Simplified PIR and CDS Protocols and Improved Linear Secret-Sharing Schemes

Bar Alon*
Department of Computer Science,
Ben Gurion University.
alonbar08@gmail.com

Amos Beimel*
Department of Computer Science,
Ben Gurion University.
amos.beimel@gmail.com

Or Lasri*
Department of Computer Science,
Ben Gurion University.
orshlomo@post.bgu.ac.il

October 8, 2024

Abstract

We consider 3 related cryptographic primitives, private information retrieval (PIR) protocols, conditional disclosure of secrets (CDS) protocols, and secret-sharing schemes; these primitives have many applications in cryptography. We study these primitives requiring information-theoretic security. The complexity of these primitives has been dramatically improved in the last few years are they are closely related, i.e., the the 2-server PIR protocol of Dvir and Gopi (J. ACM 2016) was transformed to construct the CDS protocols of Liu, Vaikuntanathan, and Wee (CRYPTO 2017, Eurocrypt 2018) and these CDS protocols are the main ingredient in the construction of the best known secret-sharing schemes. To date, the messages size required in PIR and CDS protocols and the share size required in secret-sharing schemes is not understood and there are big gaps between their upper bounds and lower bounds. The goal of this paper is to try to better understand the upper bounds by simplifying current constructions and improving their complexity.

We obtain the following two independent results:

- We simplify, abstract, and generalize the 2-server PIR protocol of Dvir and Gopi (J. ACM 2016) and the 2-server and multi-server CDS protocols of Liu et al. (CRYPTO 2017, Eurocrypt 2018) and Beimel, Farràs, and Lasri (TCC 2023). This is done by considering a new variant of matching vectors and by using a general share conversion. In addition to simplifying previous protocols, our protocols can use matching vectors over any m that is product of two distinct primes. Our construction does not improve the communication complexity of PIR and CDS protocols; however, construction of better matching vectors over *any* m that is product of two distinct primes will improve their communication complexity.
- In many applications of secret-sharing schemes it is important that the scheme is linear, e.g., by using the fact that parties can locally add shares of two secrets and obtain shares of the sum of the secrets. We provide a construction of linear secret-sharing schemes

*Partially supported by ISF grant 391/21 and by the Frankel center for computer science.

for n-party access structures with improved share size of $2^{0.7563n}$. Previously, the best share size for linear secret-sharing schemes was $2^{0.7576n}$ and it is known that for most n-party access structures the shares size is at least $2^{0.5n}$. This results is achieved by a reduction to unbalanced CDS protocols (compared to balanced CDS protocols in previous constructions).

1 Introduction

Private information retrieval (PIR) protocols, conditional disclosure of secrets (CDS) protocols, and secret-sharing schemes are cryptographic primitives that have many applications (these primitives are defined in Section 1.1). We study these primitives requiring information-theoretic security. These primitives are closely related, e.g., the same techniques are used to construct PIR and CDS protocols and CDS protocols are used to construct secret-sharing schemes for arbitrary access structures. Furthermore, the goal of these primitives is to protect the secrecy of some inputs (i.e., a database, private inputs of servers, or a secret), and they are non-interactive.

The complexity of PIR and CDS protocols and secret-sharing schemes has been dramatically improved in the last few years; yet their complexity is far from being understood. For example, the share size in the best known secret-sharing schemes for arbitrary n-party access structure is exponential, i.e., $2^{O(n)}$, while the best lower bound is $\Omega(n^2/\log n)$ [29, 28]. Determining the optimal complexity of these primitives is a major open problem. Improving the known upper bounds will lead to better complexity in the protocols that use them, and proving better lower bounds will show their limitations. Furthermore, understanding the exact complexity of these primitives may lead to understanding the optimal complexity of more complex interactive cryptographic primitives, e.g., the communication complexity required for information-theoretic secure multi-party computation (MPC) protocols. Our goal in this paper is to advance the understanding of the upper bounds for PIR and CDS protocols and secret-sharing schemes, especially, linear secret-sharing schemes. Towards this goal, we will try simplify and generalize the current constructions, provide new tools for better constructions, and construct better schemes.

1.1 PIR, CDS, and Linear Secret Sharing

Before presenting our results, we informally discuss the primitives we study in this work.

Private Information Retrieval. Private information retrieval protocols enable a user to obtain an item from a database held by two or more servers such that each server does not learn information on the retrieved item. Specifically, in a 2-server PIR protocol, a user holds an index $i \in [N]$, and two servers, Alice and Bob, each holds the same database $D \in \{0,1\}^N$. The goal is for the user to learn D_i, using one round of communication, without revealing any information about i to each server. To achieve this goal, the user computes a pair of queries q_A and q_B and sends them to Alice and Bob respectively. Each server computes an answer, based on the query that it got and the database, and sends the answer to the user, which reconstructs D_i from the index, its randomness, and the answers.

Private information retrieval protocols were introduced by Chor, Goldreich, Kushilevitz, and Sudan [26] in 1995. Since then, the communication complexity of private information retrieval protocol has been studied in a line of works [2, 42, 45, 16, 49, 64, 65, 35, 46, 25, 34]. Specifically, Efremenko [35] constructed a 3-server PIR protocol with query length $2^{O\left(\sqrt{\log N \log\log N}\right)}$ and answer length is 1, which is currently the best known 3-server PIR protocol. Dvir and Gopi [34], broke the

$N^{1/3}$ barrier for the communication complexity for 2-server PIR protocol, and showed a construction with communication complexity of $2^{O(\sqrt{\log N \log\log N})}$, which currently the best known 2-server PIR protocol. Both works of Efremenko [35], and Dvir and Gopi [34] are based on matching vector families. Beimel, Ishai, Kushilevitz, and Orlov [19] generalize the 3-server protocol of Efremenko by using share conversions; specifically, they can use matching vectors over more products of primes m. The best known lower bound on the total communication complexity of 2-server PIR protocols is $5 \log n$, proved by Wehner and de Wolf [63] (improving on [55, 50]).

Conditional Disclosure of Secrets. Conditional disclosure of secrets (CDS) protocol are a cryptographic primitive, introduced by Gertner, Ishai, Kushilevitz, and Malkin [39]. Their motivation was to construct symmetric private information retrieval protocols. CDS protocols were later used in constructions of other cryptographic applications, such as attribute based encryption [38, 10, 62] and priced oblivious transfer [1]; they are a central tool in the construction of secret-sharing schemes for arbitrary access structures [52, 6, 9, 20].

In a CDS protocol, several servers hold the same secret and a common random string, and each server holds a private input. Additionally, there is a referee who knows the private inputs of all servers. The referee should learn the secret if and only if the private inputs of the servers satisfy some condition, specified by a predicate f. To achieve this goal, each server sends a single message to the referee; the message of each server is a function of the secret, the common random string, and its private input. The referee can reconstruct the message from the messages if and only if the inputs satisfy the condition.

Constructions of CDS protocols were given in [40, 17, 38, 53, 54, 21, 4, 3]. The best known 2-server CDS protocol has message length $2^{O(\sqrt{\log N \log\log N})}$ [53]. The best known lower bounds for 2-server CDS protocols is $\Omega(\log N)$, proved by Applebaum, Arkis, Raykov, and Vasudevan [5] (see also [38, 8]). The best k-server CDS construction is due to Liu et al. [54] and has a message size of $2^{\tilde{O}(\sqrt{k \cdot \log N})}$. Applebaum and Arkis [3] (improving on [4]) constructed a CDS protocol for long secrets, where the message length is only 4 times the length of the secret.

Secret Sharing. Secret-sharing schemes, introduced by Shamir [61] and Blakley [23] for the threshold case and Ito, Saito, and Nishizeki [44] for the general case, allows a dealer holding a secret to distribute strings (called shares) to parties, such that only authorized sets of parties can reconstruct the secret, while unauthorized sets learn nothing about the secret. The collection of authorized sets is called an access structure. Secret sharing has found many applications in cryptography, distributed computing, and complexity theory (see [12]). Identifying the necessary and sufficient share size of secret-sharing schemes for general access structures is a major open problem. The best-known schemes for n-parties access structure achieve share complexity of 2^{cn} for a constant $c < 1$ [52, 6, 7, 9], with Applebaum and Nir [9] constructing the best scheme, which achieves share size $1.5^n < 2^{0.585n}$. On the negative side, the best lower bound on the total share size is $\Omega(n^2/\log n)$ due to Csirmaz [29, 28].

Linear Secret Sharing. Linear secret-sharing are schemes in which the shares are computed by applying a linear function (over some finite field) on the secret and some random elements from the field. Alternatively, these are schemes in which each share is a vector over the field and every authorized set reconstruct the secret by applying a linear function on its shares. In many applications of secret-sharing schemes it is important that the scheme is linear, e.g., they use the fact that parties can locally add shares of two secrets and obtain shares of the sum of the

secrets. Such applications include the secure multi-party computation protocol secure against an arbitrary (Q2) adversary structure [27] and the construction of public-key (multi-user) attribute-based encryption [10, 62]. Prior to our work, the best previous linear scheme of [9] has share size $2^{0.7576n}$ for any n-party access structure. On the other hand, it is known that almost all n-party access structures cannot be realized by a linear secret-sharing scheme with share size less $2^{0.5n}$, as proved by Babbai, Gal, and Wigderson [11].

