Quadratic Secret Sharing and Conditional Disclosure of Secrets*

Amos Beimel
Ben Gurion University
Beere Sheva, Israel

Hussien Othman
Ben Gurion University
Beere Sheva, Israel

Naty Peter
Georgetown University
Washington, DC, USA

February 23, 2022

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.

*A preliminary version of this paper appeared in CRYPTO 2021 [20]. The results appear in the Ph.D. thesis of the second author [47]. 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.

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 [54], Blakley [24], and Ito, Saito, and Nishizeky [39] for secure storage. Nowadays, secret-sharing schemes are used in many cryptographic tasks, see, e.g., [16] 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., [39, 22, 25, 41, 23, 19, 55]. However, in the best known secret-sharing schemes for general n-party access structures, the share size is exponential in n [44, 6, 10], 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 $\Omega(n/\log n)$ [27, 26]. There is a huge gap between the upper bounds and lower bounds; 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., [25, 41, 18, 15, 13, 31, 32, 51]. 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 $2^{0.5n-o(n)}$ [13] and there exists explicit n-party access structures requiring shares of size at least $2^{\Omega(n)}$ [53, 50, 51]. 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 [41] coincide with the access structures that have a small monotone span program). Currently, only few such constructions are known [19, 55].[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 [48] 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 [34]. In a k-server CDS protocol for a Boolean function $f : [N]^k \to \{0, 1\}$, there is a

[1] In [55] 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. See Section 5.2 for more details.

	Linear	Quadratic	Degree-d	Unrestricted
Lower bound	$\tilde{\Omega}(2^{n/2})$ [13]	$\tilde{\Omega}(2^{n/3})$ (this paper)	$\tilde{\Omega}(2^{n/(d+1)})$ (this paper)	$\Omega(n^2/\log n)$ [26]
Upper bound	$2^{0.7575+o(n)}$ [10]	$2^{0.705+o(n)}$ (this paper)	same as quadratic	$2^{0.585n+o(n)}$ [10]
Upper bound for almost all access structures	$2^{n/2+o(n)}$ [17]	$2^{n/3+o(n)}$ (this paper)	same as quadratic	$2^{\tilde{O}(\sqrt{n})}$ [17]

Table 1: Summary of previous and our results.

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 to a referee such that the referee, who knows the private inputs of 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 [44, 5, 6, 10] 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

We next describe our results and techniques. The main results are described in Table 1.

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 [48]), 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 [41, 15]. In Section 10, 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 5, we give evidence that such equivalence does not hold for polynomial secret-sharing schemes. We observe that the efficient statistical secret-sharing schemes in [55], for access structures that correspond to languages that have statistical zero-knowledge proofs with log-space verifiers and simulators, have degree-3 sharing. In particular, using this observation, we show that an access structure that corresponds to the problem of quadratic residuosity modulo a composite has efficient statistical secret-sharing scheme with degree-3 sharing. A standard assumption is that this problem is not in P/poly and, in particular, 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 4.9, every sequence of access structures that has efficient statistical secret-sharing schemes with polynomial reconstruction (of a constant degree) is in NC. Thus, under the standard assumption about quadratic residuosity modulo a composite 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. Generalizing a result of [43], 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 [51], 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. [45] constructed a quadratic two-server CDS protocol for any function $f : [N]^2 \to \{0,1\}$ with message size $O(N^{1/3})$. In Section 6, 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 [45] (denoted $\mathcal{P}_{\text{LVW}}$) to construct the k-server CDS protocol. Specifically, the k servers $Q_1,\dots,Q_k$ simulate the two servers in the CDS protocol $\mathcal{P}_{\text{LVW}}$, where Q_1 simulates the first server in $\mathcal{P}_{\text{LVW}}$ and servers $Q_2,\dots,Q_k$ simulate the second server in $\mathcal{P}_{\text{LVW}}$.

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 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 [6] and were used to construct secret-sharing schemes for arbitrary access structures. Furthermore, Applebaum et al. [6] 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 7, 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. [6] and Applebaum and Nir [10] showed transformations from k-server RCDS protocols to secret-sharing schemes for arbitrary access structures. In [10], they achieved a general secret-sharing scheme for arbitrary access structures with share size $2^{0.585n+o(n)}$. In Section 8.1, we plug our quadratic k-server RCDS protocol in the transformation of [10] 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 [10], that has share size $2^{0.7576n+o(n)}$.

Beimel and Farràs [17] 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 [13], this share size is tight for linear secret-sharing schemes. In Section 8.2, we construct quadratic secret-sharing schemes for almost all access structures. Plugging our quadratic k-server CDS protocol in the construction of [17], 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 9, we show how to transform the quadratic two-server CDS protocol of [45] 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 7 has message size $\tilde{O}(N^{8/9})$, however, it is robust for both servers. This transformation is non-black-box, and uses polynomials of degree t to mask messages, where the masks of every messages of t inputs are uniformly distributed. As proved in [8], using RCDS protocols constructed in a black-box manner from CDS protocols can only result in secret-sharing schemes for general access structures with share size $2^{\Omega(n/\log^2 n)}$. Non-black-box constructions of RCDS protocols may avoid these limitations.

1.2 Open Questions

Next, we mention a few open problems arising from this paper.

Lower Bounds. In Section 4, we show non-trivial lower bounds for secret-sharing schemes with degree-d reconstruction. Our lower bound in Corollary 4.3 applies only for 1-bits secrets. It is interesting to prove lower bounds also on the normalized share size when the secret contains many field elements. This may be done by improving our transformation from polynomial schemes to linear schemes described in Lemma 4.7.

Question 1.1. *Prove non-trivial lower bounds on the normalized share size, i.e., the information ratio (which is the share size per each bit in the secret), for secret-sharing schemes with degree-d reconstruction when the secret contains many field elements.*

It is interesting to prove lower bounds also for secret-sharing schemes with degree-d sharing. This question was originally asked in [48].

Question 1.2. *Prove lower bounds on the share size of secret-sharing schemes with degree-d sharing.*

Separation between Sharing and Reconstruction. In Section 5, we show constructions with degree-3 sharing for access structures that under a plausible conjectures do not have efficient secret-sharing schemes with degree-3 reconstruction. We would like to prove such a separation without any assumptions.

Question 1.3. *Prove (unconditionally) that there is some access structure that has an efficient secret-sharing scheme with polynomial sharing but does not have efficient secret-sharing scheme with polynomial reconstruction.*

Furthermore, it is interesting to study whether degree-d reconstruction implies degree-d sharing.

Question 1.4. *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?*

Our results indicate a partial answer to this question. That is, every secret-sharing scheme with degree-d reconstruction and share size c implies a secret-sharing scheme with linear reconstruction and share size c^d and this implies a secret-sharing scheme with linear sharing (and share size c^d), and, in particular, polynomial sharing.

Upper Bounds. In sections 6 to 8, 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 protocol is smaller.

Question 1.5. *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 polynomial secret-sharing schemes for a wide class of access structures. We show, using a construction of [55], a family of access structures that can be realized by efficient statistical secret-sharing schemes with degree-3 sharing. It is interesting to construct efficient schemes also for other classes of access structures.

Question 1.6. *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.*

Remark 1.7. By Remark 4.9, we could not hope to construct efficient secret-sharing schemes with degree-d reconstruction for a constant d that realize a larger class of access structures than the class that can be efficiently realized by linear secret-sharing schemes. However, we can hope to achieve this for non-constant d or for schemes with polynomial sharing and non-polynomial reconstruction. Furthermore, we can hope to construct more efficient secret-sharing scheme for (some) 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. [34]. 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 [33, 12, 57], priced oblivious transfer [2], and secret-sharing schemes [44, 21, 5, 6, 17, 10].

Liu et al. [45] 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 [3] (improving on [4]) 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. [46] constructed a k-server CDS protocol (for one-bit secrets) with message size $2^{\tilde{O}(\sqrt{k \log N})}$. Beimel and Peter [21] and Liu et al. [46] constructed a linear k-server CDS protocol (for one-bit secrets) with message size $O(N^{(k-1)/2})$; by [21], this bound is optimal (up to a factor of k). When the secrets are long, i.e., secrets of size $\Theta(2^{N^k})$, Applebaum and Arkis [3] 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. [33] 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. [4], Applebaum et al. [9], and Applebaum and Vasudevan [11] proved a lower bound of $\Omega(\log N)$ on the message size of two-server CDS protocols for specific functions.

Polynomial Secret-Sharing Schemes. Paskin-Cherniavsky and Radune [48] defined 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 multiplicative 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.

2.1 Notations

We denote the logarithmic function with base 2 by $\log$ and with base e by $\ln$. We denote by $[n]$ the set $\{1,\dots,n\}$. For $\alpha \in [0,1]$, we denote the binary entropy of α by $h(\alpha)$, where $h(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ for $0 < \alpha < 1$ and $h(0) = h(1) = 0$.

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 $x \in \mathcal{X}$. The statistical distance between two distributions D_1 and D_2 over the domain $\mathcal{D}$ is defined as

$$\Delta(D_1, D_2) = 1/2 \sum_{d \in \mathcal{D}} |D_1(d) - D_2(d)|.$$

In particular, if D_1 and D_2 have disjoint supports, then $\Delta(D_1, D_2) = 1$. We denote by $\binom{[N]}{m}$ the set of all subsets of $[N]$ of size m. We denote by $\tilde{O}$ the O notation ignoring poly-logarithmic factors, that is, $f(n) = \tilde{O}(g(n))$ if there exists a constant c such that $f(n) = O(g(n)\log^c(g(n)))$.

2.2 Operations in $\mathbb{F}_{2^d}$

Let d be an integer and consider addition and multiplication over the finite field $\mathbb{F}_{2^d}$. These operations can be implemented as operations in $\mathbb{F}_2$, as we next explain. Recall that an element in $\mathbb{F}_{2^d}$ can be represented as a univariate polynomial of degree $d-1$ over $\mathbb{F}_2$, where its coefficients are in $\mathbb{F}_2$, i.e., $\sum_{k=0}^{d-1} a_k \sigma^k$ (we denote the formal variable of the polynomial by σ). We next define the operations in the field of two elements $A(\sigma) = \sum_{k=0}^{d-1} a_k \sigma^k$ and $B(\sigma) = \sum_{k=0}^{d-1} b_k \sigma^k$ (where $a_0, \dots, a_{d-1}, b_0, \dots, b_{d-1} \in \{0,1\}$) in $\mathbb{F}_{2^d}$. The sum of the two elements is represented by summing their coefficients in $\mathbb{F}_2$.

The multiplication in $\mathbb{F}_{2^d}$ is defined with respect to a fixed irreducible polynomial[2] $R(\sigma) = \sum_{k=0}^{d-1} e_k \sigma^k + \sigma^d$ (where $e_0, \dots, e_{d-1} \in \{0,1\}$) of degree d over $\mathbb{F}_2$. The multiplication of two elements is done by multiplying the two polynomials and then reducing the result modulo $R(\sigma)$.

[2] A polynomial is irreducible if it not a product in $\mathbb{F}_2$ of two polynomials of degree smaller than d.

