Better Secret-Sharing via Robust Conditional Disclosure of Secrets*

Benny Applebaum
Tel-Aviv University
Tel-Aviv, Israel
benny.applebaum@gmail.com

Amos Beimel
Ben-Gurion University of the Negev
Be'er-Sheva, Israel
amos.beimel@gmail.com

Oded Nir
Tel-Aviv University
Tel-Aviv, Israel
odednir123@gmail.com

Naty Peter
Ben-Gurion University of the Negev
Be'er-Sheva, Israel
naty@post.bgu.ac.il

January 26, 2020

Abstract

A secret-sharing scheme allows to distribute a secret s among n parties such that only some predefined "authorized" sets of parties can reconstruct the secret, and all other "unauthorized" sets learn nothing about s. The collection of authorized sets is called the access structure. For over 30 years, it was known that any (monotone) collection of authorized sets can be realized by a secret-sharing scheme whose shares are of size $2^{n-o(n)}$ and until recently no better scheme was known. In a recent breakthrough, Liu and Vaikuntanathan (STOC 2018) have reduced the share size to $2^{0.994n+o(n)}$, which was later improved to $2^{0.892n+o(n)}$ by Applebaum et al. (EUROCRYPT 2019).

In this paper we improve the exponent of general secret-sharing down to 0.637. For the special case of linear secret-sharing schemes, we get an exponent of 0.762 (compared to 0.942 of Applebaum et al.).

As our main building block, we introduce a new *robust* variant of conditional disclosure of secrets (robust CDS) that achieves unconditional security even under limited form of re-usability. We show that the problem of general secret-sharing reduces to robust CDS with sub-exponential overhead and derive our main result by implementing robust CDS with a non-trivial exponent. The latter construction follows by presenting a general immunization procedure that turns standard CDS into a robust CDS.

*This work subsumes the ePrint manuscript [13]. The first and third authors are supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, and the Check Point Institute for Information Security. This work was done while the second author was visiting Georgetown University, supported by NSF grant no. 1565387, TWC: Large: Collaborative: Computing Over Distributed Sensitive Data and by ERC grant 742754 (project NTSC). The second and forth authors are also supported by ISF grant 152/17, by a grant from the Cyber Security Research Center at Ben-Gurion University of the Negev, and by the Frankel center for computer science. Part of this work was done while the second and forth authors were visiting Simons Institute for the Theory of Computing.

1 Introduction

Secret-sharing schemes, introduced by Shamir [49] and Blakley [17], are a central cryptographic tool with a wide range of applications including secure multiparty computation protocols [14, 19], threshold cryptography [24], access control [45], attribute-based encryption [33, 52], and oblivious transfer [50, 51]. In its general form [36], an n-party secret-sharing scheme for a family of authorized sets $F \subseteq 2^{[n]}$ (referred to as *access structure*) allows to distribute a secret s into n shares, $s_1, \ldots, s_n$, one for each party, such that: (1) every authorized set of parties, $A \in F$, can reconstruct s from its shares; and (2) every unauthorized set of parties, $A \notin F$, cannot reveal any partial information on the secret even if the parties are computationally unbounded. For example, in the canonical case of threshold secret-sharing the family F contains all the sets whose cardinality exceeds some certain threshold. For this case, Shamir's scheme [49] provides a solution whose complexity, measured as the total share-size $\sum_i |s_i|$, is quasi-linear, $O(n \log n)$, in the number of parties n. Moreover, Shamir's scheme is *linear*, that is, each share can be written as a linear combination of the secret and the randomness which are taken from a finite field. This form of linearity turns to be useful for many applications. (See Section 3 for a formal definition of secret sharing and linear secret sharing.)

The complexity of general secret sharing. Determining the complexity of general access structures is a basic, well-known, open problem in information-theoretic cryptography. Formally, given a (monotone) access structure[1] F we let $\mathsf{SS}(F) := \min_{\mathcal{D} \text{ realizes } F} |\mathcal{D}|$, where $|\mathcal{D}|$ denotes the total share size of a secret-sharing scheme $\mathcal{D}$.[2] For over 30 years, since the pioneering work of Ito et al. [36], all known upper-bounds on $\mathsf{SS}(F)$ are tightly related to the computational complexity of the characteristic function F. Here we think of F as the monotone function that given a vector $x \in \{0,1\}^n$ outputs 1 if and only if the corresponding characteristic set $A = \{i : x_i = 1\}$ is an authorized set. Specifically, it is known that the complexity of an access structure is at most polynomial in the representation size of F as a monotone CNF or DNF [36], as a monotone formula [15], as a monotone span program [39], or as a multi-target monotone span program [16]. This leads to an exponential upper-bound of $2^{n(1-o(1))}$ for any n-party access structure F.

On the other hand, despite much efforts, the best known lower-bound on the complexity of an n-party access structure is $\Omega(n^2/\log n)$ due to [23]. Moreover, we have no better lower-bounds even for *non-explicit* functions![3] This leaves a huge exponential gap between the upper-bound and the lower-bound. For the case of linear schemes, a counting argument (see, e.g., [9]) shows that for most monotone functions $F : \{0,1\}^n \to \{0,1\}$, the complexity of the best linear secret-sharing scheme, denoted by $\mathsf{LSS}(F)$ is at least $2^{n/2-o(n)}$.[4] Furthermore, Pitassi and Robere [47] (building on results of [48, 46]) prove that for every n there exist an explicit n-input function F such that $\mathsf{LSS}(F) = 2^{\Omega(n)}$. In his 1996 thesis [6], Beimel conjectured that an exponential lower-bound of $2^{\Omega(n)}$ also holds for the general case. Resolving this conjecture has remained one of the main open problems in the field of secret-sharing [7]. Taking a broader view, similar exponential communication-complexity gaps exist for a large family of information-theoretic secure computation tasks [27, 35, 5, 31, 11]. Among these, secret-sharing is of special interest due to its elementary nature: Secret data is only *stored and revealed* without being processed or manipulated.

[1] Monotonicity here means that for any $A \subset B$ it holds that $A \in F \Rightarrow B \in F$. It is not hard to see that a non-monotone access structure does not admit a secret-sharing scheme, and therefore this requirement is necessary.

[2] This complexity measure essentially ignores the bit-length of the secret. The alternative *information-ratio* measure normalizes the bit length of the longest share by the length of the secret, and is therefore more suitable to the case of long secrets. Indeed, recent results [2] suggest that the ratio achievable for (very) long secrets may be significantly better than the ratio achievable for short secrets.

[3] In contrast, in the computational complexity setting, counting-based methods lead to exponential lower-bounds on the complexity of most monotone functions over n-bits for various computational models including the ones mentioned above. These bounds can be shown to be tight for a random (monotone) function; see, e.g., [38].

[4] The bound holds for any finite field. From now on when the field is unspecified we take it, by default, to be the binary field. This only makes our positive results stronger.

The LV construction. In a recent breakthrough, Liu and Vaikuntanathan [40] (hereafter referred to as LV) showed, for the first time, that it is possible to construct secret-sharing schemes in which the total share size is $2^{cn+o(n)}$ with an exponent c strictly smaller than 1. In particular, they showed that every access structure can be realized by a linear scheme of complexity $2^{0.999n+o(n)}$, and by a non-linear scheme of complexity $2^{0.994n+o(n)}$. In a nutshell, for a balancing parameter $\delta > 0$, the LV construction decomposes an access structure F into three access structures:

1. The "middle slice" $F_{\mathrm{mid},\delta}$ that agrees with F on all sets whose density is in $(\frac{1}{2}-\delta, \frac{1}{2}+\delta)$ and assigns zero to sets of smaller density and one to sets of larger density.

2. Two other "extreme slices", $F_{\mathrm{bot},\delta}$ and $F_{\mathrm{top},\delta}$, that essentially agree with F on bottom inputs of density smaller than $\frac{1}{2}-\delta$ and on top inputs of density larger than $\frac{1}{2}+\delta$. (A more accurate definition appears in Appendix A.2.1.)

The extreme slices can be realized by a secret-sharing scheme with exponent smaller than 1 since they admit a monotone formula (or even a monotone CNF/DNF) of this size. Thus, the main effort in [40] is devoted to realizing $F_{\mathrm{mid},\delta}$ with a non-trivial, smaller-than-one, exponent. Towards this end, LV show that the function $F_{\mathrm{mid},\delta}$ can be computed by an exponential-size constant-depth formula with a non-trivial exponent of $\mathbf{M}_{\mathrm{LV}}(\delta) < 1$ that employs standard AND/OR-gates together with a special form of *block-regular* gates. Roughly speaking, in such a gate, $G : \{0,1\}^n \to \{0,1\}$, the n-bit input is partitioned into equal-sized blocks of size B each, and the main feature is that G is defined only on inputs $x \in \{0,1\}^n$ that hit exactly b of the indices in each block for some integer parameter b. Equivalently, the parties are partitioned into B-size committees and we can determine whether a set A is authorized or not for every set A that consists of exactly b members out of each committee.

Example 1.1. Assume that $n = 8, B = 4, b = 2$, and consider the partition to the first 4 coordinates and last 4 coordinates. There are $\binom{4}{2}^2$ inputs that G is defined on. E.g., G should be defined on the input 10100110, and is not defined on the input 11100001.

LV then show how to implement block-regular gates with sub-exponential complexity of $2^{o(n)}$ based on recent sub-exponential constructions of *Conditional Disclosure of Secrets* (CDS) problem from [42]. (We postpone the description of CDS protocols to a later point.) Taken together, this allows to realize $F_{\mathrm{mid},\delta}$ with complexity of $2^{\mathbf{M}_{\mathrm{LV}}(\delta)n}$. The final result is obtained by choosing a parameter δ that balances the cost of $F_{\mathrm{mid},\delta}$ with the cost of $F_{\mathrm{bot},\delta}$ and $F_{\mathrm{top},\delta}$.

In a follow-up work, Applebaum et al. [3] improved the LV bound to $2^{0.942n+o(n)}$ in the linear case, and to $2^{0.892n+o(n)}$ in the non-linear case. This was done by reducing the problem of realizing the extreme slices to (many) general secret-sharing problems over a smaller domain, leading to a recursive construction. However, the complexity of mid-slice access structures has remained unchanged.

2 Our Contribution

In light of the exponential gap between the lower bounds and the upper-bounds, we believe that it is both important and useful to study on the best-achievable *exponent* of secret-sharing sharing. Formally, we define the *secret-sharing exponent* $\mathbf{S}$ to be

$$\mathbf{S} = \limsup_{n\to\infty} \max_{F\in\mathcal{M}(n)} \frac{1}{n} \log \mathsf{SS}(F),$$

where $\mathcal{M}(n)$ is the family of all n-party access structures (equivalently, all monotone functions over $\{0,1\}^n$). The *linear exponent*, $\mathbf{S}_\ell$, is defined analogously except that $\mathsf{SS}(F)$ is replaced with $\mathsf{LSS}(F)$, the minimal complexity of a linear scheme that realizes F. Under this definition, it holds that $\frac{1}{2} \leq \mathbf{S}_\ell \leq 0.942$ and $0 \leq \mathbf{S} \leq 0.892$. The existence of sub-exponential secret-sharing schemes would imply that $\mathbf{S} = 0$, whereas Beimel's conjecture asserts that $\mathbf{S}$ is strictly positive.

In this work, we improve the upper-bounds on $\mathbf{S}$ and $\mathbf{S}_\ell$, and, more qualitatively, provide new directions that may eventually lead to sub-exponential solutions. Along the way, we introduce a new notion of *robust* conditional disclosure of secrets, that may be of independent interest. We proceed with a detailed account of our results.

Better secret-sharing schemes. We significantly improve the secret-sharing exponent both for the linear and non-linear case.

Theorem 2.1 (main theorem). *Every access structure over n parties can be realized by a secret-sharing scheme with a total share-size of $2^{0.637n+o(n)}$ and by a linear secret-sharing scheme with a total share size of $2^{0.762n+o(n)}$. That is, $\mathbf{S} \leq 0.637$ and $\mathbf{S}_\ell \leq 0.762$.*

The proof of Theorem 2.1 is based on a new secret-sharing schemes for mid-slice access structures. Recall that a mid-slice access structure with parameter δ is a monotone function $F_{\mathrm{mid},\delta} : \{0,1\}^n \to \{0,1\}$ that takes the value zero on inputs of Hamming weight smaller than $(\frac{1}{2}-\delta)n$, takes the value one on all inputs of Hamming weight larger than $\frac{1}{2}+\delta$, and may take arbitrary values in-between. As already mentioned, LV showed that such functions can be implemented by a formula over OR/AND gates and block-regular gates of exponential size $2^{\mathbf{M}_{\mathrm{LV}}(\delta)n}$. We begin by showing that if one considers a more powerful basis that consists of *somewhat-regular* gates (together with general threshold gates), then this can be done by a linear-size formula with only $O(n)$ gates.

A somewhat-regular gate $G : \{0,1\}^n \to \{0,1\}$ is parameterized by a pair of integers, (a,b), a block-size parameter B where $a \leq b \leq B$ and a partition Π of $[n]$ to B-size blocks. An input $x \in \{0,1\}^n$ is parsed to B-size sub-strings $(x_1,\ldots,x_{n/B})$ according to Π, and the gate can be arbitrarily programmed on inputs that each of their components x_i has Hamming weight of at least a and at most b. Such an input is referred to as (Π,a,b)-regular. (Under the committee-based terminology, in each committee the set x has at least a and at most b members.) We *do not care* what value G takes over all other inputs.[5]

Example 2.2. Assume that $n = 8, B = 4, a = 1, b = 2$, and consider the partition to the first 4 coordinates and last 4 coordinates. There are $\left(\binom{4}{1} + \binom{4}{2}\right)^2$ inputs that G is defined on. E.g., G should defined on the inputs 10100110 and 10100001, and is not defined on the input 11100001.

From somewhat-regular gates to mid-slice functions. We can realize any mid-slice access function $F_{\mathrm{mid},\delta}$ by a formula that makes use of $\ell = O(n)$ somewhat-regular gates with parameters $B = \sqrt{n}$, $a \approx (\frac{1}{2} - \delta)B$ and $b \approx (\frac{1}{2} + \delta)B$. Roughly speaking, we show (via the probabilistic method) that one can choose $\ell = O(n)$ B-partitions $\Pi_1,\ldots,\Pi_\ell$ of $[n]$ such that any input $x \in \{0,1\}^n$ of Hamming weight $\mathrm{wt}(x) \in [(\frac{1}{2} \pm \delta)n]$ is (Π_i,a,b)-regular with respect to a majority of the Π_i's. In contrast, LV used regular partitions and they needed exponentially many partitions to guarantee that an input x has exactly b ones in each part of the partition.

By programming the i-th gate G_i according to the restriction of $F_{\mathrm{mid},\delta}$ to the (Π_i,a,b)-regular inputs, we can realize $F_{\mathrm{mid},\delta}$ by computing Majority over all the ℓ somewhat-regular gates. (One still has to make sure that the formula works well for light/heavy inputs x of Hamming weight $\mathrm{wt}(x) \notin [(\frac{1}{2} \pm \delta)n]$, however, this can be achieved easily with few additional threshold gates.) Using standard secret-sharing techniques,

[5]Technically, this means that the corresponding access structure is viewed as a *partial* (or promise) access structure. Interestingly, it turns out that the freedom to work with partially-defined specifications as components (without enforcing a full specification on each such sub-component) significantly simplifies the overall construction.

the resulting formula allows us to efficiently reduce the problem of realizing a general mid-slice access structure $F_{\mathrm{mid},\delta}$ to the problem of realizing somewhat-regular access structures with parameters $B = \sqrt{n}$, $a \approx (\frac{1}{2} - \delta)B$ and $b \approx (\frac{1}{2} + \delta)B$. Our next goal is therefore to realize somewhat-regular access structures. For this, we will have to present a new notion of robust conditional disclosure of secrets.

Conditional disclosure of secrets. Conditional Disclosure of Secrets (CDS) protocols were introduced by Gartner et al. [30] in the context of private information retrieval, and since then were used in many cryptographic applications, such as attribute based encryption [29, 4, 53], priced oblivious transfer [1], and, as already mentioned, secret-sharing schemes [40, 12, 3]. In a CDS protocol, there are k servers and a referee; each server i holds a private input $x_i \in X_i$, a common secret s, and a common random string r. The referee holds all private inputs $(x_1, \ldots, x_k)$ but, prior to the protocol, it does not know neither the secret nor the random string. The goal of the protocol is to let the referee learn the secret if and only if the inputs of the servers satisfy some pre-defined condition $f : X_1 \times \cdots \times X_k \to \{0, 1\}$. The challenge is that the communication model is minimal – each server sends one message to the referee, without seeing neither the inputs of the other servers nor their messages.

Example 2.3 (CDS for equality). One can define a 2-sever CDS protocol for the equality predicate EQ : $X \times X \to \{0, 1\}$ as follows. The common randomness consists of an hash function $h : X \to \{0, 1\}$ that is sampled from a pair-wise independent hash function family, the first server sends the message $h(x_1)$ and the second server sends the message $h(x_2) \oplus s$. The second message "perfectly-encrypts" the secret under the "key" $h(x_2)$, and the first message releases the key if $x_1 = x_2$ and otherwise consists of a random independent element.

It is shown in [40] that secret-sharing for general regular gates with block-size of B can be efficiently realized based on CDS protocols for $t = (n/B)$ servers for general predicates over the domain $\{0, 1\}^B \times \cdots \times \{0, 1\}^B$. Loosely speaking, any set of secret-sharing parties $x = (x_1, \ldots, x_t) \in (\{0, 1\}^B)^t$ gets to learn, for every server $i \in [t]$, the CDS message that the i-th server computes over the input x_i. (See Section 5.1 for full details.) While this leads to an efficient implementation of regular secret-sharing schemes based on the recent CDS constructions of [41], the transformation fails to produce the more powerful form of somewhat-regular secret-sharing. The problem is that a somewhat-regular set of secret-sharing parties $x = (x_1, \ldots, x_t) \in (\{0, 1\}^B)^t$ gets to learn the CDS messages that correspond to all inputs $x' \leq x$ where $\leq$ stands for the standard partial order over binary strings. Furthermore, all these CDS messages are computed with the same randomness. In such a case, the privacy guarantees of the CDS are completely lost, even if none of the inputs satisfy the CDS predicate f!

To get a better understanding of the problem, let us consider, for example, the CDS for equality from Example 2.3. Suppose that the first server releases the CDS messages that correspond to two inputs, x_1 and x'_1, that are both unequal to the second server's input x_2. Assuming that h is implemented via a random affine function, the CDS privacy completely breaks. Given the values of $h(x_1)$ and $h(x'_1)$ together with x_1 and x'_1 (who are known to the referee) one can fully recover the description of h, evaluate it over x_2 (which is also public), and recover the secret s from $h(x_2) \oplus s$.

We remedy the situation by showing that if one starts with a stronger form of CDS protocols, then the LV transformation does lead to somewhat-regular secret-sharing schemes. Specifically, we introduce the following new notion of *robust* CDS (RCDS) that may be of independent interest.

Robust conditional disclosure of secrets. We say that a CDS protocol is robust if it provides information-theoretic privacy even if it is invoked on a bounded number of multiple inputs using the same randomness. The general notion of robustness is parameterized by the input sets over which the protocol may be re-used. For now, let us consider the special case where each server i may re-use the randomness over any set of t different inputs. Since this may happen simultaneously for all servers, the randomness may be re-used over a set of t^k inputs. We present a general transformation that takes any CDS protocol, and "immunizes" a single server. By applying the construction to each server separately, we derive t-robustness with an overhead of roughly $(t \operatorname{polylog} u)^k$, where u is the number of possible input tuples that may be re-used together by a single server. In order to explain the transformation, it will be instructive to consider the following more abstract "secure channel" setting.

How to immunize a channel? Suppose that a sender wishes to send t private messages to a receiver. The messages arrive in on online manner, one after the other, and the sender is connected to the receiver via N unidirectional channels that offer one-time privacy. That is, once a channel is being used twice all the messages that were sent over it are revealed to everyone. The goal is to minimize N as a function of t while maintaining perfect privacy for all messages. Our sender is stateless, and so it cannot even remember how many messages have been sent so far. In particular, the trivial solution of sending the i-th message over the i-th channel is inapplicable.

Fortunately, each message m_i arrives with some (non-private) unique *tag* $x_i \in X$ that is available to both parties. Therefore, we can naively solve the problem with $N = |X|$ channels by allocating a channel to each possible tag. The question is can we do better when t is significantly smaller than $|X|$? More generally, say that we know ahead of time that the sequence of t tags belong to one of u possible t-subsets $Z_1, \dots, Z_u \subset X$ which are a-priori fixed. How small can N be as a function of t and u?

A natural way to solve the problem is to secret share each message $m \in M$ to shares $(s_1, .., s_N) \in M^N$ via some secret-sharing scheme $\mathcal{D}$, and deliver these shares over a subset of the channels that is selected according to the tag x. That is, we send s_i over the i-th channel if i is in the set $H(x)$ where H is some "hash" function that maps a tag $x \in X$ to subsets of N. Correctness is guaranteed as long as $H(x)$ is an authorized set of the secret-sharing scheme for every $x \in X$. On the other hand, as long as the pair-wise intersections of $H(x)_{x \in Z}$ forms an unauthorized set, we get privacy for the set of inputs Z. The immunization question now boils down to designing such an admissible hash-function/secret-sharing pair $(\mathcal{D}, H)$ for a given sequence of t-size input sets $Z_1, \dots, Z_u \subset X$ while minimizing N.

We describe two different solutions for the problem that achieve $N = (t \operatorname{polylog} u)$ complexity. In both cases, the starting point is an inefficient construction with quadratic overhead. The first approach is based on a family of ℓ *perfect hash functions* $H = \{h_1, \dots, h_\ell : X \to [t^2]\}$ for the set family $(Z_i)_{i \in [u]}$. That is, for every $i \in [u]$ there exists a function $h \in H$ that perfectly hashes the elements of Z_i to t distinct values. We place the N channels on an $\ell \times t^2$ matrix, and given a message m labeled by x, we share m via ℓ-out-of-ℓ secret-sharing scheme to $(s_1, .., s_\ell)$, and send the secret s_i over the channel (i, j) iff $h_i(x) = j$. Accordingly, a subset of channels is authorized iff it contains at least a single channel in each row. Clearly, every message is delivered over an authorized set of channels. On the other hand, since $h_j \in H$ is perfect over Z_i, the pair-wise intersections of $H(x)_{x \in Z_i}$ completely avoids the j-th row of channels.

