(s', r)$*, where the probability distributions are over the choice of* r *from* R *with uniform distribution.*

Definition 2.3 (Threshold Secret-Sharing Schemes)**.** *Let* Π *be a secret-sharing scheme on a set of* n *parties* P*. We say that* Π *is a* t-out-of-n secret-sharing scheme *if it realizes the access structure* $\Gamma_{t,n} = \{A \subseteq P : |A| \geq t\}$*.*

Conditional Disclosure of Secrets. Next, we define k-server conditional disclosure of secrets (CDS) protocols, first presented in [30]. We consider a model where k servers[3] $Q_1, \ldots, Q_k$ hold a secret s and a common random string r; every server Q_i holds an input x_i for some k-input function f. In addition, there is a referee that holds $x_1, \ldots, x_k$ but, prior to the execution of the protocol, does not know s and r. In a CDS protocol for f, for every $i \in [k]$, server Q_i sends a single message to the referee, based on r, s, and x_i; the server does not see neither the inputs of the other servers nor their messages when computing its message. The requirements are that the referee can reconstruct the secret s if $f(x_1, \ldots, x_k) = 1$, and it cannot learn any information about the secret s if $f(x_1, \ldots, x_k) = 0$.

Definition 2.4 (Conditional Disclosure of Secrets Protocols)**.** *Let* $f : X_1 \times \cdots \times X_k \to \{0, 1\}$ *be a* k*-input function. A* k-server CDS protocol $\mathcal{P}$ *for* f*, with domain of secrets* S*, domain of common random strings* R*, and finite message domains* $M_1, \ldots, M_k$*, consists of* k *message computation functions* $\textsc{Enc}_1, \ldots, \textsc{Enc}_k$*, where* $\textsc{Enc}_i : X_i \times S \times R \to M_i$ *for every* $i \in [k]$*. For an input* $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$*, secret* $s \in S$*, and randomness* $r \in R$*, we let* $\textsc{Enc}(x, s, r) = (\textsc{Enc}_1(x_1, s, r), \ldots, \textsc{Enc}_k(x_k, s, r))$*. We say that a protocol* $\mathcal{P}$ *is a CDS protocol for* f *if it satisfies the following properties: (1) Correctness: There is a deterministic reconstruction function* $\textsc{Dec} : X_1 \times \cdots \times X_k \times M_1 \times \cdots \times M_k \to S$ *such that for every input* $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ *for which* $f(x_1, \ldots, x_k) = 1$*, every secret* $s \in S$*, and every common random string* $r \in R$*, it holds that* $\textsc{Dec}(x, \textsc{Enc}(x, s, r)) = s$*. (2) Security: For every input* $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ *for which* $f(x_1, \ldots, x_k) = 0$ *and every pair of secrets* $s, s' \in S$ *it holds that* $\textsc{Enc}(x, s, r) \equiv \textsc{Enc}(x, s', r)$*, where* r *is sampled uniformly from* R*.*

The message size *of a CDS protocol* $\mathcal{P}$ *is defined as the size of the largest message sent by the servers, i.e.,* $\max_{1 \leq i \leq k} \log |M_i|$*. In two-server CDS protocols, we sometimes refer to the servers as Alice and Bob (instead of* Q_1 *and* Q_2*, respectively).*

[3]For clarity of the presentation (especially when using CDS protocols to construct secret-sharing schemes) we denote the entities in a CDS protocol by servers and the entities in a secret-sharing scheme by parties.

Definition 2.5 (The Predicate INDEX_N^k)**.** *We define the k-input function $\mathrm{INDEX}_N^k : \{0,1\}^{N^{k-1}} \times [N]^{k-1} \to \{0,1\}$ where for every $D \in \{0,1\}^{N^{k-1}}$ (a $(k-1)$ dimensional array called the database) and every $(i_2, \dots, i_k) \in [N]^{k-1}$ (called the index), $\mathrm{INDEX}_N^k(D, i_2, \dots, i_k) = D_{i_2,\dots,i_k}$.*

Observation 2.6 ([29])**.** *If there is a k-server CDS protocol for INDEX_N^k with message size M, then for every $f : [N]^k \to \{0,1\}$ there is a k-server CDS protocol with message size M.*

We obtain the above CDS protocol for f in the following way: Server Q_1 constructs a database $D_{i_2,\dots,i_k} = f(x_1, i_2, \dots, i_k)$ for every $i_2, \dots, i_k \in [N]$ and servers $Q_2, \dots, Q_{k-1}$ treat their inputs $(x_2, \dots, x_k) \in [N]^{k-1}$ as the index, and execute the CDS protocol for $\mathrm{INDEX}_N^k(D, x_2, \dots, x_k) = f(x_1, x_2, \dots, x_k)$.

Robust Conditional Disclosure of Secrets. In the definition of CDS protocols (Definition 2.4), if a server sends messages for different inputs with the same randomness, then the security is not guaranteed and the referee can possibly learn information on the secret. In [5], the notion of robust CDS (RCDS) protocols was presented. In RCDS protocols, the security is guaranteed even if the referee receives messages of different inputs with the same randomness. Next we define the notion of t-RCDS protocols.

Definition 2.7 (Zero Sets)**.** *Let $f : X_1 \times X_2 \times \cdots \times X_k \to \{0,1\}$ be a k-input function. We say that a set of inputs $Z \subseteq X_1 \times X_2 \cdots \times X_k$ is a zero set of f if $f(x) = 0$ for every $x \in Z$. For sets $Z_1, \dots, Z_k$, we denote $\textsc{Enc}_i(Z_i, s, r) = (\textsc{Enc}_i(x_i, s, r))_{x_i \in Z_i}$ and*

$$\textsc{Enc}(Z_1 \times Z_2 \cdots \times Z_k, s, r) = (\textsc{Enc}_1(Z_1, s, r), \dots, \textsc{Enc}_k(Z_k, s, r)).$$

Definition 2.8 (t-RCDS Protocols)**.** *Let $\mathcal{P}$ be a k-server CDS protocol for a k-input function $f : X_1 \times X_2 \times \cdots \times X_k \to \{0,1\}$ and $Z = Z_1 \times Z_2 \times \cdots \times Z_k \subseteq X_1 \times X_2 \times \cdots \times X_k$ be a zero set of f. We say that $\mathcal{P}$ is robust for the set Z if for every pair of secrets $s, s' \in S$, it holds that $\textsc{Enc}(Z, s, r)$ and $\textsc{Enc}(Z, s', r)$ are identically distributed. For every integers $t_1, \dots, t_k$, we say that $\mathcal{P}$ is a $(t_1, \dots, t_k)$-RCDS protocol if it is robust for every zero set $Z_1 \times Z_2 \times \cdots \times Z_k$ such that $|Z_i| \le t_i$ for every $i \in [k]$. Finally, for every integer t, we say that $\mathcal{P}$ is a t-RCDS protocol if it is a $(t, \dots, t)$-RCDS protocol.*

3 Degree-d Secret Sharing and Degree-d CDS Protocols

In [39], polynomial secret-sharing schemes are defined as secret-sharing schemes in which the sharing function can be computed by polynomial of low degree. In this paper, we define secret-sharing schemes with polynomial reconstruction and secret-sharing schemes with both polynomial sharing and reconstruction.

Definition 3.1 (Degree of Polynomial)**.** *The degree of each multivariate monomial is the sum of the degree of all its variables; the degree of a polynomial is the maximal degree of its monomials.*

Definition 3.2 (Degree-d Mapping over $\mathbb{F}$)**.** *A function $f : \mathbb{F}^\ell \to \mathbb{F}^m$ can be computed by degree-d polynomials over $\mathbb{F}$ if there are m polynomials $Q_1, \dots, Q_m : \mathbb{F}^\ell \to \mathbb{F}$ of degree at most d s.t. $f(x_1, \dots, x_\ell) = (Q_1(x_1, \dots, x_\ell), \dots, Q_m(x_1, \dots, x_\ell))$.*

A secret-sharing scheme has a polynomial sharing if the mapping that the dealer uses to generate the shares given to the parties can be computed by polynomials, as we formalize at the following definition.

Definition 3.3 (Secret-Sharing Schemes with Degree-d Sharing [39]). *Let* Π *be a secret-sharing scheme with domain of secrets* S*. We say that the scheme* Π *has* degree-d sharing over a finite field $\mathbb{F}$ *if there are integers* $\ell, \ell_r, \ell_1, \ldots, \ell_n$ *such that* $S \subseteq \mathbb{F}^\ell$, $R = \mathbb{F}^{\ell_r}$*, and* $S_i = \mathbb{F}^{\ell_i}$ *for every* $i \in [n]$*, and* Π *can be computed by degree-*d *polynomials over* $\mathbb{F}$*.*

In Definition 3.3, we allow S to be a subset of $\mathbb{F}^\ell$ (in [39], $S = \mathbb{F}^\ell$). In particular, we will study the case where $\ell = 1$ and $S = \{0, 1\} \subseteq \mathbb{F}$.

A secret-sharing scheme has a polynomial reconstruction if for every authorized set the mapping that the set uses to reconstruct the secret from its shares can be computed by polynomials.

Definition 3.4 (Secret-Sharing Schemes with Degree-d Reconstruction). *Let* Π *be a secret-sharing scheme with domain of secrets* S*. We say that the scheme* Π *has a* degree-d reconstruction over a finite field $\mathbb{F}$ *if there are integers* $\ell, \ell_r, \ell_1, \ldots, \ell_n$ *such that* $S \subseteq \mathbb{F}^\ell$, $R = \mathbb{F}^{\ell_r}$*, and* $S_i = \mathbb{F}^{\ell_i}$ *for every* $i \in [n]$*, and* Recon_B*, the reconstruction function of the secret, can be computed by degree-*d *polynomials over* $\mathbb{F}$ *for every* $B \in \Gamma$*.*

Definition 3.5 (Degree-d Secret-Sharing Schemes). *A secret-sharing scheme* Π *is a degree-*d *secret-sharing scheme over* $\mathbb{F}$ *if it has degree-*d *sharing and degree-*d *reconstruction over* $\mathbb{F}$*.*

Definition 3.6 (CDS Protocols with Degree-d Encoding). *A CDS protocol* $\mathcal{P}$ *has a degree-*d *encoding over a finite field* $\mathbb{F}$ *if there are integers* $\ell, \ell_r, \ell_1, \ldots, \ell_k \geq 1$ *such that* $S \subseteq \mathbb{F}^\ell$*,* $R = \mathbb{F}^{\ell_r}, M_i = \mathbb{F}^{\ell_i}$ *for every* $1 \leq i \leq k$*, and for every* $i \in [k]$ *and every* $x \in X_i$ *the function* $\textsc{Enc}_{i,x} : \mathbb{F}^{\ell+\ell_r} \to M_i$ *can be computed by degree-*d *polynomials over* $\mathbb{F}$*, where* $\textsc{Enc}_{i,x}(s, r) = \textsc{Enc}_i(x, r, s)$*.*

Definition 3.7 (CDS Protocols with Degree-d Decoding). *A CDS protocol* $\mathcal{P}$ *has a degree-*d *decoding over a finite field* $\mathbb{F}$ *if there are integers* $\ell, \ell_r, \ell_1, \ldots, \ell_k \geq 1$ *such that* $S \subseteq \mathbb{F}^\ell$*,* $R = \mathbb{F}^{\ell_r}$*,* $M_i = \mathbb{F}^{\ell_i}$ *for every* $1 \leq \ell \leq k$*, and for every inputs* $x_1, \ldots, x_k$ *the function* $\textsc{Dec}_{x_1,\ldots,x_k} : \mathbb{F}^{\ell_1+\cdots+\ell_k} \to S$ *can be computed by degree-*d *polynomials over* $\mathbb{F}$*, where* $\textsc{Dec}_{x_1,\ldots,x_k}(m_1, \ldots, m_k) = \textsc{Dec}(x_1, \ldots, x_k, m_1, \ldots, m_k)$*.*

Note that in Definition 3.7, the polynomials computing the decoding can be different for every input x.

Definition 3.8 (Degree-d CDS Protocols). *A CDS protocol* $\mathcal{P}$ *is a degree-*d *CDS protocol over* $\mathbb{F}$ *if it has degree-*d *encoding and degree-*d *decoding over* $\mathbb{F}$*.*

Definition 3.9 (Linear Secret-Sharing Schemes and CDS Protocols). *A linear polynomial is a degree-*1 *polynomial. A linear secret-sharing scheme is a degree-*1 *secret-sharing scheme and* $\ell = 1$ *(i.e., the secret contains one field element). A secret-sharing scheme has a linear sharing (resp., reconstruction) if it has degree-*1 *sharing (resp., reconstruction). Similar notations hold for CDS protocols.*

Secret-sharing schemes with linear sharing are equivalent to secret-sharing schemes with linear reconstruction as shown by [33, 12].

Claim 3.10 ([33, 12]). *A secret-sharing scheme* Π *is linear if and only if for every authorized set* B *the reconstruction function* Recon_B *is a linear mapping.*

In Section 9, we generalize Claim 3.10 and show that secret-sharing schemes with degree-1 sharing (i.e., multi-linear schemes) are equivalent to secret-sharing schemes with degree-1 reconstruction.

Definition 3.11 (Quadratic Secret-Sharing Schemes and CDS Protocols). *A quadratic polynomial is a degree-*2 *polynomial. A quadratic secret-sharing scheme is a degree-*2 *secret-sharing scheme. A secret-sharing scheme has a quadratic sharing (resp., reconstruction) if it has degree-*2 *sharing (resp., reconstruction). Similar notations hold for CDS protocols.*

Let $\mathcal{A} = \{\mathcal{A}_n\}_{n\in\mathbb{N}}$ be a family of access structures, where $\mathcal{A}_n$ is an n-party access structure. We informally say that $\mathcal{A}$ can be realized by polynomial secret-sharing schemes if it can be realized by degree-$f(n)$ secret-sharing schemes where $f(n)$ is a constant or relatively small function, i.e., $\log n$.

Remark 3.12. *Observe that for every finite field, every function can be computed by a polynomial (with high degree). Therefore, every access structure can be realized by a secret-sharing scheme with polynomial reconstruction of high degree. This is not true for sharing since we require that the polynomial sharing uses uniformly distributed random elements of the field. However, by relaxing correctness and security, we can also get a statistical secret-sharing scheme with polynomial sharing of high degree (by sampling many field elements and constructing a distribution that is close to uniform on the set R of the random strings of the secret-sharing scheme).*

3.1 CDS with Degree-3 Encoding for the Non-Quadratic Residues Function

In this section we show an example of a function that can be realized by an efficient CDS protocol with degree-3 encoding, but, under the assumption that the quadratic residue modulo a prime problem is not in NC, it does not have an efficient CDS protocol with degree-d decoding (for any constant d). Our construction is built upon [16] where they construct an efficient non-linear secret-sharing scheme for an access structure that corresponds to the quadratic residue function. In the construction of [16], the random string is not uniformly distributed in the field (as we require from CDS protocols with polynomial encoding). In the following construction, in order to get a degree-d encoding, we choose the random string uniformly, resulting in a small error in the correctness.

The Quadratic Residue Modulo a Prime Problem. For a prime p, let $\mathrm{QR}_p = \{a \in \{1,\dots,p-1\} : \exists b \in \{1,\dots,p-1\}\ a \equiv b^2 \pmod p\}$. The quadratic residue modulo a prime problem is given p and a, where p is a prime, and outputs 1 if and only if $a \in \mathrm{QR}_p$. All the *known* algorithms for the quadratic residue modulo a prime problem are sequential and it is not not known if efficient parallel algorithms for this problem exist. The known algorithms are of two types; the first type requires computing a modular exponentiation and the second requires computing the gcd. Therefore, the problem is related to modular exponentiation and gcd problems, and thus according to the current state of the art, it is reasonable to assume that the problem is not in NC (see [16] for more details).

We define, for a prime p and $k = \lfloor \log p \rfloor - 1$, the function $f_{\mathrm{NQRP}_p} : \{0,1\}^k \to \{0,1\}$ such that $f_{\mathrm{NQRP}_p}(x_1,\dots,x_k) = 1$ if $(1 + \sum_{i=1}^{k} 2^i x_i) \bmod p \notin \mathrm{QR}_p$ and $f_{\mathrm{NQRP}_p}(x_1,\dots,x_k) = 0$ otherwise.[4] The function f_{NQRP_p} is realized by the CDS protocol depicted in Fig. 1. This protocol has perfect security, however, it has a one-side error $1/p$ in the correctness. Repeating this protocol t times will result in a protocol with error $O(1/p^t)$.

Lemma 3.13. *For every t, there is a k-server CDS protocol with degree-3 encoding over $\mathbb{F}_p$ for the function f_{NQRP_p} with $S = \{0,1\}$ and an error in correctness of $1/p^t$ and message size of $O(t \log p)$.*

Proof. In Fig. 1, we describe a k-server CDS protocol for f_{NQRP_p}. We next prove its correctness and security.