Until recently, most of the constructions of secret-sharing schemes were linear, e.g., [61, 24, 44, 22, 48]. In particular, linear secret-sharing schemes are equivalent to monotone span programs, a linear-algebraic model of computation introduced by Karchmer and Wigderson [48]. Linear secret-sharing schemes have many advantages, e.g., they are homomorphic, the sharing and reconstruction are efficient, and they are closed under duality. Lower bounds on the share size in linear secret-sharing schemes and monotone span programs for explicit access structures were proven in [15, 11, 36, 37, 59, 57, 58]; the best result is that there exists explicit access structures for which every linear secret-sharing scheme realizing the access structures has shares of length at least 2^{cn} for some constant $0 \leq c < 0.5$, proved by Pitassi and Robere [58]. Lower bounds for share size in linear secret-sharing schemes for almost all access structures were proven in [11, 60, 13], in particular, almost all access structures require shares of length at least $2^{0.5n}$ in any linear secret-sharing scheme realizing them [11].

All the above primitives are closely related, e.g.,

- The 2-server PIR protocol of Dvir and Gopi [33] was transformed to construct the CDS protocols of Liu et al. [53, 54].
- CDS protocols are basically a special case of secret-sharing schemes, i.e., 2-server CDS protocols are equivalent to secret-sharing schemes for forbidden bipartite graph access structures.
- CDS protocols are a central ingredient in constructing the best known secret-sharing schemes for arbitrary access structures, see [52, 7, 9]. Similarly, the best known linear secret-sharing schemes for arbitrary access structures are constructed from linear CDS protocols.

Furthermore, in all the above primitives the optimal communication complexity/share-size is not known and there are large gaps between the known lower bounds and upper bounds.

1.2 Our Results

We provide two new techniques addressing the upper bounds for the above primitives.

Abstraction of the DG PIR protocol and the LVW CDS protocol. Our first result is an abstraction of the 2-server PIR protocol of Dvir and Gopi [34] and the CDS protocol of Liu et al. [53] and its generalization by Beimel, Farràs, and Lasri [14], henceforth the DG, LVW, BFL protocols, respectively. Although we do not obtain (asymptotically) better communication, our constructions have the benefit of being both simpler and more general. The latter could potentially lead to improvements in the future.

In more detail, the DG, LVW, and BFL protocols use matching vectors [41] – a combinatorial object that was used to construct explicit Ramsey graphs [41], and later found other applications in computer science such as error-correcting codes [32], PIR protocols [35, 34], and CDS protocols [53, 54]. Roughly speaking, a matching vector family over $\mathbb{Z}_m^h$ is a collection of vectors $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$, each in $\mathbb{Z}_m^h$, such that for all $i \neq j$ it holds that $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \in S$ and $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \in T$, where S and T are

disjoint subsets of $\mathbb{Z}_m$, and the inner products are done mod m. The DG and LVW constructions use $m = 6$ and the BFL construction uses $m = p_1 \cdot p_2$ for any two primes such that $p_1 \mid p_2 - 1$. Our construction, on the other hand, works for any $m = p_1 p_2$, where $p_1 \neq p_2$ are prime numbers. Therefore, any improved construction of matching vectors (for some m, such that $\log m = 2^{o(\sqrt{\log N \log\log N})}$) immediately implies an improved construction of PIR and CDS protocols.

Theorem 1.1 (Informal, simple PIR protocols)**.** *Let $f : [N]^2 \to \{0,1\}$, let $p_1 \neq p_2$ be two prime numbers, $m = p_1 p_2$, and $h \in \mathbb{N}$. Assume there exists a matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$. Then there exist a 2-server PIR protocol with message size $O(h \cdot \log m)$.*

Theorem 1.2 (Informal, simple CDS protocols)**.** *Let $f : [N]^2 \to \{0,1\}$, let $p_1 \neq p_2$ be two prime numbers, $m = p_1 p_2$, and $h \in \mathbb{N}$. Assume there exists a matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$. Then there exist a 2-server CDS protocol with message size $O(h \cdot \log m)$.*

More importantly, our protocols abstract and generalize the previous protocols and are simpler.

A similar result holds for k-server CDS protocols, i.e., if there is a family of *decomposable* matching vectors in $\mathbb{Z}_m^h$ (for m a product of two distinct primes), then there is k-server CDS protocol with message length $O(k^2 h \log m)$.

Improved results for linear secret sharing. We generalize the reduction from secret-sharing schemes to CDS protocols. This generalization allows us to obtain better *linear* secret-sharing schemes, where the sharing and reconstruction algorithms are both linear mappings. Specifically, while the best previous linear scheme due to [9] has share size $2^{0.7576n}$ for any n-party access structure, we reduce the exponent down to $0.7563n$. Recall that best known share size for general secret-sharing schemes for arbitrary access structures is $2^{0.585n}$.

Theorem 1.3 (Informal, linear secret-sharing schemes)**.** *Every n-party access structure can be realized by a linear secret-sharing scheme with share size $2^{0.7563n}$.*

2 Our Techniques

2.1 A Simple PIR Protocol

Dvir and Gopi [34] presented a 2-server PIR protocol with communication $2^{O(\sqrt{\log N \log\log N})}$; their protocol uses matching vectors over $\mathbb{Z}_6$. Beimel, Farràs, and Lasri [14] implicitly generalized this protocol by using matching vectors over $\mathbb{Z}_m$, where m is a product of two primes p_1, p_2 such that $p_1 | p_2 - 1$. We simplify and generalize these protocols.

Warm-up. We start with describing an inefficient two server PIR protocol that will provide the motivation for our protocol. Let $(\mathbf{u}_i)_{i=1}^N \in \mathbb{F}_p^h$ be a set of orthonormal vectors over some fine field $\mathbb{F}_q$ and $h \in \mathbb{N}$, that is, $\langle \mathbf{u}_i, \mathbf{u}_i \rangle \equiv 1 \pmod p$ and $\langle \mathbf{u}_i, \mathbf{u}_j \rangle \equiv 0 \pmod p$ for every $i \neq j$. The user with index i chooses a random $\mathbf{r} \in \mathbb{F}_p^h$ and sends $\mathbf{q}_A = \mathbf{r}$ to Alice and $\mathbf{q}_B = \mathbf{r} + \mathbf{u}_i \bmod p$ to Bob. A server with query $\mathbf{q}$ and database D computes the answer $\sum_{j=1}^N \langle \mathbf{q}, \mathbf{u}_j \rangle D_j$ and sends the answer to the user, which subtracts the answer of Alice from the answer of Bob and obtains D_i as we next

explain.

$$\sum_{j=1}^{N}\langle \mathbf{r}+\mathbf{u}_i,\mathbf{u}_j\rangle D_j - \sum_{j=1}^{N}\langle \mathbf{r},\mathbf{u}_j\rangle D_j \equiv \sum_{j=1}^{N}(\langle \mathbf{r}+\mathbf{u}_i,\mathbf{u}_j\rangle - \langle \mathbf{r},\mathbf{u}_j\rangle)D_j$$
$$\equiv \sum_{j=1}^{N}\langle \mathbf{u}_i,\mathbf{u}_j\rangle D_j \equiv D_i \pmod{p},$$

where the last equality follows from the orthonormality of the vectors. The obvious problem with this construction is that the length of N orthonormal vectors is at least N and the protocol is not efficient. Following [35], we will work with vectors over $\mathbb{Z}_m$ for a composite m; specifically, m is a product of two distinct primes p_1, p_2. As every set of orthonormal vectors over $\mathbb{Z}_m$ is also orthonormal over $\mathbb{F}_{p_1}$, we need to relax the orthonormality requirements.

Matching Vectors. We use a matching vector family $((\mathbf{u}_i,\mathbf{v}_i))_{i=1}^{N}$ over $\mathbb{Z}_m^h$, where $m = p_1p_2$ for primes $p_1 < p_2$, such that for all $i \neq j$ it holds that

$$\langle \mathbf{u}_i,\mathbf{v}_i\rangle \bmod m = 1 \quad \text{and} \quad \langle \mathbf{u}_i,\mathbf{v}_j\rangle \bmod m \in \mathbb{Z}_m \setminus \mathbb{Z}_m^*.$$

Observe that this is equivalent to

$$\langle \mathbf{u}_i,\mathbf{v}_i\rangle \bmod p_1 = 1 \quad \text{and} \quad \langle \mathbf{u}_i,\mathbf{v}_i\rangle \bmod p_2 = 1$$

and

$$\langle \mathbf{u}_i,\mathbf{v}_j\rangle \bmod p_1 = 0 \quad \textit{or} \quad \langle \mathbf{u}_i,\mathbf{v}_j\rangle \bmod p_2 = 0.$$

Such a matching vector family with $h = 2^{O(\sqrt{\log N \log\log N})}$ can be constructed from the matching vector families constructed in [41, 51]. Note that this definition is a modification of the definition of matching vectors in previous papers, where it is required that $\langle \mathbf{u}_i,\mathbf{v}_i\rangle \bmod m = 0$ and the requirement for $i \neq j$ is also different. This modification allows us to simplify the protocol.