By taking ℓ to be logarithmic in u (as in the perfect hashing of [28]), we get a quasi-quadratic bound on N. In order to reduce the overhead to $t \operatorname{polylog} u$, we apply the quadratic solution over the collection $\binom{X}{\log t}$ of all $\log t$-subsets of X. This effectively upgrades one-time security into a $\log t$-security. The latter channel can be further immunized via a more liberal combination of secret-sharing/hashing pair: Only $\log t$-wise intersections of $H(x)_{x \in Z}$ should form an unauthorized set. This condition translates to a weaker version of perfect hashing that can be obtained with poly-logarithmic overhead. Overall, the resulting two-level hashing construction resembles a similar construction of [21], that was suggested in the context of traitor-tracing schemes.

We also present an alternative construction that achieves similar parameters based on "sparse-hashing". Roughly, the message is secret-shared via a threshold secret-sharing with threshold βN, and each x is mapped to a β-sparse subset of N so that the Z_i-intersection of the sets results in a sparser set of density strictly smaller than β. This construction is inspired by the a similar construction of [32] that was presented in the context of Functional Encryption.[6] To optimize the parameters, we apply it again in a two-level way.

Back to RCDS. The channel solution can be immediately adopted to the distributed CDS setting. The servers use their shared randomness to secret-share the CDS secret s according to the secret-sharing scheme $\mathcal{D}$, and use N copies of CDS with independent random strings to deliver the shares $s_1, \dots, s_N$. The i-th server, that should be immunized, sends his messages only for the CDS instances that are indexed by $H(x_i)$ and remains silent in all other instances. All other parties send their messages for all the CDS instances. Correctness and privacy follow immediately from the correctness and privacy guarantees of the channel problem. As already mentioned, by immunizing the servers one after the other, we derive a general immunization procedure that transforms a general CDS to a RCDS.

We do not know whether there is a more direct, cheaper approach for constructing a robust CDS. As a positive sign, we show that the best known linear CDS constructions already achieve some partial form of robustness. Indeed, for the linear case, it is more cheaper to use these robust schemes, than to apply the immunization procedure. The existence of similar cheaper non-linear robust CDS remains as an intriguing open question, whose resolution may lead to further improvement in the complexity of general secret-sharing schemes.

Organization. Secret-sharing schemes and CDS protocols are defined in Section 3. Some parts of the definitions are deferred to Appendix B. Robust CDS protocols are defined and constructed in Section 4. The construction of secret-sharing schemes from RCDS protocols is described in Section 5. An abstraction of the immunization construction that is used to transform a CDS protocol to a RCDS protocol as well as an alternative immunization construction appears in Section 6. A simple construction of a secret-sharing scheme with exponent less than 1 is described in Appendix A. Some additional probability background (especially, on negatively associated random variables) is presented in Appendix C. Linear CDS and RCDS protocols for arbitrary functions are discussed in Appendix D. Specifically, a proof that a variant of the linear k-server CDS protocol of [12] is already robust for half of the servers is depicted in Appendix D.1 and a construction of a more efficient linear 2-server RCDS protocol is given in Appendix D.2.

3 Preliminaries

Secret-sharing schemes. We present the definition of secret-sharing schemes, similar to [8, 22]. For the privacy of these schemes, we use the following notation: For two random variables X and Y, we say that $X \equiv Y$ if they are identically distributed.

Definition 3.1 (Partial access structures)**.** *Let $P = \{P_1, \dots, P_n\}$ be a set of parties. A* partial access structure *is a pair of collections $\Gamma = (\Gamma_{\text{no}}, \Gamma_{\text{yes}})$, where $\Gamma_{\text{no}}, \Gamma_{\text{yes}} \subseteq 2^P$ are non-empty collections of sets such that $B \not\subseteq A$ for every $A \in \Gamma_{\text{no}}, B \in \Gamma_{\text{yes}}$.*[7] *Sets in Γ_{yes} are called* authorized*, and sets in Γ_{no} are called* unauthorized*. If $\Gamma_{\text{no}} \cup \Gamma_{\text{yes}} = 2^P$ then Γ is called an access structure and will be denoted by the collection of authorized sets Γ_{yes}.*

We represent a subset of parties $A \subseteq P$ by its characteristic string $x_A = (x_1, \dots, x_k) \in \{0,1\}^n$, where for every $j \in [n]$ it holds that $x_j = 1$ if and only if $P_j \in A$. A partial access structure $\Gamma = (\Gamma_{\text{no}}, \Gamma_{\text{yes}})$ will also be described by the partial function $F : \{0,1\}^n \to \{0,1\}$, where $F(x_A) = 1$ for every subset of parties $A \in \Gamma_{\text{yes}}$ and $F(x_A) = 0$ for every set $A \in \Gamma_{\text{no}}$.

[6] Indeed, the fact that the channel abstraction captures previous scenarios suggests that this is a useful notion that may be also applied in future contexts.

[7] We do not require that $2^P \setminus \Gamma_{\text{no}}$ and Γ_{yes} are equal (this simplifies our presentation).

Definition 3.2 (Secret-sharing schemes)**.** *A* secret-sharing scheme*, with domain of secrets S, domain of random strings R, and finite domains of shares $S_1, \ldots, S_n$, is a deterministic function $\mathcal{D} : S \times R \to S_1 \times \cdots \times S_n$. A dealer distributes a secret $s \in S$ according to $\mathcal{D}$ by first sampling a random string $r \in R$ with uniform distribution, computing a vector of* shares *$\mathcal{D}(s, r) = (s_1, \ldots, s_n)$, and privately communicating each share s_i to party P_i. For a set $A \subseteq P$, we denote $\mathcal{D}_A(s, r)$ as the restriction of $\mathcal{D}(s, r)$ to its A-entries (i.e., the shares of the parties in A).*

A secret-sharing scheme $\mathcal{D}$ realizes a partial access structure $\Gamma = (\Gamma_{\text{no}}, \Gamma_{\text{yes}})$ if the following two requirements hold: (1) Perfect Correctness: The secret s can be reconstructed by any authorized set of parties. That is, for any set $B = \{P_{i_1}, \ldots, P_{i_{|B|}}\} \in \Gamma_{\text{yes}}$ 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 $r \in R$, it holds that* $\mathrm{Recon}_B(\mathcal{D}_B(s, r)) = s$*. (2) Perfect privacy: Any unauthorized set cannot learn anything about the secret from its shares. Formally, for any set $T = \{P_{i_1}, \ldots, P_{i_{|T|}}\} \in \Gamma_{\text{no}}$, every pair of secrets $s, s' \in S$, it holds that $\mathcal{D}_T(s, r) \equiv \mathcal{D}_T(s', r)$, where r is sampled with uniform distribution from R.*

The secret size *in a secret-sharing scheme $\mathcal{D}$ is defined as $\log |S|$ and the* complexity *of the scheme $\mathcal{D}$ is defined as the total share size $\sum_{1 \le i \le n} \log |S_i|$.*[8] *The scheme $\mathcal{D}$ is a* linear secret-sharing scheme over a finite field $\mathbb{F}$ *if $S = \mathbb{F}$, $R = \mathbb{F}^\ell$ for some integer $\ell \ge 1$, the sets $S_1, \ldots, S_n$ are vector spaces over $\mathbb{F}$, and the function $\mathcal{D} : \mathbb{F}^{\ell+1} \to S_1 \times \cdots \times S_n$ is a linear mapping over $\mathbb{F}$. By default, linearity is defined over the binary field $\mathbb{F}_2$.*

Conditional disclosure of secrets protocols. Next, we define k-server conditional disclosure of secrets (CDS) protocols, first presented in [30]. We consider a model where a set of k servers $Q = \{Q_1, \ldots, Q_k\}$ hold a secret s and a common random string r. In addition, every server Q_i holds an input x_i for some k-input function f. In a CDS protocol for f, for every $i \in [k]$, server Q_i sends a message to a referee, based on r, s, and x_i, such that the referee can reconstruct the secret s if $f(x_1, \ldots, x_k) = 1$, and it cannot learn any information about the secret s if $f(x_1, \ldots, x_k) = 0$.

Definition 3.3 (Conditional disclosure of secrets protocols)**.** *Let $f : X_1 \times \cdots \times X_k \to \{0, 1\}$ be a k-input function. A k-server* CDS protocol *$\mathcal{P}$ for f, with domain of secrets S, domain of common random strings R, and finite message domains $M_1, \ldots, M_k$, consists of k deterministic message computation functions* $\textsc{Enc}_1, \ldots, \textsc{Enc}_k$*, where $\textsc{Enc}_i : X_i \times S \times R \to M_i$ for every $i \in [k]$. For an input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$, secret $s \in S$, and randomness $r \in R$, we let $\textsc{Enc}(x, s, r) = (\textsc{Enc}_1(x_1, s, r), \ldots, \textsc{Enc}_k(x_k, s, r))$. We say that a protocol $\mathcal{P}$ is a CDS protocol for f if it satisfies the following properties: (1) Perfect correctness: There is a deterministic reconstruction function $\textsc{Dec} : X_1 \times \cdots \times X_k \times M_1 \times \cdots \times M_k \to S$ such that for every input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 1$, every secret $s \in S$, and every common random string $r \in R$, it holds that $\textsc{Dec}(x, \textsc{Enc}(x, s, r)) = s$. (2) Perfect privacy: For every input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 0$ and every pair of secrets $s, s' \in S$ it holds that $\textsc{Enc}(x, s, r) \equiv \textsc{Enc}(x, s', r)$, where r is sampled with uniform distribution from R.*

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

The protocol $\mathcal{P}$ is a linear CDS protocol over a finite field $\mathbb{F}$ if $S = \mathbb{F}$, $R = \mathbb{F}^\ell$ for some integer $\ell \ge 1$, $M_1, \ldots, M_k$ are vector spaces over $\mathbb{F}$, and the function $\textsc{Enc}_i : \mathbb{F}^{\ell+1} \to M_i$ is a linear function over $\mathbb{F}$ for every $i \in [k]$. By default, we take $\mathbb{F}$ to be the binary field $\mathbb{F}_2$.

Notations. We denote the logarithmic function with base 2 and base e by $\log$ and $\ln$, respectively. For $0 \le \alpha \le 1$, we denote the binary entropy of α by $\mathrm{H}_2(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$. Next, we present the standard approximation of the binomial coefficients.

[8] The complexity is sometimes defined to be the maximal share size, i.e., $\max_{1 \le i \le n}\{\log |S_i|\}$. However, since the two differ by at most a linear factor of n, the difference is not important in our context.

Fact 3.4. *Let n be an integer and let $k \in [n]$. Then, $\binom{n}{k} = \Theta(k^{-1/2} 2^{\mathrm{H}_2(k/n)n})$.*

Sets and strings. We use the notation $[n]$ to denote the set $\{1, \dots, n\}$. For a set A, we let 2^A denote the collection of all subsets of A, let $\binom{A}{k}$ denote the collection of all subsets of A of size k and let $\binom{A}{\leq k}$ denote the collection of all subsets of A of size at most k.

Given two binary strings of the same length, $a = a_1 a_2 ... a_n$ and $b = b_1 b_2 \dots b_n$, we say that $a \leq b$ if $a_i \leq b_i$ for every $i \in [n]$. We denote $\mathrm{wt}(a)$ as the Hamming weight of the string a.

4 Robust CDS: Definition and Construction

4.1 Definition of Robust CDS

In the definition of CDS protocols in [30] (as presented in Definition 3.3), if a server sends messages for two different inputs with the same randomness, then the privacy is not guaranteed and the referee can possibly learn information on the secret s. We generalize the notion of CDS protocols to robust CDS (RCDS) protocols, where the secret is hidden even if the referee sees multiple messages of servers computed on different inputs with the same randomness. Of course, this requirement makes sense only if all the corresponding inputs are zero inputs of f.

Definition 4.1 (Zero sets and robustness collections.)**.** *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a k-input function. We say that a set of inputs $Z \subseteq X_1 \times \cdots \times X_k$ is a* zero set *of f if $f(x) = 0$ for every $x \in Z$.*

*A robustness collection is a product of k collections of inputs $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_k \subseteq 2^{X_1} \times \cdots \times 2^{X_k}$, where each $\mathcal{Z}_i$ is downward closed, i.e., if $Z \in \mathcal{Z}_i$ and $Z' \subset Z$ then $Z' \in \mathcal{Z}_i$, and contains all singletons, i.e., $\{x_i\} \in \mathcal{Z}_i$ for every $x_i \in X_i$. A robustness collection is a (u,t)-*collection *if each of the collections $\mathcal{Z}_1, \dots, \mathcal{Z}_k$ contains at most u* maximal *sets, and each of these sets is of size at most t. We denote the collection by $(\mathcal{Z}_1, \dots, \mathcal{Z}_k)$.*

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

Definition 4.2 (Robust conditional disclosure of secrets (RCDS) protocols)**.** *Let $\mathcal{P}$ be a k-server CDS protocol for a k-input function $f : X_1 \times \cdots \times X_K \to \{0,1\}$ and $Z = Z_1 \times \cdots \times Z_k \subseteq X_1 \times \cdots \times X_k$ 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 $\textsc{Enc}(Z, s, r) \equiv \textsc{Enc}(Z, s', r)$. Let $\mathcal{Z}$ be a robustness collection. We say that $\mathcal{P}$ is a $\mathcal{Z}$-RCDS protocol if it is robust for every zero set $Z \in \mathcal{Z}$.*

For example, the original (non-robust) definition of privacy of CDS protocols is $\mathcal{Z}_1 \times \cdots \times \mathcal{Z}_k$-robust, where $\mathcal{Z}_i$ contains all singletons, i.e., $\mathcal{Z}_i = \{\{x_i\} : x_i \in X_i\} \cup \{\emptyset\}$. We next discuss some choices made in our definition of robustness.

Rectangles. Suppose that a 2-server CDS protocol is robust for a zero set Z that contains the inputs $(x,y), (x',y')$. This means that the messages $\textsc{Enc}((x,y), s, r) = (\textsc{Enc}_1(x, s, r), \textsc{Enc}_2(y, s, r))$, and $\textsc{Enc}((x',y'), s, r) = (\textsc{Enc}_1(x', s, r), \textsc{Enc}_2(y', s, r))$ perfectly hide the secret s. Observe that given these messages, the referee can also compute the messages $\textsc{Enc}((x',y), s, r)$ and $\textsc{Enc}((x,y'), s, r)$, which correspond to the inputs (x,y') and (x',y), i.e., the referee can try to reconstruct the secret for every input in the minimal combinatorial rectangle that contains $(x,y), (x',y')$.[9] For this reason, we always use combinatorial rectangles as sets for robustness.

[9] A set $Z \subseteq X_1 \times \cdots \times X_k$ is a combinatorial rectangle if it can be written as a product set $Z = Z_1 \times \cdots \times Z_k$, where $Z_i \subseteq X_i$.

Monotonicity. We also observe that if a protocol is robust for a zero set $Z = Z_1 \times \cdots \times Z_k$ then it is also robust for every sub-rectangle of Z. It will be convenient to keep this property even when Z is not a zero-set. That is, we will always make sure that if Z is a member of our robustness collection $\mathcal{Z}$ then so are all sub-rectangles of Z, i.e., $\mathcal{Z}_i$ will always be downward closed collection for every $i \in [k]$.

Product collections. For simplicity of notations, we focus on the case of *product collections*, i.e., $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_k$, where $\mathcal{Z}_i \subseteq 2^{X_i}$. We always require that $\mathcal{Z}_i$ contains all the singletons of X_i, thus, $\mathcal{Z}$-robustness implies privacy as defined in Definition 3.3.

Example 4.3 (t-RCDS). Consider the case where each server may output, simultaneously, messages for any subset of its inputs of size at most t. This notion, referred to as t-*RCDS*, can be captured by the product collection $(\mathcal{Z}_1, \ldots, \mathcal{Z}_k)$ where $\mathcal{Z}_i = \binom{X_i}{\leq t}$. Consequently, this is a (u, t)-robustness collection, where $u = \max_i(\binom{|X_i|}{t})$.

4.2 Construction of Robust CDS Protocols

In the rest of this section we show how to convert a CDS protocol for a function f to a $\mathcal{Z}$-RCDS protocol for f, where $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_k$ is a (u, t)-robustness collection. The message size in the resulting RCDS protocol is only $\tilde{O}(t \log u)^k$ times the message size of the CDS protocol. The construction of the RCDS protocol is done in k steps, where in the k'-th step we immunize the messages of the k' server, that is, we start with a protocol that is robust when servers $Q_1, \ldots, Q_{k'-1}$ can send messages for many inputs and servers $Q_{k'}, \ldots, Q_k$ can only send a single message and transform it to a protocol that is robust also when server $Q_{k'}$ can send messages for many inputs.

To simplify notation, we say that a CDS protocol is a $(\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_{k'})$-RCDS if it is $\mathcal{Z}$-RCDS for $\mathcal{Z} = (\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'}, \mathcal{Z}_{k'+1}, \ldots, \mathcal{Z}_k)$, where $\mathcal{Z}_i = \{\{x_i\} : x_i \in X_i\} \cup \{\emptyset\}$ for every $i \in \{k'+1, \ldots, k\}$ (i.e., $\mathcal{Z}_i$ contains all singletons).

Theorem 4.4 (Immunization Theorem). *Let $f : X_1 \times \cdots \times X_k \to \{0, 1\}$ be a function, $1 \leq k' \leq k$ be an integer, and $(\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'}) \subseteq 2^{X_1} \times \cdots \times 2^{X_{k'}}$ be a (u, t)-robustness collection. Suppose there is a $(\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'-1})$-RCDS protocol $\mathcal{P}_{k'-1}$ for a secret taken from a domain S in which the size of the messages of server Q_i, for $i \in [k]$, is c_i. Then, there is a $(\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'-1}, \mathcal{Z}_{k'})$-RCDS protocol $\mathcal{P}_{k'}$ for a secret taken from the domain S in which the size of the messages of server Q_i, for $i \in [k] \setminus \{k'\}$, is $\tilde{O}(t\, \mathrm{c}_i \log |X_i| \log u)$, and the size of the messages of server $Q_{k'}$ is $\tilde{O}_t(\mathrm{c}_{k'} \log |X_i| \log u)$. Moreover, if $\mathcal{P}_{k'-1}$ is linear, then so is $\mathcal{P}_{k'}$.*

The main idea of our construction for immunizing a server $Q_{k'}$ is to partition the input domain $X_{k'}$ of the server into sets and for each set to execute a CDS protocol with independent randomness. If the server sends encodings of a few inputs such that each input is in a different set, then the referee gets at most one encoding from each execution and the privacy of the CDS protocol implies the robustness of the new protocol. However, one partition of the inputs is not good for all sets of inputs so we use several partitions. We use a family of perfect hash functions to specify the partitions.

To reduce the message size in the construction, we use two levels of hashing. To provide robustness for t inputs, we first use a family of hash functions with range of size $2t$. This will ensure that for at least one partition, the server sends encodings of at most $\log t$ inputs in the same execution of the CDS protocol. Thus, the CDS protocol should be robust for $\log t$ inputs; this is done by a second level of hashing using a family of perfect hash functions with range of size $\log^2 t$. The idea of using two levels of hashing is similar to the construction of traitor-tracing schemes in [21].

By taking the best known CDS constructions and iteratively applying the immunization theorem, we derive the following results.

Theorem 4.5. *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a k-input function, where $|X_i| \leq 2^\ell$, and $(\mathcal{Z}_1, \ldots, \mathcal{Z}_k) \subseteq 2^{X_1} \times \cdots \times 2^{X_k}$ be a (u,t)-robustness collection. Then,*

- *There is a $(\mathcal{Z}_1, \ldots, \mathcal{Z}_k)$-RCDS protocol with a 1-bit secret, where the size of the messages of each server is $2^{\tilde{O}(\sqrt{k\ell})}\tilde{O}(t)^{k-1}(\ell \log u)^k$, and*
- *If k is odd, then there is a linear $(\mathcal{Z}_1, \ldots, \mathcal{Z}_k)$-RCDS protocol with a 1-bit secret, where the size of the messages of each server is $\tilde{O}(t2^\ell \log u)^{(k-1)/2}$.*

The rest of the section is organized as follows. In Section 4.2.1, we provide some results on families of hash functions. In Section 4.3, we prove an immunization lemma, which is used for both levels of hashing, and prove the immunization theorem (Theorem 4.4) using the immunization lemma. Finally, in Section 4.4, we show how to use the immunization theorem to construct RCDS protocols and prove Theorem 4.5.

4.2.1 Families of Hash Functions

We next present the definition of a family of t'-collision free perfect hash functions; the original definition of perfect hash functions [28] refers to the case that $t' = 1$.

Definition 4.6 (Families of t'-collision free hash functions). *A set of functions $H_{n,t,t',v} = \{h_d : [n] \to [v] : d \in [\ell]\}$ is a family of t'-collision free hash functions for a collection $\mathcal{T} \subseteq \binom{[n]}{\leq t}$ if for every set $T \in \mathcal{T}$ there exists at least one function $h \in H_{n,t,t',w}$ for which for every $b \in [v]$ it holds that $|\{x \in T : h(x) = b\}| \leq t'$, that is, h restricted to T is at most t'-to-one. A family $H_{n,t,v}$ is a family of perfect hash functions if it is a family of 1-collision free hash functions.*

The following lemma is a well-known result, which can be proved, e.g., using the probabilistic method.

Lemma 4.7. *Let n be an integer and $t \in [\sqrt{n}]$. Then, there exists a family of perfect hash functions (i.e., 1-collision free) $H_{n,t,t^2} = \{h_i : [n] \to [t^2] : i \in [\ell]\}$, where $\ell = 16t \ln n$.*

The following lemma is proved in Appendix C.2.