For correctness, assuming $r \neq 0$, when $s = 0$ the sum of the messages the referee gets is $\sum_{i=1}^{k} z_i + r^2 \equiv r^2 \bmod p$, and when $s = 1$ the sum is $r^2(1 + \sum_{i=1}^{k} 2^i x_i) \bmod p$. Recall that $r^2 \cdot a \in \mathrm{QR}_p$ iff $a \in \mathrm{QR}_p$. Therefore, when $f_{\mathrm{NQRP}_p}(x_1,\dots,x_k) = 1$, $s = 1$ iff the sum of the messages is not in QR_p. The referee can

[4] We add 1 to the input to avoid the input 0, which is neither a quadratic residue nor a quadratic non residue.

CDS protocol for f_{NQRP_p}

- The secret: A bit $s \in \{0,1\}$.
- Q_i for every $1 \le i \le k$ holds $x_i \in \{0,1\}$.
- Common randomness: $r, z_1, \ldots, z_{k-1} \in \mathbb{F}_p$.
- **The protocol**
  - Calculate $z_k = -\sum_{j=1}^{k-1} z_j$.
  - Server Q_1 sends $(z_1 + s \cdot 2^1 x_1 r^2 + r^2) \bmod p$.
  - Server Q_i for every $2 \le i \le k$ sends $(z_i + s \cdot 2^i x_i r^2) \bmod p$.

Figure 1: A k-server CDS protocol with Degree-3 Encoding for f_{NQRP_p}.

reconstruct the secret when the random element r is in $\mathbb{F}_p \setminus \{0\}$, thus the referee can reconstruct the secret with probability $1-1/p$. To amplify the correctness, we repeat the protocol t times and get correctness with probability of $1 - 1/p^t$.

In order to prove security, we prove that every k-tuples of messages for an input $x_1, \ldots, x_k$ such that $f_{\mathrm{NQRP}_p}(x_1, \ldots, x_k) = 0$ the messages are identically distributed when $s = 0$ and when $s = 1$. When $r = 0$ the messages are uniform random elements whose sum is 0 regardless of the secret. Otherwise, regardless of the secret, the sum of the messages is a uniformly random distributed quadratic residue: for $s = 0$ the sum is $r^2 \bmod p$ and for $s = 1$ the sum is $b = r^2(1 + \sum_{i=1}^{k} 2^i x_i) \bmod p \in \mathrm{QR}_p$ which is also a uniformly distributed quadratic residue. Thus, in both cases the messages are random elements in $\mathbb{F}_p$ with the restriction that their sum is a random quadratic residue.

Each message contains only one field element of size $\log p$. As we repeat the protocol t times, the message size is $t \log p$. The encoding function is $z_i + (2^i x_i) \cdot sr^2 \bmod p$ which is a degree-3 polynomial in the secret and the randomness (for every x_i). $\square$

In Lemma 4.4 we show that for any constant d, any CDS protocol with degree-d decoding and message size M can be transformed to a linear CDS protocol in which the message size is M^d. Recall that any sequence of functions $\{f_i\}_{i \in \mathbb{N}}$ that can be realized by a linear CDS protocol with polynomial message size (in the number of servers) is in NC, i.e., it has a family of circuits of poly-logarithmic depth and polynomial size (see discussion in Remark 4.6). The above is true even if there is an exponentially small error in the correctness (see Remark 9.8). Thus, we obtain the following corollary.

Corollary 3.14. *Under the assumption that $\{\mathrm{NQRP}_p\}_{p \text{ is a prime}} \notin \mathrm{NC}$, there is a sequence of functions that can be realized by an efficient CDS protocol with degree-3 encoding, but for any constant d, cannot be realized by an efficient CDS protocol with degree-d decoding.*

4 Lower Bounds for Secret Sharing with Degree-d Reconstruction

In this section, we show lower bounds for secret-sharing schemes with degree-d reconstruction.

4.1 Lower Bounds for 1-Bit Secrets for Implicit Access Structures

The following theorem was showed in [35].

Theorem 4.1 (Implied by [35]). *Let $\mathcal{F}_{\text{rec}}$ be the family of possible reconstruction functions, c be the sum of the share sizes of all the parties (i.e., the total share size), and $\mathcal{F}_{\mathcal{A}}$ be a family of n-party access structures. For all but at most $\sqrt{|\mathcal{F}_{\mathcal{A}}|}$ access structures $\Gamma \in \mathcal{F}_{\mathcal{A}}$, for any secret-sharing scheme with domain of secrets $\{0,1\}$ and reconstruction function from $\mathcal{F}_{\text{rec}}$, it holds that*

$$\log|\mathcal{F}_{\text{rec}}| \cdot c = \Omega(\log|\mathcal{F}_{\mathcal{A}}|).$$

We obtain the following two corollaries.

Corollary 4.2. *For almost all n-party access structures, any secret-sharing scheme realizing them over any finite field with domain of secrets $\{0,1\}$ and degree-d reconstruction requires total share size of $2^{n/(d+1)-o(n)}$.*

Proof. Let $\mathcal{F}_{\mathcal{A}}$ be the family of all n-party access structures. Thus, $|\mathcal{F}_{\mathcal{A}}| = 2^{\Theta(2^n/\sqrt{n})}$. We next consider the family of degree-d polynomials as the family of reconstruction functions.

Fix a finite field $\mathbb{F}$, and consider shares of total size c, hence they contain $v = c/\log|\mathbb{F}|$ field elements. In this case the reconstruction function is a polynomial of degree $\leq d$ in v variables. There are at most $(v+1)^d$ monomials of degree $\leq d$ (for each of the d variables we choose either an element from the v shares or 1 for degree smaller than d), thus less than $|\mathbb{F}|^{(v+1)^d} = 2^{\log|\mathbb{F}|\cdot(c/\log|\mathbb{F}|+1)^d} \leq 2^{(c+1)^d}$ polynomials of degree $\leq d$ (as the reconstruction function can choose any coefficient in $\mathbb{F}$ for every monomial). If $|\mathbb{F}| > 2^{2^{n/(d+1)}}$, then the share size of every secret-sharing scheme over $\mathbb{F}$ is $> 2^{n/(d+1)}$ (since $\log|\mathbb{F}| \geq 2^{n/(d+1)}$). Thus, we only need to consider at most $2^{2^{n/(d+1)}}$ fields, and consider $\mathcal{F}_{\text{rec}}$ of size $2^{2^{n/(d+1)}} \cdot 2^{(c+1)^d}$. Thus, by Theorem 4.1, $(2^{n/(d+1)} + (c+1)^d) \cdot c \geq \Omega(2^n/\sqrt{n})$, so $c^{d+1} \geq 2^{n-o(n)}$ and $c \geq 2^{n/(d+1)-o(n)}$. □

Corollary 4.3. *For almost all k-input functions $f : [N]^k \rightarrow \{0,1\}$, the message size in any degree-d CDS protocol for them over any finite field with domain of secrets $\{0,1\}$ is $\Omega(N^{(k-1)/(d+1)}/k)$.*

Proof. CDS protocols are a special case of secret-sharing schemes, where for every function $f : [N]^k \rightarrow \{0,1\}$ there is a kN-party access structure Γ_f containing all the one-inputs of the function f. The access structure Γ_f is defined as follows: The kN parties in Γ_f are partitioned into k disjoint sets $B_1, \ldots, B_k$ of size N such that $B_i = \{p_{1,i}, \ldots, p_{N,i}\}$. A set A is authorized in Γ_f iff $|A| \geq k+1$ or there exist $j_1, \ldots, j_k$ such that $f(j_1, \ldots, j_k) = 1$ and for every $1 \leq i \leq k$ it holds that $p_{j_i,i} \in A$. We ignore sets of size greater than k as we know that they are in the access structure and thus $|\mathcal{F}_A|$ will not be affected, therefore the proof of [35] still holds. The share size of a party in the secret-sharing scheme realizing Γ_f is the message size of a CDS protocol for the function (up to an additive logarithmic factor). Let α be the message size of each server in a CDS protocol and c be the total share size of the corresponding secret-sharing schemes. Thus, since for each of the k servers of the CDS protocol we have N parties in the secret-sharing scheme (for each possible input for the server), we get $c = \alpha kN$.

We take $\mathcal{F}_{\mathcal{A}}$ as the family of all possible functions $f : [N]^k \rightarrow \{0,1\}$, which is of size 2^{N^k}. Over a field $\mathbb{F}$, a minimal authorized set of size k holds $v = \alpha k/\log|\mathbb{F}|$ field elements. Similarly to the proof of Corollary 4.2, the number of polynomials of degree $\leq d$ in $v = \alpha k/\log|\mathbb{F}|$ variables over a finite field $\mathbb{F}$ is less than $|\mathbb{F}|^{(v+1)^d} \leq 2^{(\alpha k+1)^d}$. We take $\mathcal{F}_{\text{rec}}$ as the family of all polynomials of degree at most d in v variables over fields of size smaller than $2^{N^{(k-1)/(d+1)}}$; the size of $\mathcal{F}_{\text{rec}}$ is less than $2^{N^{(k-1)/(d+1)}} \cdot 2^{(\alpha k+1)^d}$. By Theorem 4.1, $(N^{(k-1)/(d+1)} + (\alpha k+1)^d) \cdot c \geq \Omega(N^k)$ (where $c = \alpha kN$), so $(\alpha k)^{d+1} \geq \Omega(N^{k-1})$ and $\alpha \geq \Omega(N^{(k-1)/(d+1)}/k)$. □

4.2 A Transformation from Secret Sharing with Degree-d Reconstruction into a Linear Secret Sharing

We start with a transformation from secret-sharing schemes with polynomial reconstruction to linear schemes. The idea of the transformation is to add random field elements to the randomness of the original polynomial scheme and generate new shares using these random elements, such that the reconstruction of the secret in the resulting scheme is a linear combination of the elements in the shares of the resulting scheme. In particular, for every monomial of degree at least two in a polynomial used for the reconstruction, we share the value of the monomial among the parties that have elements in the monomial. That is, the sharing function computes the polynomials instead of the reconstruction algorithm. As a corollary, we obtain a lower bound on the share size for schemes with polynomial reconstruction.

Lemma 4.4. *Let Γ be an n-party access structure, and assume that there exists a secret-sharing scheme Π_P realizing Γ over $\mathbb{F}$ with ℓ-elements secrets and degree-d reconstruction, in which the shares contain together c field elements. Then, there is a multi-linear secret-sharing scheme Π_L realizing Γ over $\mathbb{F}$ with ℓ-elements secrets, in which the share of each party contains $O(c^d)$ field elements. In particular, if the secret in Π_P contains one field element then Π_L is linear.*

Proof. To construct the desired scheme Π_L, the dealer first shares the secret according to scheme Π_P. Then, for every possible monomial $x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}$ in the reconstruction of some authorized set such that $2 \leq \sum_{i=1}^{d'} \ell_i \leq d$, where x_{i_j} is a field element in the share of a party P_{i_j} for every $j \in [d']$, the dealer computes the value v of the monomial (using the shares that it creates) and shares v using a d'-out-of-d' secret-sharing scheme among the parties $P_{i_1}, \ldots, P_{i_{d'}}$ (i.e., the dealer chooses d' random field elements $r_{i_1}^v, \ldots, r_{i_{d'}}^v$ such that $v = r_{i_1}^v + \cdots + r_{i_{d'}}^v$).[5] Note that the randomness of scheme Π_L contains the random elements of scheme Π_P and the random elements $r_{i_1}^v, \ldots, r_{i_{d'-1}}^v$ for every possible monomial $x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}$ of value v such that $2 \leq \sum_{i=1}^{d'} \ell_i \leq d$ as above (the dealer computes $r_{i_{d'}}^v = x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}} - r_{i_1}^v - \cdots - r_{i_{d'-1}}^v$).

We prove that the construction of Π_L realizes Γ and has linear reconstruction. By the equivalence between linear reconstruction and linear sharing (even for multi-element secrets), which is shown in Section 9, Π_L can be converted to a secret-sharing scheme with linear sharing and reconstruction while preserving the share size.

We now prove the correctness of Π_L. For an authorized set $B \in \Gamma$, denote S_B as the field elements in the shares of B, and let

$$\mathrm{Recon}_{B,j}(S_B) = \sum_{x_i \in S_B} \alpha_{x_i} x_i + \sum_{\substack{x_{i_1}, \ldots, x_{i_{d'}} \in S_B, d' \leq d, \\ 2 \leq \ell_1 + \cdots + \ell_{d'} \leq d}} \alpha_{x_{i_1}^{\ell_1}, \ldots, x_{i_{d'}}^{\ell_{d'}}} x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}$$

be the reconstruction function of B of the j-th element of the secret in scheme Π_P. Then, the set B can reconstruct the secret in scheme Π_L by applying the linear combination of the field elements in the shares

[5] If there is more than one element of some party in the monomial, the dealer can share the monomial among the parties that have elements in it, or give to such a party the sum of the shares that corresponding to its elements.

of the parties as follows:

$$\sum_{x_i \in S_B} \alpha_{x_i} x_i + \sum_{\substack{x_{i_1},\dots,x_{i_{d'}} \in S_B, d' \leq d,\\ 2 \leq \ell_1 + \cdots + \ell_{d'} \leq d}} \alpha_{x_{i_1}^{\ell_1},\dots,x_{i_{d'}}^{\ell_{d'}}} \sum_{j=1}^{d'} r^v_{i_j}$$

$$= \sum_{x_i \in S_B} \alpha_{x_i} x_i + \sum_{\substack{x_{i_1},\dots,x_{d'_{i_{d'}}} \in S_B, d' \leq d,\\ 2 \leq \ell_1 + \cdots + \ell_{d'} \leq d}} \alpha_{x_{i_1}^{\ell_1},\dots,x_{i_{d'}}^{\ell_{d'}}} x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}.$$

We next prove the security of Π_L. Let T be an unauthorized set. For every authorized subset T' it must be that $T' \not\subseteq T$, thus, the set T misses at least one random field element $r^v_{i_j}$ from any monomial for the set T', so it cannot learn information on the value of these monomials, and hence cannot learn information on the secret from these values. In the scheme Π_L, the set T can only learn its shares in scheme Π_P, and every possible monomial of at most d variables that contains elements of those shares; these additional values can be computed from the original shares of T. Thus, in scheme Π_L, the set T learns only the information it can learn in scheme Π_P, and, hence, by the security of scheme Π_P, the set T cannot learn any information about the secret.

Finally, in scheme Π_L, each party gets at most c field elements from the share of scheme Π_P, and an element from the d'-out-of-d' secret-sharing scheme, for every monomial as above $x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}$ such that $2 \leq \sum_{i=1}^{d'} \ell_i \leq d$; there are at most $\sum_{d'=2}^{d} c^{d'}$ such monomials. Overall, each party gets $c + \sum_{d'=2}^{d} c^{d'} = O(c^d)$ field elements. □

The above transformation gives us a lower bound on the share size of secret-sharing schemes with polynomial reconstruction, using any lower bound on the share size of linear secret-sharing schemes, as described next.

Corollary 4.5. *Assume that there exist an n-party access structure Γ such that the share size of at least one party in every linear secret-sharing scheme realizing Γ is c. Then, the share size of at least one party in every secret-sharing scheme realizing Γ with degree-d reconstruction is $\Omega(c^{1/d})$.*

Remark 4.6. *Recall that the class NC^i contains all Boolean functions (or problems) that can be computed by polynomial-size Boolean circuits with gates with fan-in at most two and depth $O(\log^i n)$. Following the discussion in [16], the class of access structures that have a linear secret-sharing scheme with polynomial share size contains monotone NC^1 and is contained in algebraic NC^2 and in NC^3 for small enough fields (at most exponential in polynomial of the number of parties n). Lemma 4.4 implies that the class of access structures that have a secret-sharing scheme with polynomial reconstruction and polynomial share size is also contained in NC^3.*

4.3 Lower Bounds for 1-Element Secrets for Explicit Access Structures

Now, let us recall the explicit lower bound of Pitassi and Robere [42] on the share size of linear secret-sharing schemes.

Theorem 4.7 ([42]). *There is a constant $\beta > 0$ such that for every n, there is an explicit n-party access structure Γ such that for every finite field $\mathbb{F}$, any linear secret-sharing scheme realizing Γ over $\mathbb{F}$ requires total share size of $\Omega(2^{\beta n} \log |\mathbb{F}|)$.*

The next explicit lower bound for secret-sharing schemes with polynomial reconstruction and one-element secrets follows directly from Corollary 4.5 when using Theorem 4.7.

Corollary 4.8. *There is a constant $\beta > 0$ such that for every n, there is an explicit n-party access structure Γ such that for every d and every finite field $\mathbb{F}$, any secret-sharing scheme realizing Γ over $\mathbb{F}$ with degree-d reconstruction and one-element secrets requires total share size of* $\Omega(2^{\beta n/d} \log |\mathbb{F}|)$.