Example 2.1. Consider the field $\mathbb{F}_{2^3}$ with the irreducible polynomial $\sigma^3+\sigma+1$. Note that $\sigma^3 \mod (\sigma^3+\sigma+1)=\sigma+1$ and $\sigma^4 \mod (\sigma^3+\sigma+1)=\sigma^2+\sigma$. For the two elements $\sigma^2+\sigma$ and σ^2 in $\mathbb{F}_{2^3}$, their sum is σ and the product is

$$\begin{aligned}(\sigma^2+\sigma)\cdot\sigma^2 \mod (\sigma^3+\sigma+1) &= \sigma^4+\sigma^3 \mod (\sigma^3+\sigma+1)\\ &= (\sigma^2+\sigma)+\sigma+1\\ &= \sigma^2+1.\end{aligned}$$

We next explicitly present the multiplication result. Let $c(\sigma)=\sum_{k=0}^{d-1} c_k\sigma^k$ (where $c_0,\dots,c_{d-1}\in\{0,1\}$) be the product the two polynomials $A(\sigma)$ and $B(\sigma)$ in $\mathbb{F}_{2^d}$. We next provide the formula for $c(\sigma)$, showing that each 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 polynomials such that $P_k(\sigma)=\sigma^k \bmod R(\sigma)$; write the explicit representation of $P_k(\sigma)$ as $P_k(\sigma)=\sum_{j=0}^{d-1} P_{k,j}\cdot\sigma^j$ for every $0\le k\le 2d-2$. E.g., $P_k(\sigma)=\sigma^k$ for $0\le k\le d-1$. Then,

$$c(\sigma)=A(\sigma)\cdot B(\sigma) \mod R(\sigma)=\sum_{k=0}^{2d-2}\left(\sum_{0\le i,j\le d-1, i+j=k} a_ib_j\right)\sigma^k \mod R(\sigma).$$

Let $e_k=\sum_{0\le i,j\le d-1,i+j=k} a_ib_j$ for $0\le k\le 2d-1$; this is a polynomial of degree 2 in the coefficients. . Then,

$$\begin{aligned}A(\sigma)\cdot B(\sigma) \mod R(\sigma) &= (\sum_{k=0}^{2d-2} e_k\sigma^k) \mod R(\sigma)\\ &= \sum_{k=0}^{2d-2} e_kP_k(\sigma)\\ &= \sum_{k=0}^{2d-2} e_k\sum_{j=0}^{d-1} P_{k,j}\sigma^j\\ &= \sum_{j=0}^{d-1}\left(\sum_{k=0}^{2d-2} e_kP_{k,j}\right)\sigma^j.\end{aligned} \tag{1}$$

This implies that the bivariate polynomial xy over $\mathbb{F}_{2^d}$ can be computed as polynomial of degree 2 over $\mathbb{F}_2$ with $2d$ variables, where the variables are the coefficients of the polynomials describing the elements in $\mathbb{F}_{2^d}$. To conclude, every polynomial over $\mathbb{F}_{2^d}$ with n variables and (total) degree h can be translated to a polynomial over $\mathbb{F}_2$ with nd variables and (total) degree h.

2.3 Secret Sharing

We next present the definition of secret-sharing schemes.

Definition 2.2 (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 2.3 (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 $T \notin \Gamma$ and for every pair of secrets $s, s' \in S$,*

$$\Pi_T(s, r) \text{ and } \Pi_T(s', r) \text{ are equally distributed},$$

where the probability distributions are over the choice of r from R with uniform distribution.

We next generalize secret-sharing schemes, by allowing some error in the correctness and requiring only statistical privacy.

Definition 2.4 ((ε, δ)-Secret-Sharing Schemes). *A secret-sharing sharing scheme Π with finite domain of secrets S is an (ε, δ)-secret-sharing scheme if the two following requirements hold:*

STATISTICAL CORRECTNESS. *The secret s can be reconstructed with high probability 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 Recon_B : $S_{i_1} \times \cdots \times S_{i_{|B|}} \to S$ *such that for every secret $s \in S$,*

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

where the probability is over the choice of r from R with uniform distribution.[3]

STATISTICAL SECURITY. *Every unauthorized set can learn nearly nothing about the secret from its shares.*

Formally, for any set $T \notin \Gamma$ and for every pair of secrets $s, s' \in S$,

$$\Delta(\Pi_T(s, r), \Pi_T(s', r)) \leq \delta,$$

where the probability distributions are over the choice of r from R with uniform distribution.

Definition 2.5 (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 $\Gamma_{t,n} = \{A \subseteq P : |A| \geq t\}$.*

[3] We can assume that the reconstruction function is deterministic up to a factor of 2 in the error.

2.4 Conditional Disclosure of Secrets

Next, we define k-server conditional disclosure of secrets (CDS) protocols, first presented in [34]. We start with an informal definition. We consider a model where k servers[4] $Q_1, \dots, Q_k$ 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 $x_1, \dots, x_k$ 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 $f(x_1, \dots, x_k) = 1$, and it cannot learn any information about the secret s if $f(x_1, \dots, x_k) = 0$.

Definition 2.6 (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 $M_1, \dots, M_k$, consists of k message computation functions* $\textsc{Enc}_1, \dots, \textsc{Enc}_k$*, where* $\textsc{Enc}_i : X_i \times S \times R \to M_i$ *for every $i \in [k]$. For an input $x = (x_1, \dots, x_k) \in X_1 \times \cdots \times X_k$, secret $s \in S$, and randomness $r \in R$, we let* $\textsc{Enc}(x, s, r) = (\textsc{Enc}_1(x_1, s, r), \dots, \textsc{Enc}_k(x_k, s, r))$*. We say that a protocol $\mathcal{P}$ is a CDS protocol for f if it satisfies the following properties:*

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 $x = (x_1, \dots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \dots, x_k) = 1$, every secret $s \in S$, and every common random string $r \in R$, it holds that* $\textsc{Dec}(x, \textsc{Enc}(x, s, r)) = s$.

Security. *For every input $x = (x_1, \dots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \dots, x_k) = 0$ and every pair of secrets $s, s' \in S$*

$$\textsc{Enc}(x, s, r) \textit{ and } \textsc{Enc}(x, s', r) \textit{ are equally distributed},$$

where the probability distributions are over the choice of r from R with uniform distribution.

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).*

Definition 2.7 (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 2.8 ([33])**.** *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 with input x_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 $\mathrm{INDEX}_N^k(D, x_2, \dots, x_k) = f(x_1, x_2, \dots, x_k)$.

[4] 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.

2.5 Robust Conditional Disclosure of Secrets

In the definition of CDS protocols (Definition 2.6), 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 [6], 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 2.9 (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 2.10 (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 $|Z_i| \leq t_i$ for every $i \in [k]$. Finally, for an 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 [48], 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 a multivariate monomial is the sum of the degrees 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 [48])**.** *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 $\ell, \ell_r, \ell_1, \dots, \ell_n$ 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 [48], $S = \mathbb{F}^\ell$). 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 $\ell, \ell_1, \dots, \ell_n$ such that $S \subseteq \mathbb{F}^\ell$ 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 $\text{ENC}_{i,x} : \mathbb{F}^{\ell+\ell_r} \to M_i$ can be computed by degree-d polynomials over $\mathbb{F}$, where $\text{ENC}_{i,x}(s, r) = \text{ENC}_i(x, r, s)$.*

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 $\ell, \ell_1, \ldots, \ell_k \geq 1$ such that $S \subseteq \mathbb{F}^{\ell}$ and $M_i = \mathbb{F}^{\ell_i}$ for every $1 \leq \ell \leq k$, and for every inputs $x_1, \ldots, x_k$ the function $\text{DEC}_{x_1,\ldots,x_k} : \mathbb{F}^{\ell_1+\cdots+\ell_k} \to S$ can be computed by degree-d polynomials over $\mathbb{F}$, where $\text{DEC}_{x_1,\ldots,x_k}(m_1, \ldots, m_k) = \text{DEC}(x_1, \ldots, x_k, m_1, \ldots, m_k)$.*

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 multi-linear secret-sharing scheme is a degree-1 secret-sharing scheme. A linear secret-sharing scheme is a degree-1 secret-sharing scheme with $\ell = 1$ (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 notions hold for CDS protocols.*

Secret-sharing schemes with linear sharing are equivalent to secret-sharing schemes with linear reconstruction as shown by [41, 15].

Claim 3.10 ([41, 15])**.** *Let Γ be an n-party access structure. Then,*

- *If a secret-sharing scheme Π realizing Γ has linear sharing over $\mathbb{F}$ and the secret contains one field element, then Π also has linear reconstruction.*
- *Assume there is a secret sharing realizing Γ with linear reconstruction over $\mathbb{F}$, in which the shares contain c field elements and the secret contains one field element. 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 one field element.*

In Section 10, 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 notions 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).

4 Lower Bounds for Secret Sharing with Degree-d Reconstruction

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

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

Larsen and Simkin [43] proved a lower bound on the total share size in secret-sharing schemes in which the reconstruction functions of the secret from the shares are from a given set of reconstruction functions $\mathcal{F}_{\mathrm{rec}}$. Their lower bound uses counting arguments and applies to many access structures. We generalize their proof to many access structures from a given set of access structures $\mathcal{F}_{\mathcal{A}}$. To state our lower bound, we need to recall the definition of the Vapnik–Chervonenkis (VC) dimension of a set family [56]; we provide the definition for the special case that the set family is a set of access structures (in this case, the elements are sets of parties).

Definition 4.1 (VC dimension [56]). *Let $\mathcal{F}_{\mathcal{A}}$ be a set of access structures. A sequence of sets $A_1, \dots, A_v$ is shattered by $\mathcal{F}_{\mathcal{A}}$ if for every $I \subseteq [v]$ there is an access structure $\Gamma_I \in \mathcal{F}_{\mathcal{A}}$ such that $A_i \in \Gamma_I$ if and only if $i \in I$. The VC-dimension of $\mathcal{F}_{\mathcal{A}}$ is the maximal size of a sequence shattered by $\mathcal{F}_{\mathcal{A}}$.*

Theorem 4.2 (Generalization of [43]). *Let $\mathcal{F}_{\mathrm{rec}}$ be a family of possible reconstruction functions and $\mathcal{F}_{\mathcal{A}}$ be a family of n-party access structures. Then,*

1. *For at least $2^{\mathrm{VC}(\mathcal{F}_{\mathcal{A}})} - 2^{0.5\,\mathrm{VC}(\mathcal{F}_{\mathcal{A}})}$ access structures $\Gamma \in \mathcal{F}_{\mathcal{A}}$, for any secret-sharing scheme Π realizing Γ with domain of secrets $\{0,1\}$ and reconstruction functions from $\mathcal{F}_{\mathrm{rec}}$, the sum of the share sizes of all the parties (i.e., the total share size), denoted c, satisfies*
$$c = \Omega\left(\frac{\mathrm{VC}(\mathcal{F}_{\mathcal{A}})}{\log |\mathcal{F}_{\mathrm{rec}}|}\right).$$

2. *For $\Gamma \in \mathcal{F}_{\mathcal{A}}$ let $N_{\max}(\Gamma)$ be the number of maximal unauthorized sets in Γ and $N_{\max} = \max_{\Gamma \in \mathcal{F}_{\mathcal{A}}}\{N_{\max}(\Gamma)\}$. 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 functions from $\mathcal{F}_{\mathrm{rec}}$, the sum of the share sizes of all the parties, denoted c, satisfies*
$$c = \Omega\left(\frac{\log |\mathcal{F}_{\mathcal{A}}| - 0.081 \cdot N_{\max}}{\log |\mathcal{F}_{\mathrm{rec}}|}\right).$$

For completeness, the proof of Theorem 4.2 appears in Appendix A. We obtain the following corollaries.

Corollary 4.3. *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. We prove the corollary using Item 2 of Theorem 4.2. Let $\mathcal{F}_{\mathcal{A}}$ be the family of all n-party access structures. Consider the sets of exactly $\lceil n/2 \rceil$ parties. There are $\binom{n}{\lceil n/2 \rceil}$ such sets and every subset of them can be the minimal sets of a monotone access structure. Thus, $|\mathcal{F}_{\mathcal{A}}| \geq 2^{\binom{n}{\lceil n/2 \rceil}}$. Furthermore, the maximal authorized sets of an access structure are a Sperner set (i.e., no maximal unauthorized set is contained in another set), thus, by the Sperner theorem there are at most $\binom{n}{\lceil n/2 \rceil}$ maximal unauthorized sets.

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 $|\mathbb{F}|^{(v+1)^d} = 2^{\log|\mathbb{F}| \cdot (c/\log|\mathbb{F}|+1)^d} \leq 2^{(c+\log|\mathbb{F}|)^d} \leq 2^{(2c)^d}$ polynomials of degree $\leq d$ (as the reconstruction function can choose any coefficient in $\mathbb{F}$ for every monomial and $c \geq \log|\mathbb{F}|$). If $|\mathbb{F}| > 2^{2^{n/(d+1)}}$, then the share size of every secret-sharing scheme over $\mathbb{F}$ is at least $\log|\mathbb{F}| \geq 2^{n/(d+1)}$. Thus, we only need to consider at most $2^{2^{n/(d+1)}}$ fields, and consider $\mathcal{F}_{\mathrm{rec}}$ of size at most $2^{2^{n/(d+1)}} \cdot 2^{(2c)^d}$.

Thus, by Item 2 of Theorem 4.2, for almost all access structures

$$(2^{n/(d+1)} + (2c)^d) \cdot c \geq \Omega\left(\binom{n}{\lceil n/2 \rceil} - 0.081\binom{n}{\lceil n/2 \rceil}\right) = \Omega(2^n/\sqrt{n}),$$

so $(2c)^{d+1} \geq 2^{n-o(n)}$ and $c \geq 2^{n/(d+1)-o(n)}$. □

An n-party access structure is k-uniform (for some $0 \leq k \leq n$) if all sets of size greater k are authorized, all sets of size less than k are unauthorized, and any set of size k can be either authorized or non-authorized.