Lemma 4.8. *Let n be an integer, $t \in \{15, \ldots, n/2\}$, $\mathcal{T} \subseteq \binom{[n]}{\leq t}$, and u be the number of maximal sets in $\mathcal{T}$. Then, there exists a family of $\log t$-collision free hash functions $H_{n,t,\log t,2t}$ of size $\ell = 16 \ln u$.*

4.3 The Immunization

The next lemma, called the immunization lemma, improves the immunization of server $Q_{k'}$, that is, it takes a protocol that is robust when $Q_{k'}$ sends encodings of $t' < t$ messages, and constructs a protocol that is robust when $Q_{k'}$ sends encodings of t messages. This is done using a few copies of the original protocol and a family of t'-collision free hash functions.

Lemma 4.9. *Let $f, k', \mathcal{Z}_1, \ldots, \mathcal{Z}_{k'}$, and t be as in Theorem 4.4 and $\mathcal{Z}'_{k'} = \binom{X_i}{\leq t'}$ for some integer $t' \leq t$. Suppose there is a $(\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'-1}, \mathcal{Z}'_{k'})$-RCDS protocol $\mathcal{P}$ with domain of secrets S in which the size of the messages of server Q_i, for $i \in [k]$, is c_i. Furthermore, suppose there is a family of t'-collision free hash functions $H_{|X_{k'}|,t,t',v} = \{h_1, \ldots, h_\ell\}$. Then, there is a $(\mathcal{Z}_1, \ldots, \mathcal{Z}_{k'-1}, \mathcal{Z}_{k'})$-RCDS protocol $\mathcal{P}'$ with secrets from S in which the size of the messages of server Q_i, for $i \in [k] \setminus \{k'\}$, is $O(\mathrm{c}_i\, v\ell)$ and the size of the messages of server $Q_{k'}$ is $O(\mathrm{c}_{k'}\, \ell)$. Moreover, the transformation preserves linearity.*

Proof. Let R be the set of random strings of $\mathcal{P}$ and let $\textsc{Enc}_i$ be the encoding of server Q_i in this protocol. The encoding function $\textsc{Enc}'_i$ of Q_i in the RCDS protocol $\mathcal{P}'$ for f is as follows:

- Common inputs: a secret $s \in S$, randomness r consisting of $s_1, \dots, s_{\ell-1} \in S$ and $(r_{d,j})_{d \in [\ell], j \in [v]}$, where each $r_{d,j} \in R$.

- Private input of server Q_i: $x_i \in X_i$.

- Let $s_\ell = s - (s_1 + \cdots + s_{\ell-1})$, where the sum is in S.
  (If S is not a group, choose an injective mapping from S to $\mathbb{Z}_{|S|}$ and use addition modulo $|S|$.)

- If $i \neq k'$, then $\textsc{Enc}'_i(x_i, s, r) = \big(\textsc{Enc}_i(x_i, s_d, r_{d,j})\big)_{d \in [\ell], j \in [v]}$.

- $\textsc{Enc}'_{k'}(x_{k'}, s, r) = \big(\textsc{Enc}_{k'}(x_{k'}, s_d, r_{d, h_d(x_{k'})})\big)_{d \in [\ell]}$.

Notice that the encoding of server $Q_{k'}$ contains one encoding from $\mathcal{P}$ for each s_d; this will provide the robustness. In contrast, the encoding of any other server contains many encodings from $\mathcal{P}$ for each s_d (each one with an independent random string); this will ensure the correctness of the protocol.

We first show the correctness of the RCDS protocol $\mathcal{P}'$. For input $(x_1, \dots, x_k) \in X_1 \times \cdots \times X_k$ such that $f(x_1, \dots, x_k) = 1$, the referee can reconstruct s_d, for every $d \in [\ell]$, using the decoding function of $\mathcal{P}$ on the encodings of the inputs $x_1, \dots, x_k$ with the secret s_d and random string $r_{d, h_d(x_{k'})}$. Overall, the referee can learn all the strings $s_1, \dots, s_\ell$, so it can reconstruct the secret s by summing these strings.

For the robustness of the protocol $\mathcal{P}'$, let $Z_1 \times \cdots \times Z_k$ be a zero set of f such that $Z_i \in \mathcal{Z}_i$ for every $1 \leq i \leq k'$ and $|Z_i| = 1$ for every $k'+1 \leq i \leq k$, and let $Z_{k',d,j} = \{x \in Z_{k'} : h_d(x) = j\}$ for every $d \in [\ell]$ and $j \in [v]$. Since $H_{|X_{k'}|, t, t', v}$ is a family of t'-collision free hash functions, there is at least one $d \in [\ell]$ for which h_d restricted to $Z_{k'}$ is at most t'-to-one. Fix a $j \in [v]$; by the collision freeness, $|Z_{k',d,j}| \leq t'$. The referee gets the encodings of the inputs in $Z_{k',d,j}$ from server $Q_{k'}$ in the execution of the RCDS protocol $\mathcal{P}$ for the secret s_d and random string $r_{d,j}$, i.e., it gets at most t' encodings from $Q_{k'}$ for $r_{d,j}$. Furthermore, in this execution the referee gets encoding of messages of server Q_i on inputs $Z_i \in \mathcal{Z}_i$ for $1 \leq i \leq k'-1$ and an encoding of at most one message from server Q_i for $k'+1 \leq i \leq k$. Since $\mathcal{P}$ is a $(\mathcal{Z}_1, \dots, \mathcal{Z}_{k'-1}, \mathcal{Z}'_{k'})$-RCDS protocol, the referee cannot learn any information about s_d from this execution, that is, for $Y^{d,j} = Z_1 \times \cdots \times Z_{k'-1} \times Z_{k',d,j} \times Z_{k'+1} \times \cdots \times Z_k$ and two secrets $s_d, s'_d \in S$

$$\textsc{Enc}(Y^{d,j}, s_d, r_{d,j}) \quad \equiv \quad \textsc{Enc}(Y^{d,j}, s'_d, r_{d,j}).$$

For two secrets $s, s' \in S$, fix $s_1, \dots, s_{\ell-1}$. Let $s_\ell = s - (s_1 + \cdots + s_{\ell-1})$, and $s'_e = s_e$ for $e \neq d$ and $s'_d = s_d + s' - s$. Note that $s_1, \dots, s_\ell$ are used when the secret is s, while $s'_1, \dots, s'_\ell$ are used when the secret is s'. Since the random strings $\{r_{e,j}\}_{e \in [\ell], j \in [v]}$ are statistically independent,

$$(\textsc{Enc}(Y^{e,j}, s_e, r_{e,j}))_{e \in [\ell], j \in [v]} \quad \equiv \quad (\textsc{Enc}(Y^{e,j}, s'_e, r_{e,j}))_{e \in [\ell], j \in [v]}.$$

Hence, the referee cannot learn any information on the secret s.

The message size of each server Q_e, for $e \neq k'$, in protocol $\mathcal{P}'$ is $O(v|H_{|X_{k'}|, t, t', v}|) = O(v\ell)$ times its message size in $\mathcal{P}$ and the message size of server $Q_{k'}$ in protocol $\mathcal{P}'$ is $O(|H_{|X_{k'}|, t, t', v}|) = O(\ell)$ times its message size in $\mathcal{P}$. $\square$

We next prove Theorem 4.4, the Immunization Theorem, by using two levels of the construction of Lemma 4.9.

Proof of Theorem 4.4. Let $t' = \log t$, $\mathcal{Y}_{k'} = \binom{X_{k'}}{\leq t'}$, and $H_{|X_{k'}|, t', t'^2}$ be the family of perfect hash functions of size $\ell = O(t' \log |X_{k'}|)$ guaranteed by Lemma 4.7. Note that this family is 1-collision free. By Lemma 4.9 applied to $\mathcal{P}_{k'-1}$ and $H_{|X_{k'}|, t', t'^2}$, there exists a $(\mathcal{Z}_1, \dots, \mathcal{Z}_{k'-1}, \mathcal{Y}_{k'})$-RCDS protocol $\mathcal{P}'_{k'}$, where the message size of server Q_i, for $i \neq k'$, is $O(c_i \cdot t'^2 \cdot t' \log |X_{k'}|) = O(c_i \log^3 t \log |X_{k'}|)$, and the message size of server $Q_{k'}$ is $O(c_{k'} \log t \log |X_{k'}|)$.

We next apply Lemma 4.9 using $\mathcal{P}'_{k'}$. Let $H_{|X_{k'}|,t,\log t,2t}$ be the family of $\log t$-collision free hash functions for $\mathcal{Z}_{k'}$ of size $\ell = O(\log u)$ guaranteed by Lemma 4.8 (where u is the number maximal sets in $\mathcal{Z}_{k'}$). By Lemma 4.9 applied to $\mathcal{P}'_{k'}$ and $H_{|X_{k'}|,t,\log t,2t}$, there exists a $(\mathcal{Z}_1, \dots, \mathcal{Z}_{k'-1}, \mathcal{Z}_{k'})$-RCDS protocol where the message size of server Q_i, for $i \neq k'$, is $O((c_i \log^3 t \log |X_{k'}|) \cdot 2t \cdot \log u) = \tilde{O}(c_i t \log |X_{k'}| \log u)$, and the message size of server $Q_{k'}$ is $O(c_{k'} \log t \log |X_{k'}| \log u) = \tilde{O}_t(c_{k'} \log |X_{k'}| \log u)$. $\square$

4.4 Constructing RCDS protocols

In this section we present our constructions of RCDS protocols, proving Theorem 4.5. We begin by applying Theorem 4.4 k times, immunizing all parties.

Lemma 4.10. *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a k-input function, where $|X_i| \leq 2^\ell$, and $(\mathcal{Z}_1, \dots, \mathcal{Z}_k) \subseteq 2^{X_1} \times \cdots \times 2^{X_k}$ be a (u,t)-robustness collection. Suppose there exists a CDS for f where the message size of server Q_i is c_i. Then, there is a $(\mathcal{Z}_1, \dots, \mathcal{Z}_k)$-RCDS protocol, where the size of the messages of server Q_i, for $i \in [k]$, is $c_i \cdot \tilde{O}(t)^{k-1} \cdot (\ell \log u)^k$. Moreover, this transformation preserves linearity.*

Proof. We use Theorem 4.4 k times iteratively, starting with the original CDS protocol. In the k'-th iteration, we transform a $(\mathcal{Z}_1, \dots, \mathcal{Z}_{k'-1})$-RCDS protocol to a $(\mathcal{Z}_1, \dots, \mathcal{Z}_{k'-1}, \mathcal{Z}_{k'})$-RCDS protocol. The communication overhead in each step is $\tilde{O}_t(\ell \log u)$ for the immunized server and $\tilde{O}(t\ell \log u)$ for all other servers. Since every server is immunized once, the total communication overhead is $\tilde{O}(t)^{k-1}(\ell \log u)^k$. $\square$

We move on, and prove Theorem 4.5.

Proof of Theorem 4.5. The non-linear RCDS protocol is obtained by applying Lemma 4.10 to the CDS protocol of [42], which has message size $2^{\tilde{O}(\sqrt{k\ell})}$.

For the linear RCDS protocol, we start with the variant of the linear CDS protocol of [12] with message size $O(2^{\ell(k-1)/2})$ (a protocol with a similar message size also appears in [42]), and prove in Lemma D.1 that when k is odd, this protocol is already immune for $(k+1)/2$ servers. Thus, we need to immunize, using Theorem 4.4, only $(k-1)/2$ servers. The resulting RCDS protocol has message size $\tilde{O}(t 2^\ell \log u)^{(k-1)/2}$. $\square$

4.5 (a,b)-monotone RCDS

An important example of a robustness collection is the case of *monotone robustness* in which whenever a server Q_i, holding a string $x \in \{0,1\}^\ell$, sends all the messages that correspond to inputs $x' \leq x$. In fact, we consider a more refined version of this scenario where the above happens only when x is of Hamming weight at most b and x' is of Hamming weight at least a for some integers $a < b \leq \ell$.

Definition 4.11 ((a,b)-monotone RCDS)**.** *For an input $x \in X_i$, define the set $S_{x,a} \subset X_i$ as*

$$S_{x,a} = \{x' \in X_i : x' \leq x \text{ and } \operatorname{wt}(x') \geq a\}.$$

*Furthermore, for $i \in \{1, \dots, k\}$, let $\mathcal{Z}_i = \{S_{x,a}\}_{x \in X_i, a \leq \operatorname{wt}(x) \leq b}$. An (a,b)-*monotone RCDS protocol *is a $(\mathcal{Z}_1, \dots, \mathcal{Z}_k)$-RCDS protocol.*

Let $X_i \subseteq \{0,1\}^\ell$. Each string of weight exactly b correspond to a maximal set of $\mathcal{Z}_i$. Hence, the collection $\mathcal{Z}$ is a (u,t)-collection where

$$u = \binom{\ell}{b} \qquad \textit{and} \qquad t = \sum_{i=a}^{b} \binom{b}{i}. \tag{1}$$

5 Secret-Sharing Schemes for General Access Structures

In this section, we realize any access structure based on RCDS protocols. By plugging the RCDS constructions from Theorem 4.5, we prove our main theorem – Theorem 2.1. We follow the outline sketched in the introduction: We begin by realizing somewhat-regular access structures, then move to handle mid-slice access structures, and end up by realizing general access structures. Throughout this section we will identify an access structure with its characteristic Boolean function F, as described in Definition 3.1. We also make use of partial (or promise) access structures.

5.1 Somewhat-regular Secret-Sharing from RCDS

Definition 5.1 ((k, a, b)-somewhat-regular access structure)**.** *Let Π be a partition of the set of n parties P to k equal-sized sets $(I_1, \dots, I_k)$. A (partial) access structure $\Gamma = (\Gamma_{\text{no}}, \Gamma_{\text{yes}})$ over n parties is (Π, a, b)-somewhat-regular if for every $A \in \Gamma_{\text{no}} \cup \Gamma_{\text{yes}}$ and every $i \in [k]$,*

$$a \leq |A \cap I_i| \leq b. \tag{2}$$

In other words, Γ puts no restriction on sets $A \subset [n]$ that violates (2) *for some i. We sometimes omit Π and refer to Γ as being (k, a, b)*-somewhat-regular.

Remark 5.2 (Function notation). Using the terminology of functions and strings, the partial function F describing a (k, a, b)-somewhat-regular access structure is defined only on n-bit strings $x \in \{0, 1\}^n$ with the following property. For every $i \in [k]$, the string $x[I_i]$, obtained by restricting x to the index set I_i, has Hamming weight of at least a and at most b. That is, x can be parsed to $(x_1, \dots, x_k)$ where $x_i \in \{0, 1\}^{n/k}$ and we care only about the case where the Hamming weight of each x_i is in between a and b.

Remark 5.3. We can take a fully defined access structure F and "puncture it" according to a given partition Π and parameters (a, b) and derive a (Π, a, b)-somewhat-regular version of F, denoted by $F_{\Pi,a,b}$, where $F_{\Pi,a,b}$ is undefined on inputs x for which some x_i has weight greater than b or smaller than a.

As the first step towards secret-sharing schemes for general access structures, we build secret-sharing schemes for any (k, a, b)-somewhat-regular access structures over n parties based on (a, b)-monotone robust CDS protocols for k servers. (The latter notion is defined in Definition 4.11.)

Construction 5.4. Let Π be a partition of n parties to k equal-sized sets $(I_1, \dots, I_k)$ and let F be a (Π, a, b)-somewhat-regular access structure over n parties for integers $a < b$. We share a secret s as follows.

1. Let $X_i = \{0, 1\}^{n/k}$ and let $f : X_1 \times \cdots \times X_k \to \{0, 1\}$ be some predicate that agrees with F.[10] Sample a random string r for a k-server (a, b)-monotone RCDS protocol $\mathcal{P} = (\textsc{Enc}_i)_{i \in [k]}$ for f.

2. For every $i \in k$, and $x_i \in X_i$ such that the Hamming weight $w = \text{wt}(x_i)$ of x_i is in the interval $[a, b]$, compute the message of the i-th server of $\mathcal{P}$:

$$y(i, x_i, s) := \textsc{Enc}_i(x_i, r, s),$$

   and share it between all the w parties in (the set that corresponds to) x_i via a w-out-of-w secret-sharing scheme (using fresh randomness R_{x_i} for each x_i). We denote the share of $j \in x_i$ by $y(i, x_i, s, j)$.

3. The share of the j-th party (which is in a set I_i) is $(y(i, x_i, s, j))_{x_i : j \in x_i}$.

[10] We say that a pair of partial functions f and g agree with each other if they take the same value on every input x for which both functions are defined.

Lemma 5.5. *Construction 5.4 realizes F with share sizes of $m \cdot \sum_{j=a}^{b} \binom{n/k}{j}$ for each party, where m is the message size of the underlying RCDS protocol.*

Proof. We start by proving correctness. Let $x = (x_1, \dots x_k)$ be a string with $a \leq \mathrm{wt}(x_i) \leq b$ for every $i \in [k]$ and $F(x) = 1$. For every $i \in [k]$, the parties represented by x_i can reconstruct the RCDS message $\textsc{Enc}_i(x_i, r, s)$ of the i-th server of $\mathcal{P}$. Since the RCDS predicate f agrees with F, it holds that $f(x) = 1$, and by the correctness of the RCDS protocol the parties can compute the secret s from the k RCDS messages.

We move on to prove privacy. Fix a string $x = (x_1, \dots, x_k)$ that corresponds to an unauthorized set. Consider a pair of secret s_0 and s_1. Our goal is to show that the corresponding shares of the parties in x, denoted by $\mathcal{D}_x(s_0)$ and $\mathcal{D}_x(s_1)$, are identically distributed. To see this, consider a modified construction in which for every $i \in [k]$ and every $u \in X_i$ for which $u \not\leq x_i$, the random variable $y(i, u, s_0)$ (which is shared via a $\mathrm{wt}(u)$-out-of-$\mathrm{wt}(u)$ secret-sharing scheme) is replaced with some fixed string $y(i, u)$ of appropriate length (say the all zero string). We claim that, in the modified scheme, the view $\mathcal{D}'_x(s_0)$ of the parties in x is distributed identically to $\mathcal{D}_x(s_0)$. Indeed, since $u \not\leq x_i$ there exists at least one party $j \in u$ that does not participate in x. Therefore, by the privacy of the $\mathrm{wt}(u)$-out-of-$\mathrm{wt}(u)$ secret-sharing scheme, and since each $y(i, u, s_0)$ is shared with fresh randomness, the random variables $\mathcal{D}_x(s_0)$ and $\mathcal{D}'_x(s_0)$ are identically distributed. Similarly, $\mathcal{D}_x(s_1)$ and $\mathcal{D}'_x(s_1)$ are identically distributed, and so it suffices to show that $\mathcal{D}'_x(s_0)$ is distributed identically to $\mathcal{D}'_x(s_1)$. Let us condition on some fixing of the shares of $y(i, u)$ for all $u \in X_i$ for which $u \not\leq x_i$. By the robustness of the RCDS, the remaining random variables

$$\{y(i, v, s_0) : i \in [k], v \leq x_i\} \qquad \text{and} \qquad \{y(i, v, s_1) : i \in [k], v \leq x_i\}$$

are identically distributed, so the corresponding shares are also identically distributed and privacy follows.

For every $j \in [n]$, the share of party P_j contain a share of the RCDS protocol for every string x_i of length n/k and weight between a and b such that $j \in x_i$. Then the share size of P_j is $m \cdot \sum_{j=a}^{b} \binom{n/k}{j}$. $\square$

5.2 Mid-slice Secret-Sharing from Somewhat-Regular Secret-Sharing

Definition 5.6 (Mid-slice access structure [40])**.** *An n-party access structure F is a δ* mid-slice *access structure with parameter $\delta \in \left(0, \frac{1}{2}\right)$ if:*

1. *F takes the value 0 for every input x of Hamming weight $\mathrm{wt}(x) < \left(\frac{1}{2} - \delta\right) n$;*

2. *F takes the value 1 for every input x of Hamming weight $\mathrm{wt}(x) > \left(\frac{1}{2} + \delta\right) n$;*

A "middle-slice" input x of weight $\left(\frac{1}{2} - \delta\right) n \leq \mathrm{wt}(x) \leq \left(\frac{1}{2} - \delta\right) n$ can be assigned any value (as long as monotonicity is preserved).

A partial mid-slice *access structure is defined similarly except that we drop the requirements (1) and (2), and F is undefined over light inputs ($\mathrm{wt}(x) < \left(\frac{1}{2} - \delta\right) n$) and over heavy inputs ($\mathrm{wt}(x) > \left(\frac{1}{2} + \delta\right) n$).*

We now turn to build a secret-sharing scheme for mid-slice access structures from (k, a, b)-somewhat-regular secret-sharing schemes. Fix some parameter $\delta \in \left(0, \frac{1}{2}\right)$ and set a proximity parameter ϵ to be $n^{-0.1}$. Let $\Pi = (I_1, \dots, I_k)$ be a partition of $[n]$ to $k = \sqrt{n}$ subsets of size $n/k = \sqrt{n}$ each.[11] In the following, we say that an input $x \in \{0,1\}^n$ is *good* for the i-th block of Π, if the sub-string $x_i \in \{0,1\}^{\sqrt{n}}$ is of Hamming weight at least $\left(\frac{1}{2} - \delta - \epsilon\right)\sqrt{n}$ and at most $\left(\frac{1}{2} + \delta + \epsilon\right)\sqrt{n}$. We say that x is good for the partition Π if x is *good* for all the blocks $i \in [k]$ of Π. If x is not good then it is called bad. We will use the following lemma.

[11]The choice of $\sqrt{n}$ is somewhat arbitrary and any choice of block size $\omega(\log n) < |B| < o(n)$ suffices for the final result.

Lemma 5.7. *For every constant $\delta > 0$, there exists a collection of $\ell = O(n)$ partitions $\Pi_1, \dots, \Pi_\ell$ of $[n]$ to $\sqrt{n}$ subsets of size $\sqrt{n}$ each, such that every n-bit string x of Hamming weight $\left(\frac{1}{2} - \delta\right) n \leq \mathrm{wt}(x) \leq \left(\frac{1}{2} + \delta\right) n$ is good for at least 0.7ℓ of the partitions.*

The hidden constant in the big-O notation depends on the constant δ.

Proof. We use the probabilistic method to choose at random such a collection of size $\ell = O(n)$, and see that with positive probability all inputs are good for at least 0.7ℓ of the partitions.