Recall that the information ratio (or the normalized share size) is the ratio between the share size and the secret size. Corollary 4.8 provides a lower bound on the information ratio of an explicit access structure even for large finite fields. Corollary 4.2 provides a lower bound with a better constant in the exponent, however, it only applies to implicit access structures and does not give a non-trivial lower bound on the information ratio for large finite fields.

5 Quadratic CDS Protocols

In this section, we construct a quadratic k-server CDS protocol, i.e., a CDS protocol in which the encoding and decoding are computed by degree-2 polynomials. We start by describing a quadratic two-server CDS protocol (a variant of the quadratic two-server CDS protocol of [37]) and then construct a quadratic k-server CDS protocol that "simulates" the two-server CDS protocol.

A Quadratic Two-Server CDS Protocol. As a warm-up, we describe in Fig. 2 a two-server CDS protocol in which the encoding and the decoding are computed by polynomials of degree 2 over $\mathbb{F}_2$. This protocol is a variant of the protocol of [37] using a different notation (i.e., using cubes instead of polynomials).

Lemma 5.1. *Protocol Π_2, described in Fig. 2, is a quadratic two-server CDS protocol over $\mathbb{F}_2$ for the function* INDEX_N^2 *with message size $O(N^{1/3})$.*

Proof. We start with analyzing the value of the expression in (1). When $s = 0$, Bob sends $A_1 = S_1$, $A_2 = S_2$, and $A_3 = S_3$ to the referee. Thus, when $s = 0$, we get that $m^1_{i_1} = m_1 \oplus r_{1,i_1} \oplus r_1$, $m^2_{i_2} = m_2 \oplus r_{2,i_2} \oplus r_2$, and $m^3_{i_3} = m_3 \oplus r_{3,i_3} \oplus r_3$, and the value of the expression in (1) is

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3} = r_1 \oplus r_2 \oplus r_3 = 0. \tag{2}$$

When $s = 1$, Bob sends $A_1 = S_1 \oplus \{i_1\}$, $A_2 = S_2 \oplus \{i_2\}$, and $A_3 = S_3 \oplus \{i_3\}$ to the referee. We observe the following:

$$\begin{aligned} m_1 &= \left(\bigoplus_{j_2 \in S_2 \oplus \{i_2\}, j_3 \in S_3 \oplus \{i_3\}} D_{i_1,j_2,j_3} \right) \\ &= \left(\bigoplus_{j_2 \in S_2, j_3 \in S_3 \oplus \{i_3\}} D_{i_1,j_2,j_3} \right) \oplus \left(\bigoplus_{j_3 \in S_3 \oplus \{i_3\}} D_{i_1,i_2,j_3} \right) \\ &= \left(\bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \right) \oplus \left(\bigoplus_{j_2 \in S_2} D_{i_1,j_2,i_3} \right) \oplus \left(\bigoplus_{j_3 \in S_3} D_{i_1,i_2,j_3} \right) \oplus D_{i_1,i_2,i_3}. \end{aligned} \tag{3}$$

Protocol Π_2

- The secret: A bit $s \in \{0,1\}$.
- Alice holds a database $D \in \{0,1\}^N$ and Bob holds an index $i \in [N]$ viewed as (i_1, i_2, i_3) such that $i_1, i_2, i_3 \in [N^{1/3}]$.
- Common randomness: $S_1, S_2, S_3 \subseteq [N^{1/3}]$, $r_1, r_2 \in \{0,1\}$, and $3N^{1/3}$ bits $r_{1,j_1}, r_{2,j_2}, r_{3,j_3} \in \{0,1\}$ for every $j_1, j_2, j_3 \in [N^{1/3}]$.
- **The protocol**
  - Compute $r_3 = r_1 \oplus r_2$.
  - Alice computes $3N^{1/3}$ bits:
    * $m^1_{j_1} = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus r_{1,j_1} \oplus r_1$ for every $j_1 \in [N^{1/3}]$.
    * $m^2_{j_2} = \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus r_{2,j_2} \oplus r_2$ for every $j_2 \in [N^{1/3}]$.
    * $m^3_{j_3} = \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,j_3} \oplus r_{3,j_3} \oplus r_3$ for every $j_3 \in [N^{1/3}]$.
  - Alice sends $(m^1_{j_1})_{j_1 \in [N^{1/3}]}$, $(m^2_{j_2})_{j_2 \in [N^{1/3}]}$, $(m^3_{j_3})_{j_3 \in [N^{1/3}]}$ to the referee.
  - Bob computes 3 strings $A_h = (A_h[1], \dots, A_h[N^{1/3}])$ for $h \in \{1,2,3\}$ (each string of length $N^{1/3}$), where
    * $A_h[j_h] = S_h[j_h]$ for every $j_h \neq i_h$.
    * $A_h[i_h] = S_h[i_h] \oplus s$
      (that is, if $s = 0$ then $A_h = S_h$, otherwise $A_h = S_h \oplus \{i_h\}$).
  - Bob sends $r_{1,i_1}, r_{2,i_2}, r_{3,i_3}$, and A_1, A_2, A_3 to the referee.
  - The referee computes:
    $m_1 = \bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3}$, $\quad m_2 = \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3}$,
    $m_3 = \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3}$
    and outputs

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3}. \tag{1}$$

Figure 2: A quadratic two-server CDS protocol Π_2 for the function INDEX^2_N.

Similarly,

$$m_2 = \left(\bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \right) \oplus \left(\bigoplus_{j_1 \in S_1} D_{j_1,i_2,i_3} \right) \oplus \left(\bigoplus_{j_3 \in S_3} D_{i_1,i_2,j_3} \right) \oplus D_{i_1,i_2,i_3}.$$

$$m_3 = \left(\bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3} \right) \oplus \left(\bigoplus_{j_1 \in S_1} D_{j_1,i_2,i_3} \right) \oplus \left(\bigoplus_{j_2 \in S_2} D_{i_1,j_2,i_3} \right) \oplus D_{i_1,i_2,i_3}.$$

Therefore,

$$m_1 \oplus m_2 \oplus m_3 = \left(\bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \right) \oplus \left(\bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \right) \oplus \left(\bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3} \right) \oplus D_{i_1,i_2,i_3}.$$

Thus, when $s = 1$, the value of the expression in (1) is

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3} \oplus r_1 \oplus r_2 \oplus r_3 = D_{i_1,i_2,i_3}. \tag{4}$$

Correctness. We next prove the correctness of the protocol, that is, when $D_{i_1,i_2,i_3} = 1$ the referee correctly reconstructs s. Recall that the output of the referee is the expression in (1). As explained above, when $s = 0$ the referee outputs 0 and when $s = 1$ the referee outputs $D_{i_1,i_2,i_3} = 1$.

Security. Fix inputs D and $i = (i_1, i_2, i_3)$ such that $D_{i_1,i_2,i_3} = 0$, a message of Alice $(m^1_{j_1})_{j_1 \in [N^{1/3}]}$, $(m^2_{j_2})_{j_2 \in [N^{1/3}]}$, $(m^3_{j_3})_{j_3 \in [N^{1/3}]}$, and a message of Bob $A_1, A_2, A_3, r_{1,i_1}, r_{2,i_2}, r_{3,i_3}$ such that

$$\bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3} \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3} = 0 \tag{5}$$

(no other restrictions are made on the messages). By (2) and (4), when $D_{i_1,i_2,i_3} = 0$ only such messages are possible. We next argue that the referee cannot learn any information about the secret given these inputs and messages, i.e., these messages have the same probability when $s = 0$ and when $s = 1$. In particular, we show that for every secret $s \in \{0, 1\}$ there is a unique common random string r such that Alice and Bob send these messages with the secret s. We define the common random string r as follows:

- For $h \in \{1, 2, 3\}$, define $S_h = A_h$ if $s = 0$ and $S_h = A_h \oplus \{i_h\}$ if $s = 1$. These S_1, S_2, S_3 are consistent with the message of Bob and s and are the only consistent choice. Both when $s = 0$ and $s = 1$, as $D_{i_1,i_2,i_3} = 0$, it holds that

$$\bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3} = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3}. \tag{6}$$

  This is true since when $s = 0$ the sets A_1, A_2, A_3 are the same as the sets S_1, S_2, S_3, and when $s = 1$, by (4), the two sides of the expression are differ by D_{i_1,i_2,i_3} which is 0.

- The message of Bob determines r_{1,i_1}, r_{2,i_2}, and r_{3,i_3}.

- Define

$$r_1 = m^1_{i_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus r_{1,i_1} \tag{7}$$

$$r_2 = m^2_{i_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus r_{2,i_2}. \tag{8}$$

Given the secret s, the inputs, and the messages of Alice and Bob, these values are possible and unique.

- Define $r_3 = r_1 \oplus r_2$. By (5), (6), (7), and (8), this value is possible, i.e., it satisfies

$$m^3_{i_3} = \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3} \oplus r_{3,i_3} \oplus r_3.$$

- For every $j_1 \neq i_1, j_2 \neq i_2$, and $j_3 \neq i_3$ define

$$r_{1,j_1} = m^1_{j_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus r_1,$$

$$r_{2,j_2} = m^2_{j_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus r_2,$$

$$r_{3,j_3} = m^3_{j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3} \oplus r_3.$$

Given the secret s, the inputs, and the messages of Alice and Bob, these values are possible and unique.

Recall that the common random string is uniformly distributed (i.e., the probability of each such string is $1/2^{6N^{1/3}+2}$, as it contains $6N^{1/3} + 2$ bits). Since for every pair of messages of Alice and Bob when $D_{i_1,i_2,i_3} = 0$ we have that every secret s has exactly one consistent random string, this pair has the same probability when $s = 0$ and when $s = 1$ and the security follows.

Message Size. Alice sends $3N^{1/3}$ bits and Bob sends 3 strings each of size $N^{1/3}$ and 3 random bits, so the message size is $O(N^{1/3})$.

Degree of the Protocol. The message of Alice contains an exclusive or of bits of a 3-dimension cubes, where two dimensions are determined by the common randomness (the sets S_1, S_2, S_3). That is, when we represent a set $S \subseteq [N^{1/3}]$ by $N^{1/3}$ bits $S = (S[1], \dots, S[N^{1/3}])$, then for every $j_1 \in [N^{1/3}]$

$$m^1_{j_1} = \bigoplus_{j_2 \in [N^{1/3}], j_3 \in [N^{1/3}]} S_2[j_2] \cdot S_3[j_3] \cdot D_{j_1,j_2,j_3} \oplus r_{1,j_1} \oplus r_1.$$

Thus, $m^1_{j_1}$, for every input D, is a polynomial of degree 2 over $\mathbb{F}_2$ whose variables are the bits of the random string. Similarly, $m^2_{j_2}$, $m^3_{j_3}$ are polynomials of degree 2 over $\mathbb{F}_2$. The message of Bob for every $j_h \neq i_h$ contains a polynomial of degree 1 over $\mathbb{F}_2$, since it sends $S_h[j_h]$. For the index $i_h \in [N^{1/3}]$, Bob sends $S_h[i_h] \oplus s$, which is a polynomial of degree 1 over $\mathbb{F}_2$. The decoding is also a computation of a 3-dimension cube such that only two dimensions are determined by the common randomness, i.e., the decoding is a degree-2 polynomial over $\mathbb{F}_2$. □

An Auxiliary Protocol $\Pi_{\mathbf{XOR}}$. In Fig. 4, we will describe a k-server CDS protocol, where servers $Q_2, \dots, Q_k$ simulate Bob in the two-server CDS protocol. To construct this protocol, we design a k-server protocol Π_{XOR} that simulates Bob, i.e., sends a set A, where $A = S$ if $s = 0$ and $A = S \oplus \{i\}$ if $s = 1$. In Π_{XOR}, each server Q_ℓ holds an index i_ℓ, which together determine an index $i = (i_1, i_2, \dots, i_k)$, and they need to send messages to the referee such that the referee will learn A without learning any information on s. Let $N_1, \dots, N_k$ be integers and $N = N_1 \cdot \ldots \cdot N_k$. We construct the following protocol in which server

Q_1 holds a set $S \subseteq [N]$ represented by a k-dimensional Boolean array $(Sj_1, \ldots, j_k)_{j_1 \in [N_1], \ldots, j_k \in [N_k]}$, the secret s, and an index $i_1 \in [N_1]$. Server Q_ℓ for $2 \leq \ell \leq k$ holds an index $i_\ell \in [N_\ell]$. If $s = 1$, the referee outputs $S \oplus \{(i_1, i_2, \ldots, i_k)\}$ and if $s = 0$ it outputs S (without learning any information on s). Define the function[6]

$$f_{\mathrm{XOR}}(S, s, i_1, \ldots, i_k) = \begin{cases} i_1, i_2, \ldots, i_k, S & \text{If } s = 0, \\ i_1, i_2, \ldots, i_k, S \oplus \{(i_1, i_2, \ldots, i_k)\} & \text{If } s = 1. \end{cases}$$

The protocol Π_{XOR}

- Input: Q_1 holds an array $S = (S_{j_1,\ldots,j_k})_{j_1 \in [N_1],\ldots,j_k \in [N_k]}$, a bit $s \in \{0, 1\}$, and $i_1 \in [N_1]$, and Q_ℓ, for every $2 \leq \ell \leq k$, holds an index $i_\ell \in [N_\ell]$. The referee holds $i_1, \ldots, i_k$.
- Output: An array $A = (A_{j_1,\ldots,j_k})_{j_1 \in [N_1],\ldots,j_k \in [N_k]}$ s.t. $A_{j_1,\ldots,j_k} = S_{j_1,\ldots,j_k}$ for every $(j_1, \ldots, j_k) \neq (i_1, \ldots, i_k)$ and $A_{i_1,\ldots,i_k} = S_{i_1,\ldots,i_k} \oplus s$.
- Common randomness: $r_{j_2,\ldots,j_k,\ell} \in \{0, 1\}$ for every $j_2 \in [N_2], \ldots, j_k \in [N_k]$ and every $\ell \in \{1, \ldots, k\}$.
- **The protocol**
  - Q_1 computes an $(N_1 - 1) \times N_2 \times \ldots \times N_k$ array A and two $1 \times N_2 \times \ldots \times N_k$ arrays A^0 and A^1.
    - $A_{j_1,\ldots,j_k} = S_{j_1,\ldots,j_k}$ for every $j_1 \in [N_1] \setminus \{i_1\}, j_2 \in [N_2], \ldots, j_k \in [N_k]$.
    - $A^0_{i_1,j_2,\ldots,j_k} = S_{i_1,j_2,\ldots,j_k} \oplus r_{j_2,\ldots,j_k,1}$ for every $j_2 \in [N_2], \ldots, j_k \in [N_k]$.
    - $A^1_{i_1,j_2,\ldots,j_k} = S_{i_1,j_2,\ldots,j_k} \oplus r_{j_2,\ldots,j_k,2} \oplus \cdots \oplus r_{j_2,\ldots,j_k,k} \oplus s$ for every $j_2 \in [N_2], \ldots, j_k \in [N_k]$.
  - Q_1 sends A, A^0, A^1.
  - Q_ℓ, for every $2 \leq \ell \leq k$, sends $r_{j_2,\ldots,j_k,1}$ for every $(j_2, \ldots, j_k) \in [N_2] \times \cdots \times [N_k]$ such that $j_\ell \neq i_\ell$, and $r_{j_2,\ldots,j_k,\ell}$ for every $(j_2, \ldots, j_k) \in [N_2] \times \cdots \times [N_k]$ such that $j_\ell = i_\ell$.
  - The referee completes A to an $N_1 \times N_2 \times \ldots \times N_k$ array as follows
    - $A_{i_1,i_2,\ldots,i_k} = A^1_{i_1,i_2,\ldots,i_k} \oplus r_{i_2,\ldots,i_k,2} \oplus \cdots \oplus r_{i_2,\ldots,i_k,k}$.
    - $A_{i_1,j_2,\ldots,j_k} = A^0_{i_1,j_2,\ldots,j_k} \oplus r_{j_2,\ldots,j_k,1}$ for every $(j_2, \ldots, j_k) \neq (i_2, \ldots, i_k)$.
  - The referee returns A.

Figure 3: The protocol Π_{XOR} for the function f_{XOR}.

We next define when a protocol for f_{XOR} is secure. This is a special case of security of private simultaneous messages (PSM) protocols [26, 31], that is, we require that for every two inputs for which f_{XOR} outputs the same value, the distribution of messages is the same. Observe that every possible output of f_{XOR} results from exactly two inputs.

Definition 5.2. *We say that a protocol for f_{XOR} is secure if for every $i_1 \in [N_1], \ldots, i_k \in [N_k]$, and every S, the distributions of messages of the protocol on inputs $S, s = 0, i_1, \ldots, i_k$ and inputs $S \oplus \{(i_1, i_2, \ldots, i_k)\}, s = 1, i_1, \ldots, i_k$ are the same.*

The protocol Π_{XOR} for f_{XOR} is described in Fig. 3. Next we present a high level description of the protocol. Server Q_1 sends to the referee three arrays: A, A^0, A^1. The array A contains all the indices for

[6] We include $i_1, \ldots, i_k$ in the output of f_{XOR} to be consistent with PSM protocols, in which the referee does not know the input.

which Q_1 knows that S and A are equal (i.e., indices $j_1, \ldots, j_k$ where $j_1 \neq i_1$, so $A_{j_1,\ldots,j_k} = S_{j_1,\ldots,j_k}$), the array A^0 enables the referee to compute $A_{i_1,j_2,\ldots,j_k}$ for all the indices for which there is at least one $j_\ell \neq i_\ell$ for some $2 \leq \ell \leq k$, and the array A^1 enables the referee to compute $A_{i_1,\ldots,i_k}$.