In the following, for a prime p let $\langle \mathbf{u},\mathbf{v}\rangle_p = \sum_{\ell=1}^{h}\mathbf{u}[\ell]\mathbf{v}[\ell] \bmod p$.

Protocol 2.1.
Public parameters: *Matching vectors* $((\mathbf{u}_i,\mathbf{v}_i))_{i=1}^{N}$ *over* $\mathbb{Z}_m^h$, *where* $m = p_1p_2$ *for two primes* $p_1 < p_2$.
Alice's and Bob's input: $D \in \{0,1\}^N$.
The user's input: $i \in [N]$.

- *The user chooses* $\mathbf{r} \leftarrow \mathbb{Z}_{p_1}^h$ *with uniform distribution and sends* $\mathbf{q}_A = \mathbf{r}$ *to Alice and* $\mathbf{q}_B = \mathbf{u}_i + \mathbf{r} \bmod p_1$ *to Bob.*
- *Alice and Bob compute* $\mathbf{m}_A = \sum_{j=1}^{N}(\langle \mathbf{q}_A,\mathbf{v}_j\rangle_{p_1} \cdot D_j)\mathbf{v}_j \bmod p_2$ *and* $\mathbf{m}_B = \sum_{j=1}^{N}(\langle \mathbf{q}_B,\mathbf{v}_j\rangle_{p_1} D_j)\cdot \mathbf{v}_j \bmod p_2$ *respectively and send the answers to the user (each answer is a vector in* $\mathbb{Z}_{p_2}^h$*).*
- *The user outputs 1 if*

$$\langle \mathbf{u}_i, \mathbf{m}_B - \mathbf{m}_A\rangle \not\equiv 0 \pmod{p_2}, \tag{1}$$

and 0 otherwise.

For comparison, we describe the simplest version of the PIR protocol of [34] in Appendix B. We next prove that Protocol 2.1 is a PIR protocol. Each query to a server is uniformly distributed in $\mathbb{Z}_{p_1}^h$ regardless of i and the privacy clearly holds. We next show that correctness holds; this should be carefully analyzed as we use inner product over $\mathbb{Z}_{p_1}$ and $\mathbb{Z}_{p_2}$. The user computes

$$\begin{aligned}\langle \mathbf{u}_i, \mathbf{m}_B - \mathbf{m}_A \rangle_{p_2} &\equiv \left\langle \mathbf{u}_i,\ \sum_{j=1}^{N} \langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j - \sum_{j=1}^{N} \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j \right\rangle_{p_2} \\ &\equiv \sum_{j=1}^{N} \left(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \right) \cdot \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} \cdot D_j \pmod{p_2}. \end{aligned} \tag{2}$$

We claim that this sum is equal to $\alpha \cdot D_i$ for some $\alpha \neq 0$, that is, in the sum in (2) each term for $j \neq i$ is zero and the term for $j = i$ is non-zero if and only if $D_i = 1$.

We claim that for $i \neq j$

$$\left(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \right) \cdot \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} \cdot D_j \equiv 0 \pmod{p_2}.$$

Clearly, this is true if $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} = 0$. Otherwise, $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} = 0$; thus,

$$\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \equiv \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} + \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \equiv \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \pmod{p_1}.$$

It follows that $\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} = \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}$ (since $0 \leq \langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}, \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} < p_1$). As $p_1 < p_2$, the equality holds modulo p_2.

Therefore,

$$\begin{aligned}\langle \mathbf{u}_i, \mathbf{m}_B - \mathbf{m}_A \rangle_{p_2} &\equiv \sum_{j=1}^{N} \left(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \right) \cdot \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} \cdot D_j \\ &\equiv \left(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} \right) \cdot \langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_2} \cdot D_i \pmod{p_2}. \end{aligned}$$

Now,

$$\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1} \equiv \left(\langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_1} + \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} \right) \bmod p_1 \equiv 1 + \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} \pmod{p_1}.$$

If $\langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} < p_1 - 1$ then $\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1} = 1 + \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1}$ over $\mathbb{F}_{p_2}$ and $\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} \equiv 1 \pmod{p_2}$. Otherwise, $\langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} = p_1 - 1$, and over $\mathbb{F}_{p_2}$

$$\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1} - \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} = 0 - \langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1} = 1 - p_1.$$

In either case, it follows that

$$\langle \mathbf{u}_i, \mathbf{m}_B - \mathbf{m}_A \rangle_{p_2} = \alpha \cdot D_i, \tag{3}$$

for some $\alpha \neq 0$ and the reconstruction of D_i is correct.

2.2 A Simple CDS Protocol

We construct a CDS protocol for the index function $\text{INDEX} : \{0,1\}^N \times [N] \to \{0,1\}$ defined as $\text{INDEX}(D, i) = D_i$. Note that a CDS for INDEX implies a CDS for any other function $f : X \times Y \to \{0,1\}$ by letting $D = (f(x,y))_{y \in Y}$. We make the observation that for all $s \in \{0,1\}$, $i \in [N]$, and $\mathbf{r} \in \mathbb{Z}_m^h$

$$\langle \mathbf{u}_i,\ \sum_{j=1}^{N} \langle s \cdot \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j - \sum_{j=1}^{N} \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j \rangle_{p_2} = s \cdot \alpha \cdot D_i, \tag{4}$$

where $\alpha \neq 0$. Indeed, if $s = 1$ then (4) reduces to (3), and if $s = 0$ then the two terms cancel out to the all-zero vector.

We describe the CDS protocol for INDEX in Protocol 2.2.

Protocol 2.2.
Public parameters: *Matching vectors* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ *over* $\mathbb{Z}_m^h$, *where* $m = p_1 p_2$ *for two primes* $p_1 < p_2$.
Alice's input: $D \in \{0,1\}^N$.
Bob's input: $i \in [N]$.
The secret: $s \in \{0,1\}$.
Shared randomness: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$, $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h$.

- *Alice sends* $\mathbf{m}_A = \left(\sum_{j=1}^N \langle \mathbf{r}_1, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j + \mathbf{r}_2\right) \bmod p_2$.
- *Bob sends* $\mathbf{m}_B^1 = s\mathbf{u}_i + \mathbf{r}_1 \bmod p_1$, $m_B^2 \leftarrow \langle \mathbf{u}_i, \mathbf{r}_2 \rangle \bmod p_2$
- *The referee outputs* $s = 1$ *if*

$$\left\langle \mathbf{u}_i, \left(\sum_{j=1}^N (\langle \mathbf{m}_B^1, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j)\right) - \mathbf{m}_A \right\rangle_{p_2} + m_B^2 \not\equiv 0 \pmod{p_2}, \tag{5}$$

  and $s = 0$ *otherwise.*

Correctness follows from (4),

$$\begin{aligned}
&\langle \mathbf{u}_i, \left(\sum_{j=1}^N (\langle \mathbf{m}_B^1, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j)\right) - \mathbf{m}_A \rangle_{p_2} + m_B^2 \\
&\quad \equiv \langle \mathbf{u}_i, \sum_{j=1}^N \langle s \cdot \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j - \sum_{j=1}^N \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j \rangle_{p_2} \\
&\quad \equiv s \cdot \alpha \cdot D_i \pmod{p_2}.
\end{aligned}$$

For the security, notice that by (4) for $D_i = 0$,

$$\left\langle \mathbf{u}_i, \left(\sum_{j=1}^N (\langle \mathbf{m}_B^1, \mathbf{v}_j \rangle_{p_1} \cdot D_j \mathbf{v}_j)\right) - \mathbf{m}_A \right\rangle_{p_2} + m_B^2 \equiv 0 \pmod{p)_2}.$$

The messages $\mathbf{m}_A$ and $\mathbf{m}_B^1$ are uniformly distributed (as they are masked by a one-time pad), and m_B^2 can be computed from these messages and the database held by the referee. Thus, the distribution of the view of the referee is independent of the secret.