Fix an input x of weight $\left(\frac{1}{2} - \delta\right) n \leq \mathrm{wt}(x) \leq \left(\frac{1}{2} + \delta\right) n$. We start the analysis by sampling a partition Π. We first focus on a single block i, and denote by $Y_{j,i}$ the indicator random variable that is equal to 1 if and only if the j-th bit of the i-th block is 1. For every $i \in [\sqrt{n}]$, the $\sqrt{n}$ variables $\{Y_{j,i}\}_{j \in [\sqrt{n}]}$ are negatively associated (see Claim C.6). We now denote $Y_i = \sum_j Y_{j,i}$ to be the random variable representing the number of ones of x that are placed in the i-th block of Π. Due to the linearity of expectation, and to x being of "middle" weight, we get that the expectation μ of Y_i satisfies

$$\left(\frac{1}{2} - \delta\right) \sqrt{n} \leq \mu \leq \left(\frac{1}{2} + \delta\right) \sqrt{n}$$

The probability that x is bad for the i-th block is a sum of two probabilities, that x puts too many ones or too few. These probabilities behave the same asymptotically, so we will analyze only the former probability. By the negative associativity we can use the Chernoff bound, and get

$$\Pr\left[Y_i \geq \left(\frac{1}{2} + \delta + \epsilon\right) \sqrt{n}\right] \leq e^{-\frac{\epsilon^2 c^2 \mu}{3}} = e^{-\Omega(c^2 n^{0.3})},$$

where $c = 1/\left(\frac{1}{2} + \delta\right)$ and the last equation follows since $\epsilon = n^{-0.1}$.

Now by union bound over all blocks, the probability that x is bad for the partition Π is at most

$$p = \sqrt{n} e^{-\Omega(c^2 n^{0.3})} = o(1)$$

Finally, if we independently sample ℓ partitions, the probability that x is bad for at least 0.3ℓ of the partitions is, by a Chernoff bound, at most $2^{-\Omega(\ell)}$. By taking $\ell = Cn$ for sufficiently large constant C, the latter probability is smaller than 2^{-n}, so the lemma follows by applying a union bound over all possible inputs. □

We can now realize a mid-slice access structure.

Lemma 5.8. *Let F be a mid-slice access structure over n parties with parameter $\delta \in \left(0, \frac{1}{2}\right)$. Then, F can be realized by a secret-sharing scheme with share size of $m' \cdot O(n \log n)$, assuming that any (k, a, b)-somewhat-regular access structure can be realized by a secret-sharing scheme with share size of m', where*

$$k = \sqrt{n}, \quad a = \left(\frac{1}{2} - \delta - n^{-0.1}\right) \sqrt{n}, \quad \text{and} \quad b = \left(\frac{1}{2} + \delta + n^{-0.1}\right) \sqrt{n}.$$

Proof. We start by considering a partial mid-slice access structure. Recall (Definition 5.6) that such an access structure is defined only over the middle slice, i.e., over inputs whose Hamming weight is in the interval $[0.5n \pm \delta n]$.

Construction 5.9. We realize such an access structure F as follows:

1. Let $L = (\Pi_1, \dots, \Pi_\ell)$ be the list of partitions of length $\ell = O(n)$ promised by Lemma 5.7.

2. Share s into ℓ shares $(\sigma_1, \dots, \sigma_\ell)$ via an $\ell/2$-out-of-ℓ threshold secret-sharing scheme (using fresh randomness).

3. For every $i \in [\ell]$ share each σ_i with a different random string r_i by a secret-sharing scheme realizing the (k, a, b)-somewhat-regular access structure $F_{\Pi_i,a,b}$ (as defined in Remark 5.3).

We analyze the construction. Fix some input x of Hamming weight $\left(\frac{1}{2} - \delta\right) \leq \mathrm{wt}(x) \leq \left(\frac{1}{2} - \delta\right)$. Let $I \subset [\ell]$ denote the set $\{i : x \text{ is good for } \Pi_i\}$, and recall that, by Lemma 5.7, the set I is of size at least 0.7ℓ. Observe that $F(x) = F_{\Pi_i,a,b}(x)$ for every $i \in I$. If $F(x) = 1$ then at least 0.7ℓ shares $\sigma_i, i \in I$, can be reconstructed by the parties in x and s can be recovered. If $F(x) = 0$ then at least 0.7ℓ shares $\sigma_i, i \in I$, are kept perfectly hidden (due to the privacy of $F_{\Pi_i,a,b}$) and so s remains perfectly hidden (i.e., we can perfectly simulate the view of the parties that participate in x).[12]

We use Shamir's secret-sharing to implement the threshold part and so each σ_i is of length $O(\log \ell)$. Hence, the share size per party is $O(m'\ell \log \ell) = O(m'n \log n)$ where m' is the size of shares of the underlying somewhat-regular scheme.

We move on to handle the case where F is defined over all inputs. This part of the construction is quite start-forward. Recall (Definition 5.6) that such an access structure takes the value 0 over light inputs, the value 1 over heavy inputs and may take arbitrary values over the middle slice. Letting F' denote the partial mid-slice access structure that agrees with F over the mid slice, we realize F as follows:

1. Share s via a $\left(\left(\frac{1}{2} + \delta\right) n + 1\right)$-out-of-$n$ secret-sharing scheme and give the i-th share, denoted by u_i, to the i-th party.

2. Share s via 2-out-of-2 secret sharing into s_0 and s_1.

3. Share s_0 via a $\left(\frac{1}{2} - \delta\right) n$-out-of-$n$ secret-sharing scheme and give the i-th share, denoted by v_i, to the i-th party.

4. Share s_1 to all parties according to F' (using Construction 5.9) and give the i-the share, denoted by w_i, to the i-th party.

Correctness: Any input x of weight at least $\left(\frac{1}{2} + \delta\right) n+1$ can reconstruct s via the u shares, and any middle-slice input x which is authorized (i.e., $F(x) = 1$) can recover s_0 and s_1 (via the v and w shares) and can therefore recover s. Privacy: A coalition that corresponds to a light inputs learns nothing from the u shares and from the v shares (due to the privacy of the threshold schemes) and therefore learns nothing about s. A medium-slice coalition that is unauthorized (i.e., $F(x) = 0$) learns nothing from the u shares (due to the privacy of the threshold scheme) and learns nothing from the w shares (due to the privacy of the F' scheme) and so it learns nothing on s.

Since each w_i is of length $O(m'n \log n)$ and the bit-length of u_i and v_i is $O(\log n)$, the share size per party is $O(m'n \log n) + O(\log n) = O(m'n \log n)$. □

5.3 The Exponent of Mid-slice Access Structures

Let us denote by $\mathbf{M}(\delta)$ the exponent of mid-slice access structure with parameter δ. Namely,

$$\mathbf{M}(\delta) = \limsup_{n\to\infty} \max_{F\in\mathcal{M}(\delta,n)} \frac{1}{n} \log \mathsf{SS}(F),$$

where $\mathcal{M}(\delta, n)$ is the family of all n-party δ-mid-slice access structures. The linear exponent, $\mathbf{M}_\ell(\delta)$, is defined analogously except that $\mathsf{SS}(F)$ is replaced with $\mathsf{LSS}(F)$.

[12] Note that when $i \notin I$ there are no guarantees on the share σ_i, e.g., it is possible that $F(x) = 0$ and the parties in x can recover σ_i or that $F(x) = 1$ and the parties in x would have no information on σ_i. However, since we use a threshold scheme to share s, this does not affect the correctness and privacy of the construction.

In [40] it was shown that

$$\mathbf{M}(\delta) \leq \mathrm{H}_2(0.5-\delta) + 0.2h(10\delta) + 10\delta - 0.2\log(10),$$

and for the linear case

$$\mathbf{M}_\ell(\delta) \leq \mathrm{H}_2(0.5-\delta) + 0.2\,\mathrm{H}_2(10\delta) + 2\log(26)\delta - 0.1\log(10).$$

Based on Lemma 5.8, and our constructions for robust CDS protocols we prove the following bound:

Lemma 5.10. *For every $\delta \in \left(0, \frac{1}{2}\right)$ the following holds*

$$\mathbf{M}(\delta) \leq \begin{cases} \left(\frac{1}{2}+\delta\right) \mathrm{H}_2\left(\frac{1/2-\delta}{1/2+\delta}\right) & \text{if } \delta < 1/6 \\ \left(\frac{1}{2}+\delta\right) & \text{if } \delta \geq 1/6 \end{cases}.$$

For the linear case, when $\delta < 1/6$, it holds that

$$\mathbf{M}_\ell(\delta) \leq \frac{1}{2} + \frac{1}{2}\left(\frac{1}{2}+\delta\right) \mathrm{H}_2\left(\frac{1/2-\delta}{1/2+\delta}\right).$$

Proof. Using the secret-sharing schemes with properties promised by Lemma 5.5 and Lemma 5.8, a mid-slice secret-sharing scheme for an access structure $F_{\mathrm{mid},\delta}$ over n parties can be realized from an (a,b)-monotone RCDS protocols with k servers for predicates $f : (\{0,1\}^\ell)^k \to \{0,1\}$ where $k = \ell = \sqrt{n}$, and $a = \left(\frac{1}{2}-\delta-\epsilon\right)\sqrt{n}$, $b = \left(\frac{1}{2}+\delta+\epsilon\right)\sqrt{n}$, where $\epsilon = n^{-0.1}$. Assuming that these RCDS protocols have communication complexity m, we get a secret-sharing scheme for $F_{\mathrm{mid},\delta}$ with complexity

$$m \cdot O(n\log n) \cdot \sum_{j=a}^{b} \binom{n/k}{j} = m \cdot \mathrm{poly}(n) \cdot O(2^{\sqrt{n}}) = m \cdot 2^{o(n)}.$$

Similarly, we get a linear secret-sharing of complexity $m_\ell \cdot 2^{o(n)}$ where m_ℓ is the share size of an underlying linear robust CDS protocols. So, the exponent is derived solely from the cost of the underlying robust CDS protocol. By definition (see (1)) an (a,b)-monotone collection is a (u,t)-collection where

$$u = \binom{\ell}{b} \qquad \text{and} \qquad t = \sum_{i=a}^{b} \binom{b}{i}. \tag{3}$$

So, by applying Theorem 4.5, we can realize (a,b)-monotone robust CDS with complexity of

$$m = 2^{o(n)} \cdot \left[\sum_{j=0}^{2\delta\sqrt{n}} \binom{\left(\frac{1}{2}+\delta+\epsilon\right)\sqrt{n}}{\left(\frac{1}{2}-\delta-\epsilon\right)\sqrt{n}+j}\right]^{\sqrt{n}-1} \cdot \left[\sqrt{n}\log\binom{\sqrt{n}}{\left(\frac{1}{2}+\delta+\epsilon\right)\sqrt{n}}\right]^{\sqrt{n}}.$$

For the linear case, Theorem 4.5 yields the bound $m_\ell \leq O_t(2^\ell t \log u)^{\sqrt{n}/2}$ which, under our choice of parameters, simplifies to

$$m_\ell \leq O(2^{n/2}) \cdot \left[\sum_{j=0}^{2\delta\sqrt{n}} \binom{\left(\frac{1}{2}+\delta+\epsilon\right)\sqrt{n}}{\left(\frac{1}{2}-\delta-\epsilon\right)\sqrt{n}+j} \cdot \log\binom{\sqrt{n}}{\left(\frac{1}{2}+\delta+\epsilon\right)\sqrt{n}}\right]^{\sqrt{n}/2}.$$

Both terms m and m_ℓ have different asymptotic behavior for $\delta < 1/6$ or $\delta \geq 1/6$. Specifically, recalling that $\epsilon = n^{-0.1}$, and using the standard entropy-based bound for the binomial coefficients (Fact 3.4), we get

$$m = \begin{cases} 2^{\left(\frac{1}{2}+\delta\right) \mathrm{H}_2\left(\frac{1/2-\delta}{1/2+\delta}\right)n+o(n)} & \text{if } \delta < 1/6 \\ 2^{\left(\frac{1}{2}+\delta\right)n+o(n)} & \text{if } \delta \geq 1/6 \end{cases},$$

and for linear CDS we get

$$m_\ell = 2^{\frac{1}{2}n+\frac{1}{2}\left(\frac{1}{2}+\delta\right) \mathrm{H}_2\left(\frac{1/2-\delta}{1/2+\delta}\right)n+o(n)}$$

when $\delta < 1/6$, and $m_\ell = \Omega(2^n)$ when $\delta \geq 1/6$. □

5.4 Putting it altogether (Proof of Theorem 2.1)

In [40] the exponent of general access structures was reduced to the exponent of mid-slice access structures. To realize an access structure F, they realize the mid-slice of F and in addition they realize the access structure whose minterms are the light minterms of F and the access structure whose maxterms are the heavy maxterms of F (where a minterm is a minimal authorized set and a maxterm is a maximal unauthorized set); the latter two access structures are realized by the trivial schemes. In [3] more efficient schemes realizing the latter two access structures were presented (using schemes with exponent smaller than 1). Using the current notation, we get the following lemma:

Lemma 5.11 ([3])**.** *For every $\delta \in \left(0, \frac{1}{2}\right)$ it holds that*

$$\mathbf{S} \leq \max(\mathbf{X}'(\delta), \mathbf{M}(\delta)) \quad \textit{and} \quad \mathbf{S}_\ell \leq \max(\mathbf{X}'(\delta), \mathbf{M}_\ell(\delta))$$

where

$$\mathbf{X}'(\delta) = \mathrm{H}_2\left(\frac{1}{2} - \delta\right) - \left(\frac{1}{2} - \delta\right) \log\left(\frac{\frac{1}{2} + \delta}{\frac{1}{2} - \delta}\right).$$

The proof of Theorem 2.1 now follows directly from the combination of Lemma 5.10 and Lemma 5.11 with $\delta \sim 0.1429$ for the general case, and $\delta_\ell \sim 0.09$ for the linear case. Indeed, Lemma 5.11 yields exponents of 0.637 in the general case and 0.762 in the linear case.

6 Immunizing CDS Servers: Abstraction and an Alternative Construction

In this section we provide an abstraction of the construction in Theorem 4.4, which shows how to 'immunize" a single server. The goal is to transform a CDS protocol that is robust over some collection $I \times \mathcal{Y}$ (where I consists only of singletons of the input of the first server) to a CDS protocol that is robust over $\mathcal{Z} \times \mathcal{Y}$ for some collection $\mathcal{Z}$. We begin with a general template that abstracts the work of [32] in Section 6.1. The template requires a pair of hash function and secret-sharing scheme with specific properties. These are chosen later in Section 6.2 in a way that immunizes a server over a (u, t) collection, of u maximal sets of maximal cardinality t. This will allow us to turn any CDS protocol to one that is robust over any robustness collection, by immunizing one server at a time.

6.1 A Template for Immunizing a Single Server

We follow the outline suggested in the introduction. That is, we secret-share s to N shares and send each one of them via an independent copy of a CDS protocol. The immunized server will only use a subset of these copies that will be determined based on the input of the immunized server via the aid of some hash function.

Construction 6.1. Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a predicate for a CDS protocol $\mathcal{P} = (\textsc{Enc}_i)_{i\in[k]}$ for secrets in S and randomness domain R. Let N be an integer, $H : X_1 \to 2^{[N]}$ be a mapping, and let $\mathcal{D}$ be a (randomized) sharing function that maps a secrets $s \in T$ into a vector of N shares $(s'_1, ..., s'_N) \in S^N$ using randomness from the domain R_0. Define the following new CDS for a secret $s \in T$, and randomness domain of $R_0 \times R^N$:

1. Given $s \in T$ and $(r_0, r_1, \ldots, r_L) \in R_0 \times R^N$, each server applies $\mathcal{D}$ to s and r_0 and generates the shares $(s'_1, ..., s'_N)$.
2. Each server j computes a vector v_j of N messages
$$v_j = (\textsc{Enc}_j(x_j, s'_1, r_1), \ldots, \textsc{Enc}_j(x_j, s'_N, r_N)).$$
3. The first server computes the set $H(x_1) \subseteq [N]$ and outputs only the entries $(v_1[i])_{i\in H(x_1)}$.
4. Every other servers $j > 1$ outputs its entire vector of messages v_j.

Before analyzing the construction we need the following simple definition. The set of *collisions* of a collection of sets $A_1, \ldots, A_q$ is the set of elements that appear in at least two of the sets $A_1, \ldots, A_q$.

Lemma 6.2. *Suppose that:*

1. *For every $x \in X_1$ the set $H(x)$ is an authorized set of the secret-sharing scheme $\mathcal{D}$*
2. *The underlying protocol $\mathcal{P}$ is a $\mathcal{I} \times \mathcal{Y}$-RCDS protocol, where $\mathcal{I} = \{\{x\} : x \in X_1\} \cup \emptyset$.*
3. *For every $Z \in \mathcal{Z}_1$, the set of collisions of the set system $\{H(x)\}_{x\in Z}$ is a non-authorized set of the secret-sharing scheme $\mathcal{D}$.*

Then Construction 6.1 is a $\mathcal{Z}_1 \times \mathcal{Y}$-RCDS protocol.

Proof. We claim that the protocol is perfectly correct. Indeed, fix an input (x_1, y) for which $f(x_1, y) = 1$. Then, the decoder computes the set $A = H(x_1)$ and for every $i \in A$ it applies the decoder of the original CDS to i-th component of the transcript, i.e., to $(v_1[i], \ldots, v_k[i])$. By the correctness of the underlying CDS, the value s'_i is recovered. Since A is an authorized set of the secret-sharing scheme (by the lemma's hypothesis) the shares $(s'_i)_{i\in A}$ can be used to recover the secret s.

Next, we prove that the protocol is robust over a zero-set $Z \times Y$, where $Z \in \mathcal{Z}_1$ and $Y \in \mathcal{Y}$. For secret $s \in T$, input (x_1, y), and randomness $r = (r_0, \ldots, r_L)$, let $D(x_1, y, r, s)$ denote the concatenation of the messages that are sent by all the servers. Also, let $D(x_1, y, r, s)[i]$ denote the concatenation of the i-th messages of the servers. (If $i \notin H(x_1)$ then the first entry of $D(x_1, y, r, s)[i]$ is taken to be $\perp$.) Fix a pair of secrets s and s', we show that the random variables

$$D(s) := (D(x_1, y, r, s))_{(x_1,y)\in Z}, \qquad \text{and} \qquad D(s') := (D(x_1, y, r, s'))_{(x_1,y)\in Z},$$

induced by a uniform choice of r, are distributed identically.

Let $A \subseteq [N]$ be the set of collisions of $\{H(x_1)\}_{x_1 \in Z}$. To prove that $D(s)$ and $D(s')$ are identical, we show that (1) The restriction of $D(s)$ and $D(s')$ to the indices in A is identically distributed; and (2) Conditioned on every fixing of $D(s)[A]$ and $D(s')[A]$, for every $i \notin A$ the random variables $D(s)[i]$ and $D(s')[i]$ are identically distributed, and are independent of all other i's outside A.

We prove (1). Since the set A of collisions is a non-authorized set, the A-shares $(s_i)_{i \in A}$ of s and the A-shares $(s'_i)_{i \in A}$ of s', induced by a uniform choice of r_0, are identically distributed. Hence, for uniformly chosen r_0 and $r_A = (r_i)_{i \in A}$ (and every fixing of $(r_i)_{i \notin A}$), the A-components of $D(s)$ and $D(s')$ are identically distributed.

We move on to (2). Fix some arbitrary r_0, r_A and let $(s_i)_{i \in \overline{A}}$ and $(s'_i)_{i \in \overline{A}}$ denote the resulting $\overline{A}$-shares of s and s'. For every $i \in \overline{A}$, let

$$I_i := \{x_1 \in Z : i \in H(x_1)\}$$

denote the set of inputs for which the first server "speaks" in the i-th session. By assumption, $|I_i| \leq 1$ and therefore the underlying CDS is robust over the zero-set $I_i \times Y$. It follows that the i-th components of $D(s)$ and $D(s')$ are identically distributed when r_i is uniformly chosen. Since the r_i's are chosen independently, we conclude that $D(s)$ is distributed identically to $D(s')$. $\square$

6.2 Instantiating the Template

The above template can be instantiated based on perfect hash functions as shown in Section 4.2. We provide here an alternative instantiation based on sparse hash functions.

Recall that Construction 6.1 makes use of a secret-sharing scheme over a set of N parties. In the following we let $N = N_1 \cdot N_2$, and view the set $[N]$ as $[N_1] \times [N_2]$. Correspondingly a subset M of $[N]$ can be represented as $N_1 \times N_2$ binary matrix. We will need a (partial) access structure for which M is an authorized set if at least β-fraction of the rows have at least γN_2 ones, and M is unauthorized if there are at most 0.5β fraction of the rows with more than $0.6\gamma N_2$ ones. (The parameters N_1, N_2, β, γ will be defined below.) We refer to this access structure as a $(\beta, \gamma, N_1, N_2)$-access structure. Such an access structure can be easily realized by applying a two-levels threshold secret sharing (or ramp-secret sharing). For example, distribute the secret to N_1 shares $s_1, \ldots, s_{N_1}$, say via Shamir's secret-sharing scheme with threshold $0.8\beta N_1$, and then distribute each s_i to N_2 shares $s_{i,1}, \ldots, s_{i,N_2}$ via Shamir's secret-sharing scheme with threshold of $0.8\gamma N_2$. The share $s_{i,j}$ is held by the (i,j)-th party.

Fact 6.3. *For every positive integers N_1, N_2 and reals $\beta, \gamma \in (0,1)$ and every field $\mathbb{F}$ of size at least $\max(N_1, N_2) + 1$, there exists a secret-sharing scheme that realizes the $(\beta, \gamma, N_1, N_2)$-access structure and maps a secret $s \in \mathbb{F}$ to the shares $(s_{i,j})_{i \in [N_1], j \in [N_2]} \in \mathbb{F}^{N_1 \times N_2}$.*

Next, we need a hash function H that maps $x \in X$ to subsets of $[N_1] \times [N_2]$ (or to $N_1 \times N_2$ binary matrices). It will be convenient to view H as a sequence of N_1 hash functions $h_1, \ldots, h_{N_1} : X \to 2^{[N_2]}$ one for each row. That is, $(i,j) \in H(x)$ if $j \in h_i(x)$. The collection should be compatible with the $(\beta, \gamma, N_1, N_2)$-access structure and with the collection of input tuples against which we should immunize.