Lemma 5.3. *Protocol Π_{XOR} is a correct and secure protocol for f_{XOR} with message size $O(N_1 \cdot \ldots \cdot N_k)$. The degree of the message generation and output reconstruction in the protocol (as a function of the randomness and the input S) is 1 over $\mathbb{F}_2$.*

Proof. For the correctness of the protocol, observe that for every $(j_2, \ldots, j_k) \neq (i_2, \ldots, i_k)$ there is at least one $j_\ell \neq i_\ell$, and the referee can reconstruct $A_{i_1,j_2,\ldots,j_k}$. In addition, since server Q_ℓ, for every $2 \leq \ell \leq k$, sends the bit $r_{i_2,\ldots,i_k,\ell}$ to the referee (together with other bits), the referee can reconstruct $A_{i_1,\ldots,i_k}$. By the construction, $A_{i_1,\ldots,i_k} = S_{i_1,\ldots,i_k} \oplus s$ and $A_{j_1,\ldots,j_k} = S_{j_1,\ldots,j_k}$ for every $(j_1, \ldots, j_k) \neq (i_1, \ldots, i_k)$. Thus, the correctness follows.

For the security of the protocol, fix inputs $i_1, \ldots, i_k$ and S, and denote S' as Boolean array that is identical to S except in index $i_1, \ldots, i_k$, where $S'_{i_1,\ldots,i_k} = S_{i_1,\ldots,i_k} \oplus 1$. We show a bijection ϕ between the randomness of Π_{XOR} and itself such that the messages of Π_{XOR} with $S, s = 0, i_1, \ldots, i_k$ and common randomness $\vec{r}$ is the same as the inputs $S', s = 1, i_1, \ldots, i_k$ and common randomness $\vec{r}' = \phi(\vec{r})$. Since ϕ is a bijection, the security follows. Given randomness

$$\vec{r} = \left((r_{j_2,\ldots,j_k,\ell})_{j_2 \in [N_2],\ldots,j_k \in [N_k],\ell \in \{1,\ldots,k\}}\right),$$

define $\vec{r}' = \phi(\vec{r})$ as follows:

- $r'_{j_2,\ldots,j_k,1} = r_{j_2,\ldots,j_k,1}$ for every $(j_2, \ldots, j_k) \neq (i_2, \ldots, i_k)$,
- $r'_{i_2,\ldots,i_k,1} = r_{i_2,\ldots,i_k,1} \oplus 1$,
- $r'_{i_2,\ldots,i_{\ell-1},j_\ell,\ldots,j_k,\ell} = r_{i_2,\ldots,i_{\ell-1},j_\ell,\ldots,j_k,\ell} \oplus 1$ for every $\ell \in \{2, \ldots, k\}$, every $j_\ell \neq i_\ell$, and every $j_{\ell+1}, \ldots, j_k$.
- $r'_{i_2,\ldots,i_{\ell-1},j_\ell,\ldots,j_k,\ell'} = r_{i_2,\ldots,i_{\ell-1},j_\ell,\ldots,j_k,\ell'}$ for every $\ell \in \{2, \ldots, k\}$, $\ell' \in \{2, \ldots, k\} \setminus \{\ell\}$, every $j_\ell \neq i_\ell$, and every $j_{\ell+1}, \ldots, j_k$.
- $r'_{i_2,\ldots,i_k,\ell} = r_{i_2,\ldots,i_k,\ell}$ for every $\ell \in [k]$.

Notice that no server sends either $r'_{i_2,\ldots,i_k,1}$ or $r'_{i_2,\ldots,i_{\ell-1},j_\ell,\ldots,j_k,\ell}$ for $j_\ell \neq i_\ell$, so servers $Q_2, \ldots, Q_k$ send the same messages on $\vec{r}$ and $\vec{r}'$. We next prove that server Q_1 sends the same messages with $S, s = 0, i_1, \vec{r}$ and with $S', s = 1, i_1, \vec{r}'$.

- The array A does not depend on the randomness or the bit in which S and S' differ, thus, the same array A is sent in both scenarios.
- For every $(j_2, \ldots, j_k) \neq (i_2, \ldots, i_k)$, it holds that $S'_{i_1,j_2,\ldots,j_k} = S_{i_1,j_2,\ldots,j_k}$ and $r'_{j_2,\ldots,j_k,1} = r_{j_2,\ldots,j_k,1}$, thus, the same bit $A^0_{i_1,j_2,\ldots,j_k}$ is sent in both scenarios.
- For $(i_1, \ldots, i_k)$, it holds that $S'_{i_1,\ldots,i_k} = S_{i_1,\ldots,i_k} \oplus 1$ and $r'_{i_1,\ldots,i_k,1} = r_{i_1,\ldots,i_k,1} \oplus 1$, thus, the same bit $A^0_{i_1,i_2,\ldots,i_k}$ is sent in both scenarios.
- We next argue that the array A^1 sent in both scenarios is the same. Recall that in the first scenario each bit in the array is $S_{i_1,j_2,\ldots,j_k} \oplus r_{j_2,\ldots,j_k,2} \oplus \cdots \oplus r_{j_2,\ldots,j_k,k}$, and the bit in the second scenario is $S'_{i_1,j_2,\ldots,j_k} \oplus r'_{j_2,\ldots,j_k,2} \oplus \cdots \oplus r'_{j_2,\ldots,j_k,k} \oplus 1$.

– For every $(j_2,\dots,j_k) \neq (i_2,\dots,i_k)$, there is a unique ℓ such that $r'_{j_2,\dots,j_k,\ell} = r_{j_2,\dots,j_k,\ell} \oplus 1$ and $S'_{i_1,j_2,\dots,j_k} = S_{i_1,j_2,\dots,j_k}$, so

$$S'_{i_1,j_2,\dots,j_k} \oplus r'_{j_2,\dots,j_k,2} \oplus \cdots \oplus r'_{j_2,\dots,j_k,k} \oplus 1 \\ = S_{i_1,j_2,\dots,j_k} \oplus r_{j_2,\dots,j_k,2} \oplus \cdots \oplus r_{j_2,\dots,j_k,k} \oplus 0.$$

Thus, the same bit $A^1_{i_1,j_2,\dots,j_k}$ is sent in both scenarios.

– For $(i_2,\dots,i_k)$, it holds that $r'_{i_2,\dots,i_k,\ell} = r_{i_2,\dots,i_k.\ell}$ for every $\ell \in [k]$ and $S'_{i_1,i_2,\dots,i_k} = S_{i_1,i_2,\dots,i_k} \oplus 1$, so

$$S'_{i_1,i_2,\dots,i_k} \oplus r'_{i_2,\dots,i_k,2} \oplus \cdots \oplus r'_{i_2,\dots,i_k,k} \oplus 1 \\ = S_{i_1,i_2,\dots,i_k} \oplus r_{i_2,\dots,i_k,2} \oplus \cdots \oplus r_{i_2,\dots,i_k,k} \oplus 0.$$

Thus, the same bit $A^1_{i_1,\dots,i_k}$ is sent in both scenarios.

The message size in protocol Π_{XOR} is $O(N_1 \cdot N_2 \cdot \ldots \cdot N_k)$ and the degree of the protocol is 1. $\square$

The k-Server CDS Protocol. Now we present our k-server CDS protocol for the function INDEX^k_N, assuming that $k \equiv 1 \pmod 3$. The case of $k \not\equiv 1 \pmod 3$ is somewhat more messy and can be handled as done in [17].

We next present an overview of our construction. The input of the protocol is a database $D \in \{0,1\}^{N^{k-1}}$ held by Q_1 and an index $i \in [N]^{k-1}$ jointly held by $Q_2,\dots,Q_k$. The input $i \in [N]^{k-1}$ is viewed as (i_1,i_2,i_3) where $i_1,i_2,i_3 \in [N^{(k-1)/3}]$, where i_h, for $h \in \{1,2,3\}$, contains the inputs of servers $Q_{2+(h-1)(k-1)/3},\dots,Q_{1+h(k-1)/3}$. The common randomness contains three random subsets, one for each dimension, i.e., $S_1,S_2,S_3 \subseteq [N^{(k-1)/3}]$. In the protocol, we want that the referee will be able to compute $S_1 \oplus \{i_1\}$, $S_2 \oplus \{i_2\}$, and $S_3 \oplus \{i_3\}$ when $s=1$, and S_1,S_2,S_3 when $s=0$ (as in the protocol Π_2 described in Fig. 2). For this task, we use the protocol Π_{XOR}. Servers $Q_2,\dots,Q_{1+(k-1)/3}$ execute the protocol Π_{XOR} in order to generate messages that enable the referee to learn $S_1 \oplus \{i_1\}$ when $s=1$ and S_1 when $s=0$. Similarly, servers $Q_{2+(k-1)/3},\dots,Q_{1+2(k-1)/3}$ and servers $Q_{2+2(k-1)/3},\dots,Q_k$ independently execute the protocol Π_{XOR} in order to generate messages that enable the referee to learn $S_2 \oplus \{i_2\}$ when $s=1$ and S_2 when $s=0$ and $S_3 \oplus \{i_3\}$ when $s=1$ and S_3 when $s=0$, respectively. In addition, we want the referee to learn the bits $r_{1,i_1}, r_{2,i_2}, r_{3,i_3}$ as in Π_2. To achieve this goal, we define $r_{h,j,1}\dots,r_{h,j,(k-1)/3}$ for every $j \in [N^{(k-1)/3}]$ and every $h \in \{1,2,3\}$, such that $r_{h,j,1} \oplus \cdots \oplus r_{h,j,(k-1)/3} = r_{h,j}$.

Theorem 5.4. *Protocol Π_k, described in Fig. 4, is a quadratic k-server CDS protocol over $\mathbb{F}_2$ for the function* INDEX^k_N *with message size $O(N^{(k-1)/3})$.*

Proof. We prove the correctness and the security of protocol Π_k, and analyze its degree (both of the encoding and the decoding) and its message size.

Correctness. In order to prove correctness, we show that the referee gets the messages sent in Π_2. That is, we show that the k servers simulate Alice and Bob in Π_2. First, Q_1 sends the messages of Alice. We show that $Q_2,\dots,Q_k$ send the message of Bob, namely, A_1,A_2,A_3 and $r_{1,i_1}, r_{2,i_2}, r_{3,i_3}$. By the correctness of Π_{XOR} (Lemma 5.3), the referee receives $S_h \oplus i_h$ if $s=1$ and S_h if $s=0$. Next we show that the referee receives $r_{h,i_h,1},\dots,r_{h,i_h,(k-1)/3}$ for every $h \in \{1,2,3\}$. This is true since for $i_h = (i^1_h, i^2_h, \dots, i^{(k-1)/3}_h)$,

The protocol Π_k

- The secret: A bit $s \in \{0,1\}$.
- Q_1 holds a database $D \in \{0,1\}^{N^{k-1}}$, and $Q_2, \dots, Q_k$ hold $x_2, \dots, x_k \in [N]$, respectively.
- Common randomness: $S_1, S_2, S_3 \subseteq [N^{(k-1)/3}]$, $r_1, r_2 \in \{0,1\}$, $r_{h,j,1}, \dots, r_{h,j,(k-1)/3} \in \{0,1\}$ for every $h \in \{1,2,3\}$ and every $j \in [N^{(k-1)/3}]$, and the common randomness of three independent executions of protocol Π_{XOR}.
- **The protocol**
  - Let:
    * $i_h^\ell = x_{1+(h-1)(k-1)+\ell}$ for every $h \in \{1,2,3\}$ and every $1 \leq \ell \leq (k-1)/3$.
    * $r_3 = r_1 \oplus r_2$.
  - Q_1 computes $3N^{(k-1)/3}$ bits:
    * $m_{j_1}^1 = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus r_{1,j_1,1} \oplus \cdots \oplus r_{1,j_1,(k-1)/3} \oplus r_1$ for every $j_1 \in [N^{(k-1)/3}]$.
    * $m_{j_2}^2 = \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus r_{2,j_2,1} \oplus \cdots \oplus r_{2,j_2,(k-1)/3} \oplus r_2$ for every $j_2 \in [N^{(k-1)/3}]$.
    * $m_{j_3}^3 = \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,j_3} \oplus r_{3,j_3,1} \oplus \cdots \oplus r_{3,j_3,(k-1)/3} \oplus r_3$ for every $j_3 \in [N^{(k-1)/3}]$.
  - Q_1 sends $(m_{j_1}^1)_{j_1 \in [N^{(k-1)/3}]}$, $(m_{j_2}^2)_{j_2 \in [N^{(k-1)/3}]}$, $(m_{j_3}^3)_{j_3 \in [N^{(k-1)/3}]}$ to the referee.
  - $Q_{2+(h-1)(k-1)/3}, \dots, Q_{1+h(k-1)/3}$, for every $h \in \{1,2,3\}$, execute Π_{XOR} with the set S_h held by $Q_{2+(h-1)(k-1)/3}$, the secret s, and i_h^ℓ held by $Q_{1+(h-1)(k-1)/3+\ell}$. Let $m_{\mathrm{xor},1}^h, \dots, m_{\mathrm{xor},(k-1)/3}^h$ be the messages sent in this execution of Π_{XOR}.
  - Q_ℓ, for every $2 \leq \ell \leq k$:
    * Computes $h = \lfloor 3\ell/(k-1) \rfloor$ and $\alpha = \ell - 1 - (h-1)(k-1)/3$.
    * Sends $m_{\mathrm{xor},\alpha}^h$, and for every $j = (j_1, \dots, j_{(k-1)/3}) \in [N^{(k-1)/3}]$ such that $j_\alpha = i_h^\alpha$, sends $r_{h,j,\alpha}$.
  - The referee computes:
    * A_h, for every $h \in \{1,2,3\}$, from the messages $m_{\mathrm{xor},1}^h, \dots, m_{\mathrm{xor},(k-1)/3}^h$ of Π_{XOR}.
    * $r_{h,i_h} = r_{h,i_h,1} \oplus r_{h,i_h,2} \oplus \cdots \oplus r_{h,i_h,(k-1)/3}$, for every $h \in \{1,2,3\}$.
    * $m_1 = \bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3}$, $\quad m_2 = \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3}$, $m_3 = \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3}$

    and outputs

$$m_1 \oplus m_2 \oplus m_3 \oplus m_{i_1}^1 \oplus r_{1,i_1} \oplus m_{i_2}^2 \oplus r_{2,i_2} \oplus m_{i_3}^3 \oplus r_{3,i_3}. \tag{9}$$

Figure 4: A quadratic k-server CDS protocol Π_k for the function INDEX_N^k.

for every $\alpha \in [(k-1)/3]$ server Q_ℓ for $\ell = \alpha + 1 + (h-1)(k-1)/3$ sends $r_{h,i_h,\alpha}$, thus the referee gets all bits $r_{h,i_h,1}, \dots, r_{h,i_h,(k-1)/3}$ and can compute $r_{h,i_h} = r_{h,i_h,1} \oplus \cdots \oplus r_{h,i_h,(k-1)/3}$.

Security. In order to prove security, fix inputs D and $i = (i_1, i_2, i_3)$ such that $D_{i_1,i_2,i_3} = 0$, a message of server Q_1, i.e., $(m^1_{j_1})_{j_1\in[N^{(k-1)/3}]}$, $(m^2_{j_2})_{j_2\in[N^{(k-1)/3}]}$, $(m^3_{j_3})_{j_3\in[N^{(k-1)/3}]}$, and a message of server Q_ℓ for $\ell = \alpha+1+(h-1)(k-1)/3$ for every $h \in \{1,2,3\}$ and every $\alpha \in [(k-1)/3]$, i.e., $m^h_{\text{xor},\alpha}$ and $r_{h,j,\alpha}$ for every $j = (j_1,\dots,j_{(k-1)/3}) \in [N^{(k-1)/3}]$ such that $j_\alpha = i^\alpha_h$. Let A_h be the information that the referee can learn from the messages $m^h_{\text{xor},1},\dots,m^h_{\text{xor},(k-1)/3}$. Note that when $s=0$ then $A_h = S_h$, and when $s=1$ then $A_h = S_h \oplus \{i_h\}$, thus, we are in the same situation as in Π_2. These messages must satisfy (5). We next argue that the referee cannot learn any information about the secret given these inputs and messages, i.e., these messages have the same probability when $s=0$ and when $s=1$. That is, for every $s \in \{0,1\}$, we show that there is the same number of common random strings r.