Corollary 4.4. *For almost all k-uniform n-party access structures, any secret-sharing scheme realizing them over any finite field them with domain of secrets $\{0,1\}$ and degree-d reconstruction requires total share size of $\frac{2^{h(k/n)n/(d+1)}}{(\min\{k,n-k\})^{-1/(2(d+1))}}$.*

Proof. We take $\mathcal{F}_{\mathcal{A}}$ as the family of k-uniform access structures, which is of size $2^{\binom{n}{k}} = 2^{\Theta(2^{h(k/n)n}/\min\{\sqrt{k},\sqrt{n-k}\})}$. Furthermore, the VC-dimension of $\mathcal{F}_{\mathcal{A}}$ is

$$\log|\mathcal{F}_{\mathcal{A}}| = \binom{n}{k} \geq \frac{2^{h(k/n)n}}{\min\{\sqrt{k},\sqrt{n-k}\}}$$

as the sets of size exactly k are shattered by $\mathcal{F}_{\mathcal{A}}$.

As in the proof of Corollary 4.3, the number of degree-d reconstruction functions is $|\mathcal{F}_{\mathrm{rec}}| \leq 2^{2^{h(n/k)n/(d+1)}} \cdot 2^{(2c)^d}$ (as we only need to consider fields with at most $2^{2^{h(n/k)n/(d+1)}}$ elements). By Item 1 of Theorem 4.2, for almost all k-uniform access structures

$$(2^{h(n/k)n/(d+1)} + (2c)^d) \cdot c \geq \Omega\left(\frac{2^{h(k/n)n}}{\min\{\sqrt{k},\sqrt{n-k}\}}\right),$$

so $(2c)^{d+1} \geq 2^{h(k/n)n}/\min\{\sqrt{n},\sqrt{k-n}\}$ and $c \geq 2^{h(k/n)n/(d+1)}/(\min\{k,n-k\})^{-1/(2(d+1))}$. □

Corollary 4.5. *For almost all k-input functions $f : [N]^k \to \{0,1\}$, 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 basically a special case of secret-sharing schemes, where for every function $f : [N]^k \to \{0,1\}$ there is a kN-party access structure Γ_f containing all the one-inputs of the function f. Formally, the access structure Γ_f is defined as follows: The kN parties in Γ_f are partitioned into k disjoint sets $B_1, \dots, B_k$ of size N such that $B_i = \{p_{i,1}, \dots, p_{i,N}\}$. A set A is authorized in Γ_f if and only if $|A| \geq k+1$ or there exist $j_1, \dots, j_k$ such that $f(j_1, \dots, j_k) = 1$ and $p_{i,j_i} \in A$ for every $1 \leq i \leq k$. Note that every set of size less than k is unauthorized. If there is a CDS protocol for f with message size α, then there is a secret-sharing scheme realizing Γ_f with share size $\alpha + O(\log kN)$ per party and total share size $c = O(\alpha kN)$ (for $\alpha > \log kN$).[5]

We take $\mathcal{F}_\mathcal{A}$ as the family of access structures Γ_f for all functions $f : [N]^k \to \{0,1\}$, which is of size 2^{N^k}. Furthermore, $\mathrm{VC}(\mathcal{F}_\mathcal{A}) = \log|\mathcal{F}_\mathcal{A}| = N^k$ as the sets that contain exactly one element from each B_i are shattered by $\mathcal{F}_\mathcal{A}$. 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 4.3, 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 $|\mathbb{F}|^{(v+1)^d} \leq 2^{(2\alpha k)^d}$. We take $\mathcal{F}_{\mathrm{rec}}$ 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 $\mathcal{F}_{\mathrm{rec}}$ is less than $2^{N^{(k-1)/(d+1)}} \cdot 2^{(\alpha k+1)^d}$.

By Item 1 of Theorem 4.2,

$$(N^{(k-1)/(d+1)} + (2\alpha k)^d) \cdot c \geq \Omega(N^k)$$

(where $c = \alpha kN$), so $(2\alpha k)^{d+1} \geq \Omega(N^{k-1})$ and $\alpha \geq \Omega(N^{(k-1)/(d+1)}/k)$. □

Remark 4.6. Observe that Theorem 4.2 depends only on the number of possible reconstruction functions, that is, in our case it is the number of polynomials. The lower bounds in Corollaries 4.3 to 4.5 use no other properties of the polynomials.

4.2 A Transformation from Schemes with Degree-*d* Reconstruction to Linear Schemes

We describe 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 hold elements in the monomial. That is, the task of computing the values of the monomials is done by the sharing procedure instead of the reconstruction procedure.

Lemma 4.7. *Let Γ be an n-party access structure, and assume that there exists an (ε, δ)-secret-sharing scheme Π_P realizing Γ over $\mathbb{F}$ with an ℓ-element secret and degree-d reconstruction, in which the shares contain together c field elements. Then, there is a multi-linear (ε, δ)-secret-sharing scheme Π_L realizing Γ over $\mathbb{F}$ with an ℓ-element secret, 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 the scheme Π_P. Then, for every possible monomial $x_{i_1}^{\ell_1} \cdot \ldots \cdot x_{i_{d'}}^{\ell_{d'}}$ in a reconstruction of some authorized set such that

[5] To share a secret $s \in \{0,1\}$, we first share s using a $(k+1)$-out-of-kN secret sharing scheme, then we choose k uniformly distributed bits $s_1, \dots, s_k$ and compute $s_0 = s \oplus s_1 \oplus \cdots \oplus s_k$ and give s_i to every party in B_i for every $1 \leq i \leq k$. Finally we execute the CDS protocol with secret s_0 and give party $p_{i,j}$ the message of server Q_i with input j.

$2 \leq \sum_{i=1}^{d'} \ell_i \leq d$, 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 created) and shares v using a d'-out-of-d' secret-sharing scheme among the parties $P_{i_1}, \dots, P_{i_{d'}}$ (i.e., the dealer chooses d' random field elements $r_{i_1}^v, \dots, r_{i_{d'}}^v$ such that $v = r_{i_1}^v + \dots + r_{i_{d'}}^v$).[6] Note that the randomness of scheme Π_L contains the random elements of scheme Π_P and the random elements $r_{i_1}^v, \dots, r_{i_{d'-1}}^v$ for every possible monomial $x_{i_1}^{\ell_1} \cdot \dots \cdot x_{i_{d'}}^{\ell_{d'}}$ of value v such that $2 \leq \sum_{i=1}^{d'} \ell_i \leq d$ as above (the dealer computes $r_{i_{d'}}^v = x_{i_1}^{\ell_1} \cdot \dots \cdot x_{i_{d'}}^{\ell_{d'}} - r_{i_1}^v - \dots - r_{i_{d'-1}}^v$).

We next 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 10, Π_L can be converted to a secret-sharing scheme with linear sharing and reconstruction while preserving the share size.

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

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

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 of the parties as follows:

$$\sum_{x_i \in S_B} \alpha_{x_i} x_i + \sum_{\substack{x_{i_1}, \dots, x_{i_{d'}} \in S_B, d' \leq d, \\ 2 \leq \ell_1 + \dots + \ell_{d'} \leq d}} \alpha_{x_{i_1}^{\ell_1}, \dots, x_{i_{d'}}^{\ell_{d'}}} \sum_{k=1}^{d'} r_{i_k}^v$$

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

Thus, B can reconstruct the secret in Π_L whenever it can reconstruct the secret in Π_P.

We next prove the δ-security of Π_L. Let T be an unauthorized set. For every subset T' such that $T' \not\subseteq T$, the set T misses at least one random field element $r_{i_j}^v$ from any monomial for the set T', so by the properties of d'-out-of-d' secret-sharing scheme, the values of the shares of each such monomial are uniformly distributed and independent of the secret and the other elements in the shares. Thus, 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. To conclude, in the scheme Π_L, the set T learns only the information it can learn in scheme Π_P. By the δ-security of scheme Π_P, the scheme Π_L is δ-secure.

Finally, in the 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 $x_{i_1}^{\ell_1} \cdot \dots \cdot x_{i_{d'}}^{\ell_{d'}}$ 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. □

[6] 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, i.e., give to such a party the sum of the shares corresponding to its 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 4.8. *Assume that there exists an n-party access structure Γ such that in every linear secret-sharing scheme realizing Γ the share of at least one party contains at least α field elements. Then, in every secret-sharing scheme realizing Γ with degree-d reconstruction the total number of elements in the shares is $\Omega(\alpha^{1/d})$.*

Remark 4.9. 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 [19], 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 (the class that can be computed by NC^2 circuits with gates that compute multiplication and addition over a field); for small enough fields ($|\mathbb{F}| \leq 2^{n^c}$ for some constant c) it is contained in NC^3 (as each arithmetic operation in the field can be computed by a Boolean NC^1 circuit). Lemma 4.7 implies that the class of access structures that have a secret-sharing scheme with polynomial reconstruction (of a constant degree) and polynomial share size is also contained in Boolean NC^3.

4.3 Lower Bounds for Explicit Access Structures

We use Corollary 4.8 to prove lower bounds on the share size of secret-sharing schemes with degree-d reconstruction realizing an explicit access structure. To achieve this goal, let us recall the lower bound of Pitassi and Robere [51] on the share size of linear secret-sharing schemes.

Theorem 4.10 ([51, 52])**.** *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 $\mathbb{F}$, in any linear secret-sharing scheme realizing Γ over $\mathbb{F}$ the total number of field elements in the share of at least one party is $\Omega(2^{\beta n})$.*

The next lower bound for secret-sharing schemes with polynomial reconstruction and one-element secrets follows directly from Corollary 4.8 when using Theorem 4.10.

Corollary 4.11. *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 4.11 provides a lower bound on the information ratio of an explicit access structure even for large finite fields. Corollary 4.3 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 Separation between Polynomial Sharing and Polynomial Reconstruction

In this section we show secret-sharing schemes with degree-3 sharing. The constructions are built on previous constructions of [19, 55]. We use these constructions to separate between polynomial sharing and polynomial reconstruction (under some assumptions).

5.1 CDS with Degree-3 Encoding for the Quadratic Non-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 [19], 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 [19], 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, 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 known if efficient parallel algorithms for this problem exist. The known algorithms (e.g., [1, 28]) are of two types; the first type requires computing a modular exponentiation (for a survey see [35]) and the second requires computing the Jacobi symbol (see [14] for details). Therefore, the problem is related to modular exponentiation and computing Jacobi symbol problems, and thus according to the current state of the art, it is reasonable to assume that the problem is not in NC (see [19] 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.[7] The function f_{NQRP_p} is realized by the CDS protocol depicted in Figure 1. This protocol has perfect security, however, it has a one-sided error $1/p$ in the correctness. Repeating this protocol t times will result in a protocol with error $O(1/p^t)$.

Lemma 5.1. *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 $S = \{0,1\}$, an error in correctness of $1/p^t$, and message size $O(t \log p)$.*

Proof. In Figure 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 $f_{\mathrm{NQRP}_p}(x_1,\dots,x_k) = 1$, the secret is $s = 0$ iff the sum of the messages is in QR_p. The referee can 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 (each time with independent randomness) and get correctness with probability of $1 - 1/p^t$.

In order to prove security, we prove that for every k-tuples of messages for an input $x_1,\dots,x_k$ such that $f_{\mathrm{NQRP}_p}(x_1,\dots,x_k) = 0$, 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 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

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

A CDS Protocol for f_{NQRP_p}

- The secret: A bit $s \in \{0,1\}$.
- Server Q_i, for every $1 \leq i \leq k$, holds $x_i \in \{0,1\}$.
- Common randomness: $r, z_1, \dots, z_{k-1} \in \mathbb{F}_p$.
- **The protocol**
  - Calculate $z_k = -\sum_{j=1}^{k-1} z_j \mod p$.
  - Server Q_1 sends $m_1 = (z_1 + s \cdot 2^1 x_1 r^2 + r^2) \bmod p$.
  - Server Q_i, for every $2 \leq i \leq k$, sends $m_i = (z_i + s \cdot 2^i x_i r^2) \bmod p$.
  - If $\sum_{i=1}^{k} m_i \bmod p$ is a quadratic residue the referee outputs $s = 0$; otherwise it outputs $s = 1$.