Definition 6.4. *Let X be a set and let $Z_1, \ldots, Z_u$ be a sequence of t-subsets of X, i.e., $Z_i \in \binom{X}{t}$ for every i. A $(\beta, \gamma, N_1, N_2)$ hash function family $H = \{h_1, \ldots, h_{N_1}\}, h_i : X \to 2^{[N_2]}$ for $\mathcal{Z}$ satisfies the following properties:*

1. *For every $x \in X$ exactly βN_1 functions in H, map x to a subset of $[N_2]$ of size at least γN_2.*
2. *For every set of inputs Z_i, $i \in [u]$, for all but $\beta/2$-fraction of the hash functions $h_j, j \in [N_1]$ it holds that the family of sets $\{h_j(x)\}_{x \in Z_i}$ has at most $0.6\gamma N_2$ collisions.*

In Section 6.3 we will prove the following lemma.

Lemma 6.5. *For every X and $Z_1, \ldots, Z_u \in \binom{X}{t}$, there exists $(\beta, \gamma, N_1, N_2)$-hash function H with $N_1 = O(t \log u)$ and $N_2 = \log t \cdot \mathrm{polylog}(t, \log u)$ where for every x the set $H(x) = \bigcup_i h_i(x)$ is of size exactly $\beta\gamma N_1 N_2$, and behaves asymptotically as $\log^2 u \cdot \mathrm{polylog}(t, \log u)$.*

We can now immunize a single server (the $(k'+1)$-th server) against u different t-subsets $Z_1, \ldots, Z_u$ of the server's input domain (and any subset of these sets). The communication overhead will be $\log^2 u \cdot \mathrm{polylog}(t, \log u)$ for the immunized server, and $\tilde{O}(t \log^2 u)$ for the others. Formally, we prove the following theorem.

Theorem 6.6. *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a predicate. Suppose that $\mathcal{P}$ is a CDS for f that achieves robustness over some k'-product collection $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_{k'})$ for some $0 \le k' < k$. Let $\mathcal{Z} \in 2^{X_{k'+1}}$ be a downward-closed collection of subsets of $X_{k'+1}$ that contains at most u maximal sets $Z_1, \ldots, Z_u$ each of cardinally of at most t. Then the protocol $\mathcal{P}$ can be converted into a CDS $\mathcal{P}'$ for f which is robust over the $(k'+1)$-product collection $(\mathcal{Y}, \mathcal{Z})$ in which the communication complexity of the $k'+1$ server grows by a factor of $\log^2 u \cdot \mathrm{polylog}(t, \log u)$, and for all the servers grows by a factor of $\tilde{O}(t \log^2 u)$.*

Proof. We immunize the $(k'+1)$-th server against the sets $Z \in \mathcal{Z}$ by applying Construction 6.1 to the original CDS (while treating the $k'+1$ party as the first party).

Let H be $(\beta, \gamma, N_1, N_2)$-hash family for $(Z_1, \ldots, Z_u)$ as promised by Lemma 6.5 where $N_1 = O(t \log u)$, $N_2 = \log u \cdot \mathrm{polylog}(t, \log u)$ and β, γ are as promised by the lemma. Take $N = N_1 \times N_2 = \tilde{O}(t \log^2 u)$. Let $\mathbb{F}$ be a finite field of size at least $\max(N_1, N_2) + 1 < O(t \log u)$ and let $\mathcal{D}$ denote a secret sharing that realizes the $(\beta, \gamma, N_1, N_2)$-access structure (as promised in Fact 6.3) that maps a secret $s \in \mathbb{F}$ to $\mathbb{F}^{N_1 \times N_2}$. Furthermore, let us slightly modify the underlying CDS into a CDS $\mathcal{P}_1$ that supports secrets from $\mathbb{F}$. If the original domain S is larger than $\mathbb{F}$ we can simply take $\mathcal{P}$ as $\mathcal{P}_1$. Otherwise, this can be achieved by concatenating several elements from S. This modification increases the communication complexity by a factor of at most $\log |\mathbb{F}| = O(\log t + \log \log u)$.

Instantiate Construction 6.1 with the underlying CDS, with the mapping H, and with the secret-sharing $\mathcal{D}$. (Recall that we view subsets of $[N]$ as $N_1 \times N_2$ binary matrices and we abuse notation and let $H(x)$ denote the matrix whose (i,j)-th cell is 1 iff $j \in h_i(x)$.) Fix some zero-set $Z \times Y$ where $Z \in \mathcal{Z}$, and $Y \subset \bigtimes_{j<k'} \mathcal{Y}_j \times \bigtimes_{j>k'} \binom{X_j}{1}$. Recall that Z is a subset of Z_i for some $i \in [u]$. Lemma 6.2 shows that the protocol is robust over $(Z \times Y)$, since the following three conditions hold:

1. By assumption, the underlying CDS protocol is robust for every set of the form $I \times Y$ for every singleton $I \subset Z$.

2. Since H is $(\beta, \gamma, N_1, N_2)$-hash function for the maximal sets $(Z_1, \ldots, Z_u)$ and since $Z \subseteq Z_i$ for some $i \in [u]$, it follows that the set of collisions of $\{H(x)\}_{x \in Z}$ is a non-authorized set of the scheme $\mathcal{D}$. (Written as a matrix, this set has at most $0.5\beta N_1$ rows that have at least $0.6\gamma N_2$ ones.)

3. For every $x \in X_{k'+1}$ the set $H(x)$ is an authorized set of the secret-sharing scheme $\mathcal{D}$.

A promised by Lemma 6.5, the immunized server sends $\beta\gamma N$ messages of the protocol $\mathcal{P}_1$ and each other server sends N CDS messages. Compared to the original CDS $\mathcal{P}$, the communication complexity of the immunized server grows by a multiplicative factor of $O(\beta\gamma N \log |\mathbb{F}|) = \log^2 u \cdot \mathrm{polylog}(t, \log u)$, and for all other servers by a factor of $O(N \log |\mathbb{F}|) = \tilde{O}(t \log^2 u)$. □

By iterating over all parties, we derive the following corollary.

Corollary 6.7. *Let $\mathcal{P}$ be a k-server CDS protocol for a predicate $f : X_1 \times \cdots \times X_k \to \{0,1\}$ in which each server has a communication complexity ℓ. Let $\mathcal{Z} = (\mathcal{Z}_1, \cdots, \mathcal{Z}_k)$ be a (t, q)-product collection. Then, the CDS $\mathcal{P}$ can be transformed into a robust CDS $\mathcal{P}'$ which is robust over the collection $\mathcal{Z}$ with multiplicative communication overhead of $\tilde{O}(q^{k-1} \log^{2k} t)$. Furthermore, if the original $\mathcal{P}$ is linear then so is $\mathcal{P}'$.*

Proof. We use Theorem 6.6 k times iteratively starting with the original CDS. At the i-th iteration, we transform a CDS which is robust over the $(i-1, u, t)$-product collection $(\mathcal{Z}_1, \cdots, \mathcal{Z}_{i-1})$ into a CDS which is robust over the (i, u, t)-product collection $(\mathcal{Z}_1, \cdots, \mathcal{Z}_i)$. The communication overhead in each step is $\log^2 u \cdot \mathrm{polylog}(t, \log u)$ for the immunized server, and $\tilde{O}(t \log^2 u)$ for all others. Since every server is immunized once, the overall communication grows by a factor of $\tilde{O}(t^{k-1} \log^{2k} u)$.

Finally, assume that the original protocol $\mathcal{P}$ is linear over a field $\mathbb{F}$. In order to preserve $\mathbb{F}$-linearity, it suffices to employ an $\mathbb{F}$-linear secret sharing in Fact 6.3. This is immediate when $\mathbb{F}$ is larger than $\max(N_1, N_2) + 1$; When $\mathbb{F}$ is smaller this can be achieved by combining a pair of ramp-secret sharing (e.g., by using random linear codes) over $\mathbb{F}$ or by using an implementation over a larger extension field. □

6.3 Proof of Lemma 6.5

The proof is via the probabilistic method. We define the family H in two steps. We start by selecting for each x a random βN_1-subset $I_x \subset [N_1]$. In addition, we select N_1 random functions $h'_i, \forall i \in [N_1]$ from X to γN_2-subsets of $[N_2]$. The final mapping is defined as follows: For every $i \in [N_1]$ let $h_i(x)$ be the empty set if $i \notin I_x$ and otherwise let $h_i(x) = h'_i(x)$. In matrix terminology, the 1-cells of the matrix $H(X)$ that corresponds to x are selected by first choosing a random βN_1 rows and then choosing random γN_2 cells in each of these rows. Clearly, the mapping satisfies the first property of Definition 6.4 and the "Moreover" part. Let

$$N_1 = 2t \log u, \quad \beta = \frac{c \log N_1}{t}, \quad N_2 = c'(2\beta t)^2 \log(3tN_1) = O(\log^2 N_1 \log(N_1 u)),$$

and $\gamma = 1/(4c^2 \log^2 N_1)$ where c and c' are some positive constants. Fix some tuple of inputs $Z = (x_1, \ldots, x_t)$. We show that, except with probability $1/(3u)$, the collection H satisfies the second property of Definition 6.4 for the input tuple Z, and so the lemma follows by a union bound over all input tuples.

Analyzing I. We say that a "row" $j \in [N_1]$ gets a copy of x if $j \in I_x$. Since each $x \in Z$ is placed in a random βN_1 subset, each row j gets a copy of every $x \in Z$ independently with probability β. Call a row j *bad* if it holds more than $2\beta t$ of the elements of Z, and let χ_j be an indicator random variable that takes the value 1 if the j-th row is bad. By a Chernoff bound, χ_j gets the value 1 with probability $p < e^{-\beta t/3}$. Say that $I = \{I_x\}$ is good for Z if there are less than $\log t < 0.5\beta N_1$ bad rows. Since the random variables $\chi_1, \ldots, \chi_{N_1}$ are negatively associated (see proof in Claim C.5), the probability of having at least $\log u$ bad cells can be upper-bounded by

$$\binom{N_1}{\log u} p^{\log u} \leq (N_1 \cdot p)^{\log u} \leq \frac{1}{3u},$$

where the last inequality holds as long as $p < 1/10N_1$ which holds for $\beta = c \log N_1 / q$ for sufficiently large constant c.

Analyzing h'_i. Fix some I and consider a good row $i \in [N_1]$ (i.e., has at most $\ell = 2\beta t$ inputs). We say that h'_i is *good* for Z, if h'_i induces at most $0.6\gamma N_2$ colliding cells. Here an index $j \in [N_2]$ is counted as a collision if at least two distinct inputs $x \neq x' \in Z$ have $i \in h(x) \cap h(x')$ and $j \in h'_i(x) \cap h'_i(x')$. (In matrix notation, both x and x' put 1 in their (i, j)-th cell.) In the above we restrict the attention to good rows; If the i-th row is bad then we treat h'_i as being vacuously good.

Claim 6.8. *Let $\epsilon = 1/(3uN_1)$, and recall that $\gamma = 1/\ell^2$ and that $N_2 = c'\ell^2 \log(1/\epsilon)$. Then, for sufficiently large constant c', the following holds. For every $i \in [N_1]$, with probability $1 - \epsilon$ over the choice of h'_i, the function h'_i is good (i.e., it yields at most $0.6\gamma N_2$ collisions).*

Proof of claim. Fix some good $i \in [N_1]$. The function $h'_i(x)$ maps every input x independently to a random γN_2-subsets of $[N_2]$. Therefore every index $j \in [N_2]$ gets x (i.e., $j \in h'_i(x)$) independently with probability γ. We define an indicator random variable ζ_j that takes the value 1 if the j'th index in the i'th row gets more than one input mapped into it. By a union bound,

$$\Pr[\zeta_j = 1] \leq \sum_{x \neq x' : i \in I_x \cap I_{x'}} \Pr[j \in h'_i(x)] \cdot \Pr[j \in h'_i(x')].$$

Since i is a good row, there are at most $\binom{\ell}{2} < \ell^2/2$ pairs $x \neq x'$ for which i is in both I_x and $I_{x'}$. Hence,

$$\Pr[\zeta_j = 1] < 0.5\ell^2\gamma^2.$$

Next we define the random variable $\zeta = \sum_j \zeta_j$ representing the number of collisions in the i'th row. By the linearity of expectation and since $\gamma\ell^2 = 1$ it holds that

$$\mathrm{E}[\zeta] < 0.5\ell^2\gamma^2 N_2 = 0.5\gamma N_2.$$

Finally, since the variables ζ_j are negatively associated (from the same reasons as the variables in Claim C.5), we can apply the Chernoff bound and conclude that

$$\Pr[\zeta > 0.6\gamma N_2] \leq \exp(-\Omega(\gamma N_2)) < \epsilon$$

where the inequality holds for sufficiently large constant c'. □

Since $\epsilon = 1/(3tN_1)$, we can apply union bound over all $i \in [N_1]$ we conclude that all h_i's are good except with probability $1/3u$. Overall, the event that I and $h'_1, \ldots, h'_{N_1}$ are all good for all the u predefined input vectors $Z_1, \ldots, Z_u$ happens with probability $1 - u(1/3u + 1/3u) > 1/3$, and the proof follows.

References

[1] B. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In B. Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 118–134. Springer-Verlag, 2001.

[2] B. Applebaum and B. Arkis. On the power of amortization in secret sharing: d-uniform secret sharing and CDS with constant information rate. In A. Beimel and S. Dziembowski, editors, *TCC 2018*, volume 11239 of *LNCS*, pages 317–344. Springer, 2018.

[3] B. Applebaum, A. Beimel, O. Farràs, O. Nir, and N. Peter. Secret-sharing schemes for general and uniform access structures. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 441–471. Springer-Verlag, 2019.

[4] N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer-Verlag, 2014.

[5] D. Beaver, J. Feigenbaum, J. Kilian, and P. Rogaway. Security with low communication overhead. In *CRYPTO '90*, volume 537 of *LNCS*, pages 62–76, 1990.

[6] A. Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion, 1996.

[7] A. Beimel. Secret-sharing schemes: A survey. In *Coding and Cryptology – Third International Workshop, IWCC 2011*, volume 6639 of *LNCS*, pages 11–46. Springer, 2011.

[8] A. Beimel and B. Chor. Universally ideal secret-sharing schemes. *IEEE Trans. on Information Theory*, 40(3):786–794, 1994.

[9] A. Beimel, O. Farràs, Y. Mintz, and N. Peter. Linear secret-sharing schemes for forbidden graph access structures. In Y. Kalai and L. Reyzin, editors, *TCC 2017*, volume 10678 of *LNCS*, pages 394–423. Springer-Verlag, 2017.

[10] A. Beimel, O. Farràs, and N. Peter. Secret sharing schemes for dense forbidden graphs. In V. Zikas and R. De Prisco, editors, *SCN 2016*, volume 9841 of *LNCS*, pages 509–528, 2016.

[11] A. Beimel, A. Gabizon, Y. Ishai, and E. Kushilevitz. Distribution design. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 81–92, 2016.

[12] A. Beimel and N. Peter. Optimal linear multiparty conditional disclosure of secrets protocols. In T. Peyrin and S. D. Galbraith, editors, *ASIACRYPT 2018*, volume 11274 of *LNCS*, pages 332–362. Springer-Verlag, 2018.

[13] A. Beimel and N. Peter. Secret-sharing from robust conditional disclosure of secrets. Cryptology ePrint Archive, Report 2019/522, 2019. `https://eprint.iacr.org/2019/522`.

[14] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for noncryptographic fault-tolerant distributed computations. In *20th STOC*, pages 1–10, 1988.

[15] J. C. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In S. Goldwasser, editor, *CRYPTO '88*, volume 403 of *LNCS*, pages 27–35. Springer-Verlag, 1988.

[16] M. Bertilsson and I. Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In J. Seberry and Y. Zheng, editors, *AUSCRYPT '92*, volume 718 of *LNCS*, pages 67–79. Springer-Verlag, 1992.

[17] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. of the 1979 AFIPS National Computer Conference*, volume 48 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.

[18] G. R. Blakley and C. A. Meadows. Security of ramp schemes. In G. R. Blakley and D. Chaum, editors, *CRYPTO '84*, volume 196 of *LNCS*, pages 242–268. Springer-Verlag, 1984.

[19] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *20th STOC*, pages 11–19, 1988.

[20] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, and V. Vaikuntanathan. Secure computation from random error correcting codes. In M. Naor, editor, *EUROCRYPT 2007*, volume 4515 of *LNCS*, pages 291–310. Springer-Verlag, 2007.

[21] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Information Theory*, 46(3):893–910, 2000.

[22] B. Chor and E. Kushilevitz. Secret sharing over infinite domains. *J. of Cryptology*, 6(2):87–96, 1993.

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

[24] Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. In J. Feigenbaum, editor, *CRYPTO '91*, volume 576 of *LNCS*, pages 457–469. Springer-Verlag, 1991.

[25] D. Dubhashi, V. Priebe, and D. Ranjan. Negative dependence through the FKG inequality, 1996.

[26] D. Dubhashi and D. Ranjan. Balls and bins: A study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998.

[27] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation. In *26th STOC*, pages 554–563, 1994.

[28] M. L. Fredman, J. Komlós, and E. Szemerédi. Storing a sparse table with 0(1) worst case access time. *J. ACM*, 31(3):538–544, 1984.

[29] R. Gay, I. Kerenidis, and H. Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In R. Gennaro and M. Robshaw, editors, *CRYPTO 2015*, volume 9216 of *LNCS*, pages 485–502. Springer-Verlag, 2015.

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

[31] M. Göös, T. Pitassi, and T. Watson. Zero-information protocols and unambiguity in arthur-merlin communication. *Algorithmica*, 76(3):684–719, 2016.

[32] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Functional encryption with bounded collusions via multi-party computation. In *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, 2012.

[33] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *13th CCS*, pages 89–98, 2006.

[34] V. Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20:7186, 2000.

[35] Y. Ishai and E. Kushilevitz. On the hardness of information-theoretic multiparty computation. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 439 – 455. Springer-Verlag, 2004.

[36] M. Ito, A. Saito, and T. 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).

[37] K. Joag-Dev and F. Proschan. Negative association of random variables with applications. *The Annals of Statistics*, 11(1):286–295, 1983.

[38] S. Jukna. *Boolean Function Complexity – Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[39] M. Karchmer and A. Wigderson. On span programs. In *8th Structure in Complexity Theory*, pages 102–111, 1993.

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

[41] T. Liu, V. Vaikuntanathan, and H. Wee. Conditional disclosure of secrets via non-linear reconstruction. In J. Katz and H. Shacham, editors, *CRYPTO 2017*, volume 10401 of *LNCS*, pages 758–790. Springer-Verlag, 2017.

[42] T. Liu, V. Vaikuntanathan, and H. Wee. Towards breaking the exponential barrier for general secret sharing. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 567–596. Springer-Verlag, 2018.

[43] C. McDiarmid. On the method of bounded differences. *Surveys in combinatorics*, 141(1):148–188, 1989.

[44] M. Mitzenmacher and E. Upfal. *Probability and Computing*. Cambridge University Press, 2005.

[45] M. Naor and A. Wool. Access control and signatures via quorum secret sharing. In *3rd CCS*, pages 157–167, 1996.

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

[47] T. Pitassi and R. Robere. Lifting nullstellensatz to monotone span programs over any field. In *50th STOC*, pages 1207–1219, 2018.

[48] R. Robere, T. Pitassi, B. Rossman, and S. A. Cook. Exponential lower bounds for monotone span programs. In *57th FOCS*, pages 406–415, 2016.

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

[50] B. Shankar, K. Srinathan, and C. Pandu Rangan. Alternative protocols for generalized oblivious transfer. In S. Rao, M. Chatterjee, P. Jayanti, C. S. Ram Murthy, and S. K. Saha, editors, *9th ICDCN*, volume 4904 of *Lecture Notes in Computer Science*, pages 304–309. Springer-Verlag, 2008.

[51] T. Tassa. Generalized oblivious transfer by secret sharing. *Designs, Codes and Cryptography*, 58(1):11–21, 2011.

[52] B. Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 53–70. Springer-Verlag, 2011.

[53] H. Wee. Dual system encryption via predicate encodings. In Y. Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 616–637. Springer-Verlag, 2014.

A A Simple General Schemes with Exponent Less Than One

In this section we present a relatively simple construction of a secret-sharing scheme for an arbitrary n-party access structure with share size 2^{cn} for a constant $c < 1$. To achieve this goal, we present a simple 2-server RCDS protocol in Appendix A.1 and a reduction from secret-sharing to 2-server RCDS protocols in Appendix A.2. The purpose of this section is pedagogical and its aim is to give a full description of this scheme without relying on undescribed schemes, e.g., on the CDS scheme of [42] (which uses a construction of matching vectors of [34]). To understand this section, the reader needs the definitions given in Sections 3 and 4.1; no material from other sections is needed.

A.1 A 2-Server Robust CDS Protocols

We say that an RCDS protocol is a (t_1, t_2)-RCDS if it is robust for every zero-set $Z = Z_1 \times Z_2$ such that $|Z_1| \le t_1$ and $|Z_2| \le t_2$. In this section, we present a 2-server $(|X|, t)$-RCDS protocol for a function $f : X \times Y \to \{0, 1\}$ (that is, the robustness is guaranteed when server Q_1 can send unbounded number of messages and server Q_2 can send at most t messages). We start by showing that a CDS protocol described in [12] (inspired by the protocol of [29]) is robust when server Q_1 can send unbounded number of messages and server Q_2 can send only one message.

Protocol $\mathcal{P}_2$

The secret: A bit $s \in \{0, 1\}$.
Inputs: Q_1 and Q_2 hold the inputs $x \in X$ and $y \in Y$, respectively.
Common randomness: The two servers hold $|Y| + 1$ uniformly distributed and independent random bits $r_0, r_1, \ldots, r_{|Y|} \in \{0, 1\}$.
The protocol:

1. Q_1 sends to the referee the bit $m_x = s \oplus r_0 \oplus \bigoplus_{i \in Y, f(x,i)=0} r_i$.
2. Q_2 sends to the referee the bits $m_y = (r_0, r_1, \ldots, r_{y-1}, r_{y+1}, \ldots, r_{|Y|})$.
3. If $f(x, y) = 1$, the referee computes $m_x \oplus r_0 \oplus \bigoplus_{i \in Y, f(x,i)=0} r_i$.