- For every $h \in \{1,2,3\}$, define $S_h = A_h$ if $s=0$ and $S_h = A_h \oplus \{i_h\}$ if $s=1$. These S_1,S_2,S_3 are consistent with the messages of servers $Q_1,\dots,Q_k$ and are the only consistent choice. Both when $s=0$ and when $s=1$, (6) holds.
- By the security of Π_{XOR} (Lemma 5.3), the messages $m^h_{\text{xor},1},\dots,m^h_{\text{xor},(k-1)/3}$ determine the common random string of Π_{XOR} and there is the same number of such random strings for $s=0$ and $s=1$.
- The messages of Q_ℓ, for every $2+(h-1)(k-1)/3 \le \ell \le 1+h(k-1)/3$, determine $r_{h,i_h,1},\dots,r_{h,i_h,(k-1)/3}$.
- Define $r_{h,i_h} = r_{h,i_h,1} \oplus \cdots \oplus r_{h,i_h,(k-1)/3}$.
- Define
$$r_1 = m^1_{i_1} \oplus \bigoplus_{j_2\in S_2, j_3\in S_3} D_{i_1,j_2,j_3} \oplus r_{1,i_1} \tag{10}$$
and
$$r_2 = m^2_{i_2} \oplus \bigoplus_{j_1\in S_1, j_3\in S_3} D_{j_1,i_2,j_3} \oplus r_{2,i_2}. \tag{11}$$
Given the secret s, the inputs, and the messages of $Q_1,\dots,Q_k$, these values are possible and unique.
- Define $r_3 = r_1 \oplus r_2$. By (5), (6), (10), and (11), this value is possible, i.e., it satisfies
$$m^3_{i_3} = \bigoplus_{j_1\in S_1, j_2\in S_2} D_{j_1,j_2,i_3} \oplus r_{3,i_3} \oplus r_3.$$
- For every $j_1 \neq i_1$, $j_2 \neq i_2$, and $j_3 \neq i_3$, define
$$r_{1,j_1,1} \oplus \cdots \oplus r_{1,j_1,(k-1)/3} = m^1_{j_1} \oplus \bigoplus_{j_2\in S_2, j_3\in S_3} D_{i_1,j_2,j_3} \oplus r_1,$$
$$r_{2,j_2,1} \oplus \cdots \oplus r_{2,j_2,(k-1)/3} = m^2_{j_2} \oplus \bigoplus_{j_1\in S_1, j_3\in S_3} D_{j_1,i_2,j_3} \oplus r_2,$$
and
$$r_{3,j_3,1} \oplus \cdots \oplus r_{3,j_3,(k-1)/3} = m^3_{j_3} \oplus \bigoplus_{j_1\in S_1, j_2\in S_2} D_{j_1,j_2,i_3} \oplus r_3.$$
Given the secret s, the inputs, and the messages of $Q_1,\dots,Q_k$, these values are possible and unique. Note that the number of options for $r_{h,j_h,1},\dots,r_{h,j_h,(k-1)/3}$ is the same when the XOR is 1 or 0. Therefore, there is the same number of common random strings for each secret.

Degree of Encoding and Decoding. The message of server Q_1 is simply the message of Alice in Π_2 thus it is can be computed by quadratic polynomials over $\mathbb{F}_2$. The messages of the other servers are the messages in the protocol Π_{XOR}, thus can be computed by degree-1 polynomials over $\mathbb{F}_2$. The decoding is quadratic over $\mathbb{F}_2$ since it is the same function as in Π_2, but using the decoding of Π_{XOR} which is of degree-1 over $\mathbb{F}_2$.

Message Size. Server Q_1 sends $3N^{(k-1)/3}$ bits. Server Q_ℓ, for every $2 \leq \ell \leq k$, sends its message from the protocol Π_{XOR}, which is of size $O(N^{(k-1)/3})$, and additional $O(N^{(k-1)/3})$ random bits. $\square$

Corollary 5.5. *Every function $f : [N]^k \to \{0,1\}$ has a quadratic k-server CDS protocol over $\mathbb{F}_2$ with message size $O(N^{(k-1)/3})$.*

6 A Quadratic Robust CDS Protocol

In this section, we construct a quadratic k-server t-RCDS protocol, which is a CDS protocol in which the referee gets no information on the secret even if each server sends messages on multiple inputs with the same common randomness.

6.1 An Improved Analysis of the Transformation of [5]

We first show an improved analysis of the transformation of [5] from t'-RCDS protocols to t-RCDS protocols for $t' < t$; in particular, from CDS protocols (i.e., $t' = 1$) to t-RCDS protocols. In the transformation of [5], the servers independently execute $O(t^{k-1})$ copies of the underlying RCDS protocol for $f : [N]^k \to \{0,1\}$. This is done in a way that ensures that even if a server sends messages of many inputs, in at least some of the executions of the underlying RCDS protocol the referee gets messages of few inputs. We observe that the input domain in each execution of the underling RCDS is $[N/t]$ (as opposed to $[N]$), and this will reduce the total message size. In Lemma 6.2, we present the improved analysis.

We start with an overview of the ideas behind our analysis. Following the construction of the linear two-server RCDS protocol in [6] (the full version of [5]), when making a server Q_i robust, we divide the domain of inputs of Q_i using a hash function $h : [N] \to [v]$ (actually we do this for several hash functions, as will be explained later); for every $b \in [v]$, the servers execute the underlying CDS protocol where the input of Q_i is restricted to the inputs $\{x_i : h(x_i) = b\}$. We next define families of hash functions that we use in the transformation.

Definition 6.1 (Families of m'-Collision-Free Hash Functions)**.** *A set of functions $\mathcal{H}_{N,m,m',v} = \{h_d : [N] \to [v] : d \in [\ell]\}$ (where ℓ is the number of functions in the family) is a* family of m'-collision-free hash functions *if for every set $T \in \binom{N}{[m]}$ there exists at least one function $h \in \mathcal{H}_{N,m,m',v}$ for which for every $b \in [v]$ it holds that $|\{x \in T : h(x) = b\}| \leq m'$, that is, h restricted to T is at most m'-to-one. A family $\mathcal{H}_{N,m,1,v}$ is a* family of perfect hash functions *if it is a family of 1-collision-free hash functions. A family $\mathcal{H}_{N,m,m',v}$ is* output-balanced *if $|\{x \in [N] : h(x) = a\}| \leq \lceil N/v \rceil$ for every $a \in [v]$ and $h \in \mathcal{H}_{N,m,m',v}$, i.e., each h divides $[N]$ to v sets of almost the same size.*

Lemma 6.2. *Let $f : [N]^k \to \{0,1\}$ be a k-input function and t and t' be integers such that $t' < t \leq N$. Assume that there is a k-server t'-RCDS protocol $\mathcal{P}'$ for f, in which for every $N' \leq N$ and for every restriction of f with input domain $A_1 \times \ldots, \times A_k$, where $A_i \subseteq [N]$ is of size N' for $1 \leq i \leq k$, the message size is $c(N')$. In addition, assume that there is a family of an output-balanced t'-collision-free hash functions $\mathcal{H}_{N,kt,t',v} = \{h_1, \ldots, h_\ell\}$ of size ℓ. Then, there is a k-server t-RCDS protocol $\mathcal{P}$ for f with* *message size $O(\ell v^{k-1} \cdot c(N/v))$. This transformation preserves the degree of the encoding and the decoding of the underlying RCDS protocol.*

Proof. The desired protocol $\mathcal{P}$ is described in Fig. 5. This is actually the transformation of [5] with the following difference. Instead of executing $\mathcal{P}'$ with domain of inputs of size N per server, we execute it with a restriction of f with domain of inputs of size $\lceil N/v \rceil$ per server.[7] The correctness and robustness of the protocol follows from the proof of the transformation of [5].

Next, we analyze the message size. Observe that for each $h \in \mathcal{H}_{N,kt,t',v}$, each server sends messages in v^{k-1} copies of $\mathcal{P}'$, where each copy is for a restriction of f with input domain of size $\max_{a\in[v]} |S_a|$ per server, where $S_a = \{x \in [N] : h(x) = a\}$. Since $\mathcal{H}_{N,kt,t',v}$ is output balanced, it holds that $\max_{a\in[v]} |S_a| \leq \lceil N/v \rceil$ and since $|\mathcal{H}_{N,kt,t',v}| = \ell$, the message size is $O(\ell v^{k-1} \cdot c\left(\lceil N/v \rceil\right))$. We next argue that the degree of the encoding and decoding in the transformation does not change when S is the additive group of the field in the protocol $\mathcal{P}'$ (see Fig. 5). In the encoding, the servers execute a linear operation on the secret and the field elements $s_1, \dots, s_{\ell-1}$ in order to generate s_ℓ. Then, they encode each s_d by executing the underlying RCDS protocol. That is, the encoding is computed by the degree-d polynomials that compute the encoding in the underlying RCDS protocol. For the decoding, the referee first executes the decoding procedure of the underlying RCDS protocol in order to learn $s_1, \dots, s_\ell$ and then by summing them up the referee learns the secret. That is, the decoding is actually summing up the degree-d polynomials that compute the decoding of the ℓ copies of the underlying RCDS protocol. Therefore, the degree of the encoding and the decoding of the transformation are the same as for the underlying RCDS protocol. $\square$

A t-RCDS protocol

The secret: $s \in S$, where, w.l.o.g., S is a group (e.g., $S = \mathbb{Z}_m$ for some m).
The protocol

- Choose $\ell - 1$ random elements $s_1, \dots, s_{\ell-1} \in S$ and let $s_\ell = s - (s_1 + \cdots + s_{\ell-1})$ (addition is in the group).
- For every $d \in [\ell]$:
  - Let $S_a = \{x \in [N] : h_d(x) = a\}$ for every $a \in [v]$.
  - For every $a_1, \dots, a_k \in [v]$, independently execute the k-server t'-RCDS protocol $\mathcal{P}'$ for the restriction of f to $S_{a_1} \times \cdots \times S_{a_k}$ with the secret s_d, that is, for every $i \in [k]$, server Q_i with input x_i sends a message for the restriction of f to $S_{a_1} \times \cdots \times S_{a_{i-1}} \times S_{h_d(x_i)} \times S_{a_{i+1}} \times \cdots \times S_{a_k}$ for every $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k \in [v]$.

Figure 5: A transformation of a t'-RCDS protocol to a t-RCDS protocol for $t' < t$.

6.2 The Construction of the Quadratic t-RCDS Protocol

We next construct a quadratic k-server t-RCDS protocol. Our construction uses the improved analysis in Lemma 6.2 of the transformation of [5] for converting a t'-RCDS protocol into a t-RCDS protocol for

[7] in [5], they do not deal with restrictions of the domain of inputs since it does not improve the asymptotic message size of their protocols.

$t' < t$. Applying the transformation of [5] without our improved analysis starting from our quadratic k-server CDS protocol in Theorem 5.4 will result in a quadratic k-server t-RCDS protocol with message size $\tilde{O}(N^{(k-1)/3}t^{k-1})$. Using our improved analysis, we get better message size of $\tilde{O}(N^{(k-1)/3}t^{2(k-1)/3+1})$.

We start by quoting the following two lemmas that we use in order to instantiate Lemma 6.2. Both lemmas can be proved by a simple probabilistic argument. Their proofs can be found in [40].

Lemma 6.3. *Let N be an integer and $m \in [\sqrt{N}]$. Then, there exists an output-balanced family of perfect hash functions $\mathcal{H}_{N,m,1,m^2} = \{h_i : [N] \to [m^2] : i \in [\ell]\}$, where $\ell = 16m \ln N$. such that for every subset $T \in \binom{N}{[m]}$ there are at least $\ell/4$ functions $h \in \mathcal{H}_{N,m,1,m^2}$ for which $|h(T)| = |T|$.*[8]

Lemma 6.4. *Let N be an integer and $m \in \{15, \dots, N/2\}$. Then, there exists an output-balanced family of $\log m$-collision-free hash functions $\mathcal{H}_{N,m,\log m,2m} = \{h_i : [N] \to [2m] : i \in [\ell]\}$, where $\ell = 16m \ln N$. such that for every subset $T \in \binom{N}{[m]}$ there are at least $\ell/4$ functions $h \in \mathcal{H}_{N,m,\log m,2m}$ such that for every $b \in [2m]$ it holds that $|\{a \in T : h(a) = b\}| < \log m$.*

Theorem 6.5. *Let $t < \min\left\{N/2k, 2^{\sqrt{N}/k}\right\}$. Then, there is a quadratic k-server t-RCDS protocol over $\mathbb{F}_2$ with message size*

$$O(N^{(k-1)/3}t^{2(k-1)/3+1} \cdot k^{2k} \cdot \log^2 N \cdot \log^{(4k-1)/3} t) = \tilde{O}(N^{(k-1)/3}t^{2(k-1)/3+1} \cdot k^{2k}).$$

Proof. Similarly to [5], we construct the protocol in two stages. In the first stage, we transform our quadratic k-server CDS protocol from Fig. 4 into a quadratic k-server $\log t$-RCDS protocol, and then, in the second stage, we transform this protocol into a quadratic k-server t-RCDS protocol.

For the first stage, we use the output-balanced family $\mathcal{H}_{N,k\log t,1,k^2\log^2 t}$ of perfect hash functions with $O(k \log t \log N)$ hash functions promised by Lemma 6.3. Applying the transformation of Lemma 6.2 with $\mathcal{H}_{N,k\log t,1,k^2\log^2 t}$ and our quadratic (non-robust) k-server CDS protocol described in Theorem 5.4 as the underlying protocol (this protocol has message size $O(N^{(k-1)/3})$) results in a quadratic k-server $\log t$-RCDS protocol, which we denote by $\mathcal{P}'$, with message size $c'(N) = O(N^{(k-1)/3} \cdot (k \log t)^{(4k-1)/3} \cdot \log N)$.

For the second stage, we apply Lemma 6.2 with the $\log t$-RCDS protocol $\mathcal{P}'$ and the output-balanced family of $(\log t)$-collision-free hash functions, denoted by $\mathcal{H}_{N,kt,\log t,2kt}$ with $O(kt \log N)$ hash functions promised by Lemma 6.4; therefore, we get message size of

$$O(kt \log N \cdot (2kt)^{k-1} \cdot c'(N/2kt)) = O(N^{(k-1)/3} t^{\frac{2(k-1)}{3}+1} \cdot k^{2k} \cdot \log^2 N \cdot \log^{\frac{4k-1}{3}} t).$$

□

7 A Quadratic Secret Sharing for General Access Structures

In this section, we use our results described in Section 5 and Section 6.2 to construct improved quadratic secret-sharing schemes. Our upper bounds are better than the best known upper bounds for linear schemes. In addition, our upper bounds imply a separation between quadratic and linear secret-sharing schemes for almost all access structures.

[8] We use the fact that there are $\ell/4$ "good" functions in Section 8 to construct two-server RCDS protocols.

A Construction for All Access Structures. Next we use our quadratic k-server RCDS protocol in the construction of general secret sharing of [8].

Theorem 7.1 (Implied by [8]). *Assume that for every function $f : [N]^k \to \{0,1\}$ there is a k-server t-RCDS protocol with message size $c(k,N,t)$, then there is a secret-sharing scheme realizing an arbitrary n-party access structure with share size*

$$\max\left\{\max_{0<\beta\leq 0.5} c(\sqrt{n}, 2^{\sqrt{n}}, 2^{\beta\sqrt{n}}),\right.$$
$$\left.\max_{0.5<\beta\leq 1} c\left(\sqrt{2n(1-\beta)}, 2^{\sqrt{2n(1-\beta)}}, 2^{\sqrt{n(1-\beta)/2}}\right)\cdot 2^{H_2(\beta)n-2(1-\beta)n}\right\}\cdot 2^{o(n)}.$$

Furthermore, the degree of sharing and reconstruction of this secret-sharing scheme is the degree of encoding and decoding, respectively, of the underlying RCDS protocol.[9]

In the construction of [8], they use a t-RCDS protocol that is robust only for some of the subsets of size t (rather than all subsets). In our construction, we can avoid the more complex definition of robustness and use a t-RCDS protocol that is robust against all subsets of size at most t.[10]

Theorem 7.2. *Every n-party access structure can be realized by a quadratic secret-sharing scheme over $\mathbb{F}_2$ with share size $2^{0.705n+o(n)}$.*

Proof. The theorem follows from Theorem 7.1 using our quadratic t-RCDS protocol with message size $\tilde{O}(N^{(k-1)/3}t^{2(k-1)/3+1}\cdot k^{2k})$ from Theorem 6.5. We get share size

$$\max\left\{\max_{0<\beta\leq 0.5} 2^{n(2\beta+1)/3}, \max_{0.5<\beta\leq 1} 2^{H_2(\beta)n-2/3(1-\beta)n}\right\}\cdot 2^{o(n)}.$$

The maximum value of this expression is at $\beta \approx 0.613512$ and it is $2^{0.705n}$. □

In comparison, Applebaum and Nir [8] construct a linear secret-sharing scheme over $\mathbb{F}_2$ with share size $2^{0.7576n+o(n)}$ and a general (non-polynomial) secret-sharing scheme with share size $2^{0.585n+o(n)}$.