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

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 $z_i + (2^i x_i) \cdot s r^2 \bmod p$ which is a degree-3 polynomial in the secret and the randomness (for every x_i). □

As in [19], we can translate the function $f_{\mathrm{NQRP}_p} : \{0,1\}^k \to \{0,1\}$ to an access structure Γ_{NQRP_p} with $n = 2k$ parties such that Γ_{NQRP_p} has a $(1/p^t, 0)$-secret-sharing scheme with one bit secrets, shares of size $O(tn)$, and degree-3 sharing over $\mathbb{F}_2$.

5.2 Efficient Statistical Secret Sharing with Degree-3 Sharing

The quadratic non-residue problem is an example of a problem that has a statistical zero-knowledge proof. Vaikuntanathan and Vasudevan [55] generalized the construction of [19] to all languages that have statistical zero-knowledge in which the verifier and the simulator use logarithmic space (see [55] for the definition of these zero-knowledge proofs). Following [19], they defined for every $n \in \mathbb{N}$ an access structure Γ_L^n that corresponds to a language L. Next we present the definition of Γ_L^n.

Definition 5.2 (The Access Structure Γ_L^n [55])**.** *For a language L and an integer $n \in \mathbb{N}$, the access structure Γ_L^n is an access structure over $2n$ parties $\{p_{i,b}\}_{i \in [n], b \in \{0,1\}}$ whose minimal authorized sets are:*

- $\{p_{i,0}, p_{i,1}\}$ *for every* $i \in [n]$.
- $\{p_{1,x_1}, \dots, p_{n,x_n}\}$ *for every* $x = (x_1, \dots, x_n) \in L \cap \{0,1\}^n$.

In [55], they constructed an efficient statistical secret-sharing scheme for Γ_L^n, where L is a language that has a statistical zero-knowledge proof with verifier and simulator that run in logarithmic space. In the heart of the secret-sharing scheme of [55], the sharing function applies the encoding scheme of the partial randomized encoding of [38] for a randomized function $f(x; r)$, where the random input r is uniformly distributed.

In the construction of [38] for partial randomized encoding, generalizing the construction of [37], the encoding is done by polynomials of degree-3 over $\mathbb{F}_2$ and the randomness used in the scheme is uniformly distributed. The transformations used in [55] preserve the degree of the encoding of the partial randomized encoding, i.e., the degree of the sharing is the degree of the encoding of the underlying partial randomized encoding scheme. Therefore, since the randomness is uniformly distributed both in the randomized function $f(x;r)$ and in its partial randomized encoding scheme of [38], the resulting efficient statistical secret-sharing scheme has degree-3 sharing.

Theorem 5.3 (Implicit in [55])**.** *Let L be a language that has a statistical zero-knowledge proof with a verifier and simulator that run in logarithmic space. Then, for every $n \in \mathbb{N}$, the access structure Γ_L^n can be realized by an $(n^{-\omega(1)}, n^{-\omega(1)})$-secret-sharing scheme with one bit secrets, shares of size polynomial in n, and degree-3 sharing over $\mathbb{F}_2$.*

5.3 The Separation between Polynomial Sharing and Polynomial Reconstruction

In Lemma 4.7, we showed that for any constant d, any (ε, δ)-secret-sharing scheme with degree-d reconstruction and total share size c can be transformed to a linear (ε, δ)-secret-sharing scheme in which the maximum share size is $O(c^d)$. Let $\{\Gamma_i\}_{i\in\mathbb{N}}$ be a sequence of access structures. Recall that if $\{\Gamma_i\}_{i\in\mathbb{N}}$ can be realized by a linear secret-sharing scheme with polynomial (in the number of parties) share size, then $\{\Gamma_i\}_{i\in\mathbb{N}}$ is in NC, i.e., it has a family of Boolean circuits of poly-logarithmic depth and polynomial size (see discussion in Remark 4.9). The above is true even if the security is only statistical and there is an exponentially small error in the correctness (see Claim 10.10 and Theorem 10.9). Note that we can execute the secret-sharing scheme in Theorem 5.3 $O(n)$ times and an authorized set returns the value that is reconstructed in the majority of the executions. Thus, we achieve a secret-sharing scheme with statistical security, exponentially small error in the correctness, polynomial share size, and degree-3 sharing. We obtain the following corollaries.

Corollary 5.4. *There is a sequence of access structures $\{\Gamma_n\}_{n\in\mathbb{N}}$ that can be realized by an efficient secret-sharing scheme with perfect security, exponentially small error in the correctness, and degree-3 sharing, but, for any constant d, under the assumption that $\{\mathrm{NQRP}_p\}_{p \text{ is a prime}} \notin \mathrm{NC}$, the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ cannot be realized by an efficient secret-sharing scheme with degree-d reconstruction.*

Corollary 5.5. *Under the assumption that there is a language $L \notin \mathrm{NC}$ that has statistical zero-knowledge with verifier and simulator that run in logarithmic space, there is a sequence of access structures $\{\Gamma_n\}_{n\in\mathbb{N}}$ that can be realized by an efficient secret-sharing scheme with statistical security, exponentially small error in correctness, and degree-3 sharing, but for any constant d, the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ cannot be realized by an efficient secret-sharing scheme with degree-d reconstruction.*

An example of a language that has a statistical zero-knowledge proof with verifier and simulator that run in logarithmic space is the quadratic residue modulo a composite N, denoted by QR_N. A standard assumption is that $\mathrm{QR}_N \notin \mathrm{P/poly}$ and in particular $\mathrm{QR}_N \notin \mathrm{NC}$.[8]

Corollary 5.6. *There is a sequence of access structures $\{\Gamma_n\}_{n\in\mathbb{N}}$ that can be realized by an efficient secret-sharing scheme with statistical security, exponentially small error in the correctness, and degree-3 sharing, but, for any constant d, under the (standard) assumption that $\{\mathrm{QR}_N\}_{N \text{ is a composite}} \notin \mathrm{NC}$, the sequence $\{\Gamma_n\}_{n\in\mathbb{N}}$ cannot be realized by an efficient secret-sharing scheme with degree-d reconstruction.*

[8] We only assume worst-case hardness of this problem.

6 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 [45]) and then construct a quadratic k-server CDS protocol that "simulates" the two-server CDS protocol.

6.1 A Quadratic Two-Server CDS Protocol

As a warm-up, we describe in Figure 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 [45] using a different notation (i.e., using cubes instead of polynomials).

Protocol $\mathbf{\Pi_2}$

- 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

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

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

Lemma 6.1. *Protocol Π_2, described in Figure 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 (2). 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 (2) is

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

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:

$$\begin{aligned} m_1 &= \left(\bigoplus_{j_2 \in S_2 \oplus \{i_2\}, j_3 \in S_3 \oplus \{i_3\}} D_{i_1,j_2,j_3} \right) \\ &= \left(\bigoplus_{j_2 \in S_2, j_3 \in S_3 \oplus \{i_3\}} D_{i_1,j_2,j_3} \right) \oplus \left(\bigoplus_{j_3 \in S_3 \oplus \{i_3\}} D_{i_1,i_2,j_3} \right) \\ &= \left(\bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \right) \oplus \left(\bigoplus_{j_2 \in S_2} D_{i_1,j_2,i_3} \right) \oplus \left(\bigoplus_{j_3 \in S_3} D_{i_1,i_2,j_3} \right) \oplus D_{i_1,i_2,i_3}. \end{aligned} \tag{4}$$

Similarly,

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

and

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

Therefore,

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

Thus, when $s = 1$, the value of the expression in (2) is

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3} \oplus r_1 \oplus r_2 \oplus r_3 = D_{i_1,i_2,i_3}. \tag{5}$$

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 (2). 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

$$\bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3} \oplus m^1_{i_1} \oplus r_{1,i_1} \oplus m^2_{i_2} \oplus r_{2,i_2} \oplus m^3_{i_3} \oplus r_{3,i_3} = 0 \quad (6)$$

(no other restrictions are made on the messages). By (3) and (5), 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

$$\bigoplus_{j_2 \in A_2, j_3 \in A_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D_{j_1,j_2,i_3} = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D_{j_1,j_2,i_3}. \quad (7)$$

  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 (5), 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

$$r_1 = m^1_{i_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus r_{1,i_1} \quad (8)$$

  and

$$r_2 = m^2_{i_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus r_{2,i_2}. \quad (9)$$

  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 (6), (7), (8), and (9), 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

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

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

and

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

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 $\mathbb{F}_2$. □

6.2 An Auxiliary Protocol Π_{XOR}

In protocol Π_2, Bob 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 $(S_{j_1,\dots,j_k})_{j_1 \in [N_1],\dots,j_k \in [N_k]}$, 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 $S \oplus \{(i_1, i_2, \dots, i_k)\}$ and if $s = 0$ it outputs S (without learning any information on s). Define the function[9]

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

We next define when a protocol for f_{XOR} is secure. This is a special case of security of private simultaneous messages (PSM) protocols [30, 36], 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.

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

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}.

Definition 6.2. *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 Figure 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 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 6.3. *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 $j_\ell \neq i_\ell$, so 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 $i_1, \dots, i_k$ and S, and denote S' as Boolean array that is identical to S except in index $i_1, \dots, i_k$, where $S'_{i_1,\dots,i_k} = S_{i_1,\dots,i_k} \oplus 1$. We show a bijection ϕ between the randomness of Π_{XOR} and itself such that the messages of Π_{XOR} with $S, s = 0, i_1, \dots, i_k$ and common randomness $\vec{r}$ is the same as the inputs $S', s = 1, i_1, \dots, i_k$ and common randomness $\vec{r}' = \phi(\vec{r})$. Since ϕ is a bijection, the security follows. Given randomness

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

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

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

Notice that no server sends either $r'_{i_2,\dots,i_k,1}$ or $r'_{i_2,\dots,i_{\ell-1},j_\ell,\dots,j_k,\ell}$ for $j_\ell \neq i_\ell$, so servers $Q_2, \dots, Q_k$ 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 $(j_2, \dots, j_k) \neq (i_2, \dots, i_k)$, it holds that $S'_{i_1,j_2,\dots,j_k} = S_{i_1,j_2,\dots,j_k}$ and $r'_{j_2,\dots,j_k,1} = r_{j_2,\dots,j_k,1}$, thus, the same bit $A^0_{i_1,j_2,\dots,j_k}$ is sent in both scenarios.
- For $(i_1, \dots, i_k)$, it holds that $S'_{i_1,\dots,i_k} = S_{i_1,\dots,i_k} \oplus 1$ and $r'_{i_1,\dots,i_k,1} = r_{i_1,\dots,i_k,1} \oplus 1$, thus, the same bit $A^0_{i_1,i_2,\dots,i_k}$ 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 $S_{i_1,j_2,\dots,j_k} \oplus r_{j_2,\dots,j_k,2} \oplus \dots \oplus r_{j_2,\dots,j_k,k}$, and the bit in the second scenario is $S'_{i_1,j_2,\dots,j_k} \oplus r'_{j_2,\dots,j_k,2} \oplus \dots \oplus r'_{j_2,\dots,j_k,k} \oplus 1$.
  - 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

$$\begin{aligned} &S'_{i_1,j_2,\dots,j_k} \oplus r'_{j_2,\dots,j_k,2} \oplus \dots \oplus r'_{j_2,\dots,j_k,k} \oplus 1 \\ &\qquad = S_{i_1,j_2,\dots,j_k} \oplus r_{j_2,\dots,j_k,2} \oplus \dots \oplus r_{j_2,\dots,j_k,k} \oplus 0. \end{aligned}$$

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

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

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

  Thus, the same bit $A^1_{i_1,\ldots,i_k}$ 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. □

6.3 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 [21] (see [47] for details).