In Section 4, we show more general PIR and CDS protocols that do not assume $p_1 < p_2$. Note that if $p_2 | p_1 - 1$, then (4) does not necessarily hold. To resolve this, we abstract the properties of the operator $\langle \mathbf{q}, \mathbf{v} \rangle_{p_1}$ that we use in Protocol 2.1 and Protocol 2.2; instead the servers apply a "so-called" *share conversion* $C(\langle \mathbf{q}, \mathbf{v} \rangle)$. There are a few reasons to describe the protocols using a general share conversion:

- The proofs are somewhat cleaner using this notation.
- This provides a more general protocol; in particular, it captures the BFL protocol (which generalizes the DG protocol and the LVW protocol).
- In some scenarios, we will want the servers in Protocol 2.1 to send answers over a specific field, e.g., given matching vectors over $\mathbb{Z}_{21}$, we will want to work in $\mathbb{F}_7$. We construct share conversions that enable such property.

Remark 2.3. *In [54], they use the notion of PIR encoding to describe their CDS protocol, which looks similar to our protocol. However, the PIR encoding they construct for 2-servers is actually the same as in [34, 53] and their construction and proofs are more complicated than our construction and proofs.*

2.3 An Improved Linear Secret Sharing

Our construction of an improved linear secret-sharing scheme follows the ideas of the recent constructions of linear secret-sharing schemes [52, 6, 7, 9], specifically, we use the blue-print of Applebaum and Nir [9]. This construction uses robust CDS protocols to realize downslice access structures (see definition below). It then uses covering and bootstrapping techniques to construct better linear secret-sharing schemes for downslice access structures. As every n-party access structure can be written as an intersection of n downslice access structures, this implies a linear secret-sharing scheme for every access structure.

Let $0 < b < n$. A b-downslice access structure Γ is an access structure where all maximal unauthorized sets are of size b, that is there are some sets $A_1, \ldots, A_\ell$ of size b such that $A \notin \Gamma$ if and only if there is some $1 \leq i \leq \ell$ such that $A \subseteq A_i$.

Example 2.4. *Consider the access structure with n parties $\{p_1, \ldots, p_n\}$, where a set is A authorized if and only if it contains at least one party from $\left\{p_1, \ldots, p_{n/2}\right\}$ and at least one party from $\left\{p_{n/2+1}, \ldots, p_n\right\}$; this is an $n/2$-downslice in which the maximal unauthorized sets are $\left\{p_1, \ldots, p_{n/2}\right\}$ and $\left\{p_{n/2+1}, \ldots, p_n\right\}$.*

Every access structure Γ can be written as $\cap_{1 \leq b \leq n} \Gamma_b$, where Γ_b is the b-downslice access structure whose maximal unauthorized sets are the maximal unauthorized sets of Γ of size b.

Our construction is implied by a better construction of linear secret-sharing schemes for n-party b-downslice access structures for $b > n/2$. We next explain the idea of our improvement. We achieve this goal by a reduction to CDS protocols; specifically to robust CDS protocols (as introduced in [7]). Previous construction of linear secret-sharings schemes using robust CDS protocols use either reductions to 2-server protocols [7] or to $\sqrt{n}$-server protocols [7, 9]; our improvement is via reduction to 2-server CDS protocols.

We first recall the notion of robust CDS protocols. The security requirement of a CDS protocol for a function f ensures that if Alice sends its message for an input x and Bob sends its message for an input y such that $f(x, y) = 0$, then the referee does not learn any information on the secret from these messages. Assume that Alice also sends a message for an inputs $x' \neq x$ with the same shared randomness, such that also $f(x', y) = 0$; the CDS protocol does not guarantee that the referee does not learn any information on the secret from the 3 messages. In a (t_1, t_2)-robust CDS protocol the security is guaranteed even if Alice sends messages for a set X_1 with at most t_1 inputs and Bob sends messages for a set X_2 with at most t_2 inputs (such that $f(x, y) = 0$ for every $x \in X_1, y \in X_2$), then the referee does not learn information on the secret from the $t_1 + t_2$ messages.

Warm-up. We show in Scheme 2.5 that, given an access structure Γ, an efficient (N, N)-robust CDS protocol for a function f_Γ defined below implies a good secret-sharing scheme for Γ. Let $\{p_1, \ldots, p_n\}$ be the parties of Γ, $P_1 = \left\{p_1, \ldots, p_{n/2}\right\}$ and $P_2 = \left\{p_{n/2+1}, \ldots, p_n\right\}$. Define $f_\Gamma : 2^{P_1} \times 2^{P_2} \to \{0,1\}$, where for $A_1 \subseteq P_1, A_2 \subseteq P_2$

$$f_\Gamma(A_1, A_2) = 1 \quad \iff \quad A_1 \cup A_2 \in \Gamma;$$

the size of the domain of each server is $N = 2^{n/2}$.

Scheme 2.5.
Public parameters: *An (N, N)-robust CDS protocol $\mathcal{P}$ for f_Γ.*
The secret: $s \in \{0,1\}$.

- *Choose a random string r for the CDS protocol.*
- *For every $A_1 \in 2^{P_1}$, compute m_{A_1} – the message in $\mathcal{P}$ of Alice with input A_1 and random string r, share m_{A_1} in a $|A_1|$-out-of-$|A_1|$ secret-sharing scheme, and give each share to a party in A_1.*
- *For every $A_2 \in 2^{P_2}$, compute m_{A_2} – the message in $\mathcal{P}$ of Bob with input A_2 and random string r, share m_{A_2} in a $|A_2|$-out-of-$|A_2|$ secret-sharing scheme, and give each share to a party in A_2.*

For a set A, let $A_1 = A \cap P_1, A_2 = A \cap P_2$. If $A \in \Gamma$, $f_\Gamma(A_1, A_2) = 1$, thus the secret can be reconstructed from m_{A_1}, m_{A_2}. As the parties in A can reconstruct these messages, the correctness follows. For the security, consider a set $A \notin \Gamma$. The parties in A can reconstruct only the messages M_{B_1}, M_{B_2} for every $B_1 \subseteq A_1$ and $B_2 \subseteq A_2$. As Γ is monotone, $B_1 \cup B_2 \notin \Gamma$ for every such B_1, B_2, thus, $f_\Gamma(B_1, B_2) = 0$. By the (N, N)-robustness of the CDS protocol, the parties in A do not know any information on the secret.

The problem in the above construction is the share size. The share of each party $p_i \in P_1$ is a share of the message of each subset $A_1 \in P_1$ such that $p_i \in A_1$, there are $2^{n/2-1}$ such sets. The best known (N, N)-robust CDS protocol for a function $f : [N] \times [N] \to \{0,1\}$ has message size $O(N/\log N)$; in our case $N = 2^{n/2}$. Thus, the total share of each party is $O(2^{n/2-1} \cdot N/\log N) = 2^{n-o(n)}$. This share size is too big as our goal is share size 2^{cn} for some constant $c < 1$.

Linear secret-sharing schemes for somewhat regular access structures. As in previous papers, to improve the share size we reduce the question of constructing a linear secret-sharing scheme for an arbitrary access structure Γ to the question of constructing a (t, N_2)-robust CDS protocol for functions $f : [N_1] \times [N_2] \to \{0,1\}$ for some $t = o(N_1)$; the message size of Alice in the best known linear CDS protocol for such f is $\tilde{O}(N_1/t)$ and the message size of Bob is $\tilde{O}(t)$ (up to polynomial factors).[1] Notice that the message sizes of Alice and Bob are not the same when $t \not\approx \sqrt{N_1}$.

We use a different approach, which results in a better share size. Rather than consider $|P_1| = |P_2| = n/2$ as in the warm-up and [7], we take P_1, P_2 such that $|P_1| = \mu n$ and $|P_2| = (1-\mu)n$ for some parameter $0 < \mu < 1$ and define $f_\Gamma : \{0,1\}^{\mu n} \times \{0,1\}^{(1-\mu)n} \to \{0,1\}$ with respect to

[1] To get this message size we need a more refined notion of robustness called $(\mathcal{Z}, N)$-robustness. See Definition 3.9 for a definition of this notion and Section 5.1 for the details why we can use it.

this partition. Following [7], we show in Section 5.1 that to realize b-downslice access structure it suffices to consider somewhat regular access structure, which are access structures in which every maximal unauthorized set A is "well partitioned", i.e., $|A \cap P_1| \leq \mu b$.[2] We execute Scheme 2.5 with the modified f_Γ using a (t, N)-robust CDS protocol for $t = 2^{\mu b}$.

The correctness follows as in the warm-up. For the security, consider an unauthorized set A; recall that $|A \cap P_1| \leq \mu b$, thus, the parties in A learn at most $t = 2^{\mu b}$ messages in the CDS protocol $\mathcal{P}$ (one message for each $A' \subseteq A$). The share of each $p_i \in P_1$ contains $2^{\mu n}$ shares of messages of Alice, each message of size $\tilde{O}(N_1/t) = 2^{\mu n}/2^{\mu b}$; i.e., the share size is $\tilde{O}(2^{\mu(2n-b)})$. Similarly, the share of each $p_i \in P_2$ contains $2^{(1-\mu)n}$ shares of messages of Bob, each message of size $\tilde{O}(t) = 2^{\mu b}$; i.e., the share size is $\tilde{O}(2^{n-\mu(n-b)})$. To minimize the maximum share size, we take μ such that $2^{\mu(2n-b)} = 2^{n-\mu(n-b)}$, i.e., $\mu = \frac{n}{3n-2b}$; this results in share size $\tilde{O}(2^{n(2n-b)/(3n-2b)})$. For $b > n/2$ our scheme improves on the scheme of [9].