A Construction for Almost All Access Structures. It was shown in [14] that almost all access structures can be realized by a general secret-sharing scheme with shares of size $2^{o(n)}$ and by a linear secret-sharing scheme with share size $2^{n/2+o(n)}$. Furthermore, it was shown in [11] that almost all access structures require share size $2^{n/2-o(n)}$ in any linear secret-sharing scheme even with 1-bit secrets over all finite fields $\mathbb{F}_q$. Following [14], we show that almost all access structures can be realized by a quadratic secret-sharing scheme with 1-bit secrets over $\mathbb{F}_2$ with share size $2^{n/3+o(n)}$, proving a separation between quadratic and linear schemes for almost all access structures.

Theorem 7.3. *Almost all access structures can be realized by a quadratic secret-sharing scheme with 1-bit secrets over $\mathbb{F}_2$ and with share size $2^{n/3+o(n)}$.*

[9] In the transformation in [5], which is used in [8], the secret is shared by Shamir's scheme over field with more than n elements, and then each share is treated as the secret in the underlying RCDS. In our construction, we use the field $\mathbb{F}_{2^{\lceil \log n\rceil}}$ and execute our quadratic RCDS protocol for every bit in the share. This will add only logarithmic multiplication factor to the share size. The addition and multiplication operations in $\mathbb{F}_{2^{\lceil \log n\rceil}}$ can be computed as operations in $\mathbb{F}_2$.

[10] If we make each server robust by an independent stage as in Theorem 4.5 in [5] then the more complex condition is required. However, if we make each server robust simultaneously, as it is done in Appendix D in [6] (the full version of [5]) and as we do in Lemma 6.2, the simpler condition is sufficient.

Proof. We say that Γ is an $[a, b]$-slice access structure if for every set of parties A it holds that if $|A| < a$, then $A \notin \Gamma$ and if $|A| > b$, then $A \in \Gamma$.

By [34], almost all access structures are $[n/2 - 1, n/2 + 2]$-slice access structure, thus it suffices to construct secret-sharing schemes for them. Let $c(k, N)$ be the message size in a quadratic k-server protocol for any function $f : [N]^k \to \{0, 1\}$. By [36], for every k there is a secret-sharing scheme for $[a, b]$-slice access structure with share size $\dfrac{c(k,N) \cdot 2^{(b-a+1)n/k} O(n) \binom{n}{a}}{\binom{n/k}{a/k}^k}$. In our case, $a = \lfloor n/2 \rfloor - 1$ and $b = \lfloor n/2 \rfloor + 2$, and by taking $k = \sqrt{n/\log n}$ we get share size $c(k, N) \cdot 2^{O(\sqrt{n \log n})}$. Using our quadratic k-server CDS protocol described in Theorem 5.4 with $c(k, N) = N^{(k-1)/3}$ and $N = \binom{n/k}{a/k} < 2^{n/k}$, the share size is $2^{n/3+o(n)}$. $\square$

8 Improved Quadratic Two-Server RCDS Protocols

In this section we construct quadratic two-server RCDS protocols that for some range of parameters are better than the protocols constructed in Section 6. We use specific properties of the quadratic two-server CDS protocol of [37] to construct these RCDS protocols (unlike the construction in Section 6 that uses the CDS protocol in a blackbox manner).

A Quadratic Two-Server $(t, 1)$-RCDS Protocol. We next construct a quadratic two-server RCDS protocol that is robust for the first server. That is, the protocol is secure when the referee receives messages of at most t inputs from the first server and a message of one input from the second server. The protocol, denoted by Π_2^{robust}, is described in Fig. 6. Next we review the ideas in the protocol. Our protocol is built on the CDS protocol Π_2 (described in Fig. 2). In protocol Π_2, the message of Alice for each input is masked with the same random bits. When the referee gets one message from Alice, this mask prevents it from learning information. However, if the referee gets messages from Alice for two inputs, the same mask is used and the referee can learn the secret. In order to overcome this vulnerable point, in Π_2^{robust}, Alice uses different random bits to mask messages of different inputs. To get good message size, we cannot use independent masks for each input; we only need the masks of every t inputs to be independent. Thus, we use t-wise independent random variables. This is achieved by having univariate polynomial Q of degree $t - 1$ in the common randomness of Alice and Bob and Alice uses the mask $Q(x)$ for the message generated for the input x. The protocol uses many polynomials over $\mathbb{F}_{2^{\lceil \log M \rceil}}$, denoted by $Q_{h,j}$ for every $h \in \{1, 2, 3\}$ and $j \in [N^{1/3}]$. Alice masks her messages with $\mathrm{LSB}(Q_{h,j}(x))$ (that is, least significant bit of the polynomial $Q_{h,j}$ evaluated at x) and Bob sends the coefficients of only the 3 polynomials that correspond to his input, namely, $Q_{1,i_1}, Q_{2,i_2,}, Q_{3,i_3}$. The security follows from a similar argument as in the protocol of Π_2 and the fact that t points determine a unique polynomial of degree $t - 1$ and less than t points give no information on the polynomial of degree t. In the protocol we consider a function $f : [M] \times [N] \to \{0, 1\}$. The message size in the protocol only depends on the logarithm of the size of the input domain of Alice (i.e., on $\log M$).

Theorem 8.1. *Protocol Π_2^{robust} described in Fig. 6 is a quadratic two-server $(t, 1)$-RCDS protocol over $\mathbb{F}_2$ for a function $f : [M] \times [N] \to \{0, 1\}$ in which the message sizes of Alice and Bob are $O(N^{1/3})$ and $O(t \log M + N^{1/3})$, respectively.*

Proof. We next prove the correctness and robustness of the protocol described in Fig. 6. Similarly to the

Protocol Π_2^{robust}

- The secret: A bit $s \in \{0,1\}$.
- Alice holds $x \in [M]$ and Bob holds $i = (i_1, i_2, i_3) \in [N]$ such that $i_1, i_2, i_3 \in [N^{1/3}]$.
- Common randomness: $S_1, S_2, S_3 \subseteq [N^{1/3}]$, $r_{1,x}, r_{2,x} \in \{0,1\}$ for every $x \in [M]$, and polynomials $Q_{1,j_1}, Q_{2,j_2}, Q_{3,j_3}$ over $\mathbb{F}_{2^{\lceil \log M \rceil}}$ of degree $t-1$ for every $j_1, j_2, j_3 \in [N^{1/3}]$.
- **The protocol**
  - Alice and the referee compute a database $D \in \{0,1\}^N$ where $D_\ell = f(x, \ell)$ for $1 \le \ell \le N$.
  - Alice computes $r_{3,x} = r_{1,x} \oplus r_{2,x}$.
  - Alice computes $3N^{1/3}$ bits:
    * $m^1_{j_1} = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus \text{LSB}(Q_{1,j_1}(x)) \oplus r_{1,x}$ for every $j_1 \in [N^{1/3}]$.
    * $m^2_{j_2} = \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,j_2,j_3} \oplus \text{LSB}(Q_{2,j_2}(x)) \oplus r_{2,x}$ for every $j_2 \in [N^{1/3}]$.
    * $m^3_{j_3} = \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,j_3} \oplus \text{LSB}(Q_{3,j_3}(x)) \oplus r_{3,x}$ for every $j_3 \in [N^{1/3}]$.
  - Alice sends $(m^1_{j_1})_{j_1 \in [N^{1/3}]}$, $(m^2_{j_2})_{j_2 \in [N^{1/3}]}$, $(m^3_{j_3})_{j_3 \in [N^{1/3}]}$ to the referee.
  - Bob computes 3 strings $A_h = (A_h[1], \dots, A_h[N^{1/3}])$ for $h \in \{1,2,3\}$ (each string of length $N^{1/3}$).
    * $A_h[j_h] = S_h[j_h]$ for every $j_h \neq i_h$.
    * $A_h[i_h] = S_h[i_h] \oplus s$ (that is, if $s = 0$ then $A_h = S_h$, otherwise $A_h = S_h \oplus \{i_h\}$).
  - Bob sends the t coefficients of $Q_{1,i_1}, Q_{2,i_2}, Q_{3,i_3}$, and A_1, A_2, A_3 to the referee.
  - The referee computes:
    $m_1 = \bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3}$, $\quad m_2 = \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3}$,
    $m_3 = \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3}$
    and outputs

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m^2_{i_2} \oplus \text{LSB}(Q_{2,i_2}(x)) \oplus m^3_{i_3} \oplus \text{LSB}(Q_{3,i_3}(x)). \tag{12}$$

Figure 6: A quadratic two-server $(t,1)$-RCDS protocol Π_2^{robust} for an arbitrary function $f : [M] \times [N] \to \{0,1\}$.

proof of Lemma 5.1, when $s = 0$ the output of the protocol (i.e., the value of the expression in (12)) is

$$m_1 \oplus m_2 \oplus m_2 \oplus m^1_{i_1} \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m^2_{i_2} \oplus \text{LSB}(Q_{2,i_2}(x)) \oplus m^3_{i_3} \oplus \text{LSB}(Q_{3,i_3}(x)) = r_1 \oplus r_2 \oplus r_3 = 0, \tag{13}$$

and when $s = 1$, the output (i.e., the value of the expression in (12)) is

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m^2_{i_2} \oplus \text{LSB}(Q_{2,i_2}(x)) \oplus m^3_{i_3} \oplus \text{LSB}(Q_{3,i_3}(x)) = D_{i_1,i_2,i_3}. \tag{14}$$

When $f(x, (i_1, i_2, i_3)) = D_{i_1,i_2,i_3} = 1$, the correctness follows directly from (13) and (14).

Next we prove the robustness of the scheme. Fix $t' \leq t$ inputs $x^1, \dots, x^{t'}$ and their corresponding databases $D^1, \dots, D^{t'}$, respectively, and $i = (i_1, i_2, i_3)$ such that $f(x^\ell, (i_1, i_2, i_3)) = D^\ell_{i_1,i_2,i_3} = 0$ for every $1 \leq \ell \leq t'$. Furthermore, fix the t' messages of Alice $(m^{1,\ell}_{j_1})_{j_1 \in [N^{1/3}]}$, $(m^{2,\ell}_{j_2})_{j_2 \in [N^{1/3}]}$, $(m^{3,\ell}_{j_3})_{j_3 \in [N^{1/3}]}$ for $1 \leq \ell \leq t'$ and the message of Bob $A_1, A_2, A_3, Q_{1,i_1}, Q_{2,i_2}, Q_{3,i_3}$ such that for every $1 \leq \ell \leq t'$

$$\begin{aligned}\bigoplus_{j_2 \in A_2, j_3 \in A_3} D^\ell_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D^\ell_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D^\ell_{j_1,j_2,i_3} \oplus m^{1,\ell}_{i_1} \\ \oplus \mathrm{LSB}(Q_{1,i_1}(x^\ell)) \oplus m^{2,\ell}_{i_2} \oplus \mathrm{LSB}(Q_{2,i_2}(x^\ell)) \oplus m^{3,\ell}_{i_3} \oplus \mathrm{LSB}(Q_{3,i_3}(x^\ell)) = 0. \quad (15)\end{aligned}$$

By (13) and (14), when $D^\ell_{i_1,i_2,i_3} = 0$ only such messages are possible (no other restrictions are made on the messages). We next argue that the referee cannot learn any information about the secret given these inputs and messages. We show that these messages have the same probability given $s = 0$ and $s = 1$. That is, we show that for every $s \in \{0, 1\}$ there is the same number of common random strings r such that Alice and Bob send these messages with the secret s. We characterize the common random strings r that are consistent with these messages and a secret s as follows:

- For $h \in \{1, 2, 3\}$, define $S_h = A_h$ if $s = 0$ and $S_h = A_h \oplus \{i_h\}$ if $s = 1$. These S_1, S_2, S_3 are consistent with the messages of Bob and s and are the only consistent choice. Both when $s = 0$ and $s = 1$, as $D^\ell_{i_1,i_2,i_3} = 0$, it holds that for every $1 \leq \ell \leq t'$

$$\begin{aligned}&\bigoplus_{j_2 \in A_2, j_3 \in A_3} D^\ell_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D^\ell_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D^\ell_{j_1,j_2,i_3} \\ &= \bigoplus_{j_2 \in S_2, j_3 \in S_3} D^\ell_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D^\ell_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D^\ell_{j_1,j_2,i_3}. \quad (16)\end{aligned}$$

  This is true since when $s = 0$, the sets A_1, A_2, A_3 are the same as S_1, S_2, S_3, and when $s = 1$, by (4), the value of the expression for every $1 \leq \ell \leq t'$ is $D^\ell_{i_1,i_2,i_3}$ which is 0.

- The message of Bob determines Q_{1,i_1}, Q_{2,i_2}, and Q_{3,i_3}.

- Define for $1 \leq \ell \leq t'$

$$r_{1,x^\ell} = m^{1,\ell}_{i_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D^\ell_{i_1,j_2,j_3} \oplus \mathrm{LSB}(Q_{1,i_1}(x^\ell)), \quad (17)$$

  and

$$r_{2,x^\ell} = m^{2,\ell}_{i_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D^\ell_{j_1,i_2,j_3} \oplus \mathrm{LSB}(Q_{2,i_2}(x^\ell)). \quad (18)$$

  Given the secret s, the inputs, and the messages of Alice and Bob, these values are possible and unique.

- Define $r_{3,x^\ell} = r_{1,x^\ell} \oplus r_{2,x^\ell}$. By (15), (16),(17), and (18), this value is possible, i.e., it satisfies

$$m^{3,\ell}_{i_3} = \bigoplus_{j_1 \in S_1, j_2 \in S_2} D^\ell_{j_1,j_2,i_3} \oplus \mathrm{LSB}(Q_{3,i_3}(x^\ell)) \oplus r_{3,x^\ell}.$$

- Let $j_1 \neq i_1, j_2 \neq i_2$, and $j_3 \neq i_3$. Furthermore, let $y_h^1, y_h^2, \ldots, y_h^{t'}$ for $h \in \{1,2,3\}$ be any elements in $\mathbb{F}_{2^{\lceil \log M \rceil}}$ s.t. for every $1 \leq \ell \leq t'$:

$$\mathrm{LSB}(y_1^\ell) = m_{j_1}^{1,\ell} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{j_1,j_2,j_3}^\ell \oplus r_{1,x^\ell},$$

$$\mathrm{LSB}(y_2^\ell) = m_{j_2,\ell}^{2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,j_2,j_3}^\ell \oplus r_{2,x^\ell},$$

and

$$\mathrm{LSB}(y_3^\ell) = m_{j_3}^{3,\ell} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,j_3}^\ell \oplus r_{3,x^\ell}.$$

Let Q_{h,j_h} for $h \in \{1,2,3\}$ be a polynomial such that $Q_{h,j_h}(x_\ell) = y_h^\ell$ for every $1 \leq \ell \leq t'$. Since $t' \leq t$ such polynomial exists and, in fact, there are exactly $|\mathbb{F}_{2^{\lceil \log M \rceil}}|^{t-t'}$ such polynomials. Given the secret s, the inputs, and the messages of Alice and Bob, the values $\mathrm{LSB}(y_h^1), \ldots, \mathrm{LSB}(y_h^{t'})$ for $h \in \{1,2,3\}$ are possible and unique. Therefore, only such $y_h^1, \ldots, y_h^{t'}$ can define the polynomial Q_{h,j_h} and thus these are the only options for $Q_{j_h}^h$. Since the polynomials are over a finite field with characteristic 2, the LSB is uniformly distributed, therefore the number of options of $y_h^1, \ldots, y_h^{t'}$ is the same for $s = 0$ and $s = 1$. Hence, we get that the number of possible polynomials of Q_{h,j_h} is the same for $s = 0$ and $s = 1$.

Recall that the common random string is uniformly distributed. Since for every pair of messages of Alice and Bob when $D_{i_1,i_2,i_3} = 0$ for every secret s has the same number of consistent random strings, these messages have the same probability when $s = 0$ and when $s = 1$ and the security follows.

The message of Bob contains coefficients of three polynomials over $\mathbb{F}_{2^{\lceil \log M \rceil}}$ of degree $t-1$. Thus, since each polynomial has t coefficients in $\mathbb{F}_{2^{\lceil \log M \rceil}}$, the size of the message of Bob is $O(t \log M + N^{1/3})$. The message of Alice contains $N^{1/3}$ bits as in Π_2.