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 $Q_2,\ldots,Q_k$. 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 $Q_{2+(h-1)(k-1)/3},\ldots,Q_{1+h(k-1)/3}$. 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 Figure 2). For this task, we use the protocol Π_{XOR}. Servers $Q_2,\ldots,Q_{1+(k-1)/3}$ 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 $Q_{2+(k-1)/3},\ldots,Q_{1+2(k-1)/3}$ and servers $Q_{2+2(k-1)/3},\ldots,Q_k$ 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 $r_{h,j,1}\ldots,r_{h,j,(k-1)/3}$ 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 6.4. *Protocol Π_k, described in Figure 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 $Q_2,\ldots,Q_k$ 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 6.3), the referee receives $S_h \oplus i_h$ if $s=1$ and S_h if $s=0$. Next we show that the referee receives $r_{h,i_h,1},\ldots,r_{h,i_h,(k-1)/3}$ for every $h \in \{1,2,3\}$. This is true since for $i_h = (i_h^1, i_h^2, \ldots, i_h^{(k-1)/3})$, 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 $r_{h,i_h,1},\ldots,r_{h,i_h,(k-1)/3}$ 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 $j=(j_1,\ldots,j_{(k-1)/3}) \in [N^{(k-1)/3}]$ such that $j_\alpha = i^\alpha_h$. Let A_h be the information that the referee can learn from the messages $m^h_{\text{xor},1},\ldots,m^h_{\text{xor},(k-1)/3}$. Note that when $s=0$ then $A_h = S_h$, and when $s=1$

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 $m_{\text{xor},1}^h, \dots, m_{\text{xor},(k-1)/3}^h$ 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 $m_{\text{xor},\alpha}^h$, 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 $m_{\text{xor},1}^h, \dots, m_{\text{xor},(k-1)/3}^h$ 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

$$m_1 \oplus m_2 \oplus m_3 \oplus m_{i_1}^1 \oplus r_{1,i_1} \oplus m_{i_2}^2 \oplus r_{2,i_2} \oplus m_{i_3}^3 \oplus r_{3,i_3}. \tag{10}$$

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

then $A_h = S_h \oplus \{i_h\}$, thus, we are in the same situation as in Π_2. These messages must satisfy (6). 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 $Q_1, \ldots, Q_k$ and are the only consistent choice. Both when $s = 0$ and when $s = 1$, (7) holds.
- By the security of Π_{XOR} (Lemma 6.3), the messages $m^h_{\text{xor},1}, \ldots, m^h_{\text{xor},(k-1)/3}$ 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 $2 + (h-1)(k-1)/3 \leq \ell \leq 1 + h(k-1)/3$, determine $r_{h,i_h,1}, \ldots, r_{h,i_h,(k-1)/3}$.
- Define $r_{h,i_h} = r_{h,i_h,1} \oplus \cdots \oplus r_{h,i_h,(k-1)/3}$.
- Define
$$r_1 = m^1_{i_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D_{i_1,j_2,j_3} \oplus r_{1,i_1} \tag{11}$$
and
$$r_2 = m^2_{i_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D_{j_1,i_2,j_3} \oplus r_{2,i_2}. \tag{12}$$
Given the secret s, the inputs, and the messages of $Q_1, \ldots, Q_k$, these values are possible and unique.
- Define $r_3 = r_1 \oplus r_2$. By (6), (7), (11), and (12), 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 $Q_1, \ldots, Q_k$, these values are possible and unique. Note that the number of options for $r_{h,j_h,1}, \ldots, r_{h,j_h,(k-1)/3}$ 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 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 they 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. □

Corollary 6.5. *Every function $f : [N]^k \to \{0,1\}$ has a quadratic k-server CDS protocol over $\mathbb{F}_2$ with message size $O(N^{(k-1)/3})$.*

7 A Quadratic Robust CDS Protocol

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

7.1 An Improved Analysis of the Transformation of [6]

We first show an improved analysis of the transformation of [6] 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 [6], the servers independently execute $O(t^k)$ 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. Following the construction of the linear two-server RCDS protocol in [7] (the full version of [6]), we divide the domain of inputs using a hash function $h : [N] \to [t]$ (actually we do this for several hash functions, as will be explained later); for every $a_1, \dots, a_k \in [t]$, the servers execute the underlying CDS protocol where the input of each Q_i is restricted to the inputs $\{x_i : h(x_i) = a_i\}$. 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 7.2, we present the improved analysis. We next define families of hash functions that will be used in the transformation.

Definition 7.1 (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 7.2. *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 $A_1 \times \dots, \times A_k$, 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 output-balanced t'-collision-free hash functions $\mathcal{H}_{N,kt,t',v} = \{h_1, \dots, h_\ell\}$ 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 Figure 5. This is actually the transformation of [6] 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.[10] The correctness and robustness of the protocol follows from the proof of the transformation of [6].

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

[10] In [6], they do not deal with restrictions of the domain of inputs since it does not improve the asymptotic message size of their protocols.

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 Figure 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 resulting RCDS protocol are the same as for the underlying RCDS protocol. □

A t-RCDS Protocol $\mathcal{P}$

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 S).
- 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 $t' < t$.

7.2 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 7.2 of the transformation of [6] for converting a t'-RCDS protocol into a t-RCDS protocol for $t' < t$. Applying the transformation of [6] without our improved analysis starting from our quadratic k-server CDS protocol in Theorem 6.4 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 use the following two lemmas. Both lemmas can be proved by a simple probabilistic argument. Their proofs can be found in [49].

Lemma 7.3. *Let N be an integer and $m \in [\sqrt{N}]$. Then, there exists an output-balanced family of perfect hash functions $\mathcal{H}_{N,m,1,m^2} = \{h_d : [N] \to [m^2] : d \in [\ell]\}$, 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 $|h(T)| = |T|$.*[11]

Lemma 7.4. *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 $\mathcal{H}_{N,m,\log m,2m} = \{h_d : [N] \to [2m] : d \in [\ell]\}$, where $\ell = 16m \ln N$,*

[11] We use the fact that there are $\ell/4$ "good" functions in Section 9 to construct two-server RCDS protocols.

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 7.5. *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 [6], we construct the protocol in two stages. In the first stage, we transform our quadratic k-server CDS protocol from Figure 4 to a quadratic k-server $\log kt$-RCDS protocol, and then, in the second stage, we transform this protocol to a quadratic k-server t-RCDS protocol.

For the first stage, we use the output-balanced family $\mathcal{H}_{N,k\log kt,1,k^2\log^2 kt}$ of perfect hash functions with $O(k \log kt \log N)$ hash functions promised by Lemma 7.3. Applying the transformation of Lemma 7.2 with $\mathcal{H}_{N,k\log kt,1,k^2\log^2 kt}$ and our quadratic (non-robust) k-server CDS protocol described in Theorem 6.4 as the underlying protocol (this protocol has message size $O(N^{(k-1)/3})$) results in a quadratic k-server $\log kt$-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 7.2 with the $\log kt$-RCDS protocol $\mathcal{P}'$ and the output-balanced family $\mathcal{H}_{N,kt,\log kt,2kt}$ of $(\log kt)$-collision-free hash functions with $O(kt \log N)$ hash functions promised by Lemma 7.4; therefore, we get message size of

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

□

8 A Quadratic Secret Sharing for General Access Structures

In this section, we use our results described in Section 6 and Section 7.2 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.

8.1 A Construction for All Access Structures

Next we use our quadratic k-server RCDS protocol in the construction of general secret-sharing of [10].

Theorem 8.1 (Implied by [10])**.** *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.[12]

[12] In the transformation in [6], which is used in [10], 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 $\mathbb{F}_2$.

In the construction of [10], 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.[13]

Theorem 8.2. *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 8.1 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 7.5. 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 [10] 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 $2^{0.585n+o(n)}$.

8.2 A Construction for Almost All Access Structures

It was shown in [17] 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 [13] 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 [17], we show that 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 $2^{n/3+o(n)}$, proving a separation between quadratic and linear schemes for almost all access structures.

Theorem 8.3. *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 $2^{n/3+o(n)}$.*

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 [42], 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 [44], 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 6.4 with $c(k, N) = N^{(k-1)/3}$ and $N = \binom{n/k}{a/k} < 2^{n/k}$, the share size is $2^{n/3+o(n)}$. □

[13] If we make each server robust by an independent stage as in Theorem 4.5 in [6] then the more complex condition is required. However, if we make each server robust simultaneously, as it is done in Appendix D in [7] (the full version of [6]) and as we do in Lemma 7.2, the simpler condition is sufficient.

9 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 7. We use specific properties of the quadratic two-server CDS protocol of [45] to construct these RCDS protocols (unlike the construction in Section 7 that uses the CDS protocol in a black-box manner). As proved in [8], black-box constructions of RCDS protocols from CDS protocols have limitations. The ideas used in this construction might be useful to bypass these limitations.

9.1 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 Figure 6. Next we review the ideas in the protocol. Our protocol is built on the CDS protocol Π_2 (described in Figure 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, where 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 $\text{LSB}(Q_{h,j}(x))$ (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 protocol 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 $f : [M] \times [N] \to \{0, 1\}$.

Theorem 9.1. *Protocol Π_2^{robust}, described in Figure 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 protocol Π_2^{robust} described in Figure 6. Similarly to the proof of Lemma 6.1, when $s = 0$ the output of the protocol (i.e., the value of the expression in (13)) is

$$m_1 \oplus m_2 \oplus m_2 \oplus m_{i_1}^1 \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m_{i_2}^2 \oplus \text{LSB}(Q_{2,i_2}(x)) \\ \oplus m_{i_3}^3 \oplus \text{LSB}(Q_{3,i_3}(x)) = r_{1,x} \oplus r_{2,x} \oplus r_{3,x} = 0, \quad (14)$$

and when $s = 1$, the output (i.e., the value of the expression in (13)) is

$$m_1 \oplus m_2 \oplus m_3 \oplus m_{i_1}^1 \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m_{i_2}^2 \oplus \text{LSB}(Q_{2,i_2}(x)) \\ \oplus m_{i_3}^3 \oplus \text{LSB}(Q_{3,i_3}(x)) = D_{i_1,i_2,i_3}. \quad (15)$$

When $f(x, (i_1, i_2, i_3)) = D_{i_1,i_2,i_3} = 1$, the correctness follows directly from (14) and (15).

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 $1 \leq \ell \leq N$.
  - 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 $A_h = (A_h[1], \ldots, A_h[N^{1/3}])$ 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

$$m_1 \oplus m_2 \oplus m_3 \oplus m^1_{i_1} \oplus \text{LSB}(Q_{1,i_1}(x)) \oplus m^2_{i_2} \oplus \text{LSB}(Q_{2,i_2}(x)) \oplus m^3_{i_3} \oplus \text{LSB}(Q_{3,i_3}(x)). \qquad (13)$$

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

Next we prove the robustness of the protocol. Fix $t' \leq t$ inputs $x^1, \ldots, x^{t'}$ and their corresponding databases $D^1, \ldots, D^{t'}$, 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 $1 \leq \ell \leq t'$,

$$\bigoplus_{j_2 \in A_2, j_3 \in A_3} D^\ell_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D^\ell_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D^\ell_{j_1,j_2,i_3} \oplus m^{1,\ell}_{i_1} \oplus \text{LSB}(Q_{1,i_1}(x^\ell)) \oplus m^{2,\ell}_{i_2} \oplus \text{LSB}(Q_{2,i_2}(x^\ell)) \oplus m^{3,\ell}_{i_3} \oplus \text{LSB}(Q_{3,i_3}(x^\ell)) = 0. \qquad (16)$$

By (14) and (15), 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 $1 \leq \ell \leq t'$,

$$\begin{aligned}\bigoplus_{j_2 \in A_2, j_3 \in A_3} D^{\ell}_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_3 \in A_3} D^{\ell}_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in A_1, j_2 \in A_2} D^{\ell}_{j_1,j_2,i_3} \\ = \bigoplus_{j_2 \in S_2, j_3 \in S_3} D^{\ell}_{i_1,j_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D^{\ell}_{j_1,i_2,j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D^{\ell}_{j_1,j_2,i_3}. \quad (17)\end{aligned}$$

  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 (15), the values of the expressions for every $1 \leq \ell \leq t'$ are differ by $D^{\ell}_{i_1,i_2,i_3}$, which is 0.