Example 2.6. *Consider $b = 0.5412n$ (this is the value that we will use in our construction). In this case $\mu = \frac{1}{3-2\cdot 0.5412} = 0.5214$ and the share size is $\tilde{O}(2^{0.7607n})$. Notice that in this case more parties are in P_1, the set of parties attached to Alice.*

Consider $b = 0.4n$. In this case $\mu = 0.4545 < 0.5$ and the share size is $\tilde{O}(2^{0.7272n})$. Notice that in this case less parties are in P_1, the set of parties attached to Alice.

Linear secret-sharing schemes for arbitrary access structures. We use the fact that any access structure Γ is the intersection of n downslices to realize an arbitrary access structure; however as the share size for b-downslices approaches 2^n as b approaches n, we cannot use the above schemes for downslices as is. We use two techniques from previous papers to reduce the share size. The first technique is the covering technique [6, 9], which shows that for every $a < b$, every b-downslice access structure with n parties can be realized as the intersection of roughly $\binom{n}{n-b}/\binom{n-a}{n-b}$ access structures, each one of them is a $b-a$-downslice access structure with $n-a$ parties. The second technique is the boosting technique from [9] showing that for linear secret-sharing schemes each b-downslice with n parties can realized by realizing b-downslice access structures with fewer parties (and some multislice access structures). This results in a recursive construction that uses a scheme for b-downslice access structures and results in a scheme for b-downslice access structures with better share size.

Following [9], our scheme for an arbitrary access structure will have the following structure:

- Write Γ as $\cap_{b=1}^{n} \Gamma_b$, where Γ_b is a b-downslice access structure.
- For every $b < 0.5n$, use the linear scheme of [9] to realize the b-downslice Γ_b.
- Use the boosting technique to realize $0.5n$ downslice access structures.
- For every $0.5n \leq b \leq 0.554n$, use the covering technique to realize the b-downslice Γ_b using the scheme for $0.5n$ downslice access structures.
- Use the covering technique to realize 0.554-downslice access structures using our scheme for $0.5214n$-downslice access structures.

[2] Actually, the bounds on the sizes that we get are bigger by a factor of $n^{0.2}$; we ignore this factor in this section. Furthermore, following [7], we will also require that minimal authorized sets are well partitioned; we do not use this property in this paper.

- Use the resulting scheme for $0.554n$-downslice access structures and the boosting technique to get a better scheme for $0.554n$-downslice access structures.
- For every $0.554n \leq b \leq n$, use the covering technique to realize the b-downslice Γ_b using the scheme for $0.554n$ downslice access structures.

The scheme in [9] had different constant instead of 0.554 and $0.5214n$. Finding the exact constants that optimize our scheme was done using a computer programs; this program also found the parameters in the boosting.

Remark 2.7. *In [9] and in this paper, we partition the possible values of $1 \leq b \leq n$ to 3 intervals. It might seem that taking more intervals can reduce the share size; however, this is not true. For example, the construction for $0.592n$-downslice access structures (via covering) results in the highest share size. For $b = 0.592n$, using the covering technique with $0.554n$ downslice access structures is optimal, i.e., using the covering technique with a different a-downslice will result is bigger share size. Although for other values $0.554n < b < n$ covering to a different downslice may improve the share size for b-downslice access structure, it will not improve the maximum share size for Γ.*

3 Preliminaries

3.1 Notations

For $n \in \mathbb{N}$ we use the notation $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We denote by log the logarithmic function with base 2. For strings $x, y \in \{0,1\}^n$ we write $x \leq y$ if $x_i \leq y_i$ for every $i \in [n]$. We denote the concatenated string as $x||y$. We let $\mathrm{wt}(x)$ denote the Hamming weight of x. We denote the *binary entropy function* by $h : [0,1] \to [0,1]$ and it is defined as $h(x) = -x \log x - (1-x)\log(1-x)$ for all $x \in (0,1)$, and $h(0) = h(1) = 0$. For every $p \in \mathbb{Z}$, we denote $\langle \cdot, \cdot \rangle_p$ the inner product modulo p, i.e., for every two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^h$, $\langle \mathbf{x}, \mathbf{y} \rangle_p = \sum_{\ell=1}^{h} \mathbf{x}[\ell] \cdot \mathbf{y}[\ell] \bmod p$. For every $m \in \mathbb{N}$, we define the equivalence relation over $\mathbb{Z} \times \mathbb{Z}$, for every $a, b \in \mathbb{Z}$, we say that $a \equiv b \pmod m$ if $a \bmod m = b \bmod m$.

3.2 Secret-Sharing Schemes

We start by defining (perfect) secret-sharing schemes.

Definition 3.1 (Access structures)**.** *Let $P = \{p_1, \ldots, p_n\}$ be a set of parties. A collection $\Gamma \subseteq 2^{\{p_1,\ldots,p_n\}}$ is monotone if $B \in \Gamma$ and $B \subseteq C$ imply that $C \in \Gamma$. An access structure $\Gamma \subseteq 2^{\{p_1,\ldots,p_n\}}$ is a monotone collection of non-empty sets. Sets in Γ are called authorized, and sets not in Γ are called unauthorized. We will also represent an n-party access structure by a function $f : \{0,1\}^n \to \{0,1\}$, where an input (i.e., a string) $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_n \in \{0,1\}^n$ represents the set $A_\sigma = \{p_i : i \in [n], \sigma_i = 1\}$, and $f(\sigma) = 1$ if and only if $A \in \Gamma$. We will also call f an access structure.*

A secret-sharing scheme defines a way to distribute shares to parties. Such a scheme is said to realize an access structure Γ if the shares held by any authorized set of parties (i.e., a set in the access structure) can be used to reconstruct the secret, and the shares held by any unauthorized set of parties reveal nothing about the secret. The formal definition is given as follows.

Definition 3.2 (Secret-sharing schemes)**.** *A secret-sharing scheme Π over a set of parties $P = \{p_1, \ldots, p_n\}$ with domain of secrets S and domain of random strings R is a mapping from $S \times R$*

to a set of n-tuples $S_1 \times S_2 \times \cdots \times S_n$ (the set S_j is called the domain of shares of p_j). A dealer distributes a secret $s \in S$ according to Π by first sampling a random string $r \in R$ with uniform distribution, computing a vector of shares $\Pi(s;r) = (\mathsf{sh}_1, \ldots, \mathsf{sh}_n)$, and privately communicating each share sh_j to party p_j. For a set $A \subseteq \{p_1, \ldots, p_n\}$, we denote $\Pi_A(s;r)$ as the restriction of $\Pi(s;r)$ to its A-entries (i.e., the shares of the parties in A).

A secret-sharing scheme Π with domain of secrets S realizes an access structure Γ if the following two requirements hold:

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

Security. *Every unauthorized set cannot learn anything about the secret from its shares. Formally, for any set $T \notin \Gamma$, every two secrets $s_1, s_2 \in S$, and for a uniformly distributed $r \leftarrow R$,*

$$\left(\Pi_T(s_1;r) = \langle \mathsf{sh}_j \rangle_{p_j \in T}\right) \equiv \left(\Pi_T(s_2;r) = \langle \mathsf{sh}_j \rangle_{p_j \in T}\right).$$

The size of the share of party p_j is defined as $\log |S_j|$ and the size of the shares of Π is defined as $\max_{1 \le j \le n} \log |S_j|$. The total share size of Π is defined as $\sum_{j=1}^{n} \log |S_j|$. The scheme is called linear 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 fields over $\mathbb{F}$, and the mapping Π is a linear mapping of the secret and the coordinates of the random string $r = (r_1, \ldots, r_\ell)$. Finally, for an access structure Γ, let $\mathsf{LSS}(\Gamma)$ denote the minimum total share size, where the minimum is taken over all linear secret-sharing schemes realizing Γ (over all finite fields).*

3.2.1 Multislice and Downslice Access Structures

We next introduce the multislic and downslice of an access structure. They have found use in constructing secret-sharing schemes for general access structures [52, 6, 7, 9]. We first define multislices. Roughly, the $(a:b)$-multislice of an access structure Γ agrees with Γ on all sets of size between a and b, all sets of size less than a are unauthorized, and all sets of size greater than b are authorized. Formally, it is defined as follows.