For the degree of the protocol, observe that addition and multiplication of field element with a constant in $\mathbb{F}_{2^{\lceil \log M \rceil}}$ can be computed as degree-1 polynomials over $\mathbb{F}_2$ with the same degree (see Section 2). Therefore, $\mathrm{LSB}(Q(x))$ can be computed by degree-1 polynomials over $\mathbb{F}_2$ (since we use only addition and multiplication with constant). Hence, by the same argument in as Π_2, the degree of the encoding and decoding is 2 over $\mathbb{F}_2$. $\square$

Remark 8.2. *We construct the protocol in Fig. 6 for an arbitrary function. This is in contrast to the protocols in previous sections, where we constructed protocols for INDEX and from it we got a protocol for every function. The problem in constructing this protocol for INDEX is that there are 2^N possible databases and for each database we need to evaluate Q on a different field element, thus the polynomials should be over $\mathbb{F}_{2^N}$. Hence, the message size of Bob would be $O(tN)$ which is inefficient (compared to the trivial protocol with message size $O(N)$).*

Comparison to the linear protocols. By [17], we know that for almost all functions $f : [N]^2 \to \{0,1\}$ every linear two-server CDS protocol requires messages of size at least $\Omega(\sqrt{N})$ (and by [29] all functions $f : [N]^2 \to \{0,1\}$ have such protocol). Therefore, our protocol is more efficient than any possible linear two-server $(t,1)$-RCDS protocol (e.g., [29, 17]) for every $t < \sqrt{N}$. However, as proved in [5], the linear CDS protocol of [17], with message size $\Theta(\sqrt{N})$, is also a linear two-server $(t,1)$-RCDS protocol for every t. Thus, for $t > \sqrt{N}$ the linear RCDS protocol of [17] is better than our protocol.

A Quadratic Two-Server (t_1, t_2)-RCDS Protocol for Long Secrets. In Theorem 8.3 we construct a quadratic two-server (t_1, t_2)-robust CDS protocol for secrets of size $O(t_2 \log N \log t_2)$ (using the $(t_1, 1)$ RCDS of Theorem 8.1). The construction follows the transformation described in Lemma 6.2. However, instead of sharing the secret by an ℓ-out-of-ℓ threshold scheme (i.e., generate ℓ random bits $s_1, \dots, s_\ell$ such that $s = \oplus_{i=1}^{\ell} s_i$), we share it by a ramp scheme ([21]), following [6]. In addition, starting from a scheme that is $(t_1, 1)$-robust, we only need to immunize Bob, i.e., enable him to send messages of t_2 inputs such that the referee will not learn the secret from theses messages and t_1 messages of Alice (provided that the messages correspond to a zero-set of inputs).

Theorem 8.3. *There is a quadratic two-server (t_1, t_2)-RCDS protocol over $\mathbb{F}_2$ for any function $f : [M] \times [N] \to \{0, 1\}$, with secrets of size $O(t_2 \log N \log t_2)$ bits, such that the message size of Alice is $\tilde{O}(N^{1/3} t_2^{5/3})$ and the message size of Bob is $\tilde{O}(t_1 t_2 + N^{1/3} t_2^{2/3})$, that is, Alice and Bob send $\tilde{O}(N^{1/3} t_2^{2/3})$ and $\tilde{O}(t_1 + N^{1/3}/t_2^{1/3})$ bits per bit of secret, respectively.*

Before proving Theorem 8.3, we define ramp secret-sharing schemes.

Definition 8.4 (Ramp secret-sharing scheme [21]). *A secret-sharing scheme (as defined in Definition 2.2) is a (b, g, n)-ramp secret-sharing scheme if*

- *Every subset A of parties of size at least g can reconstruct the secret.*
- *Every subset A of parties of size at most b should learn no information about the secret.*

In contrast to Definition 2.3, there are no requirements on subsets A such that $b < |A| < g$.

Proof (of Theorem 8.3). Starting from a scheme that is $(t_1, 1)$-robust, we only need to immunize Bob, i.e., enable him to send messages of t_2 inputs such that the referee will not learn the secret from these messages and t_1 messages of Alice (provided that the messages correspond to a zero-set of inputs). In Fig. 7, we describe the transformation that immunizes Bob. As in previous protocols, we will use this transformation twice. Next we prove the correctness and the robustness of the transformation.

For the correctness of the transformation, let $x \in [M], y \in [N]$ such that $f(x, y) = 1$. For every $i \in [\ell]$, both Alice and Bob send their message in the copy of $\mathcal{P}$ with the secret $s_{i,k}$, where the inputs are restricted to $[M] \times B_{h_i(x)}$. Since $x \in [M]$ and $y \in B_{h_i(x)}$, the referee can reconstruct $s_{i,k}$ from the messages in this copy of $\mathcal{P}$ for inputs x and y for every $i \in [\ell]$. Hence, by the correctness of Π_{ramp}, the referee reconstructs the secret s.

For the robustness, we assume that $\mathcal{P}$ is a (t_1, t'_2)-RCDS protocol and prove that the resulting protocol is a (t_1, t_2)-RCDS protocol. Let (Z_1, Z_2) be a zero-set of f such that $|Z_1| \leq t_1$ and $|Z_2| \leq t_2$. Using the family of hash functions in Lemma 6.3 and Lemma 6.4, there are at least $\ell/4$ hash functions $h \in \mathcal{H}_{N,t_2,t'_2,v}$ such that $h(Z_2)$ is at most t'_2-to-one. Let h_i be a t'_2-to-one hash function on Z_2. Thus, each t'_2 inputs of Z_2 are in a different subset B_j in the partition induced by h_i. Therefore, the referee gets at most t'_2 messages of Bob in each copy of $\mathcal{P}$, and since $\mathcal{P}$ is a (t_1, t'_2)-RCDS protocol, the referee cannot learn any information about s_i from any copy of $\mathcal{P}$ for the restriction of f to $[M] \times B_j$ with secret $s_{i,k}$, for every $j \in [v]$. As each copy is executed with independent randomness, the referee cannot learn any information about s_i. Since this holds for at least $\ell/4$ hash functions, the referee does not get any information on at least $\ell/4$ shares of the ramp scheme, and, hence, by the security of the $(3\ell/4, \ell, \ell)$-ramp scheme, the referee cannot learn any information about the secret.

Next we construct the quadratic two-server (t_1, t_2)-RCDS protocol. Observe that we use a linear $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme over $\mathbb{F}_{2^{\lceil \log \ell \rceil}}$. This linear ramp scheme can be obtained from the

A (t_1, t_2)-RCDS protocol

- Denote by $\mathcal{P}$ an underlying 2-server (t_1, t_2')-RCDS protocol.
- Let $\mathcal{H}_{N,t_2,t_2',v} = \{h_1, \dots, h_\ell\}$ be a set of hash function.
- Let $\mathbb{F}$ be a finite field and Π_{ramp} be a $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme over $\mathbb{F}$.
- The secret: A vector $s = (s_1', \dots, s_{\ell/4}') \in \mathbb{F}^{\ell/4}$.
- **The protocol**
  - Let $s_1, \dots, s_\ell \in \mathbb{F}$ be the shares of the $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme Π_{ramp} for the secret s. Let $|s_i|$ be the size of s_i and denote $s_i = (s_{i,1}, \dots, s_{i,|s_i|})$ for every $i \in [\ell]$.
  - For every $i \in [\ell]$ do:
    * Let $B_j = \{y \in [N] : h_i(y) = j\}$ for every $j \in [v]$.
    * For every $j \in [v], k \in [|s_i|]$, independently execute protocol $\mathcal{P}$ for the restriction of f to $[M] \times B_j$ with the secret $s_{i,k}$. That is, Alice with input x sends a message for the restriction of f to $[M] \times B_j$ with secret $s_{i,k}$ for every $j \in [v]$ and $k \in [|s_i|]$, and Bob with input y sends message only for the restriction of f to $[M] \times B_{h_i(x)}$ with secret $s_{i,k}$ for every $k \in [|s_i|]$.

Figure 7: A two-server (t_1, t_2)-RCDS protocol from a two-server (t_1, t_2')-RCDS protocol for a function $f : [M] \times [N] \to \{0, 1\}$.

threshold secret-sharing scheme of Shamir by fixing the last $\ell/4$ coefficients to be the secret. The share size in this scheme is one field element, that is, the size of s_i for $i \in [\ell]$ is $\log \ell$.

Similarly to Theorem 6.5, we construct the protocol in two stages. For the first stage, we use the output-balanced family $\mathcal{H}_{N,\log t_2,1,\log^2 t_2}$ of perfect hash functions with $\ell = O(\log t_2 \log N)$ hash functions promised by Lemma 6.3. Applying the transformation in Fig. 7 with $\mathcal{H}_{N,\log t_2,1,\log^2 t_2}$ and our quadratic two-server $(t_1, 1)$-RCDS protocol of Theorem 8.1 as the underlying protocol, results in a quadratic two-server $(t_1, \log t_2)$-RCDS protocol, denoted by $\mathcal{P}$, in which the message size of Alice is $\tilde{O}(N^{1/3})$ and the message size of Bob is $\tilde{O}(N^{1/3} + t_1)$.

For the second stage, we apply the transformation of Fig. 7 with the $(t_1, \log t_2)$-RCDS protocol $\mathcal{P}$ and the output-balanced family of $(\log t_2)$-collision-free hash functions, denoted by $\mathcal{H}_{N,t_2,\log t_2,2t_2}$, with $\ell = O(t_2 \log N)$ hash functions, promised by Lemma 6.4. Therefore, since the input domain of Bob in each copy of the underlying RCDS protocol is of size N/t_2, the message size of Alice is $\tilde{O}(t_2^2(N/t_2)^{1/3}) = \tilde{O}(N^{1/3}t_2^{5/3})$ and the message size of Bob is $\tilde{O}(t_2((N/t_2)^{1/3} + t_1)) = \tilde{O}(N^{1/3}t_2^{2/3} + t_1t_2)$.

For the degree of the protocol, we use a linear $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme over a field $\mathbb{F}_{2^{\lceil \log \ell \rceil}}$. Since operations (addition and multiplication) in $\mathbb{F}_{2^{\lceil \log \ell \rceil}}$ can be implemented as operations in $\mathbb{F}_2$ with the same degree (see Section 2), the ramp scheme we use is over $\mathbb{F}_2$. Therefore, using our quadratic two-server $(t_1, 1)$-RCDS protocol over $\mathbb{F}_2$ of Theorem 8.1, and with a similar argument as in Lemma 6.2, the degree of the protocol is 2 for encoding and decoding. □

Comparison to Linear Protocols. The linear two-server (t_1, t_2)-RCDS protocol (which is also an (M, t_2)-RCDS protocol) with secrets of size $O(t_2 \log N \log t_2)$ of [6] requires message size of $\tilde{O}(t_2 + \sqrt{N})$ per bit of secret. Therefore, the message size per bit of secret of our protocol for both Alice and Bob is better than the linear protocol when $t_1 < \sqrt{N}$ and $t_2 < N^{1/4}$.

9 Sharing and Reconstruction for Multi-Linear Secret Sharing

Karchmer and Wigderson [33] showed that linear sharing implies linear reconstruction. Beimel [12] showed that linear reconstruction implies linear sharing one-element secrets. In this section we show that this holds also for multi-linear schemes, that is, we show that linear sharing and linear reconstruction are equivalent for multi-element secrets. Furthermore, this holds also if the linear reconstruction has a small error. Our proof generalizes the proof of [12].

9.1 From Linear Sharing to Linear Reconstruction

We start by showing that every secret-sharing scheme with linear sharing has also linear reconstruction. This generalizes the ideas of [33].

Lemma 9.1. *Let Γ be an n-party access structure and Π be a secret-sharing scheme with linear sharing realizing Γ. Then, Π is a secret-sharing scheme with linear reconstruction.*

Proof. Denote the secret by $s = (s_1, \ldots, s_\ell)$, and let $B \in \Gamma$ be an authorized set. Each coordinate of each share of the parties in B is a linear combination of the random elements and $s_1, \ldots, s_\ell$. As in [12], we can represent these linear combinations as a system of linear equations in which the variables are the random elements and $s_1, \ldots, s_\ell$. Since B is an authorized set that can reconstruct the secret, for every $i \in [\ell]$, there is only one element $s_{i,0}$ such that there exists a solution to the system in which the i-th elements of the secret equals to $s_{i,0}$. Thus, for every $i \in [\ell]$, the equation $s_i = s_{i,0}$ is a linear combination of the equations in the system, and the i-th element of the secret is a linear combination of the coordinates of the shares of the parties in B. $\square$

9.2 From Linear Reconstruction to Linear Sharing

Next, we show that for any secret-sharing scheme with linear reconstruction there is an equivalent secret-sharing scheme with linear sharing. We first prove that for any secret-sharing scheme with linear reconstruction there is a multi-target monotone span program (defined in Definition 9.2) for the dual access structure that has the same size. Then, we use a claim from [13], which shows that for any multi-target monotone span program there is a secret-sharing scheme with linear sharing and linear reconstruction for the same access structure that has the same size. We apply the same transformation again to get a secret-sharing scheme with linear sharing for the dual of the dual access structure, i.e., for the original access structure. The construction of the dual multi-target monotone span program borrows ideas from the construction of the dual span program of Fehr [25].

We start by quoting a definition from [13] of a generalization of monotone span programs, called multi-target monotone span programs. Multi-linear schemes, introduced by [19] are based on this generalization.

Definition 9.2 (Multi-Target Monotone Span Programs [13])**.** *A* multi-target monotone span program *is a quadruple $\widehat{M} = \langle \mathbb{F}, M, \delta, V \rangle$, where $\mathbb{F}$ is a finite field, M is an $a \times b$ matrix over $\mathbb{F}$, $\delta : \{1, \ldots, a\} \to P$ (where P is a set of parties) is a mapping labeling each row of M by a party, and $V = \{\mathbf{v_1}, \ldots, \mathbf{v}_\ell\}$ is a set of independent non-zero vectors in $\mathbb{F}^b$, for some $1 \leq \ell < b$, such that for every $A \subset P$ one of the following holds:*

1. *The rows of M_A span each vector in V. In this case, we say that $\widehat{M}$ accepts A.*

2. *The rows of M_A span no non-zero vector in the linear space spanned by the vectors in V. In this case, we say that $\widehat{M}$ rejects A.*

The size of $\widehat{M}$ is the number of rows of M (i.e., a). We say that $\widehat{M}$ accepts an access structure Γ where $\widehat{M}$ accepts a set A if and only if $A \in \Gamma$.

Note that we need to construct $\widehat{M}$ for which there are no subsets A such that M_A does not satisfy items 1 and 2 in Definition 9.2. By applying a linear transformation to the rows of M, the set of vectors can be changed to any set of independent non-zero vectors without changing the size of $\widehat{M}$.

Lemma 9.3 ([13])**.** *Let Γ be an n-party access structure and $\widehat{M} = \langle \mathbb{F}, M, \delta, V \rangle$ be a multi-target monotone span program of size c with ℓ target vectors in V that accepts Γ. Then, there is a secret-sharing scheme Π realizing Γ with linear sharing and linear reconstruction over $\mathbb{F}$, in which the shares contain c field elements and the secret contains ℓ field elements.*

Now, we prove that for every secret-sharing scheme with linear reconstruction realizing some access structure, there is a multi-target monotone span program accepting its dual access structure. Recall that for linear reconstruction we have a reconstruction vector for every (minimal) authorized subset A and every element of the secret s_i; this vector contains the coefficients of the linear combination of the shares of A that recover s_i. We represent a reconstruction vector for A by a vector in which the number of coordinates is the total number of elements in the shares, and it is non-zero only in coordinates correspond to shares of parties in A.

Definition 9.4 (Dual Access Structure)**.** *For an access structure $\Gamma \subseteq 2^P$, its* dual *access structure $\Gamma^\perp \subseteq 2^P$ is defined as*

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

Construction 9.5 (Dual Multi-Target Monotone Span Program)**.** *Let Π be a secret-sharing scheme with linear reconstruction realizing Γ over $\mathbb{F}$, where the secret contains ℓ field elements. Construct a multi-target monotone span program $\widehat{M^\perp} = \langle \mathbb{F}, M^\perp, \delta, V \rangle$ for Π such that:*

- *the number of rows of $M^\perp$ is the number of elements c in the shares generated by the dealer in Π,*
- *the label of a row j, i.e., $\delta(j)$, is the party that gets the j-th element in the shares for every $j \in [c]$,*
- *for every minimal authorized set $A \in \Gamma$ and every $i \in [\ell]$ there is a column $(\mathbf{r_{A,i}})^T$ in $M^\perp$, where $\mathbf{r_{A,i}}$ is a reconstruction vector of the i-th element in the secret for A in Π, and these columns are ordered according to $i \in [\ell]$ (i.e., we first have a block of columns for $i = 1$, and then a block of columns for $i = 2$, etc), and*
- *$V = \{\mathbf{v_1}, \ldots, \mathbf{v_\ell}\}$, where $\mathbf{v_i}$ consist of ℓ blocks of coordinates, the size of each of them is the number of minimal authorized sets of Γ, and all of them contain only zero's except for the i-th block, which contains only one's, for every $i \in [\ell]$.*

The multi-target monotone span program $\widehat{M^\perp}$ is called the dual *multi-target monotone span program of* Π.