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

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

$$r_{1,x^{\ell}} = m^{1,\ell}_{i_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D^{\ell}_{i_1,j_2,j_3} \oplus \text{LSB}(Q_{1,i_1}(x^{\ell})), \quad (18)$$

  and

$$r_{2,x^{\ell}} = m^{2,\ell}_{i_2} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D^{\ell}_{j_1,i_2,j_3} \oplus \text{LSB}(Q_{2,i_2}(x^{\ell})). \quad (19)$$

  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 (16), (17), (18), and (19), this value is possible, i.e., it satisfies

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

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

$$\text{LSB}(y^{\ell}_1) = m^{1,\ell}_{j_1} \oplus \bigoplus_{j_2 \in S_2, j_3 \in S_3} D^{\ell}_{j_1,j_2,j_3} \oplus r_{1,x^{\ell}},$$

$$\text{LSB}(y^{\ell}_2) = m^{2}_{j_2,\ell} \oplus \bigoplus_{j_1 \in S_1, j_3 \in S_3} D^{\ell}_{j_1,j_2,j_3} \oplus r_{2,x^{\ell}},$$

  and

$$\text{LSB}(y^{\ell}_3) = m^{3,\ell}_{j_3} \oplus \bigoplus_{j_1 \in S_1, j_2 \in S_2} D^{\ell}_{j_1,j_2,j_3} \oplus r_{3,x^{\ell}}.$$

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 $\text{LSB}(y_h^1), \dots, \text{LSB}(y_h^{t'})$ for $h \in \{1,2,3\}$ are possible and unique. Therefore, only such $y_h^1, \dots, y_h^{t'}$ can define the polynomial Q_{h,j_h} and thus these are the only options for Q_{h,j_h}. Since the polynomials are over a finite field with characteristic 2, the LSB is uniformly distributed, therefore the number of options of $y_h^1, \dots, y_h^{t'}$ is the same for $s = 0$ and $s = 1$. Hence, we get that the number of possible polynomials Q_{h,j_h}, for $h \in \{1,2,3\}$, 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 $D_{i_1,i_2,i_3} = 0$, 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 elements with a constant in $\mathbb{F}_{2^{\lceil \log M \rceil}}$ can be computed as degree-1 polynomial over $\mathbb{F}_2$ with the same degree (see Section 2.2). Therefore, $\text{LSB}(Q(x))$ can be computed by degree-1 polynomials over $\mathbb{F}_2$, since we use only addition and multiplication with constants (that depend on x). Hence, by the same argument as in Π_2, the degree of the encoding and decoding is 2 over $\mathbb{F}_2$. □

Remark 9.2. We construct the protocol in Figure 6 for an arbitrary function. This is in contrast to the protocol Π_2, which is for the function INDEX, and from it we got a protocol for every function. The problem in constructing Π_2^{robust} for INDEX is that there are 2^N possible databases and for each database we need to evaluate a polynomial 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 Linear Protocols. By [21], 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 [33] 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., [33, 21]) for every $t < \sqrt{N}$. However, as proved in [6], the linear CDS protocol of [21], with message size $\Theta(\sqrt{N})$, is also a linear two-server $(N,1)$-RCDS protocol. Thus, for $t > \sqrt{N}$ the linear RCDS protocol of [21] is better than our protocol.

9.2 A Quadratic Two-Server (t_1, t_2)-RCDS Protocol for Longer Secrets

In Theorem 9.3, 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 protocol of Theorem 9.1). The construction follows the transformation that was described in Lemma 7.2. 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 secret-sharing scheme, following [7]. In addition, starting from a protocol 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).

Theorem 9.3. *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 $\tilde{O}(t_1 t_2 \log M + N^{1/3} t_2^{2/3})$, that is, Alice and Bob send $\tilde{O}(N^{1/3} t_2^{2/3})$ and $\tilde{O}(t_1 \log M + N^{1/3}/t_2^{1/3})$ bits per bit of secret, respectively.

Example 9.4. To understand the message size in the RCDS protocol of Theorem 9.3, we consider two examples for t_1, t_2. The message size of Alice and Bob in the two-server $(N^{5/12}, N^{1/8})$-RCDS protocol for $f : [N] \times [N] \to \{0, 1\}$ is $\tilde{O}(N^{13/24})$ times the length of the secret. This is more efficient than the message size in the linear RCDS protocol of [6], which requires message size of $\tilde{O}(N^{15/24})$ times the length of the secret. However, the message size of Alice and Bob in the two-server $(N^{1/2}, N^{1/4})$-RCDS protocol in Theorem 9.3 is $\tilde{O}(N^{3/4})$ times the length of the secret, and this the same as the message size in the linear RCDS protocol of [6]. For larger parameters, the linear RCDS protocol of [6] is more efficient. Namely, the message size in the protocol in Theorem 9.3 is better than the linear RCDS protocol of [6] when $t_1 < \sqrt{N}$ and $t_2 < N^{1/4}$. However, the linear RCDS protocol is an (N, t_2)-RCDS protocol.

Proof (of Theorem 9.3). Starting from a protocol 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 Figure 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.

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 functions.
- 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 $i \in [\ell]$ and $k \in [|s_i|]$ do:
    - Let $B_j = \{y' \in [N] : h_i(y') = j\}$ for every $j \in [v]$.
    - For every $j \in [v]$, 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 Bob with input y sends a message only for the restriction of f to $[M] \times B_{h_i(y)}$ with the secret $s_{i,k}$.

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

For the correctness of the transformation, let $x \in [M]$ and $y \in [N]$ 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 $[M] \times B_{h_i(y)}$; 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 can reconstruct 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 provided that the protocol is using the family of hash functions from Lemma 7.3

or Lemma 7.4. 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 guaranteed in Lemma 7.3 or Lemma 7.4, 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, at most t'_2 inputs of Z_2 are in a unique 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 $s_{i,k}$ 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, hence, by the security of the $(3\ell/4, \ell, \ell)$-ramp scheme, the referee cannot learn any information about the secret s.

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 threshold secret-sharing scheme of Shamir. The share size in this scheme is one field element, that is, the size of s_i for $i \in [\ell]$ is $\log \ell$. In the protocol in Figure 7, Alice send messages of $\ell v \log \ell$ executions of the underlying protocol $\mathcal{P}$ for a function $f : [M] \times [N/v] \to \{0, 1\}$ and Bob sends messages of $\ell \log \ell$ executions of the protocol $\mathcal{P}$. Hence, the message size of Alice and Bob in the protocol in Figure 7 is $\ell v \log \ell \cdot c_1(M, [N/v])$ and $\ell \log \ell \cdot c_2(M, [N/v])$, respectively, where $c_1(M, [N/v])$ and $c_2(M, [N/v])$ are the message size of Alice and Bob in the protocol $\mathcal{P}$, respectively.

Next we construct the quadratic two-server (t_1, t_2)-RCDS protocol. 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 7.3. Applying the transformation in Figure 7 with $\mathcal{H}_{N,\log t_2,1,\log^2 t_2}$ (here, $v = \log^2 t_2$) and our quadratic two-server $(t_1, 1)$-RCDS protocol of Theorem 9.1 as the underlying protocol $\mathcal{P}$ 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 $\ell v \log \ell \cdot O((N/v)^{1/3}) = \tilde{O}(N^{1/3})$, and the message size of Bob is $\ell \log \ell \cdot O(t_1 \log M + (N/v)^{1/3}) = \tilde{O}(t_1 \log M + N^{1/3})$.

For the second stage, we apply the transformation of Figure 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 7.4. Therefore, using the message size of Alice and Bob in $\mathcal{P}'$, the message size of Alice is (now, $v = 2t_2$) $\ell v \log \ell \cdot \tilde{O}((N/v)^{1/3}) = \tilde{O}(t_2^2 (N/t_2)^{1/3}) = \tilde{O}(N^{1/3} t_2^{5/3})$ and the message size of Bob is $\ell \log \ell \cdot \tilde{O}((N/v)^{1/3} + t_1 \log M) = \tilde{O}(t_2((N/t_2)^{1/3} + t_1 \log M)) = \tilde{O}(N^{1/3} t_2^{2/3} + t_1 t_2 \log M)$.

Next, we argue that the degree of the encoding and decoding is as in the underlying RCDS protocol. In the encoding, the servers execute a linear $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme over a field $\mathbb{F}_{2^{\lceil \log \ell \rceil}}$ in order to generate the shares $s_1 \dots s_\ell$, and encode each s_i is executing the underlying RCDS protocol for every bit in s_i. Since operations (addition and multiplication with a constant) in $\mathbb{F}_{2^{\lceil \log \ell \rceil}}$ can be implemented as operations in $\mathbb{F}_2$ with the same degree (see Section 2.2), the ramp scheme that the servers use is over $\mathbb{F}_2$. Therefore, the degree of the encoding is as the degree of the underlying RCDS protocol.

For the decoding, the referee first executes the decoding procedure of the underlying RCDS protocol in order to learn some of $s_1, \dots, s_\ell$ and then by executing the reconstruction procedure of the linear ramp secret-sharing scheme, the referee learns the secret. As the reconstruction procedure in the linear ramp secret-sharing scheme can be implemented in $\mathbb{F}_2$ (by Section 2.2), we conclude that the decoding is actually applying linear operations on the degree-2 polynomials that compute the decoding of some of the many copies of the underlying RCDS protocol. Thus, the degree of the encoding and the decoding of the resulting protocol are the same as for the underlying RCDS protocol. Thus, using our quadratic two-server $(t_1, 1)$-RCDS protocol over $\mathbb{F}_2$ of Theorem 9.1, 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 for a function $f : [M] \times [N] \to \{0,1\}$ (which is also an (M, t_2)-RCDS protocol) with secrets of size $O(t_2 \log N \log t_2)$ of [7] 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 $t_2 < N^{1/4}$.

10 Sharing and Reconstruction for Multi-Linear Secret Sharing

In [40, 15], it was shown that linear sharing and linear reconstruction are equivalent for 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. Our proof generalizes the proofs of [40, 15].

10.1 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 [41].

Lemma 10.1. *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 $s = (s_1, \dots, s_\ell)$, 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 the elements of the secret $s_1, \dots, s_\ell$. As in [15], we can reone-sided linear combinations as a system of linear equations in which the variables are the random elements and the elements of the secret $s_1, \dots, s_\ell$. 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. □

10.2 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 10.2) for its dual access structure; the size of this program is equal to the number of field elements in the secret-sharing scheme. Then, we use a claim from [16], 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; the number of field elements in the shares of this secret-sharing scheme equals to the size of the program. 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 [29]. We prove that even if we start with a $(0, 1/2)$-secret-sharing scheme (i.e., the correctness is perfect, however, the statistical distance between the shares of any two secrets is at most $1/2$), then the resulting scheme has linear sharing and reconstruction, perfect correctness, and perfect security.

We start by quoting a definition from [16] of a generalization of monotone span programs, called multi-target monotone span programs. Multi-linear schemes, introduced by [23], are based on this generalization.

Definition 10.2 (Multi-Target Monotone Span Programs [16])**.** *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 $V = \{\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}\}$ is a set of independent non-zero vectors in $\mathbb{F}^b$, for some $1 \leq \ell < b$, such that for every $A \subseteq P$ exactly 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.*

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 $\widehat{M}$ such that there are no subsets A such that M_A does not satisfy items 1 and 2 in Definition 10.2. Also note that by applying a linear transformation to the rows of M, the set of vectors V can be changed to any set of independent non-zero vectors without changing the size of $\widehat{M}$.

In [16] it was shown that a multi-target monotone span program implies a multi-linear secret-sharing scheme.

Lemma 10.3 ([16])**.** *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 perfectly-secure and perfectly-correct 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.*

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.

Definition 10.4. *Let Π be a secret-sharing scheme with linear reconstruction in which the shares contain c elements in $\mathbb{F}$. For every (minimal) authorized subset A and every element of the secret s_i, we define a reconstruction vector $r_{A,i} \in \mathbb{F}^c$, which contains the coefficients of the linear combination of the shares of A that recover s_i; the size of any reconstruction vector for A is the number of elements in the shares and it is non-zero only in coordinates corresponding to the shares of parties in A. In particular, for every vector of shares $(m_1, \ldots, m_c)$ of Π*

$$(m_1, \ldots, m_c) r_{A_i} = s_i.$$

Definition 10.5 (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 10.6 (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 $(\boldsymbol{r_{A,i}})^T$ in $M^\perp$, where $\boldsymbol{r_{A,i}}$ is the 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 block of columns for $i = 1$, and then block of columns for $i = 2$, etc), and*

- *$V = \{\boldsymbol{v_1}, \dots, \boldsymbol{v_\ell}\}$, where, for every $i \in [\ell]$, $\boldsymbol{v_i}$ consist of ℓ blocks of coordinates, the size of each of them is the number of minimal authorized sets of Γ, such that the i-th block contains ones and all other blocks contain zeros.*

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

Example 10.7. Let $P = \{P_1, P_2, P_3, P_4\}$ be a set of parties, $\{\{P_1, P_2\}, \{P_2, P_3\}, \{P_3, P_4\}\}$ be the minimal authorized sets in an access structure Γ, 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), the share of P_2 is $(r_1 \oplus s_1, r_3 \oplus s_2, r_4)$, 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., $(0, \dots, 0, 1, 0, 1)^T$. The full matrix is as follows:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

The target vectors are $(1, 1, 1, 0, 0, 0)$ and $(0, 0, 0, 1, 1, 1)$.