Definition 3.3 (Multislices)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be an n-party access structure and let $a \le b \in [n]$. The $(a:b)$-multislice of f is the access structure $F : \{0,1\}^n \to \{0,1\}$ defined as*

$$F(x) = \begin{cases} 0 & \text{if } \mathrm{wt}(x) < a \\ f(x) & \text{if } \mathrm{wt}(x) \in [a,b] \\ 1 & \text{if } \mathrm{wt}(x) > b \end{cases}.$$

An access structure is called an $(a:b,n)$-multislice access structure *if it is the $(a:b)$-multislice of some access structure over n parties. For $\alpha < \beta \in [0,1]$, we denote the linear exponent of $(\alpha n : \beta n, n)$-multislice access structure by*

$$\mathbf{M}_\ell(\alpha:\beta) = \limsup_{n \to \infty} \max_{F \in \mathcal{M}(\alpha:\beta,n)} \frac{1}{n} \log \mathsf{LSS}(F),$$

where $\mathcal{M}(\alpha:\beta,n)$ is the set of all $(\alpha n : \beta n, n)$-multislice access structures.

We next define a b-downslice of an access structure Γ. It is defined as the access structure that agrees with Γ on all sets of size b, and whose max-terms are all of size b. That is, a set is unauthorized if and only if it is contained in an unauthorized set of Γ of size b.

Definition 3.4 (Downslices)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be an n-party access structure and let $b \in [n]$. The b-downslice of f is the access structure $F : \{0,1\}^n \to \{0,1\}$ defined as*

$$F(x) = \begin{cases} 0 & \text{if } \exists x' : \mathrm{wt}(x') = b, x \leq x', f(x') = 0 \\ 1 & \text{otherwise} \end{cases}.$$

An access structure is called a (b,n)-downslice access structure *if it is the b-downslice of some access structure over n parties. For $\beta \in [0,1]$, we denote the linear exponent of $(\beta n, n)$-downslice access structure by*

$$\mathbf{D}_\ell(\beta) = \limsup_{n\to\infty} \max_{F \in \mathcal{D}(\beta,n)} \frac{1}{n} \log \mathsf{LSS}(F),$$

where $\mathcal{D}(\beta, n)$ is the set of all $(\beta n, n)$-downslice access structures.

3.3 Conditional Disclosure of Secrets

In a 2-server CDS protocol, there are 2 servers, denote Alice and Bob, holding the same secret s and a common random string. Additionally, Alice holds a private input x and Bob holds a private input y. In addition, there is a referee, which knows x and y but, prior to the execution of the protocol, the referee does not know the secret and the common random string. In a CDS protocol, each server sends a single message to the referee. The message of each server is a function of the secret, the common random string, and its private input; the message is independent of the input and the messages computed by the other server. The referee should learn the secret s iff $f(x,y) = 1$, for a fixed predicate f.

Definition 3.5 (Conditional disclosure of secrets (CDS) protocols)**.** *Let $f : X \times Y \to \{0,1\}$ be a function. A 2-server CDS protocol $\mathcal{P}$ for f with domain of secrets S is a pair of functions (A, B) with the following syntax. Alice and Bob hold private inputs $x \in$ Alice and $y \in Bc$ respectively. In addition, both Alice and Bob receive the same secret $s \in S$ and randomness r as common inputs. Alice and Bob compute the message $\mathsf{A}(x, s; r)$ and $\mathsf{B}(y, s; r)$ respectively.*

We say that a protocol $\mathcal{P}$ is a CDS protocol for f if it satisfies the following properties.

Correctness: *There exists a deterministic reconstruction algorithm C receiving the private inputs x and y of Alice and Bob, respectively, and their outputs, that is able to reconstruct s if $f(x,y) = 1$. That is, for every inputs $x \in X$ and $y \in Y$ for which $f(x,y) = 1$, every secret $s \in S$, and every common random string $r \in R$, it holds that $\mathsf{C}(x, y, \mathsf{A}(x, s; r), \mathsf{A}(y, s; r)) = s$.*

Security: *If $f(x,y) = 0$ then C does not learn any information about the secret. Formally, for every pair of secrets $s_1, s_2 \in S$, for all $x \in X$ and $y \in Y$ satisfying $f(x,y) = 0$, and for a uniform random string $r \leftarrow R$,*

$$(\mathsf{A}(x, s_1; r), \mathsf{B}(y, s_1; r)) \equiv (\mathsf{A}(x, s_2; r), \mathsf{B}(y, s_2; r)).$$

The message size of a CDS protocol $\mathcal{P}$ is defined as the size of the largest message sent by the servers, i.e., $\max_{x\in X, y\in Y, s\in S, r\in R}\{\mathsf{A}(x, s; r), \mathsf{B}(y, s; r)\}$.

Definition 3.6 (The predicate $\mathbf{INDEX_N}$)**.** *We define the function* $\mathrm{INDEX}_N : \{0,1\}^N \times [N] \to \{0,1\}$ *as follows. For every array* $D \in \{0,1\}^N$ *(called a database) and every index* $i \in [N]$*, we let* $\mathrm{INDEX}_N(D,i) = D_i$*.*

Observation 3.7. *If there is a CDS protocol for* INDEX_N *with message size* M*, then for every* $f : [N]^2 \to \{0,1\}$ *there is a CDS protocol with message size* M*.*

3.3.1 Robust Conditional Disclosure of Secrets

Observe that in the security definition of a CDS protocol, if a server sends several messages for different inputs with the same randomness, then it is not guaranteed that the referee does not learn anything about the secret. Toward constructing better secret-sharing schemes, Applebaum et al. [7] introduced the notion of robust CDS (RCDS) protocol, where the security is guaranteed to hold even if the servers send several messages for different inputs that do not satisfy the predicate. Before formally defining RCDS protocols, we first define the notion of a zero set, which is simply a subset of inputs that are mapped to 0.

Definition 3.8 (Zero sets.)**.** *Let* $f : X \times Y \to \{0,1\}$ *be a function. We say that a set* $Z \subseteq X \times Y$ *is a* zero set of f *if* $f(x,y) = 0$ *for all* $(x,y) \in Z$*.*

Definition 3.9 ($\mathcal{Z}$-robust CDS protocols)**.** *Let* $\mathcal{P}$ *be a CDS protocol for a function* $f : X \times Y \to \{0,1\}$*, and let* $Z = Z_1 \times Z_2 \subseteq X \times Y$ *be a zero set of* f*. Let* $\mathsf{A}(Z_1, s; r) = (A(x,s;r))_{x \in Z_1}$ *and* $\mathsf{B}(Z_2, s; r) = (\mathsf{B}(y,s;r))_{y \in Z_2}$*. We say that* $\mathcal{P}$ *is robust for the set* Z *if for every pair of secrets* $s_1, s_2 \in S$ *and for* $r \leftarrow R$ *it holds that*

$$(\mathsf{A}(Z_1, s_1; r), \mathsf{B}(Z_2, s_1; r)) \equiv (\mathsf{A}(Z_1, s_1; r), \mathsf{B}(Z_2, s_2; r)).$$

Let $\mathcal{Z}_1 \subseteq 2^X, \mathcal{Z}_2 \subseteq 2^Y$*. We say that* $\mathcal{P}$ *is* $(\mathcal{Z}_1, \mathcal{Z}_2)$*-robust if it is robust for every zero set* $Z = Z_1 \times Z_2$ *such that* $Z_1 \in \mathcal{Z}_1, Z_2 \in \mathcal{Z}_2$*. Furthermore, for an integer* t_1 *we say that* $\mathcal{P}$ *is* $(\mathcal{Z}_1, t_2)$*-robust if it is* $(\mathcal{Z}_1, \mathcal{Z}_2)$*-robust for* $\mathcal{Z}_2 = \{A \subseteq\in Y : |A| \leq t_2\}$*;* (t_1, t_2)*-robustness is defined similarly. Finally, for an integer* t*, we say that* $\mathcal{P}$ *is* t*-robust if it is* (t,t)*-robust.*

3.4 Private Information Retrieval (PIR)

A 2-server PIR protocol involves two servers, Alice and Bob and a user Charlie. Each server hold the same database $D \in \{0,1\}^N$, and the user who holds an index $i \in [N]$ wants to retrieve the bit D_i without revealing i. In a 2-server PIR protocol, Charlie sends each server a random query, each server Alice and Bob send Charlie back an answer which is a function of the query the server received and the database D. Given the answers from the servers Charlie reconstruct D_i.