Figure 1: A 2-server CDS protocol $\mathcal{P}_2$ for a function $f : X \times Y \to \{0, 1\}$.

Lemma A.1. *Let $f : X \times Y \to \{0, 1\}$ be a function. Then, protocol $\mathcal{P}_2$, described in Figure 1, is a 2-server $(|X|, 1)$-RCDS protocol for f in which the message size of Q_1 is 1 and the message size of Q_2 is $|Y|$.*

Proof. For the correctness of the protocol $\mathcal{P}_2$, consider inputs x, y such that $f(x, y) = 1$. In this case r_y is not part of the exclusive-or in the message m_x sent by Q_1 and server Q_2 sends all r_i's except for r_y. Thus, the referee can recover s from m_x and m_y as described in $\mathcal{P}_2$.

For the robustness of the protocol, assume that Q_2 sends the message of input $y \in Y$ and Q_1 sends multiple messages for a subset of inputs $Z \subseteq X$, such that $f(x, y) = 0$ for every $x \in Z$. We prove below that the probability of these messages is the same for $s = 0$ and $s = 1$. Recall that the referee gets the bits $r_0, \ldots, r_{|Y|}$ except for r_y from Q_2 and the bit

$$m_x = s \oplus r_0 \oplus \bigoplus_{i \in Y, f(x,i)=0} r_i = (s \oplus r_y) \oplus r_0 \oplus \bigoplus_{i \in Y \setminus \{y\}, f(x,i)=0} r_i$$

for every $x \in Z$ from Q_1. For every $x \in Z$, the element r_y acts as a one-time-pad protecting s in m_x, that is, if the messages $(m_x)_{x \in Z}, m_y$ are generated with common randomness $r_0, r_1, \ldots, r_{|Y|}$ and the secret $s = 0$, then the same messages are generated from the common randomness $r_0, r_1, \ldots, r_{y-1}, \overline{r_y}, r_{y+1}, \ldots, r_{|Y|}$ and the secret $s = 1$. $\square$

Next, we show how to transform the above CDS protocol to a $(|X|, t)$-RCDS protocol for $|Y|^{1/4} \le t \le |Y|^{1/2}$. This is done by immunizing Q_2 using a family of perfect hash functions $H_{|Y|,t,t^2}$ (introduced in [28]), that is, a family of functions $h : Y \to [t^2]$ such that for every set $T \in \binom{Y}{t}$ there exists at least one $h \in H_{|Y|,t,t^2}$ such that h restricted to T is one-to-one, i.e., $h(y_1) \ne h(y_2)$ for every distinct $y_1, y_2 \in T$. The following lemma is proved by a simple probabilistic argument (i.e., choosing the hash functions with uniform distribution from the functions satisfying (4)); we omit its proof.

Lemma A.2. *Let n be an integer and $t \in [\sqrt{n}]$. Then, there exists a family of perfect hash functions $H_{n,t,t^2} = \{h_i : [n] \to [t^2] : i \in [\ell]\}$, where $\ell = 16t \ln n$, such that for every $i \in [\ell]$ and every $b \in [t^2]$ it holds that*

$$|\{a \in [n] : h_i(a) = b\}| \leq \lceil n/t^2 \rceil. \tag{4}$$

Lemma A.3. *Let $f : X \times Y \to \{0,1\}$ be a function and $|Y|^{1/4} \leq t \leq |Y|^{1/2}$ be an integer. Then, there is a 2-server $(|X|, t)$-RCDS protocol for f with one-bit secrets in which the message size is $O(t^3 \log |Y|)$.*

Proof. The desired protocol $\mathcal{P}_2^t$ is described in Figure 2. Let $H_{|Y|,t,t^2} = \{h_i : Y \to [t^2] : i \in [\ell]\}$, where $\ell = \Theta(t \log |Y|)$, be the family of perfect hash functions promised by Lemma A.2.

Protocol $\mathcal{P}_2^t$

The secret: A bit $s \in \{0,1\}$.
Inputs: Q_1 and Q_2 hold the inputs $x \in X$ and $y \in Y$, respectively.
Common randomness: The two servers hold $\ell - 1$ uniformly distributed and independent random bits $s_1, \ldots, s_{\ell-1}$ and ℓt^2 common random strings for the CDS protocol $\mathcal{P}_2$.
The protocol:

1. Compute $s_\ell = s \oplus s_1 \oplus \cdots \oplus s_{\ell-1}$.
2. For every $i \in [\ell]$ do:
   - Let $Y_j = \{y \in Y : h_i(y) = j\}$, for every $j \in [t^2]$.
   - For every $j \in [t^2]$, independently execute the CDS protocol $\mathcal{P}_2$ of Lemma A.1 for the restriction of f to $X \times Y_j$ with the secret s_i. That is, Q_1 with input x sends a message for the restriction of f to $X \times Y_j$, for every $j \in [t^2]$, and Q_2 with input y sends a message only for the restriction of f to $X \times Y_{h_i(y)}$.

Figure 2: A 2-server $(|X|, t)$-RCDS protocol $\mathcal{P}_2^t$ for a function $f : X \times Y \to \{0,1\}$.

For the correctness of the protocol, let $x \in X$ and $y \in Y$ for which $f(x,y) = 1$. Then, for every $i \in [\ell]$, the input y is in $Y_{h_i(y)}$, so the referee can reconstruct s_i using the messages on the inputs x, y in the CDS protocol $\mathcal{P}_2$ for the restriction of f to the inputs of $X \times Y_{h_i(y)}$ with the secret s_i. Overall, the referee can learn all the bits $s_1, \ldots, s_\ell$, so it can reconstruct the secret s by xoring these bits.

For the robustness of the protocol, let (Z_1, Z_2) be a zero set of f such that $|Z_2| \leq t$. By Lemma A.2, there is at least one $i \in [\ell]$ for which $|h_i(Z_2)| = |Z_2|$. We prove that the referee cannot learn any information on s_i, and, thus, cannot learn the secret s.

Since h_i is one-to-one on Z_2, each input of Z_2 is in a different subset Y_j in the partition induced by h_i, and the referee gets at most one message of Q_2 in each execution of the CDS protocol $\mathcal{P}_2$ for the restriction of f to $X \times Y_j$ with the secret s_i. Since the CDS protocol $\mathcal{P}_2$ is a $(|X|, 1)$-RCDS protocol and $f(x,y) = 0$ for every $(x,y) \in Z_1 \times Z_2$, the referee cannot learn any information about s_i from any execution of the CDS protocol $\mathcal{P}_2$ for the restriction of f to the inputs of $X \times Y_j$ with the secret s_i, for every $j \in [t^2]$. Since each execution of $\mathcal{P}_2$ for each function h_i is done with independent common random strings, the referee cannot learn any information on s_i, and, hence, it cannot learn any information on the secret s.

We next provide an analyzing of the message size. Consider the execution of step 2 of $\mathcal{P}_2^t$. By Lemma A.2, $|Y_j| = O(|Y|/t^2)$ for every $j \in [t^2]$. The message size of Q_1 is t^2 times the message size of Q_1 in $\mathcal{P}_2$, i.e., it is t^2. The message size of Q_2 is the message size of Q_2 in $\mathcal{P}_2$, i.e., it is $O(|Y|/t^2)$. Since there are $\ell = \Theta(t \log |Y|)$ hash functions and $t \geq |Y|^{1/4}$, the message size of both servers is $O(t^3 \log |Y|)$. ∎

A.2 A Secret-Sharing Scheme from a 2-Server RCDS

A.2.1 Liu and Vaikuntanathan Decomposition of Access Structures

As in [40], we decompose an access structure F to three parts, depending on a parameter $\delta \in (0, \frac{1}{2})$: A bottom part $F_{\mathrm{bot},\delta}$, which handles small sets, a middle part $F_{\mathrm{mid},\delta}$, which handles medium-size sets, and a top part $F_{\mathrm{top},\delta}$, which handles large sets. This decomposition presented in the following proposition.

Proposition A.4 (Liu and Vaikuntanathan [40]). *Let F be an access structure over a set of n parties and $\delta \in (0, \frac{1}{2})$. Define the following access structures $F_{\mathrm{top},\delta}, F_{\mathrm{bot},\delta}$, and $F_{\mathrm{mid},\delta}$.*

$$\begin{aligned}
A \notin F_{\mathrm{top},\delta} &\iff \exists A' \notin F, A \subseteq A' \textit{ and } |A'| > \left(\frac{1}{2}+\delta\right)n,\\
A \in F_{\mathrm{bot},\delta} &\iff \exists A' \in F, A' \subseteq A \textit{ and } |A'| < \left(\frac{1}{2}-\delta\right)n,\\
A \in F_{\mathrm{mid},\delta} &\iff A \in F \textit{ and } \left(\frac{1}{2}-\delta\right)n \leq |A| \leq \left(\frac{1}{2}+\delta\right)n, \textit{ or } |A| > \left(\frac{1}{2}+\delta\right)n.
\end{aligned}$$

Then, $F = F_{\mathrm{top},\delta} \cap (F_{\mathrm{mid},\delta} \cup F_{\mathrm{bot},\delta})$. Therefore, if $F_{\mathrm{top},\delta}, F_{\mathrm{bot},\delta}$, and $F_{\mathrm{mid},\delta}$ can be realized by secret-sharing schemes with share size $O(2^{cn})$ then also F can be realized by a secret-sharing scheme with share size $O(2^{cn})$.

As mentioned in Proposition A.4, $F = F_{\mathrm{top},\delta} \cap (F_{\mathrm{mid},\delta} \cup F_{\mathrm{bot},\delta})$. Thus, by standard closure properties of secret-sharing schemes, realizing F can be reduced to realizing $F_{\mathrm{top},\delta}$, $F_{\mathrm{bot},\delta}$, and $F_{\mathrm{mid},\delta}$ (that is, choose a random bit s_1, share $s_1 \oplus s$ with a scheme realizing $F_{\mathrm{top},\delta}$ and independently share s_1 with schemes realizing $F_{\mathrm{mid},\delta}$ and $F_{\mathrm{bot},\delta}$). In [40], the access structures $F_{\mathrm{bot},\delta}$ was realized by sharing the secret independently for each minimal authorized set, resulting in a scheme realizing $F_{\mathrm{bot},\delta}$ with share size $\binom{n}{(\frac{1}{2}-\delta)n} \leq O(2^{\mathrm{H}_2(\frac{1}{2}-\delta)n})$ (where $\mathrm{H}_2(\cdot)$ is the binary entropy function). A similar construction was used for $F_{\mathrm{top},\delta}$. The properties of the resulting scheme for F are stated in the following lemma.

Lemma A.5 ([40]). *Let F be an access structure and $\delta \in (0, \frac{1}{2})$, and assume that $F_{\mathrm{mid},\delta}$ can be realized by secret-sharing schemes with share size $2^{M(\delta)n}$. Then, F can be realized by a secret-sharing scheme with share size $2^{\left(\max\{\mathrm{H}_2(\frac{1}{2}-\delta), M(\delta)\}\right)n}$.*

A.2.2 Secret-Sharing Schemes Realizing the Access Structure $F_{\mathrm{mid},\delta}$

Our main construction in this section is a secret-sharing scheme realizing the middle access structure $F_{\mathrm{mid},\delta}$ whose exponent is smaller than 1. Towards this construction, we defined balanced access structures in Definition A.6, represent $F_{\mathrm{mid},\delta}$ as a union of a polynomial number of balanced access structures, and show how to realize each such access structure using an RCDS protocol. By closure properties of secret-sharing schemes, we can realize $F_{\mathrm{mid},\delta}$ using the schemes for the balanced access structures, and, hence, we can realize F with a smaller share size.

Definition A.6 (The access structure $F_{B,\mathrm{mid},\delta}$). *Let F be an access structure with n parties, $\delta \in (0, \frac{1}{2})$, and B be a subset of parties. The access structure $F_{B,\mathrm{mid},\delta}$ is the access structure that contains all subsets of parties of size greater than $(\frac{1}{2}+\delta)n$, and all subsets of parties that contain authorized subsets $A' \in F$ of* *size between $(\frac{1}{2}-\delta)n$ and $(\frac{1}{2}+\delta)n$ that contain exactly $\lfloor |A'|/2 \rfloor$ of their parties from B. That is,*

$$
\begin{aligned}
F_{B,\mathrm{mid},\delta} &= \{A : \exists A' \in F, A' \subseteq A, \left(\frac{1}{2}-\delta\right)n \leq |A'| \leq \left(\frac{1}{2}+\delta\right)n, \text{ and } |A' \cap B| = \lfloor |A'|/2 \rfloor\} \\
&\cup \{A : |A| > \left(\frac{1}{2}+\delta\right)n\}.
\end{aligned}
$$

Following the above definition, we present our main secret-sharing scheme, which realizes the access structure $F_{B,\mathrm{mid},\delta}$.

Lemma A.7. *Let F be a an access structure over a set of n parties, $\delta \in (0.027, \frac{1}{6})$, and B be a subset of parties such that $|B| = n/2$. Then, there is a secret-sharing scheme realizing $F_{B,\mathrm{mid},\delta}$ with a one-bit secret in which the share size is $2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n}$.*

Proof. Assume without loss of generality that n is even (this can be done by adding dummy parties). Define

$$
\mathcal{B}_1 = \left\{S_1 \subseteq B : \left(\frac{1}{4}-\frac{\delta}{2}\right)n \leq |S_1| \leq \left(\frac{1}{4}+\frac{\delta}{2}\right)n\right\}
$$

and

$$
\mathcal{B}_2 = \left\{S_2 \subseteq \overline{B} : \left(\frac{1}{4}-\frac{\delta}{2}\right)n \leq |S_2| \leq \left(\frac{1}{4}+\frac{\delta}{2}\right)n\right\}.
$$

Note that $|\mathcal{B}_1| = |\mathcal{B}_2| < 2^{n/2}$. Moreover, define the function $f : \mathcal{B}_1 \times \mathcal{B}_2 \to \{0,1\}$, where $f(S_1, S_2) = 1$ if and only if $S_1 \cup S_2 \in F$, $(\frac{1}{2}-\delta)n \leq |S_1 \cup S_2| \leq (\frac{1}{2}+\delta)n$, and $|S_1| = |S_2|$ or $|S_1| = |S_2| - 1$. The scheme $\mathcal{D}_{\mathrm{B,mid}}$ realizing $F_{\mathrm{B,mid}}$ is described in Figure 3.

Scheme $\mathcal{D}_{B,\mathrm{mid},\delta}$

The secret: A bit $s \in \{0,1\}$.
The scheme:

1. Share the secret s among the n parties using a $((\frac{1}{2}+\delta)n+1)$-out-of-$n$ secret-sharing scheme.
2. Choose a random bit $s_1 \in \{0,1\}$ and define $s_2 = s \oplus s_1$.
3. Let $t = O\left(n2^{\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{1}{4}+\frac{\delta}{2})n}\right)$ (this choice of t will be explained later).
4. Execute the 2-server $(2^{|\mathcal{B}_1|}, t)$-RCDS protocol of Lemma A.3 for the function f with the secret s_1; for every $S_1 \in \mathcal{B}_1$ (respectively, $S_2 \in \mathcal{B}_2$) share the message of Q_1 (respectively, Q_2) when holding the input S_1 (respectively, S_2) among the parties of S_1 (respectively, S_2) using an $|S_1|$-out-of-$|S_1|$ (respectively, $|S_2|$-out-of-$|S_2|$) secret-sharing scheme.
5. Execute the 2-server $(t, 2^{|\mathcal{B}_2|})$-RCDS protocol of Lemma A.3 for the function f with the secret s_2; for every $S_1 \in \mathcal{B}_1$ (respectively, $S_2 \in \mathcal{B}_2$) share the message of Q_1 (respectively, Q_2) when holding the input S_1 (respectively, S_2) among the parties of S_1 (respectively, S_2) using an $|S_1|$-out-of-$|S_1|$ (respectively, $|S_2|$-out-of-$|S_2|$) secret-sharing scheme.

Figure 3: A secret-sharing scheme $\mathcal{D}_{B,\mathrm{mid},\delta}$ realizing the access structure $F_{B,\mathrm{mid},\delta}$.

For the correctness of the scheme, take a minimal authorized set $A \in F_{B,\mathrm{mid},\delta}$, that is, $A = S_1 \cup S_2$ for some $S_1 \subseteq B, S_2 \subseteq \overline{B}$ such that $S_1 \cup S_2 \in F$, $(\frac{1}{2}-\delta)n \leq |S_1 \cup S_2| \leq (\frac{1}{2}+\delta)n$, and $|S_1| = |S_2|$ or

$|S_1| = |S_2| - 1$. The parties in $A = S_1 \cup S_2$ can reconstruct the messages of Q_1 and Q_2 when holding the inputs S_1 and S_2, respectively, in the first RCDS protocol (i.e., the protocol of step 4), and can reconstruct s_1 from these messages using the reconstruction function of this protocol (since $f(S_1, S_2) = 1$). By symmetric arguments, the parties in A can reconstruct s_2 (using the protocol of step 5), and, thus, the parties in A can reconstruct the secret s by xoring s_1 and s_2. Authorized sets of size greater than $(\frac{1}{2} + \delta)n$ can reconstruct the secret s using the $((\frac{1}{2} + \delta)n + 1)$-out-of-$n$ secret-sharing scheme (i.e., the scheme of step 1).

For the privacy of the scheme, take an unauthorized set $A \notin \Gamma_{B,\mathrm{mid},\delta}$, that is, $A = S_1 \cup S_2$ such that $S_1 \subseteq B, S_2 \subseteq \overline{B}$ and $|S_1 \cup S_2| \leq (\frac{1}{2} + \delta)n$ (subsets of size greater than $(\frac{1}{2} + \delta)n$ are authorized), and assume without loss of generality that $|S_1| \leq (\frac{1}{4} + \frac{\delta}{2})n$ (otherwise, $|S_2| \leq (\frac{1}{4} + \frac{\delta}{2})n$ and we consider the second RCDS protocol, i.e, the protocol of step 5). In the first RCDS protocol (i.e, the protocol of step 4), the parties in S_1 know a message of Q_1 on an input $S_1' \in \mathcal{B}_1$ if and only if $S_1' \subseteq S_1$. That is, they can reconstruct the messages of the inputs (which are sets) in $\mathcal{B}_1$ for the set $T_{S_1} \triangleq \{S_1' \in \mathcal{B}_1 : S_1' \subseteq S_1, |S_1'| \geq (\frac{1}{4} - \frac{\delta}{2})n\}$. The number of subsets in T_{S_1} is at most

$$t \triangleq \sum_{i=(\frac{1}{4}-\frac{\delta}{2})n}^{(\frac{1}{4}+\frac{\delta}{2})n} \binom{(\frac{1}{4} + \frac{\delta}{2})n}{i}.$$

Since $\delta < \frac{1}{6}$, we have that $(\frac{1}{4} - \frac{\delta}{2})n > 1/2(\frac{1}{4} + \frac{\delta}{2})n$ and

$$t = O\left(n \cdot \binom{\frac{1}{4} + \frac{\delta}{2})n}{(\frac{1}{4} - \frac{\delta}{2})n}\right) = O\left(n 2^{\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{1}{4}+\frac{\delta}{2})n}\right).$$

For every $S_1' \subseteq S_1$ and $S_2' \subseteq S_2$, we have that (S_1', S_2') is a zero set of f, and the parties in $A = S_1 \cup S_2$ (which learn the messages on the inputs of T_{S_1} of Q_1 and possibly many messages of Q_2) learn only messages of the zero set $T_{S_1} \times \{S_2' \in \mathcal{B}_2 : S_2' \subseteq S_2\}$ in the first RCDS protocol. Thus, by the robustness of the RCDS protocol, the parties in A cannot learn any information on s_1, and, hence, they cannot learn any information on the secret s.

Overall, in the resulting scheme each party P_i gets a share of size $\log n$ from the threshold scheme of step 1 and less than $|\mathcal{B}_1| = |\mathcal{B}_2| < 2^{n/2}$ shares from the threshold schemes of step 4 (respectively, step 5), one for each message of the RCDS protocol for f on an input S such that $P_i \in S$. Thus, since $\delta > 0.027$ implies that $t = 2^{\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{1}{4}+\frac{\delta}{2}+o(1))n} > 2^{n/8} > |\mathcal{B}_2|^{1/4}$, the message size of the RCDS protocols osstep 4 and step 5 is $O(t^3 n)$, and the share size of each party in the scheme $\mathcal{D}_{B,\mathrm{mid},\delta}$ is $O(2^{n/2} \cdot t^3 n) = 2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n}$. □

We use the following family of subsets, in which every set of medium size is equally partitioned by one of the subsets in the family (a similar family appears in [3]). The proof of the claim is by a simple use of the probabilistic method.

Claim A.8. *Let P be a set of n parties for some even n and $\delta \in (0, \frac{1}{2})$. Then, there are $\ell = \Theta(n^{3/2})$ subsets $B_1, \ldots, B_\ell \subseteq P$, each of them of size $n/2$, such that for every subset $A \subseteq P$ for which $(\frac{1}{2} - \delta)n \leq |A| \leq (\frac{1}{2} + \delta)n$ it holds that $|A \cap B_i| = \lfloor |A|/2 \rfloor$ for at least one $i \in [\ell]$.*

We use the above scheme and the family of "balancing" subsets of Claim A.8 to construct a scheme that realizes the access structure $F_{\mathrm{mid},\delta}$.