Claim 10.8. *Let Π be a $(0, 1/2)$-secret-sharing scheme realizing Γ with linear reconstruction over $\mathbb{F}$, where the secret contains ℓ field elements. The dual multi-target monotone span program $\widehat{M^{\perp}}$ of Π, as defined in Construction 10.6, 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 $\boldsymbol{r_{A,i}}$ 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 $\boldsymbol{v_i}$, since in the column $(\boldsymbol{r_{A,i}})^T$, 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, in every non-trivial combination of the target vectors all entries in at least one block $i \in [\ell]$ are non-zero. Thus, the rows labeled by $B = P \setminus A$ cannot span any non-trivial combination of the target vectors $V = \{\boldsymbol{v_1}, \dots, \boldsymbol{v_\ell}\}$.

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 vectors $\boldsymbol{v_j}$ for every $j \in [\ell]$. Assume by contradiction that there is a target vector $\boldsymbol{v_j}$ 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 $\boldsymbol{u}$ such that $\boldsymbol{v_j} \cdot \boldsymbol{u} = 1$ and $M^{\perp}_B \cdot \boldsymbol{u} = \boldsymbol{0}$.

Denote the secret for scheme Π by $s = (s_1, \ldots, s_\ell)$ and let Π_s be a vector of 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

$$\begin{aligned}\Pi_s \cdot (M^\perp \cdot \boldsymbol{u}) &= (\Pi_s \cdot M^\perp) \cdot \boldsymbol{u} = (s_1, \ldots, s_1, \ldots, s_\ell, \ldots, s_\ell) \cdot \boldsymbol{u} \\ &= \left(\sum_{i=1}^{\ell} (s_i \cdot \boldsymbol{v_i})\right) \cdot \boldsymbol{u} = \sum_{i=1}^{\ell} s_i \cdot (\boldsymbol{v_i} \cdot \boldsymbol{u}).\end{aligned} \tag{20}$$

Moreover, since $M_B^\perp \cdot \boldsymbol{u} = \boldsymbol{0}$, then $M^\perp \cdot \boldsymbol{u}$ is non-zero only in rows labeled by A, so by the above computation, the parties of A can compute $\Pi_s \cdot (M^\perp \cdot \boldsymbol{u}) = \sum_{i=1}^{\ell} s_i \cdot (\boldsymbol{v_i} \cdot \boldsymbol{u})$, which is, a non-trivial linear combination of the elements of the secret (since $\boldsymbol{v_j} \cdot \boldsymbol{u} \neq 0$). Take two secrets $s, s' \in \mathbb{F}^\ell$ that differ in this linear combination. As argued above, the shares of the unauthorized set A on these secrets are disjoint, contradicting the security requirement of the $(0, 1/2)$-secret-sharing scheme Π. □

Using two applications of Construction 10.6 and by Claim 10.8 and Lemma 10.3, we get the following theorem.

Theorem 10.9. *Let Γ be an n-party access structure and Π be a $(0, 1/2)$-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 $(0, 0)$-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 $(0, 1/2)$-secret-sharing scheme Π realizing Γ, we use Construction 10.6 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 10.8, $\widehat{M^\perp}$ accepts the dual access structure $\Gamma^\perp$. Then, by Lemma 10.3, there is a $(0, 0)$-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 10.6 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 10.8, 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 10.3, we get the desired secret-sharing scheme with linear sharing and linear reconstruction over $\mathbb{F}$ realizing Γ, in which the shares contain c field elements and the secret contains ℓ field elements. □

We next show that we can trade an exponentially small error in the correctness for statistical distance in the security. This claim implies that if a secret-sharing with linear reconstruction has an exponentially small error in the reconstruction, then by Theorem 10.9, it can be converted to a multi-linear secret-sharing scheme without increasing the share size.

Claim 10.10. *Let $\varepsilon < \frac{1}{2^{n+2}}$ and Π be an (ε, δ)-secret-sharing scheme realizing an n-party access structure Γ with linear reconstruction over $\mathbb{F}$ for 1-bits secrets. Then, there is a $(0, 2\delta + 2^{n+1}\varepsilon)$-secret-sharing scheme Π' realizing Γ with the same share size as in Π and with linear reconstruction over $\mathbb{F}$.*

Proof. We get Π' by modifying the sharing function of Π as follows: For every possible vector of shares $\boldsymbol{v}$ in Π for sharing a secret $b \in \{0, 1\}$, if the reconstruction algorithm for some $A \in \Gamma$ reconstructs $1 - b$ from $\boldsymbol{v}$, then do not give the vector $\boldsymbol{v}$ when sharing b. That is, these vectors have probability zero given b, and we normalize the probability of all other vectors of shares such that their sum given b is 1. Since we removed the vectors of shares in which the reconstruction in Π errs, every authorized set can reconstruct the secret b in Π' without any error using the reconstruction function of Π. Next we analyze the security of Π'. Let S_b for $b \in \{0,1\}$ be the vector of shares that are not used to share b. Let $\mathcal{D}_b(\boldsymbol{v})$ be the distribution of vector of shares $\boldsymbol{v}$ for sharing secret b in Π. By the correctness of Π, the probability that for a given set $A \in \Gamma$ and for a secret b the reconstruction algorithm reconstructs the secret $b' \neq b$ is at most ε. Thus, by a union bound over all authorized sets, $\mathcal{D}_0(S_0) \leq \varepsilon \cdot |\Gamma| \leq 2^n\varepsilon$ and similarly, $\mathcal{D}_1(S_1) \leq 2^n\varepsilon$. We add additional vectors to each set S_b such that $\mathcal{D}_0(S_0) = 2^n\varepsilon$ and $\mathcal{D}_1(S_1) = 2^n\varepsilon$, while maintaining the fact that S_0 and S_1 are disjoint. Let $\mathcal{D}'_b(\boldsymbol{v})$ be the distribution of vector of shares $\boldsymbol{v}$ for sharing secret b in Π', i.e.,

$$\mathcal{D}'_b(\boldsymbol{v}) = \begin{cases} 0 & \text{If } \boldsymbol{v} \in S_b, \\ \dfrac{\mathcal{D}_b(\boldsymbol{v})}{1-2^n\varepsilon} & \text{If } \boldsymbol{v} \notin S_b. \end{cases}$$

We next prove the statistical security of Π'. Fix an unauthorized set $A \notin \Gamma$. For a secret $b \in \{0,1\}$ let $\mathcal{D}_{b,A}(\boldsymbol{w})$ and $\mathcal{D}'_{b,A}(\boldsymbol{w})$ be the probability that the shares of the parties in A are $\boldsymbol{w}$ in Π and Π' respectively. Denoting the shares of the parties not in A by $\boldsymbol{z}$, we obtain

$$\begin{aligned} \mathcal{D}'_{b,A}(\boldsymbol{w}) &= \sum_{\boldsymbol{z}} \mathcal{D}'_b(\boldsymbol{wz}) \\ &= \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\notin S_b} \mathcal{D}'_b(\boldsymbol{wz}) + \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_b} \mathcal{D}'_b(\boldsymbol{wz}) \\ &= \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\notin S_b} \mathcal{D}'_b(\boldsymbol{wz}) \\ &= \frac{1}{1-2^n\varepsilon} \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\notin S_b} \mathcal{D}_b(\boldsymbol{wz}) \qquad (21) \\ &= \frac{1}{1-2^n\varepsilon} \left(\sum_{\boldsymbol{z}} \mathcal{D}_b(\boldsymbol{wz}) - \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_b} \mathcal{D}_b(\boldsymbol{wz}) \right) \\ &= \frac{1}{1-2^n\varepsilon} \left(\mathcal{D}_{b,A}(\boldsymbol{w}) - \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_b} \mathcal{D}_b(\boldsymbol{wz}) \right). \end{aligned}$$

Using (21), we bound the statistical distance between the shares of A for the secret 0 and the shares of A for the secret 1.

$$
\begin{aligned}
2\Delta(\mathcal{D}'_{0,A},\mathcal{D}'_{1,A}) &= \sum_{\boldsymbol{w}} \left|\mathcal{D}'_{0,A}(\boldsymbol{w}) - \mathcal{D}'_{1,A}(\boldsymbol{w})\right| \\
&= \frac{1}{1-2^n\varepsilon} \sum_{\boldsymbol{w}} \left|\mathcal{D}_{0,A}(\boldsymbol{w}) - \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_0} \mathcal{D}_0(\boldsymbol{wz}) - \mathcal{D}_{1,A}(\boldsymbol{w}) + \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_1} \mathcal{D}_1(\boldsymbol{wz})\right| \\
&\leq \frac{1}{1-2^n\varepsilon} \left(\sum_{\boldsymbol{w}} |\mathcal{D}_{0,A}(\boldsymbol{w}) - \mathcal{D}_{1,A}(\boldsymbol{w})| + \sum_{\boldsymbol{w}} \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_0} \mathcal{D}_0(\boldsymbol{wz}) + \sum_{\boldsymbol{w}} \sum_{\boldsymbol{z}\,:\,\boldsymbol{wz}\in S_1} \mathcal{D}_1(\boldsymbol{wz})\right) \\
&\leq \frac{1}{1-2^n\varepsilon} \left(2\delta + \sum_{\boldsymbol{v}\in S_0} \mathcal{D}_0(\boldsymbol{v}) + \sum_{\boldsymbol{v}\in S_1} \mathcal{D}_1(\boldsymbol{v})\right) \\
&= \frac{1}{1-2^n\varepsilon} \left(2\delta + \mathcal{D}_0(S_0) + \mathcal{D}_1(S_1)\right) \\
&= \frac{1}{1-2^n\varepsilon} \left(2\delta + 2\cdot 2^n\varepsilon\right) \\
&\leq 2\left(2\delta + 2\cdot 2^n\varepsilon\right),
\end{aligned}
$$

where the last inequality follows since $\varepsilon < \frac{1}{2^{n+2}}$. Thus, Π' is a perfectly correct secret-sharing scheme, in which the for every unauthorized set the statistical distance between the shares of 0 and 1 is at most $2\delta + 2^{n+1}\varepsilon$. □

References

[1] Leonard M. Adleman and Kireeti Kompella. Using smoothness to achieve parallelism. In *20th STOC*, pages 528–538, 1988.

[2] William Aiello, Yuval Ishai, and Omer Reingold. Priced oblivious transfer: How to sell digital goods. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 118–134, 2001.

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

[4] Benny Applebaum, Barak Arkis, Pavel Raykov, and Prashant Nalini Vasudevan. Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. *SIAM J. Comput.*, 50(1):32–67, 2021.

[5] Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 441–471, 2019.

[6] Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. In *52nd STOC*, pages 280–293, 2020.

[7] Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. Cryptology ePrint Archive, Report 2020/080, 2020.

[8] Benny Applebaum, Amos Beimel, Oded Nir, Naty Peter, and Toniann Pitassi. Secret sharing, slice formulas, and monotone real circuits. In *ITCS 2022*, volume 215 of *LIPIcs*, pages 8:1–8:23, 2022.

[9] Benny Applebaum, Thomas Holenstein, Manoj Mishra, and Ofer Shayevitz. The communication complexity of private simultaneous messages, revisited. In *EUROCRYPT 2018*, volume 10401 of *LNCS*, pages 261–286, 2018.

[10] Benny Applebaum and Oded Nir. Upslices, downslices, and secret-sharing with complexity of 1.5^{n}. In *CRYPTO 2021*, volume 12827 of *LNCS*, pages 627–655, 2021.

[11] Benny Applebaum and Prashant Nalini Vasudevan. Placing conditional disclosure of secrets in the communication complexity universe. In *10th ITCS*, pages 4:1–4:14, 2019.

[12] Nuttapong Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577, 2014.

[13] László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Combinatorica*, 19(3):301–319, 1999.

[14] E. Bach and J. Shalit. *Algorithmic Number Theory*, volume 1: Efficient Algorithms. MIT press, 1996.

[15] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion, 1996. `www.cs.bgu.ac.il/~beimel/pub.html`.

[16] Amos Beimel. Secret-sharing schemes: A survey. In *IWCC 2011*, volume 6639 of *LNCS*, pages 11–46, 2011.

[17] Amos Beimel and Oriol Farràs. The share size of secret-sharing schemes for almost all access structures and graphs. In *TCC 2020*, volume 12552 of *LNCS*, pages 499–529, 2020.