Definition 3.10 (Private information retrieval (PIR) protocols)**.** *A* 2*-server PIR protocol* $\mathcal{P}$ *for a database* $D \in \{0,1\}^N$ *and an index* $i \in [N]$ *consists of two randomized query functions* $\mathcal{Q}_A, \mathcal{Q}_B$*, two answer functions* $\mathcal{A}_A, \mathcal{A}_B$ *and a reconstruction function* $\mathcal{C}$*. The user Charlie holds the index* i*, and sends a query to the servers Alice and Bob* $q_A \leftarrow \mathcal{Q}_A(i;r)$*,* $q_B \leftarrow \mathcal{Q}_B(i;r)$ *respectively. The servers Alice and Bob hold the database* D*, receive a query from Charlie and send each an answer* $a_A \leftarrow \mathcal{A}_A(q_A, D), a_B \leftarrow \mathcal{A}_B(q_B, D)$ *respectively. Finally, Charlie computes his output by applying the reconstruction function,* $\mathcal{C}(a_A, a_B, i)$*. We say that a protocol* $\mathcal{P} = (\mathcal{Q}_A, \mathcal{Q}_B, \mathcal{A}_A, \mathcal{A}_B, \mathcal{C})$ *is a* 2*-server PIR protocol if it satisfies the following properties.*

Correctness: *Charlie outputs the correct value. Formally, for every database* $D \in \{0,1\}^N$*, index* $i \in [N]$*, and a random string* $r \in R$*,*

$$\mathcal{C}(\mathcal{A}_A(\mathcal{Q}_A(i;r),D),\mathcal{A}_B(\mathcal{Q}_B(i;r),D),i) = D_i.$$

Security: *Each server learns no information about* i *from its query. Formally, for every pair of indexes* i_1, i_2*, and for a uniform random string* $r \leftarrow R$*, it holds that*

$$\mathcal{Q}_A(i_1;r) \equiv \mathcal{Q}_A(i_2;r) \text{ and } \mathcal{Q}_B(i_1;r) \equiv \mathcal{Q}_B(i_2;r).$$

The communication complexity of a 2-server PIR protocol is the largest message communicated between the servers and the user, i.e., $\max\{|q_A|,|q_B|,|a_A|,|a_B|\}$ *as defined above of over all* $D \in \{0,1\}^N, i \in [N], r \in R$*.*

3.5 Matching Vectors

We next define matching vectors (MV). Matching vectors are the key tool for the constructions of the best known non-linear CDS protocols in [53, 54, 14] and PIR protocol in [35, 34]. Here, we present a definition of matching vectors that generalizes the known definition from [53, 54, 35, 34]. In our definition, the matching vector family is defined by two disjoint sets S and T in which the inner products of the vectors lies, whereas in the former definitions the family only defined by one set S and the inner products lies in $S \cup \{0\}$.

Afterwards, we specify the type of matching vector which we are going to use in our CDS construction, and show the equivalence to the previous definitions, and henceforth deriving an efficient construction for our definition.

Definition 3.11 (Matching vectors)**.** *Let* $N, m, h > 0$ *be positive integers, and let* $S, T \subseteq \mathbb{Z}_m$ *be subsets such that* $S \cap T = \emptyset$*. The vectors family* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$*, where* $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{Z}_m^h$*, is called* (S,T)-matching vectors *if the following hold.*

1. $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_m \in T$ *for every* $i \in [N]$*.*
2. $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m \in S$ *for every* $i \neq j \in [N]$*.*

If $T = \{1\}$ *then we call the vector family* S*-matching vectors.*

Let $m = p_1 p_2$ where p_1 and p_2 are distinct prime numbers. In previous constructions of CDS and PIR protocols based on matching vector families [35, 34, 53, 14] $(S_{\text{can}}, 0)$-matching vectors and $(S_{\text{one}}, 0)$-matching vectors were used, where

$$S_{\text{can}} = \{a \in \mathbb{Z}_m : a \bmod p_1, a \bmod p_2 \in \{0,1\}\} \setminus \{0\}, \text{ and}$$
$$S_{\text{one}} = \{a \in \mathbb{Z}_m : a \equiv 1 \pmod{p_1} \vee a \equiv 1 \pmod{p_2}\}.$$

To simplify the construction of 2-server CDS and PIR protocols, we use a slightly different matching vectors,

$$S_{\text{zero}} = \mathbb{Z}_m \setminus \mathbb{Z}_m^* = \{a \in \mathbb{Z}_m : a \equiv 0 \pmod{p_1} \vee a \equiv 0 \pmod{p_2}\}.$$

We next prove the length of $(S_{\text{one}}, 0)$-matching vectors and S_{zero}-matching vectors are equivalent up to an addition of one entry.

Claim 3.12. *If there is an $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length h, then there is an S_{zero}-matching vector family over $\mathbb{Z}_m$ of length $h+1$. If there is an S_{zero}-matching vector family over $\mathbb{Z}_m$ of length h, then there is an $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length $h+1$.*

Proof. Given $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$, define $\mathbf{u}'_i = (1, -\mathbf{u}_i), \mathbf{v}'_i = (1, \mathbf{v}_1)$. Then, if $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ is an $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length h, then, $((\mathbf{u}'_i, \mathbf{v}'_i))_{i\in[N]}$ is an S_{zero}-matching vector family over $\mathbb{Z}_m$ of length $h+1$, and if $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ is an S_{zero}-matching vector family over $\mathbb{Z}_m$ of length h, then $((\mathbf{u}'_i, \mathbf{v}'_i))_{i\in[N]}$ is an $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length $h+1$. This follows since, for every i, j, $\langle \mathbf{u}'_i, \mathbf{v}'_j \rangle = 1 - \langle \mathbf{u}_i, \mathbf{v}_j \rangle$ therefore

$$\begin{aligned}
&\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m = 0 \Rightarrow \langle \mathbf{u}'_i, \mathbf{v}'_j \rangle_m = 1,\\
&\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m = 1 \Rightarrow \langle \mathbf{u}'_i, \mathbf{v}'_j \rangle_m = 0,\\
&\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m \in S_{\text{one}} \Rightarrow \langle \mathbf{u}'_i, \mathbf{v}'_j \rangle_m \in S_{\text{zero}}, \text{ and}\\
&\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m \in S_{\text{zero}} \Rightarrow \langle \mathbf{u}'_i, \mathbf{v}'_j \rangle_m \in S_{\text{one}}.
\end{aligned}$$

□

Since $S_{\text{can}} \subset S_{\text{one}}$, a family of $(S_{\text{can}}, 0)$-matching vectors is a family of $(S_{\text{one}}, 0)$-matching vectors; Claim 3.12 and the known results for S_{can} matching vectors [41, 51] imply the following corollary

Corollary 3.13. *For every distinct primes p_1, p_2, there is an S_{zero} matching vector family over $\mathbb{Z}_m$ for $m = p_1 \cdot p_2$, of size N and length $2^{O(\sqrt{\log N \log\log N})}$.*

As we have seen, $(S_{\text{can}}, 1)$-matching vectors imply S_{zero}-matching vectors of the same length. We do not know whether such implication exists in the other direction or there is a better construction for S_{zero}-matching vectors, which would improve the communication complexity of 2-server PIRa and CDS protocols.

4 Simplified PIR and CDS Protocols using Share Conversions

In this section we will show a 2-server PIR, 2-server CDS and multi-server CDS protocols for INDEX_N; these protocol are simplifications and generalizations of the protocols of Dvir and Gopi [33], Liu et al. [54], and Beimel et al. [14]. Our protocol modifies the previous protocols in two ways. First, we use an S_{zero}-matching vector family over $\mathbb{Z}_m$ for a general composite $m = p_1 \cdot p_2$, where [53] uses $(S_{\text{can}}, 0)$-matching vector family over $\mathbb{Z}_6$ and [14] uses $(S_{\text{one}}, 0)$-matching vector family for $m = p_1 \cdot p_2$ s.t. $p_1 | p_2 - 1$. The second generalization is the use of a general share conversion as defined below in Section 4.1 and constructed in Section 4.5.

4.1 Share Conversion

We start by giving the definition of share conversion.

Definition 4.1 (Share conversion)**.** *Let p_1 and p_2 be two prime numbers. We say that $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ is a share conversion if for all $s_1, s_2 \in \mathbb{F}_{p_2}$ such that $s_1 - s_2 \equiv 1 \pmod{p_1}$ then $C(s_1) - C(s_2) \not\equiv 0 \pmod{p_2}$.*

The definition is a special case of share conversion for 2-servers as defined in [19].

In our PIR protocol the queries to Alice and Bob are $\mathbf{u}_i + \mathbf{r}$ and $\mathbf{r}$ respectively; these queries are used to compute to following expression

$$C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}).$$

We observe the following about that expression which was the motivation for the definition of share conversion.

Observation 4.2. *Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ be an S_{zero}-matching vector family over $\mathbb{Z}_m^h$ for $m = p_1 p_2$, and let $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ be a share conversion. Then, for every $i, j \in [N]$,*

- *If $i \neq j$ and $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_1} = 0$ then*

$$C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \equiv 0 \pmod{p_2}.$$

- *If $i = j$, then*

$$C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_i \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_i \rangle_{p_1}) \not\equiv 0 \pmod{p_2}.$$

Proof. Let $i, j \in [N]$.