Theorem A.9. *Let F be an access structure over a set of n parties and $\delta \in (0.027, \frac{1}{6})$. Then, there is a secret-sharing scheme realizing $F_{\mathrm{mid},\delta}$ with a one-bit secret in which the share size is $2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n}$.*

Proof. As in Lemma A.7, assume without loss of generality that n is even. By Claim A.8, there exist $\ell = \Theta(n^{3/2})$ subsets $B_1, \dots, B_\ell \subseteq P$, where $|B_i| = n/2$ for every $i \in [\ell]$, such that for every subset A such that $(\frac{1}{2} - \delta)n \leq |A| \leq (\frac{1}{2} + \delta)n$, it holds that $|A \cap B_i| = \lfloor |A|/2 \rfloor$ for at least one $i \in [\ell]$. Thus, we get that $F_{\mathrm{mid},\delta} = \cup_{i=1}^{\ell} F_{B_i,\mathrm{mid},\delta}$. By Lemma A.7, for every $i \in [\ell]$ there is a secret-sharing scheme $\mathcal{D}_{B_i,\mathrm{mid},\delta}$ realizing the access structure $F_{B_i,\mathrm{mid},\delta}$ with a one bit secret in which the share size is $2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n}$. For every $i \in [\ell]$, we independently share the secret s using the secret-sharing scheme $\mathcal{D}_{B_i,\mathrm{mid},\delta}$ realizing the access structure $F_{B_i,\mathrm{mid},\delta}$. The combined scheme is a secret-sharing scheme realizing the access structure $F_{\mathrm{mid},\delta}$ in which the share size is $O(n^{3/2}) \cdot 2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n} = 2^{(\frac{1}{2}+\mathrm{H}_2(\frac{1-2\delta}{1+2\delta})(\frac{3}{4}+\frac{3\delta}{2})+o(1))n}$. □

A.2.3 Secret-sharing Schemes Realizing any Access Structure

Theorem A.10. *There exists a constant $c < 1$ such that for every n and every n-party access structure F there is a secret-sharing scheme realizing F with a one-bit secret in which the share size is $2^{(c+o(1))n}$.*

Proof. By Lemma A.5 and Theorem A.9, for every $\delta \in (0.027, \frac{1}{6})$ the access structure F can be realized by a secret-sharing scheme with share size

$$2^{\left(\max\{\mathrm{H}_2\left(\frac{1}{2}-\delta\right), \frac{1}{2}+\mathrm{H}_2\left(\frac{1-2\delta}{1+2\delta}\right)\left(\frac{3}{4}+\frac{3\delta}{2}\right))\}+o(1)\right)n}. \tag{5}$$

By taking $\delta \approx 0.04063789$ (i.e., $t \approx O(2^{0.165076564n})$) the above two expressions in the exponent are equal, and we achieve share size of $2^{(c+o(1))n}$ for $c = 0.99523$. □

B Known Secret-Sharing Schemes

In this section we present known results on threshold and ramp secret-sharing schemes and closure properties of secret-sharing schemes. First, we define threshold secret-sharing schemes, and provide some known result for such schemes.

Definition B.1 (Threshold secret-sharing schemes)**.** *We say that an n-party secret-sharing scheme is a k-*out-of-n secret-sharing scheme *if it realizes the access structure $\Gamma_{k,n} = \{A \subseteq P : |A| \geq k\}$.*

Claim B.2 ([49])**.** *For every $k \in [n]$ there is a linear k-out-of-n secret-sharing scheme realizing $\Gamma_{k,n}$ for secrets of size m in which the share size is* $\max\{m, O(\log n)\}$.

Now, we define ramp secret-sharing schemes as in [18] and present results about an efficient ramp secret-sharing scheme implicit in [20].

Definition B.3 (Ramp secret-sharing schemes)**.** *Let $\mathcal{D}$ be a secret-sharing scheme on a set of n parties and let $0 \leq k_1 < k_2 \leq n$. The scheme $\mathcal{D}$ is a (k_1, k_2, n)-*ramp secret-sharing scheme *if each subset of parties of size at least k_2 can reconstruct the secret and each subset of parties of size at most k_1 cannot learn any information about the secret. There are no restrictions on other subsets of parties.*

Claim B.4. *For every constants $0 \leq b < a \leq 1$ there is p_0 such that for every prime-power $q > p_0$, there is a linear (bn, an, n)-ramp secret-sharing scheme over the field $\mathbb{F}_q$ in which each share is a field element (where p_0 is independent of n).*

Finally, we present the following claim, dealing with decomposition of secret-sharing schemes.

Claim B.5 ([15])**.** *Let $\Gamma_1, \dots, \Gamma_t$ be access structures over the same set of n parties, and let $\Gamma = \Gamma_1 \cup \cdots \cup \Gamma_t$ and $\Gamma' = \Gamma_1 \cap \cdots \cap \Gamma_t$. If there exist secret-sharing schemes realizing $\Gamma_1, \dots, \Gamma_t$ with share size at most c, then there exist secret-sharing schemes realizing Γ and Γ' with share size at most ct. Moreover, if the former schemes are linear over a finite field $\mathbb{F}$, then there exist linear secret-sharing schemes over $\mathbb{F}$ realizing Γ and Γ' with share size at most ct.*

C Some Probabilistic Facts

C.1 Negative Association

Definition C.1 (Negative association [37]). *Let $X := (X_1, \ldots, X_\ell)$ be a vector of random variables. The random variables X are negatively associated if for every two disjoint index sets, $I, J \in [\ell]$,*

$$\mathrm{E}[f(X_i, i \in I) \cdot g(X_j, j \in J)] \le \mathrm{E}[f(X_i, i \in I)] \cdot \mathrm{E}[g(X_j, j \in J)]$$

for all functions $f : \mathbb{R}^{|I|} \to \mathbb{R}$ and $g : \mathbb{R}^{|J|} \to \mathbb{R}$ that are both non-decreasing or both non-increasing.

We are interested in this definition since the Chernoff-Hoeffding bounds are applicable to sums of variables that satisfy the negative association condition [43] (see also [26, Proposition 5]). We will also use the following facts. The first one is presented in [25] in the context of applications of negative associativity in statistical physics, and the description of the Fermi-Dirac occupancy numbers for particle ensembles. Consider the so-called Fermi-Dirac model, in which m balls are thrown into ℓ bins with the restriction that each bin contains at most one ball. Let X_i denote the random variable that counts the number of balls in the i-th bin. The following fact asserts that the X_i are negatively associated.

Fact C.2 ([25, Theorem 10]). *Let $X := (X_1, \ldots, X_\ell)$ be random variables that take values in $\{0, 1\}$ and are distributed uniformly over m-weight vectors where $m \le \ell$. That is, for every $x = (x_1, \ldots, x_\ell) \in \{0, 1\}^\ell$ of Hamming weight m it holds that*

$$\Pr[X = x] = \binom{\ell}{m}^{-1},$$

and $\Pr[X = x] = 0$ for every x of Hamming weight $\mathrm{wt}(x) \neq m$. Then, the random variables $(X_1, \ldots, X_\ell)$ are negatively associated.

Fact C.3 ([37, Property 7]). *If two vectors of negatively associated random variables X and Y are mutually independent, then the random variables (X, Y) are negatively associated.*

Fact C.4 ([37, Property 6]). *Let $X := (X_1, \ldots, X_\ell)$ be negatively associated random variables, and $I_1, \ldots, I_k \subseteq [\ell]$ disjoint index sets, for some positive integer k. For $j \in [k]$, let $h_j : \mathbb{R}^{|I_k|} \to \mathbb{R}$ be functions that are all non-decreasing or all non-increasing. Then the random variables $Y_1, \ldots Y_k$ defined as $Y_j := h_j(X_i, i \in I_j)$ are also negatively associated. That is, non-decreasing (or non-increasing) functions of disjoint subsets of negatively associated variables are also negatively associated.*

Now we turn to prove that the random variables $\chi_1, \ldots, \chi_{N_1}$ defined in the proof of Section 6.3 are negatively associated. Recall that these random variables are defined via the following experiment. Given N_1 rows and a list of q inputs, each input u is placed in a random βN_1 subset of the rows. We call a row j *bad* if it has more than $2\beta q$ inputs mapped into it, and let χ_j be an indicator random variable that takes the value 1 if the j-th row is bad. For ease of notation we use N to denote N_1.

Claim C.5. *The random variables $\chi_1, \ldots, \chi_N$ are negatively associated.*

Proof. We call $X_{i,u}$ the indicator random variable that takes the value 1 if the input u was mapped to the i-th row. We notice that when we observe a specific u' that is mapped uniformly to βN out of N rows, the variables $X_{i,u'}$s with $i \in [N]$ behave like the random variables described in Fact C.2 with $\ell = N$ and $m = \beta N$. Therefore they are negatively associated.

Since every input is mapped independently, the q vectors $X_{1,u}, \ldots X_{N,u}$ each defined by different inputs u are mutually independent, and therefore due to Fact C.3, for every input u in the list and $i \in [N]$, the random variables $X_{i,u}$ are negatively associated.

For the last step we use Fact C.4. We take the random variables $X_{i,u}$ and look at disjoint sets $I_i = X_{i,u_1}, \ldots, X_{i,u_q}$ for all $i \in N$. We define a non-decreasing mapping

$$h_i(I_i) = \begin{cases} 1 & \text{if } \sum_{j=1}^{q} X_{i,u_j} \geq 2\beta q \\ 0 & otherwise, \end{cases}$$

and get that the random variables $Y_i = h_i(I_i)$ are negatively associated. We finish the proof by noticing that each Y_i is distributed like an indicator random variable that takes the value 1 if the i-th row is bad. $\square$

Next we prove a probabilistic statement from the proof of Lemma 5.7.

Claim C.6. *Given a fixed n-bit string x and a uniformly chosen partitions Π of $[n]$ to $\sqrt{n}$ subsets of size $\sqrt{n}$ each, denote by Y_{j,B_i} the random variable that takes the value 1 when the j-th bit in the i-th block is 1. Then for every index $i \in [\sqrt{n}]$, the $\sqrt{n}$ variables $\{Y_{j,B_i}\}_{1 \leq j \leq \sqrt{n}}$ are negatively associated.*

Proof. The $\sqrt{n} \cdot \sqrt{n}$ random variables Y_{j,B_i} for $1 \leq i, j \leq \sqrt{n}$ satisfy the conditions for Fact C.2 for $\ell = n$, $m = \mathrm{wt}(x)$, and they are therefore negatively associated. It is easy to see that any subset of negatively associated variables are also negatively associated [37, Property 4], and with that the proof is completed. $\square$

C.2 Family of $\log t$-Collision Free Hash Functions

We next prove a strong version of Lemma 4.8, i.e., show the existence of a family of $\log t$-collision free hash functions $H_{n,t,\log t,2t}$ of size $\ell = 16t \ln u$, as in the lemma, with some additional properties (we need this stronger version for our RCDS protocols in Appendix D.2).

Lemma C.7. *Let n be an integer, $t \in \{15, \ldots, n/2\}$, $\mathcal{T} \subseteq \binom{[n]}{\leq t}$, and u be the number of maximal sets in $\mathcal{T}$. Then, there exists a family of of $\log t$-collision free hash functions $H_{n,t,\log t,2t} = \{h_1, \ldots, h_\ell\}$ of size $\ell = 16 \ln u$, such that for every $i \in [\ell]$ and every $b \in [2t]$ it holds that $|\{a \in [n] : h_i(a) = b\}| \leq \lceil n/2t \rceil$, and for every subset $T \in \mathcal{T}$ there are at least $\ell/4$ functions $h \in H_{n,t,2t}$ such that for every $b \in [2t]$ it holds that $|\{a \in T : h(a) = b\}| < \log t$.*

Proof. Without loss of generality, we assume that t divides n (this can be achieved by increasing n by at most $t-1$), and let $t' = \log t$. We show that there exists a family of hash function $H_{n,t,t',2t}$ as above with $\ell = 16 \ln u$ functions using the probabilistic method. As a first step in the proof, we choose at random a function $h : [n] \to [2t]$ such that for every $b \in [2t]$ it holds that $|\{a \in [n] : h(a) = b\}| \leq \lceil n/2t \rceil$, and fix a subset $T \in \max(\mathcal{T})$, where $\max(\mathcal{T})$ is the set of maximal sets in $\mathcal{T}$ (recall that $|\max(\mathcal{T})| = u$). The probability that for some $b \in [2t]$ it holds that $|\{a \in T : h(a) = b\}| \geq \log t$ is

$$\begin{aligned}
&\Pr[\, \exists_{b \in [2t]} |\{a \in T : h(a) = b\}| \geq \log t \,] \\
&\quad = \Pr[\, \exists j_1 \neq \cdots \neq j_{\log t} \in T : h(j_1) = \cdots = h(j_{\log t}) \,] \\
&\quad \leq \sum_{j_1 \neq \cdots \neq j_{\log t} \in T} \Pr[\, h(j_1) = \cdots = h(j_{\log t}) \,] < \binom{t}{\log t} \cdot \left(\frac{1}{2t}\right)^{\log t - 1} \\
&\quad \leq \left(\frac{et}{\log t}\right)^{\log t} \cdot \frac{1}{(2t)^{\log t - 1}} = \left(\frac{e}{2 \log t}\right)^{\log t} \cdot 2t < \frac{1}{2}
\end{aligned}$$

(where the last inequality holds since $t \geq 15$).

Next, we claim that if we choose at random $\ell = 16 \ln u$ functions as above, we get the desired family $H_{n,t,2t} = \{h_1, \dots, h_\ell\}$. We bound the probability that for a given subset $T \in \max(\mathcal{T})$ of size at most t, there exist at most $\ell/4$ functions $h \in H_{n,t,t',2t}$ that we choose at random, such that for every $b \in [2t]$ it holds that $|\{a \in T : h(a) = b\}| < t'$.

For every $i \in [\ell]$, let X_i be a Boolean random variable such that $X_i = 1$ if for every $b \in [2t]$ it holds that $|\{a \in T : h_i(a) = b\}| < \log t$ and $X_i = 0$ otherwise. Additionally, let $X = \sum_{i=1}^{\ell} X_i$, i.e., X is the number of hash functions h_i, for $i \in [\ell]$, such that for every $b \in [2t]$ it holds that $|\{a \in T : h_i(a) = b\}| < \log t$. As we have shown above, $\Pr[\, X_i = 0\,] = \Pr[\, \exists b \in [2t] : |\{a \in T : h(a) = b\}| \geq \log t\,] < \frac{1}{2}$, so by linearity of expectation, $\mathrm{E}(X) = \sum_{i=1}^{\ell} \mathrm{E}(X_i) = \sum_{i=1}^{\ell} \Pr[\, X_i = 1\,] > \ell \cdot \frac{1}{2} = \frac{\ell}{2}$.

Using a Chernoff bound [44] ($\Pr[\, X \leq (1-\delta) \cdot \mathrm{E}(X)\,] \leq e^{-\mathrm{E}(X)\cdot\delta^2/2}$ for all $0 < \delta < 1$) we get that

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

By the union bound, the probability that there exists a subset $T \in \max(\mathcal{T})$ with at most $\ell/4$ functions h_i, for $i \in [\ell]$, such that $\exists b \in [2t] : |\{a \in T : h(a) = b\}| \geq \log t$, is less than 1. Thus, there exists a family $H_{n,t,t',2t}$ with $\ell = 16 \ln u$ hash functions as required. □

Moreover, using similar arguments we can prove the following strong version of Lemmas 4.7 and A.2.

Lemma C.8. *Let n be an integer and $t \in [\sqrt{n}]$, and $\mathcal{T} \subseteq \binom{[n]}{\leq t}$. Then, there exists a family of hash functions $H_{n,t,t^2} = \{h_i : [n] \to [t^2] : i \in [\ell]\}$, where $\ell = 16 \ln |\mathcal{T}|$, such that for every $i \in [\ell]$ and every $b \in [t^2]$ it holds that $|\{a \in [n] : h_i(a) = b\}| \leq \lceil n/t^2 \rceil$, and for every subset $T \in \mathcal{T}$ there are at least $\ell/4$ functions $h \in H_{n,t,t^2}$ for which $|h(T)| = |T|$.*

D Linear Robust CDS Protocols

D.1 A k-Server CDS Protocol

We show that for an odd k a (non-optimized) variant of the linear CDS protocol of [12] is robust when half of the servers can send an unbounded number of messages (a variant of this protocol has similar properties when k is even). We assume without loss of generality that for the k-input function f, for every $j \in \{(k+3)/2, \dots, k\}$ there exists an input $a_j \in X_j$ such that $f(i_1, \dots i_{j-1}, a_j, i_{j+1}, \dots, i_k) = 0$ for every $i_1 \in X_1, \dots, i_{j-1} \in X_{j-1}, i_{j+1} \in X_{j+1}, \dots, i_k \in X_k$ (this can be done by adding a dummy element to the input domain of server Q_j).

Lemma D.1. *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a function, where $|X_i| \leq 2^\ell$, for some odd integer $k > 2$. Then, for every finite field $\mathbb{F}$, protocol $\mathcal{P}_k$, described in Figure 4, is a linear k-server $\left(2^{X_1}, \dots, 2^{X_{(k+1)/2}}\right)$-RCDS protocol for f with domain of secrets $\mathbb{F}$, in which the message size is $2^{\ell(k-1)/2} \log |\mathbb{F}|$.*

Proof. For proving the correctness of the protocol $\mathcal{P}_k$ let $x_1, \dots, x_k$ be inputs such that $f(x_1, \dots, x_k) = 1$. In this case, server Q_1 sends $s_{x_1,\dots,x_k}$ to the referee (in addition to other elements), server Q_j, for $2 \leq j \leq k'$, sends $q^j_{x_1,\dots,x_{k'}}$ to the referee, and for every $(i_{k'+1}, \dots, i_k) \neq (x_{k'+1}, \dots, x_k)$ at least one server Q_j (where $k'+1 \leq j \leq k$) sends $r_{i_{k'+1},\dots,i_k}$. Since $f(x_1, \dots, x_k) = 1$, the random element $r_{x_{k'+1},\dots,x_k}$ does not appear in $s_{x_1,\dots,x_k}$. Thus, the referee can compute the expression in step 5 of $\mathcal{P}_k$, which equals s.

For the robustness of the protocol, recall that $k' = (k+1)/2$. Assume that servers $Q_{k'+1}, \dots, Q_k$ send messages of inputs $x_{k'+1} \in X_{k'+1}, \dots, x_k \in X_k$, respectively,[13] and servers $Q_1, \dots, Q_{k'}$

[13]If such server does not send any message, we will assume that it sends the message of the dummy input a_j.

Protocol $\mathcal{P}_k$

The secret: An element $s \in \mathbb{F}$.
Inputs: Servers $Q_1, \ldots, Q_k$ hold the inputs $x_1 \in X_1, \ldots, x_k \in X_k$, respectively.
Common randomness: Let $k' = (k+1)/2$. The k servers hold the following uniformly distributed and independent random elements.

- $q^j_{i_1,\ldots,i_{k'}} \in \mathbb{F}$ for every $j \in \{2, \ldots, k'\}$ and every $i_1 \in X_1, \ldots, i_{k'} \in X_{k'}$.
- $r_{i_{k'+1},\ldots,i_k} \in \mathbb{F}$ for every $i_{k'+1} \in X_{k'+1}, \ldots, i_k \in X_k$.

The protocol:

1. Define $q_{i_1,\ldots,i_{k'}} = \sum_{j=2}^{k'} q^j_{i_1,\ldots,i_{k'}}$ for every $i_1 \in X_1, \ldots, i_{k'} \in X_{k'}$.
2. Server Q_1 sends to the referee the elements
$$s_{x_1,i_2,\ldots,i_{k'}} = s + q_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1}\in X_{k'+1},\ldots,i_k \in X_k, f(x_1,i_2\ldots,i_k)=0} r_{i_{k'+1},\ldots,i_k}$$
for every $i_2 \in X_2, \ldots, i_{k'} \in X_{k'}$.
3. For every $j \in \{2, \ldots, k'\}$, server Q_j sends to the referee the elements $q^j_{i_1,i_2,\ldots,i_{j-1},x_j,i_{j+1}\ldots,i_{k'}}$ for every $i_1 \in X_1, \ldots, i_{j-1} \in X_{j-1}, i_{j+1} \in X_{j+1}, \ldots, i_{k'} \in X_{k'}$.
4. For every $j \in \{k'+1, \ldots, k\}$, server Q_j sends to the referee the elements $r_{i_{k'+1},\ldots,i_k}$ for every $i_{k'+1} \in X_{k'+1}, \ldots, i_k \in X_k$ such that $i_j \neq x_j$.
5. If $f(x_1, \ldots, x_k) = 1$, the referee computes
$$s_{x_1,x_2,\ldots,x_{k'}} - q_{x_1,\ldots,x_{k'}} - \sum\nolimits_{i_{k'+1}\in X_{k'+1},\ldots,i_k\in X_k, f(x_1,\ldots,x_{k'},i_{k'+1},\ldots,i_k)=0} r_{i_{k'+1},\ldots,i_k}.$$

Figure 4: A linear k-server CDS protocol $\mathcal{P}_k$ for a function $f : X_1 \times \cdots \times X_k \to \{0, 1\}$.

send multiple messages for subsets of inputs $Z_1 \subseteq X_1, \ldots, Z_{k'} \subseteq X_{k'}$, respectively, such that $f(x_1, \ldots, x_{k'}, x_{k'+1}, \ldots, x_k) = 0$ for every $(x_1, \ldots, x_{k'}) \in Z_1 \times \cdots \times Z_{k'}$. We prove below that these messages are statistically independent from the secret.

The referee gets the following messages:

The messages of Q_1. The elements

$$s_{x_1,i_2,\ldots,i_{k'}} = s + q_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1}\in X_{k'+1},\ldots,i_k\in X_k, f(x_1,i_2,\ldots,i_{k'},i_{k'+1},\ldots,i_k)=0} r_{i_{k'+1},\ldots,i_k}$$

for every $(i_2, \ldots, i_{k'}) \in X_2 \times \cdots \times X_{k'}$ and every $x_1 \in Z_1$.

The messages of $Q_2, \ldots, Q_{k'}$. The elements $q^j_{i_1,i_2,\ldots,i_{k'}}$ for every $j \in \{2, \ldots, k'\}$ and every $i_1 \in X_1, \ldots, i_{k'} \in X'_k$ such that $i_j \in Z_j$. In particular, for every $(x_1, \ldots, x_{k'}) \in Z_1 \times \cdots \times Z_{k'}$, the referee gets the elements $q^2_{x_1,\ldots,x_{k'}}, q^3_{x_1,\ldots,x_{k'}}, \ldots, q^{k'}_{x_1,\ldots,x_{k'}}$, thus it can compute $q_{x_1,\ldots,x_{k'}}$.