[18] Amos Beimel, Anna Gál, and Mike Paterson. Lower bounds for monotone span programs. *Computational Complexity*, 6(1):29–45, 1997.

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

[20] Amos Beimel, Hussien Othman, and Naty Peter. Quadratic secret sharing and conditional disclosure of secrets. In *CRYPTO 2021*, volume 12827 of *LNCS*, pages 748–778, 2021.

[21] Amos Beimel and Naty Peter. Optimal linear multiparty conditional disclosure of secrets protocols. In *ASIACRYPT 2018*, volume 11274 of *LNCS*, pages 332–362, 2018.

[22] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *CRYPTO '88*, volume 403 of *LNCS*, pages 27–35, 1988.

[23] Michael Bertilsson and Ingemar Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In *AUSCRYPT '92*, volume 718 of *LNCS*, pages 67–79, 1992.

[24] George Robert Blakley. Safeguarding cryptographic keys. In *Proc. of the 1979 AFIPS National Computer Conference*, volume 48, pages 313–317, 1979.

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

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

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

[28] Shawna Meyer Eikenberry and Jonathan P. Sorenson. Efficient algorithms for computing the Jacobi symbol. *Journal of Symbolic Computation*, 26(4):509–523, 1998.

[29] Serge Fehr. Efficient construction of the dual span program. Manuscript, 1999.

[30] Uri Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation. In *26th STOC*, pages 554–563, 1994.

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

[32] Anna Gál and Pavel Pudlák. Monotone complexity and the rank of matrices. *Inform. Process. Lett.*, 87:321–326, 2003.

[33] Romain Gay, Iordanis Kerenidis, and Hoeteck Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In *CRYPTO 2015*, volume 9216 of *LNCS*, pages 485–502, 2015.

[34] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *JCSS*, 60(3):592–629, 2000.

[35] Daniel M. Gordon. A survey of fast exponentiation methods. *Journal of Algorithms*, 27(1):129–146, 1998.

[36] Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *5th Israel Symp. on Theory of Computing and Systems*, pages 174–183, 1997.

[37] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304, 2000.

[38] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In *41st ICALP*, volume 8572 of *LNCS*, pages 650–662, 2014.

[39] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structure. In *Globecom 87*, pages 99–102, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. of Cryptology*, 6(1), 15-20, 1993.

[40] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *20th STOC*, pages 539–550, 1988.

[41] Mauricio Karchmer and Avi Wigderson. On span programs. In *8th Structure in Complexity Theory*, pages 102–111, 1993.

[42] Aleksei Dmitrievich Korshunov. On the number of monotone Boolean functions. *Probl. Kibern*, 38:5–108, 1981.

[43] Kasper Green Larsen and Mark Simkin. Secret sharing lower bound: Either reconstruction is hard or shares are long. In *SCN 2020*, volume 12238 of *LNCS*, pages 566–578, 2020.

[44] Tianren Liu and Vinod Vaikuntanathan. Breaking the circuit-size barrier in secret sharing. In *50th STOC*, pages 699–708, 2018.

[45] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In *CRYPTO 2017*, volume 10401 of *LNCS*, pages 758–790, 2017.

[46] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 567–596, 2018.

[47] Hussien Othman. *Secret Sharing: Polynomial Schemes and Evolving Schemes*. PhD thesis, Ben-Gurion University, 2022.

[48] Anat Paskin-Cherniavsky and Artiom Radune. On polynomial secret sharing schemes. In *ITC 2020*, volume 163 of *LIPIcs*, pages 12:1–12:21, 2020.

[49] Naty Peter. *Secret-Sharing Schemes and Conditional Disclosure of Secrets Protocols*. PhD thesis, Ben-Gurion University of the Negev, 2020. `http://aranne5.bgu.ac.il/others/PeterNaty19903.pdf`.

[50] Toniann Pitassi and Robert Robere. Strongly exponential lower bounds for monotone computation. In *49th STOC*, pages 1246–1255, 2017.

[51] Toniann Pitassi and Robert Robere. Lifting Nullstellensatz to monotone span programs over any field. In *50th STOC*, pages 1207–1219, 2018.

[52] Robert Robere. *Unified Lower Bounds For Monotone Computation*. PhD thesis, University of Toronto, 2018. `https://www.cs.mcgill.ca/~robere/thesis.pdf`.

[53] Robert Robere, Toniann Pitassi, Benjamin Rossman, and Stephen A. Cook. Exponential lower bounds for monotone span programs. In *57th FOCS*, pages 406–415, 2016.

[54] Adi Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[55] Vinod Vaikuntanathan and Prashant Nalini Vasudevan. Secret sharing and statistical zero knowledge. In *ASIACRYPT 2015*, pages 656–680, 2015.

[56] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. Available at: *Measures of Complexity: Festschrift for Alexey Chervonenkis*, pages 11-30, 2015.

[57] Hoeteck Wee. Dual system encryption via predicate encodings. In *TCC 2014*, volume 8349 of *LNCS*, pages 616–637, 2014.

A Proof of Theorem 4.2

We next prove Theorem 4.2, that is, for a family of possible reconstruction functions $\mathcal{F}_{\text{rec}}$ and a family of n-party access structures $\mathcal{F}_{\mathcal{A}}$, for many access structures in $\mathcal{F}_{\mathcal{A}}$ the total share size in any secret-sharing scheme realizing the access structure with reconstruction functions in $\mathcal{F}_{\text{rec}}$ is $\Omega\left(\frac{\mathrm{VC}(\mathcal{F}_{\mathcal{A}})}{\log|\mathcal{F}_{\text{rec}}|}\right)$ (where $\mathrm{VC}(\mathcal{F}_{\mathcal{A}})$ is the VC-dimension of $\mathcal{F}_{\mathcal{A}}$).

Proof of Theorem 4.2. We first prove Item 1 of the theorem. Let $v = \mathrm{VC}(\mathcal{F}_{\mathcal{A}})$ and $A_1, \dots, A_v$ be a sequence of sets shattered by $\mathcal{F}_{\mathcal{A}}$. Furthermore, let $\mathcal{F}'_{\mathcal{A}} \subseteq \mathcal{F}_{\mathcal{A}}$ be a collection of 2^v access structures that shatter $A_1, \dots, A_v$.

Consider an access structure $\Gamma \in \mathcal{F}'_{\mathcal{A}}$ and a secret-sharing scheme Π that realizes Γ with domain of secrets $\{0,1\}$, domain of random strings R, domain of shares $\{0,1\}^c$, and reconstruction functions from $\mathcal{F}_{\text{rec}}$. We prove that Γ can be described by $c\log|\mathcal{F}_{\text{rec}}| + 0.081v$ bits. As the descriptions of almost all access structures $\Gamma \in \mathcal{F}'_{\mathcal{A}}$ require at least $v/2$ bits, the theorem follows.

Let T be a parameter to be fixed later. We consider an experiment where we choose secrets $b_1, \dots, b_T \in \{0,1\}$ independently with uniform distribution and choose random strings $r_1, \dots, r_T \in R$ independently with uniform distribution. Since Π realizes Γ, the two following conditions hold:

- If $B \in \Gamma$, then there exists a function $f \in \mathcal{F}_{\text{rec}}$ such that $f(\Pi_B(b_i, r_i)) = b_i$ for every $i \in \{1, \dots, T\}$.
- If $B \notin \Gamma$, then for every $i \in \{1, \dots, T\}$ and for every function $f \in \mathcal{F}_{\text{rec}}$
$$\Pr_{b_i, r}[f(\Pi_B(b_i, r)) = b_i] = 1/2.$$
This follows from the security requirement of the secret-sharing scheme (i.e., from the fact that $\Pi_B(0, r)$ and $\Pi_B(1, r)$ are equally distributed).

Hence, we get that when $B \in \Gamma$, there exists at least one function $f \in \mathcal{F}_{\text{rec}}$ such that $|\{i : f(\Pi_B(b_i, r_i)) = b_i\}| = T$. On the other hand, when $B \notin \Gamma$, for a given $f \in \mathcal{F}_{\text{rec}}$ the probability that $f(\Pi_B(b_i, r_i)) = b_i$ for every $i \in \{1, \dots, T\}$ is 2^{-T}. By the union bound, if we take $T = \log|\mathcal{F}_{\text{rec}}| + 7$, then the probability that there is at least one $f \in \mathcal{F}_{\text{rec}}$ such that $f(\Pi_B(b_i, r_i)) = b_i$ for every $i \in \{1, \dots, T\}$ is at most
$$2^{-T} \cdot |\mathcal{F}_{\text{rec}}| = 2^{-\log|\mathcal{F}_{\text{rec}}| - 7} \cdot |\mathcal{F}_{\text{rec}}| < 0.01.$$

By an averaging argument, there are $b_1, \dots, b_T$ and $r_1, \dots, r_T$ such that for at least 0.99 of the sets $A_1, \dots, A_v$ that are not in Γ it holds that $|\{i : f(\Pi_{A_i}(b_i, r_i)) = b_i\}| < T$ for all $f \in \mathcal{F}_{\text{rec}}$. We use these values to describe Γ:

- Write down $b_1, \dots, b_T$ and $\Pi_1(b_i, r_i), \dots, \Pi_n(b_i, r_i)$ for every $i \in \{1, \dots, T\}$.
- Let
$$X = \{j : A_j \notin \Gamma \text{ and } \exists_{f \in \mathcal{F}_{\text{rec}}} |\{i : f(\Pi_{A_i}(b_i, r_i)) = b_i\}| = T\},$$
that is, the indices of the unauthorized sets that "pass the test" of being in the access structure.

  The size of X is at most $0.01 \cdot v$. The number of options of X is $\binom{v}{\leq 0.01v} \leq 2^{h(0.01)\cdot v}$ (where $h(\cdot)$ is the binary entropy). Thus, X can be encoded with $h(0.01) \cdot v$ bits.

Therefore, Γ can be encoded by at most $(c+1)(\log|\mathcal{F}_{\text{rec}}|+7)+h(0.01)\cdot v$ bits. The number of access structures in $\mathcal{F}'_{\mathcal{A}}$ that can be described by $0.5\cdot\log|\mathcal{F}'_{\mathcal{A}}| = 0.5\cdot v$ bits is at most $2^{0.5\cdot\log|\mathcal{F}'_{\mathcal{A}}|} = 2^{0.5\cdot v}$. For all other sets $\mathcal{F}'_{\mathcal{A}}$ it holds that

$$(c+1)(\log|\mathcal{F}_{\text{rec}}|+7)+h(0.01)\cdot v \geq 0.5\cdot v,$$

that is, $c=\Omega\left(\frac{v}{\log|\mathcal{F}_{\text{rec}}|}\right)$.

We next outline the changes in the above proof needed to prove Item 2 of the theorem. In this case we consider all access structures in $\mathcal{F}_{\mathcal{A}}$ and in the description of an access structure $\Gamma\in\mathcal{F}_{\mathcal{A}}$ we write the maximal unauthorized sets of Γ that "pass the test" of being in the access structure. That is, we describe the access structure as follows:

- Write down $b_1,\ldots,b_T$ and $\Pi_1(b_i,r_i),\ldots,\Pi_n(b_i,r_i)$ for every $i\in\{1,\ldots,T\}$.
- Let

  $$X=\{A : A \text{ is a maximal unauthorized set of } \Gamma \text{ and } \exists_{f\in\mathcal{F}_{\text{rec}}}|\{i : f(\Pi_A(b_i,r_i))=b_i\}|=T\}.$$

  As above, we choose $b_1,\ldots,b_T,r_1,\ldots,r_T$ such that the size of X is at most $0.01\cdot N_{\max}(\Gamma)$. The number of options of X is $\binom{N_{\max}(\Gamma)}{\leq 0.01\cdot N_{\max}(\Gamma)}\leq 2^{h(0.01)\cdot N_{\max}(\Gamma)}$. Thus, X can be encoded with $h(0.01)\cdot N_{\max}(\Gamma)\leq 0.081\cdot N_{\max}(\Gamma)$ bits.

Thus, for all but $\sqrt{\mathcal{F}_{\mathcal{A}}}$,

$$(c+1)(\log|\mathcal{F}_{\text{rec}}|+7)+h(0.01)\cdot N_{\max}(\Gamma)\geq 0.5\cdot\log|\mathcal{F}_{\mathcal{A}}|,$$

and Item 2 of the theorem follows. □