Example 9.6. *Let $P = \{P_1, P_2, P_3, P_4\}$, $\Gamma = \{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}\}$, and Π be a secret-sharing scheme realizing Γ with linear reconstruction (and linear sharing) for two-bit secrets (s_1, s_2) and 4 random bits r_1, r_2, r_3, r_4 such that the share of P_1 is $(r_1, r_3 \oplus s_2)$, the share of P_2 is $(r_1 \oplus s_1, r_3, r_4)$, the share of P_3 is $(r_1, r_2, r_4 \oplus s_2)$, and the share of P_4 is $(r_2 \oplus s_1, r_4)$.*

Then, the multi-target monotone span program $\widehat{M^\perp} = \langle \mathbb{F}, M^\perp, \delta, V \rangle$ for Π will contain a 10×6 binary matrix $M^\perp$ for which the first 2 rows are labeled by P_1, the next 3 rows are labeled by P_2, the next 3 rows are labeled by P_3, and the last 2 rows are labeled by P_4. For example, the first column is the reconstruction vector of s_1 for $\{P_1, P_2\}$, i.e., $(1, 0, 1, 0, \dots, 0)^T$, and the last column is the reconstruction vector of s_2 for $\{P_3, P_4\}$, i.e., $(0, \dots, 0, 1, 0, 1)^T$.

Claim 9.7. *Let Π be a secret-sharing scheme realizing Γ with linear reconstruction over $\mathbb{F}$, where the secret contains ℓ field elements. Then, the dual multi-target monotone span program $\widehat{M^\perp}$ of Π, as defined in Construction 9.5, is a multi-target monotone span program accepting the dual access structure $\Gamma^\perp$. Moreover, the size of $\widehat{M^\perp}$ is the number of elements in the shares of Π.*

Proof. We begin by proving that for every authorized set $A \in \Gamma$, the set $B = P \setminus A$ is rejected by $\widehat{M^\perp}$. It suffices to consider only minimal authorized sets $A \in \Gamma$. For every $i \in [\ell]$, the reconstruction vector $\mathbf{r_{A,i}}$ of the i-th secret element for A in Π is a column of $M^\perp$, and has non-zero entries only in rows labeled by A, i.e., it has zero entries in all rows labeled by B. Thus, for every $i \in [\ell]$, the rows labeled by $B = P \setminus A$ cannot span $\mathbf{v_i}$, since in the column $(\mathbf{r_{A,i}})^T$, which is on the i-th block of columns of $M^\perp$, all entries labeled by B are zero. Moreover, by the structure of the target vectors, every non-zero combination of the target vectors contains non-zero entries in at least one block $i \in [\ell]$. Thus, the rows labeled by $B = P \setminus A$ cannot span any non-zero vector in the linear space spanned by the vectors in $V = \{\mathbf{v_1}, \dots, \mathbf{v_\ell}\}$.

Now, assume that $A \notin \Gamma$. We prove that the rows of $M^\perp$ labeled by $B = P \setminus A$, denoted by $M^\perp_B$, linearly span all the target vectors of V, that is, the rows of $M^\perp_B$ span the vector $\mathbf{v_i}$ for every $i \in [\ell]$. Assume by contradiction that there is a target vector $\mathbf{v_j}$ that is not spanned by the rows of $M^\perp_B$ for some $j \in [\ell]$. Then, by orthogonality arguments, there is a column vector $\mathbf{u}$ such that $\mathbf{v_j} \cdot \mathbf{u} = 1$ and $M^\perp_B \cdot \mathbf{u} = \mathbf{0}$. Denote the secret for scheme Π by $s = (s_1, \dots, s_\ell)$ and let Π_s be the shares of Π for the secret s. Thus, since the i-th block of columns contains only reconstruction vectors for s_i in Π for every $i \in [\ell]$, we have that

$$\begin{aligned} \Pi_s \cdot (M^\perp \cdot \mathbf{u}) &= (\Pi_s \cdot M^\perp) \cdot \mathbf{u} = (s_1, \dots, s_1, \dots, s_\ell, \dots, s_\ell) \cdot \mathbf{u} \\ &= \sum_{i=1}^{\ell} (s_i \cdot \mathbf{v_i}) \cdot \mathbf{u} = \sum_{i=1}^{\ell} s_i \cdot (\mathbf{v_i} \cdot \mathbf{u}). \end{aligned} \tag{19}$$

Moreover, since $M^\perp_B \cdot \mathbf{u} = \mathbf{0}$, then $M^\perp \cdot \mathbf{u}$ is non-zero only in rows labeled by A, so by the above computation the parties of A can compute $\Pi_s \cdot (M^\perp \cdot \mathbf{u}) = \sum_{i=1}^{\ell} s_i \cdot (\mathbf{v_i} \cdot \mathbf{u})$, which is a non-trivial linear combination of the elements of the secret, since $\mathbf{v_j} \cdot \mathbf{u} \neq 0$, which is a contradiction to the fact that $A \notin \Gamma$. ☐

Remark 9.8. *We say that a secret-sharing scheme Π realizing an n-party access structure Γ has an error of ε in the reconstruction if the correctness requirement is relaxed to the following one: The secret can be reconstructed by any authorized set of parties with probability at least $1 - \varepsilon$. That is, for any set $B = \{P_{i_1}, \dots, P_{i_{|B|}}\} \in \Gamma$ there exists a* reconstruction function $\mathrm{Recon}_B : S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ *such that for every secret $s \in S$,*

$$\Pr\left[\, \mathrm{Recon}_B\left(\Pi_B(s, r)\right) = s \,\right] \geq 1 - \varepsilon,$$

where the probability is over the choice of r, the random string of Π, from R with uniform distribution.

Then, as we show below, when Π is a secret-sharing scheme with linear reconstruction over $\mathbb{F}$ with an error of at most $2^{-(n+1)}$ in the reconstruction of the elements of the secret (i.e., all the elements of the secret are reconstructed correctly with probability $1 - 2^{-(n+1)}$), then Construction 9.5 together with Lemma 9.3 imply a dual secret-sharing scheme $\Pi^{\perp}$ realizing the dual access structure $\Gamma^{\perp}$ with linear sharing and linear reconstruction over $\mathbb{F}$ without an error in the reconstruction and with the same share size as the share size of Π.

This follows since if there is an error of $2^{-(n+1)}$ in the reconstruction of the secret for some authorized set, then by the union bound the probability that there is an error for some authorized set is less than $2^{-(n+1)}$ times the number of authorized sets, which is less than $1/2$. Then, following the proof of Claim 9.7, for every vector of shares Π_s for the secret s for which the reconstruction is correct for all authorized sets, equation (19) holds. Thus, with probability more than $1/2$, if the contradicting assumption was true than the set A can reconstruct a non-trivial linear combination of the elements of the secret, which is a contradiction to the fact that $A \notin \Gamma$.

Using two applications of Construction 9.5 and by Claim 9.7 and Lemma 9.3, we get the result below.

Corollary 9.9. *Let Γ be an n-party access structure and Π be a secret-sharing scheme realizing Γ with linear reconstruction over $\mathbb{F}$, in which the shares contain c field elements and the secret contains ℓ field elements. Then, there is a secret-sharing scheme realizing Γ with linear sharing and linear reconstruction over $\mathbb{F}$, in which the shares contain c field elements and the secret contains ℓ field elements.*

Proof. Given the secret-sharing scheme Π realizing Γ, we use Construction 9.5 to get a multi-target monotone span program $\widehat{M^{\perp}} = \langle \mathbb{F}, M^{\perp}, \delta, V \rangle$ of size c with ℓ target vectors in V. By Claim 9.7, $\widehat{M^{\perp}}$ accepts the dual access structure $\Gamma^{\perp}$. Then, by Lemma 9.3, there is a secret-sharing scheme $\Pi^{\perp}$ with linear sharing and linear reconstruction over $\mathbb{F}$ realizing $\Gamma^{\perp}$, in which the shares contain c field elements and the secret contains ℓ field elements.

Next, we again use Construction 9.5 on the scheme $\Pi^{\perp}$ to get a multi-target monotone span program $\widehat{M} = \langle \mathbb{F}, M, \delta, V \rangle$ of size c with ℓ target vectors in V. Again by Claim 9.7, we get that $\widehat{M}$ accepts the dual access structure of $\Gamma^{\perp}$, which is Γ, since the dual of a dual access structure is the original access structure, that is, $(\Gamma^{\perp})^{\perp} = \Gamma$. Finally, again by Lemma 9.3, we get the desired secret-sharing scheme with linear sharing and linear reconstruction over $\mathbb{F}$ realizing $\Gamma^{\perp}$, in which the shares contain c field elements and the secret contains ℓ field elements. $\square$

Combining Lemma 9.1 and Corollary 9.9, we obtain the following corollary.

Corollary 9.10. *Let Γ be an access structure and $\mathbb{F}$ be a finite field. Then, there is a secret-sharing scheme realizing Γ with linear reconstruction over $\mathbb{F}$, in which the shares contain c field elements and the secret contains ℓ field elements, if and only if there is a secret-sharing scheme realizing Γ with linear sharing (and linear reconstruction) over $\mathbb{F}$, in which the shares contain c field elements and the secret contains ℓ field elements. In that case, we say that the later scheme is a* multi-linear secret-sharing scheme.

References

[1] Aiello, W., Ishai, Y., Reingold, O.: Priced oblivious transfer: How to sell digital goods. In: EUROCRYPT 2001. LNCS, vol. 2045, pp. 118–134 (2001)

[2] Applebaum, B., Arkis, B.: On the power of amortization in secret sharing: *d*-uniform secret sharing and CDS with constant information rate. ACM Trans. Comput. Theory **12**(4), 24:1–24:21 (2020)

[3] Applebaum, B., Arkis, B., Raykov, P., Vasudevan, P.N.: Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. SIAM J. Comput. **50**(1), 32–67 (2021)

[4] Applebaum, B., Beimel, A., Farràs, O., Nir, O., Peter, N.: Secret-sharing schemes for general and uniform access structures. In: EUROCRYPT 2019. LNCS, vol. 11478, pp. 441–471 (2019)

[5] Applebaum, B., Beimel, A., Nir, O., Peter, N.: Better secret sharing via robust conditional disclosure of secrets. In: STOC 2020. pp. 280–293 (2020)

[6] Applebaum, B., Beimel, A., Nir, O., Peter, N.: Better secret sharing via robust conditional disclosure of secrets. Cryptology ePrint Archive, Report 2020/080 (2020)

[7] Applebaum, B., Holenstein, T., Mishra, M., Shayevitz, O.: The communication complexity of private simultaneous messages, revisited. In: EUROCRYPT 2018. LNCS, vol. 10401, pp. 261–286 (2018)

[8] Applebaum, B., Nir, O.: Upslices, downslices, and secret-sharing with complexity of 1.5^{n}. IACR Cryptol. ePrint Arch. **2021**, 470 (2021), `https://eprint.iacr.org/2021/470`, to appear in CRYPTO 2021.

[9] Applebaum, B., Vasudevan, P.N.: Placing conditional disclosure of secrets in the communication complexity universe. In: 10th ITCS. pp. 4:1–4:14 (2019)

[10] Attrapadung, N.: Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In: EUROCRYPT 2014. LNCS, vol. 8441, pp. 557–577 (2014)

[11] Babai, L., Gál, A., Wigderson, A.: Superpolynomial lower bounds for monotone span programs. Combinatorica **19**(3), 301–319 (1999)

[12] Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D. thesis, Technion (1996), `www.cs.bgu.ac.il/~beimel/pub.html`

[13] Beimel, A.: Secret-sharing schemes: A survey. In: IWCC 2011. LNCS, vol. 6639, pp. 11–46 (2011)

[14] Beimel, A., Farràs, O.: The share size of secret-sharing schemes for almost all access structures and graphs. In: TCC 2020. LNCS, vol. 12552, pp. 499–529 (2020)

[15] Beimel, A., Gál, A., Paterson, M.: Lower bounds for monotone span programs. Computational Complexity **6**(1), 29–45 (1997)

[16] Beimel, A., Ishai, Y.: On the power of nonlinear secret-sharing. SIAM J. on Discrete Mathematics **19**(1), 258–280 (2005)

[17] Beimel, A., Peter, N.: Optimal linear multiparty conditional disclosure of secrets protocols. In: ASIACRYPT 2018. LNCS, vol. 11274, pp. 332–362 (2018)

[18] Benaloh, J.C., Leichter, J.: Generalized secret sharing and monotone functions. In: CRYPTO '88. LNCS, vol. 403, pp. 27–35 (1988)

[19] Bertilsson, M., Ingemarsson, I.: A construction of practical secret sharing schemes using linear block codes. In: AUSCRYPT '92. LNCS, vol. 718, pp. 67–79 (1992)

[20] Blakley, G.R.: Safeguarding cryptographic keys. In: Proc. of the 1979 AFIPS National Computer Conference. vol. 48, pp. 313–317 (1979)

[21] Blakley, G.R., Meadows, C.A.: Security of ramp schemes. In: CRYPTO '84. LNCS, vol. 196, pp. 242–268 (1984)

[22] Brickell, E.F.: Some ideal secret sharing schemes. Journal of Combin. Math. and Combin. Comput. **6**, 105–113 (1989)

[23] Csirmaz, L.: The dealer's random bits in perfect secret sharing schemes. Studia Sci. Math. Hungar. **32**(3–4), 429–437 (1996)

[24] Csirmaz, L.: The size of a share must be large. J. of Cryptology **10**(4), 223–231 (1997)

[25] Fehr, S.: Efficient construction of the dual span program (1999), manuscript

[26] Feige, U., Kilian, J., Naor, M.: A minimal model for secure computation. In: 26th STOC. pp. 554–563 (1994)

[27] Gál, A.: A characterization of span program size and improved lower bounds for monotone span programs. Computational Complexity **10**(4), 277–296 (2002)

[28] Gál, A., Pudlák, P.: Monotone complexity and the rank of matrices. Inform. Process. Lett. **87**, 321–326 (2003)

[29] Gay, R., Kerenidis, I., Wee, H.: Communication complexity of conditional disclosure of secrets and attribute-based encryption. In: CRYPTO 2015. LNCS, vol. 9216, pp. 485–502 (2015)

[30] Gertner, Y., Ishai, Y., Kushilevitz, E., Malkin, T.: Protecting data privacy in private information retrieval schemes. JCSS **60**(3), 592–629 (2000)

[31] Ishai, Y., Kushilevitz, E.: Private simultaneous messages protocols with applications. In: 5th Israel Symp. on Theory of Computing and Systems. pp. 174–183 (1997)

[32] Ito, M., Saito, A., Nishizeki, T.: Secret sharing schemes realizing general access structure. In: Globecom 87. pp. 99–102 (1987), Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1), 15-20, 1993.

[33] Karchmer, M., Wigderson, A.: On span programs. In: 8th Structure in Complexity Theory. pp. 102–111 (1993)

[34] Korshunov, A.D.: On the number of monotone Boolean functions. Probl. Kibern **38**, 5–108 (1981)

[35] Larsen, K.G., Simkin, M.: Secret sharing lower bound: Either reconstruction is hard or shares are long. In: SCN 2020. LNCS, vol. 12238, pp. 566–578 (2020)

[36] Liu, T., Vaikuntanathan, V.: Breaking the circuit-size barrier in secret sharing. In: 50th STOC. pp. 699–708 (2018)

[37] Liu, T., Vaikuntanathan, V., Wee, H.: Conditional disclosure of secrets via non-linear reconstruction. In: CRYPTO 2017. LNCS, vol. 10401, pp. 758–790 (2017)

[38] Liu, T., Vaikuntanathan, V., Wee, H.: Towards breaking the exponential barrier for general secret sharing. In: EUROCRYPT 2018. LNCS, vol. 10820, pp. 567–596 (2018)

[39] Paskin-Cherniavsky, A., Radune, A.: On polynomial secret sharing schemes. In: ITC 2020. LIPIcs, vol. 163, pp. 12:1–12:21 (2020)

[40] Peter, N.: Secret-sharing schemes and conditional disclosure of secrets protocols. Thesis at Ben-Gurion Universiy (2020), https://aranne5.bgu.ac.il/others/PeterNaty19903.pdf

[41] Pitassi, T., Robere, R.: Strongly exponential lower bounds for monotone computation. In: 49th STOC. pp. 1246–1255 (2017)

[42] Pitassi, T., Robere, R.: Lifting Nullstellensatz to monotone span programs over any field. In: 50th STOC. pp. 1207–1219 (2018)

[43] Robere, R., Pitassi, T., Rossman, B., Cook, S.A.: Exponential lower bounds for monotone span programs. In: 57th FOCS. pp. 406–415 (2016)

[44] Shamir, A.: How to share a secret. Communications of the ACM **22**, 612–613 (1979)

[45] Vaikuntanathan, V., Vasudevan, P.N.: Secret sharing and statistical zero knowledge. In: ASIACRYPT 2015. pp. 656–680 (2015)

[46] Wee, H.: Dual system encryption via predicate encodings. In: TCC 2014. LNCS, vol. 8349, pp. 616–637 (2014)