The messages of $Q_{k'+1}, \ldots, Q_k$. The elements $r_{i_{k'+1},\ldots,i_k}$ for every $i_{k'+1} \in X_{k'+1}, \ldots, i_k \in X_k$ except for $r_{x_{k'+1},\ldots,x_k}$.

Intuitively, for every $(x_1, \ldots, x_{k'}) \in Z_1 \times \cdots \times Z_{k'}$, the element $r_{x_{k'+1},\ldots,x_k}$ acts as a one-time-pad protecting s in $s_{x_1,x_2,\ldots,x_{k'}}$, and for every $x_1 \in Z_1$ and every $(i_2, \ldots, i_{k'}) \notin Z_2 \times \cdots \times Z_{k'}$, the element $q_{x_1,i_2,\ldots,i_{k'}}$ acts as a one-time-pad protecting s in $s_{x_1,i_2,\ldots,i_{k'}}$.

Formally, for every two secrets $s, s' \in \mathbb{F}$ we show a bijection ϕ from the randomness of $\mathcal{P}_k$ to itself such that the messages in $\mathcal{P}_k$ with secret s and randomness r are the same as the messages in $\mathcal{P}_k$ with secret s' and randomness $r' = \phi(r)$. Consider the random elements

$$r = \Big((q^j_{i_1,\ldots,i_{k'}})_{j\in\{2,\ldots,k'\},i_1\in X_1,\ldots,i_{k'}\in X_{k'}}, (r_{i_{k'+1},\ldots,i_k})_{i_{k'+1}\in X_{k'+1},\ldots,i_k\in X_k} \Big)$$

and the messages generated from them for the secret s. The bijection is as follows:

1. Define $r'_{i_{k'+1},\ldots,i_k} = r_{i_{k'+1},\ldots,i_k}$ for every $(i_{k'+1}, \ldots, i_k) \neq (x_{k'+1}, \ldots, x_k)$ and $r'_{x_{k'+1},\ldots,x_k} = r_{x_{k'+1},\ldots,x_k} + s - s'$. Since no server sends $r_{x_{k'+1},\ldots,x_k}$, these values and the secret s' generate the same messages of $Q_{k'+1}, \ldots, Q_k$ as the messages in $\mathcal{P}_k$ with the secret s and randomness r.

2. For every $x_1 \in Z_1$ and every $i_2 \in X_2, \ldots, i_{k'} \in X_{k'}$ such that $f(x_1, i_2, \ldots, i_{k'}, x_{k'+1}, \ldots, x_k) = 0$ (in particular, for every $(x_1, i_2, \ldots, i_{k'}) \in Z_1 \times \cdots \times Z_{k'}$): Define $q'_{x_1,i_2,\ldots,i_{k'}} = q_{x_1,i_2,\ldots,i_{k'}}$ and for every $j \in \{2, \ldots, k'\}$ let $q'^j_{x_1,i_2,\ldots,i_{k'}} = q^j_{x_1,i_2,\ldots,i_{k'}}$.

   Since $f(x_1, i_2, \ldots, i_{k'}, x_{k'+1}, \ldots, x_k) = 0$, the element $r_{x_{k'+1},\ldots,x_k}$ appears in the sum in $s_{x_1,i_2,\ldots,i_{k'}}$ and

$$\begin{aligned} s_{x_1,i_2,\ldots,i_{k'}} &= s + q_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1}\in X_{k'+1},\ldots,i_k\in X_k, f(x_1,i_2,\ldots,i_{k'},i_{k'+1},\ldots,i_k)=0} r_{i_{k'+1},\ldots,i_k} \\ &= s' + q'_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1}\in X_{k'+1},\ldots,i_k\in X_k, f(x_1,i_2,\ldots,i_{k'},i_{k'+1},\ldots,i_k)=0} r'_{i_{k'+1},\ldots,i_k}. \end{aligned}$$

   Thus, the same message $s_{x_1,i_2,\ldots,i_{k'}}$ is generated for s and s' with r and $r' = \phi(r)$, respectively.

3. For every $x_1 \in Z_1$ and every $i_2 \in X_2, \ldots, i_{k'} \in X_{k'}$ such that $f(x_1, i_2, \ldots, i_{k'}, x_{k'+1}, \ldots, x_k) = 1$, let let j_0 be the minimal index such that $i_{j_0} \notin Z_{j_0}$ and define $q'^{j_0}_{x_1,i_2,\ldots,i_{k'}} = q^{j_0}_{x_1,i_2,\ldots,i_{k'}} + s - s'$ and $q'^j_{x_1,i_2,\ldots,i_{k'}} = q^j_{x_1,i_2,\ldots,i_{k'}}$ for every $j \neq j_0$, thus,

$$q'_{x_1,i_2,\ldots,i_{k'}} = \sum_{j=2}^{k'} q'^j_{x_1,i_2,\ldots,i_{k'}} = \sum_{j=2}^{k'} q^j_{x_1,i_2,\ldots,i_{k'}} + s - s' = q_{x_1,i_2,\ldots,i_{k'}} + s - s'.$$

Since $f(x_1, i_2, \ldots, i_{k'}, x_{k'+1}, \ldots, x_k) = 1$, the element $r_{x_{k'+1},\ldots,x_k}$ does not appear in the sum in $s_{x_1,i_2,\ldots,i_{k'}}$ and

$$\begin{aligned} s_{x_1,i_2,\ldots,i_{k'}} &= s + q_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1} \in X_{k'+1},\ldots,i_k \in X_k, f(x_1,i_2,\ldots,i_{k'},i_{k'+1},\ldots,i_k)=0} r_{i_{k'+1},\ldots,i_k} \\ &= s' + q'_{x_1,i_2,\ldots,i_{k'}} + \sum\nolimits_{i_{k'+1} \in X_{k'+1},\ldots,i_k \in X_k, f(x_1,i_2,\ldots,i_{k'},i_{k'+1},\ldots,i_k)=0} r'_{i_{k'+1},\ldots,i_k}. \end{aligned}$$

Thus, the same message $s_{x_1,i_2,\ldots,i_{k'}}$ is generated for s and s' with r and $r' = \phi(r)$, respectively. Notice that $q'^{j_0}_{x_1,i_2,\ldots,i_{k'}}$ is not sent by Q_{j_0} these values and the secret s' generate the same messages of $Q_2, \ldots, Q_{k'}$ as the messages in $\mathcal{P}_k$ with the secret s and randomness r.

4. For every $i_1 \notin Z_1, i_2 \in X_2, \ldots, i_{k'} \in X_{k'}$ and every $j \in \{2, \ldots, k'\}$, let $q'^{j}_{i_1,i_2,\ldots,i_{k'}} = q^{j}_{i_1,i_2,\ldots,i_{k'}}$.

To conclude, the messages have the same probability for s and s'. which implies the robustness. □

D.2 A Linear 2-Server CDS Protocol

In this section, we present a linear 2-server RCDS protocol that is robust for every zero-set $Z_1 \times Z_2$, where Z_1 can be an arbitrary subset of the inputs of Q_1 and Z_2 can be an arbitrary subset of size at most t of the inputs of Q_2, in which the message size is $\tilde{O}((t + 2^{\ell/2}) \log |\mathcal{Z}_2|)$. This protocol can be used to construct an alternative linear secret-sharing scheme for arbitrary n-party access structures with the same share size as the linear secret-sharing scheme of Theorem 2.1. Furthermore, when the secret contains $\log |\mathcal{Z}_2|$ field elements, the share size remains $\tilde{O}((t + 2^{\ell/2}) \log |\mathcal{Z}_2|)$, i.e, the normalized share size (the share size divided by the secret size) is only $\tilde{O}(t + 2^{\ell/2})$.

We start with the protocol $\mathcal{P}_2$, as described in Figure 5. Similarly to Lemma D.1, the protocol $\mathcal{P}_2$ is (2^X)-RCDS protocol, that is it robust when Q_1 sends an unbounded number of messages and Q_2 sends one message. This protocol is a simple generalization of the protocol described in Figure 1 to an arbitrary field; its properties, as described in Lemma D.2, follow from the same arguments as in the proof of Lemma A.1.

Protocol $\mathcal{P}_2$

The secret: An element $s \in \mathbb{F}$.
Inputs: Q_1 and Q_2 hold the inputs $x \in X$ and $y \in Y$, respectively.
Common randomness: The two servers hold $|Y|$ uniformly distributed and independent random elements $r_0, r_1, \ldots, r_{|Y|} \in \mathbb{F}$.
The protocol:

1. Q_1 sends to the referee the element $m_A = s + r_0 + \sum_{i \in Y, f(x,i)=0} r_i$.
2. Q_2 sends to the referee the elements $m_B = (r_0, r_1, \ldots, r_{y-1}, r_{y+1}, \ldots, r_{|Y|})$.
3. If $f(x, y) = 1$, the referee computes $m_A - r_0 - \sum_{i \in Y, f(x,i)=0} r_i$.

Figure 5: A linear 2-server CDS protocol $\mathcal{P}_2$ for a function $f : X \times Y \to \{0, 1\}$.

Lemma D.2. *Let $f : X \times Y \to \{0, 1\}$ be a function. Then, for every finite field $\mathbb{F}$, protocol $\mathcal{P}_2$, described in Figure 5, is a linear 2-server (2^X)-RCDS protocol for f with domain of secrets $\mathbb{F}$ secrets, in which the message size of Q_1 is $\log |\mathbb{F}|$ and the message size of Q_2 is $|Y| \log |\mathbb{F}|$.*

The next protocol, originally appearing in [10], balances the sizes of messages of Q_1 and Q_2. Its idea is to partition the set of inputs of Q_2 to disjoint sets and execute the protocol $\mathcal{P}_2$ independently for every set of inputs.

Claim D.3 ([10])**.** *Let $f : X \times Y \to \{0,1\}$ be a function. Then, for every finite field $\mathbb{F}$ and every $d \leq |Y|$ there is a linear 2-server (2^X)-RCDS protocol $\mathcal{P}_2^{\text{balanced}}$ for f with one-element secrets in which the message size of Q_1 is $O(d \log |\mathbb{F}|)$ and the message size of Q_2 is $O((|Y|/d) \log |\mathbb{F}|)$.*

Proof. The description of the protocol $\mathcal{P}_2^{\text{balanced}}$ is as follows: Let s be the secret, and partition the set Y to d disjoint sets $Y_1, \dots, Y_d$ of size at most $\lceil |Y|/d \rceil$, that is, every input $y \in Y$ is in exactly one set Y_i. For every $i \in [d]$, we execute the linear CDS protocol $\mathcal{P}_2$ independently for the restriction of f to the inputs of $X \times Y_i$ with the secret s. Server Q_1, when holding the input $x \in X$, sends the messages in all the above independent protocols. Server Q_2, when holding the input $y \in Y$, only sends the message in the protocol for the restriction of f to the inputs of $X \times Y_i$ for which $y \in Y_i$.

For the correctness of the protocol, if $f(x, y) = 1$ then the referee can reconstruct the secret from the messages of the CDS protocol for the restriction of f to the inputs of $X \times Y_i$ for which $y \in Y_i$. For the robustness of the protocol, let $Z_1 \subseteq X$ and $y \in Y$ such that $f(x, y) = 0$ for every $x \in Z_1$. The referee cannot learn any information on the secret from the messages on y and the inputs of Z_1 from each of the above independent protocols, which follows by the robustness of each of these protocols. Thus, the resulting protocol $\mathcal{P}_2^{\text{balanced}}$ is (2^X)-RCDS protocol.

The message of Q_1 contains d field elements and the message of Q_2 contains at most $\lceil |Y|/d \rceil$ field elements (since it sends a message in one execution of $\mathcal{P}_2$ in which the input domain size of Q_2 is at most $\lceil |Y|/d \rceil$). $\square$

Next, we show how to transform the above CDS protocol to a $(2^X, \binom{Y}{\leq t})$-RCDS protocol. This is done by immunizing Q_2 as in Theorem 4.4, that is, we use two levels of hashing. However, in this case we optimize the share size by using the fact that each copy of the CDS protocol is applied to a function with a smaller domain of Q_2.

Lemma D.4. *Let $f : X \times Y \to \{0,1\}$ be a function, and $t \leq \sqrt{|Y|}$ be an integer. Then, for every finite field $\mathbb{F}$, there is a linear 2-server $(2^X, \binom{Y}{\leq t})$-RCDS protocol for f with one-element secrets in which the message size of Q_1 is $O((t^3 + |Y|t/\sqrt{|X|}) \log |Y| \log |\mathbb{F}|)$ and the message size of Q_2 is $O(\sqrt{|X|} t \log |Y| \log |\mathbb{F}|)$. Furthermore, there is p_0 such that for every prime-power $q > p_0$, there is a multi-linear 2-server $(2^X, \binom{Y}{\leq t})$-RCDS protocol for f over $\mathbb{F}_q$ with secrets of size $\Theta(qt \log |Y|)$ in which the normalized message size of Q_1 is $O(t^2 + |Y|/\sqrt{|X|})$ and the normalized message size of Q_2 is $O(\sqrt{|X|})$.*

Proof. The desired protocol $\mathcal{P}_2^t$, described in Figure 6, is a special case of the protocol of Lemma 4.9, when $k = k' = 2, t' = 1$, and $\mathcal{P}$ is the protocol of Claim D.3 with $d = \max\{1, |Y|/(\sqrt{|X|}t^2)\}$. Let $H_{|Y|,t,t^2} = \{h_i : Y \to [t^2] : i \in [\ell]\}$, where $\ell = \Theta(t \log |Y|)$, be the family of perfect hash functions promised by Lemma C.8.

The correctness and privacy of $\mathcal{P}_2^t$ follow by Lemma 4.9. We next provide a refined analyzing of its message size. Consider the execution of step 2 of $\mathcal{P}_2^t$. By Lemma C.8, $|Y_j| = O(|Y|/t^2)$ for every $j \in [t^2]$. The message of Q_1 contains t^2 messages of Q_1 in $\mathcal{P}_2^{\text{balanced}}$, i.e., it contains $t^2 d = t^2 \cdot \max\{1, |Y|/(\sqrt{|X|}t^2)\} = O(t^2 + |Y|/\sqrt{|X|})$ field elements, The message of Q_2 contains one message of Q_1 in $\mathcal{P}_2^{\text{balanced}}$, i.e., it contains $O(|Y_{h_i(y)}|/d) = O\left(\frac{|Y|/t^2}{\max\{1, |Y|/(\sqrt{|X|}t^2)\}}\right) = O(\min\{|Y|/t^2, \sqrt{|X|}\}) \leq O(\sqrt{|X|})$ field elements. Since there are $\ell = \Theta(t \log |Y|)$ hash functions, the sizes of the messages is as in the lemma.

Protocol $\mathcal{P}_2^t$

The secret: An element $s \in \mathbb{F}$.
The protocol:

1. Choose ℓ random elements $s_1, \dots, s_\ell \in \mathbb{F}$ such that $s = s_1 + \cdots + s_\ell$.
2. For every $i \in [\ell]$ do:
   - Let $Y_j = \{y \in Y : h_i(y) = j\}$, for every $j \in [t^2]$.
   - For every $j \in [t^2]$, independently execute the CDS protocol $\mathcal{P}_2^{\text{balanced}}$ of Claim D.3 for the restriction of f to $X \times Y_j$ with the secret s_i and $d = \max\{1, |Y|/(\sqrt{|X|}t^2)\}$. That is, Q_1 with input x sends a message for the restriction of f to $X \times Y_j$, for every $j \in [t^2]$, and Q_2 with input y sends a message only for the restriction of f to $X \times Y_{h_i(y)}$.

Figure 6: A linear 2-server $(2^X, \binom{Y}{\leq t})$-RCDS protocol $\mathcal{P}_2^t$ for a function $f : X \times Y \to \{0,1\}$.

To construct the desired protocol for long secrets, let $s = (s'_1, \dots, s'_{\ell/4}) \in \mathbb{F}_q^{\ell/4}$ be the secret. We change step 1 in the protocol $\mathcal{P}_2^t$ (described in Figure 6) such that $s_1, \dots, s_\ell \in \mathbb{F}_q$ are the shares of a $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme of the secret $s = (s'_1, \dots, s'_{\ell/4}) \in \mathbb{F}_q^{\ell/4}$, where p_0 is the constant from Claim B.4 and $q > p_0$.

As before, for every inputs $x \in X, y \in Y$ such that $f(x,y) = 1$, the referee can learn all the secrets in those ℓ protocols from the messages on the inputs x, y, so it can reconstruct the secret s using the reconstruction function of the ramp scheme. Moreover, for every (Z_1, Z_2) that is a zero set of f such that $|Z_2| \leq t$, by Lemma 4.7, there are at least $\ell/4$ values of $i \in [\ell]$ for which $|h_i(Z_2)| = |Z_2|$. Thus, the referee cannot learn any information on at least $\ell/4$ of the shares $s_1, \dots, s_\ell$ in the above ℓ protocols from the messages on the inputs of Z_1, Z_2. By the security of the ramp scheme, the referee cannot learn any information on the secret s. $\square$

We improve our linear robust 2-server CDS protocol using the family of hash functions of Lemma 4.8.

Theorem D.5. *Let $f : X \times Y \to \{0,1\}$ be a function, where $|X| = |Y| = 2^\ell$. Then, for every finite field $\mathbb{F}$, every integer $t \leq |X|/(2\log^2|X|) \leq 2^{\ell-1}/\ell^2$, and every $\mathcal{Z}_2 \subseteq \binom{|Y|}{\leq t}$, there is a linear 2-server $(2^X, \mathcal{Z}_2)$-RCDS protocol for f with one-element secrets in which the message size is $O((t\log^2 t + 2^{\ell/2})\ell \log t \log|\mathcal{Z}_2| \log|\mathbb{F}|)$. Furthermore, there is p_0 such that for every prime-power $q > p_0$, there is a multi-linear 2-server $(\binom{X}{\leq |X|}, \mathcal{Z}_2)$-RCDS protocol for f over $\mathbb{F}_q$ with secrets of size $\Theta(q\ell \log t \log|\mathcal{Z}_2|)$ in which the normalized message size is $O(t\log^2 t + 2^{\ell/2})$.*

Proof. As the protocol of Lemma D.4, the desired protocol $\mathcal{P}_{\text{L2RCDS}}$ is a special case of the protocol of Lemma 4.9, when $k = k' = 2, t' = \log t$, and $\mathcal{P}$ is the protocol $\mathcal{P}_2^t$ of Lemma D.4. Let $H_{|Y|,t,\log t,2t} = \{h_i : Y \to [2t] : i \in [\ell]\}$, where $\ell = \Theta(\log|\mathcal{Z}_2|)$, be the family of hash functions promised by Lemma C.7 for $\mathcal{Z}_2$ (that is, for every $Z_2 \in \mathcal{Z}_2$, at least $\ell/4$ hash functions prevent a collision of $\log t$ elements of Z_2).

The protocol $\mathcal{P}_{\text{L2RCDS}}$ is described explicitly in Figure 7. It contains $2t\ell = O(t\log|\mathcal{Z}_2|)$ executions of the protocol $\mathcal{P}_2^{t'}$ with $t' = \log t$ and $|Y'| = |Y|/(2t)$ (since $t \leq |X|/(2\log^2|X|)$, we have that $\log t \leq$

Protocol $\mathcal{P}_{\mathrm{L2RCDS}}$

The secret: An element $s \in \mathbb{F}$.
The protocol:

1. Choose ℓ random elements $s_1, \dots, s_\ell \in \mathbb{F}$ such that $s = s_1 + \cdots + s_\ell$.
2. For every $i \in [\ell]$ do:
   - Let $Y_j = \{y \in Y : h_i(y) = j\}$, for every $j \in [2t]$.
   - For every $j \in [2t]$, independently execute the linear 2-server $(2^X, \binom{Y}{\leq \log t})$-RCDS protocol $\mathcal{P}_2^{\log t}$ of Lemma D.4 for the restriction of f to $X \times Y_j$ with the secret s_i. That is, Q_1 with input x sends a message for the restriction of f to $X \times Y_j$ for every $j \in [2t]$, and Q_2 with input y sends a message only for the restriction of f to $X \times Y_{h_i(y)}$.

Figure 7: A linear 2-server $(2^X, \mathcal{Z}_2)$-RCDS protocol $\mathcal{P}_{\mathrm{L2RCDS}}$ for a function $f : X \times Y \to \{0, 1\}$.

$\sqrt{|X|/(2t)}$ as required). Since Q_1 sends $2t$ messages of $\mathcal{P}_2^{t'}$ for every $h_i \in H_{|Y|,t,2t}$, her message contains

$$O\left((\log^3 t + \frac{2^\ell \log t/(2t)}{2^{\ell/2}})\ell \cdot 2t \log |\mathcal{Z}_2|\right) = O((t \log^2 t + 2^{\ell/2})\ell \log t \log |\mathcal{Z}_2|)$$

field elements. Since Q_2 sends only one message of $\mathcal{P}_2^{t'}$ for every $h_i \in H_{|Y|,t,2t}$, his message contains $O(2^{\ell/2}\ell \log t \log |\mathcal{Z}_2|)$ field elements.

To construct the desired protocol for long secrets, let $s = (s'_1, \dots, s'_{\ell/4}) \in (\mathbb{F}^{\ell'})^{\ell/4}$ be the secret, where $\ell' = \Theta(\ell \log t)$. Similarly to the multi-linear protocol of Lemma D.4, we change step 1 in the protocol $\mathcal{P}_{\mathrm{L2RCDS}}$ such that $s_1, \dots, s_\ell \in \mathbb{F}^{\ell'}$ are the shares of a $(3\ell/4, \ell, \ell)$-ramp secret-sharing scheme of the secret $s = (s'_1, \dots, s'_{\ell/4}) \in (\mathbb{F}^{\ell'})^{\ell/4}$, but now in step 2 we execute the multi-linear 2-server $(2^X, \binom{Y}{\leq \log t})$-RCDS protocol of Lemma D.4, instead of the linear protocol.

Overall, we results in a 2-server $(2^X, \mathcal{Z}_2)$-RCDS protocol for f with secrets of size $\Theta(\ell' \ell \log |\mathbb{F}|) = \Theta(\ell \log t \log |\mathcal{Z}_2| \log |\mathbb{F}|)$ in which the normalized message size is $O(t \log^2 t + 2^{\ell/2})$. $\square$