- If $i \neq j$ and $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_1} = 0$ then

$$\begin{aligned} C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) &\equiv C(\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} + \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \\ &\equiv C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \\ &\equiv 0 \pmod{p_2}. \end{aligned}$$

- If $i = j$, then $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_m = 1$, in particular $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_1} = 1$, thus

$$\begin{aligned} C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) &\equiv C(\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} + \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \\ &\equiv C(1 + \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \\ &\not\equiv 0 \pmod{p_2}. \end{aligned}$$

□

4.2 The 2-Server PIR Protocol

In this section we present the simplified 2-server PIR protocol using share conversion. A concrete implementation given in Protocol 2.1.

Protocol 4.3.
Public parameters: *An S_{zero}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ for $m = p_1 p_2$.*
Alice and Bob's input: $D \in \{0,1\}^N$.
Charlie's input: $i \in [N]$.
Charlie's randomness: $\mathbf{r} \in \mathbb{F}_{p_1}^h$.
Notations: *Let $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ be a share conversion, and let $V : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}^h$ where $V(\mathbf{w}) = \sum_{j=1}^N C(\langle \mathbf{w}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \mathbf{v}_j \bmod p_2$..*

- *Charlie sends queries $\mathbf{q}_A \leftarrow \mathbf{u}_i + \mathbf{r} \bmod p_1$ and $\mathbf{q}_B \leftarrow \mathbf{r}$ to Alice and Bob respectively.*

- *Alice and Bob send answers* $\mathbf{a}_A \leftarrow V(\mathbf{q}_A)$ *and* $\mathbf{a}_B \leftarrow V(\mathbf{q}_B)$ *respectively to Charlie.*
- *Charlie outputs 1 if*

$$\langle \mathbf{u}_i, \mathbf{a}_A - \mathbf{a}_B \rangle_{p_2} \neq 0, \tag{6}$$

and 0 otherwise.

We next show two lemmas about the inner products and share conversion that are used in the proof of correctness and security of our PIR and CDS protocols.

Lemma 4.4. *Let* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ *be an* S_{zero}*-matching vector family over* $\mathbb{Z}_m^h$ *for* $m = p_1 p_2$*, and let* $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ *be a share conversion. Then for every* $\mathbf{r} \in \mathbb{F}_{p_1}^h$*, and* $i, j \in [N]$

$$(C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1})) \cdot \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} \equiv 0 \pmod{p_2}$$

if and only if $i \neq j$*.*

Proof. Let $i, j \in [N]$, and $\mathbf{r} \in \mathbb{F}_{p_1}^h$. Observe that

- If $i \neq j$, then $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m \in S_{\text{zero}}$. If $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} = 0$, the claim holds. Otherwise, $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} = 0$, thus from Observation 4.2 $C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \equiv 0 \pmod{p_2}$ and the claim holds.
- If $i = j$, then, $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_m = 1$, therefore, $\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} = 1$ and $\langle \mathbf{u}_i, \mathbf{v}_i \rangle_{p_2} = 1$, thus, from Observation 4.2

$$C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \not\equiv 0 \pmod{p_2}$$

and the claim holds.

□

Lemma 4.5. *Let* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ *be an* S_{zero}*-matching vector family over* $\mathbb{Z}_m^h$ *for* $m = p_1 p_2$*,* $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ *be a conversion and let*

$$V(\mathbf{w}) = \sum_{j=1}^{N} \mathbf{v}_j C(\langle \mathbf{w}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \mod p_2.$$

Then, for every $i \in [N]$*, and* $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$

$$\langle \mathbf{u}_i, V(\mathbf{u}_i + \mathbf{r}) - V(\mathbf{r}) \rangle_{p_2} = D_i \cdot \alpha,$$

for some $\alpha \not\equiv 0 \pmod{p_2}$*.*

Proof. Observe that

$$\begin{aligned}
\langle \mathbf{u}_i, V(\mathbf{u}_i + \mathbf{r}) - V(\mathbf{r}) \rangle_{p_2} &\equiv \langle \mathbf{u}_i, \sum_{j=1}^{N} \mathbf{v}_j C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \\
&\quad - \sum_{j=1}^{N} \mathbf{v}_j C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \rangle_{p_2} \\
&\equiv \langle \mathbf{u}_i, \sum_{j=1}^{N} \mathbf{v}_j \left(C(\langle \mathbf{u}_i + \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \right) \cdot D_j \rangle_{p_2} \\
&\equiv \sum_{j=1}^{N} \left(C(\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_1} + \langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \right. \\
&\quad \left. - C(\langle \mathbf{r}, \mathbf{v}_j \rangle_{p_1}) \right) \cdot \langle \mathbf{u}_i, \mathbf{v}_j \rangle_{p_2} \cdot D_j \pmod{p_2}.
\end{aligned}$$

Then, from Lemma 4.4, the coefficient of D_j in the sum modulo p_2 is 0 if and only if $i \neq j$, thus we get that $\langle \mathbf{u}_i, V(\mathbf{u}_i + \mathbf{r}) - V(\mathbf{r}) \rangle_{p_2} = D_i \cdot \alpha$ where $\alpha \not\equiv 0 \pmod{p_2}$. □

Finally, we can prove that Protocol 4.3 is a 2-server PIR protocol.

Theorem 4.6. *Let p_1, p_2 two distinct primes and let $m = p_1 p_2$. Protocol 4.3 is a 2-server PIR protocol over $\mathbb{F}_{p_2}$ with message size h, where h is the length of the matching vectors used in the protocol.*

Proof. In order to prove the theorem, we need to show that Protocol 4.3 satisfies correctness and security.

Correctness. The correctness follows directly from Lemma 4.5, since Charlie returns 1 iff $\langle \mathbf{u}_i, \mathbf{a}_A - \mathbf{a}_B \rangle_{p_2} \not\equiv 0 \pmod{p_2}$ and

$$\begin{aligned} \langle \mathbf{u}_i, \mathbf{a}_A - \mathbf{a}_B \rangle_{p_2} &= \langle \mathbf{u}_i, V(\mathbf{q}_A) - V(\mathbf{q}_B) \rangle_{p_2} \\ &= \langle \mathbf{u}_i, V(\mathbf{u}_i + \mathbf{r}) - V(\mathbf{r}) \rangle_{p_2} \\ &= D_i \cdot \alpha \end{aligned}$$

for $\alpha \not\equiv 0 \pmod{p_2}$ and Charlie outputs the correct D_i.

Security. For a uniformly distributed random $\mathbf{r}$, the query $q_B \leftarrow \mathbf{r}$ is uniformly distributed, and so is the query $q_A \leftarrow \mathbf{u}_i + \mathbf{r}$ since it is masked with $\mathbf{r}$. Thus, each query is are identically distributed for every index i. □

4.3 The 2-Server CDS Protocol

In Protocol 4.7, we next present our simplified 2-server CDS protocol using share conversion. A concrete implementation given in Protocol 2.2. Protocol 4.7 is obtained by exchanging the roles of the parties compared to the PIR protocol (Protocol 4.3). That is, the randomness of the user is now that shared random string of the servers; Alice computes the same message as her answer in the PIR protocol and Bob computes the query q_B and send it to Charlie, which computes the answer of Bob in the PIR protocol (Charlie holds the database, while Bob does not hold it). Furthermore, some masking is added to prevent Charlie from learning information on the common random string.

Protocol 4.7.
Public parameters: *An S_{zero}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ for $m = p_1 p_2$.*
Alice's input: $D \in \{0,1\}^N$.
Bob's input: $i \in [N]$.
The secret: $s \in \{0,1\}$.
Shared randomness: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$, $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h$.
Notations: *Let $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ be a share conversion, and let $V : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}^h$ where $V(\mathbf{w}) = \sum_{j=1}^N C(\langle \mathbf{w}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \mathbf{v}_j \bmod p_2$..*

- *Alice sends $\mathbf{m}_A \leftarrow V(\mathbf{r}_1) + \mathbf{r}_2 \bmod p_2$.*
- *Bob sends $\mathbf{m}_B^1 \leftarrow (s\mathbf{u}_i + \mathbf{r}_1 \bmod p_1)$, $m_B^2 \leftarrow \langle \mathbf{u}_i, \mathbf{r}_2 \rangle_{p_2}$.*

- *Charlie outputs 1 if*
$$\langle \mathbf{u}_i, V(\mathbf{m}_B^1) - \mathbf{m}_A \rangle_{p_2} + m_B^2 \not\equiv 0 \pmod{p_2}, \tag{7}$$
and 0 otherwise.

Theorem 4.8. *Let p_1, p_2 be two distinct primes and let $m = p_1 p_2$. Protocol 4.7 is a 2-server CDS protocol over $\mathbb{F}_{p_2}$ for INDEX_N^2 with message size $h \cdot \log m$, where h is the length of the matching vectors used in the protocol.*

Proof. Before proving the correctness and security we will show that for some $\alpha \not\equiv 0 \pmod{p_2}$
$$\langle \mathbf{u}_i, V(\mathbf{m}_B^1) - \mathbf{m}_A \rangle_{p_2} + m_B^2 = s \cdot D_i \cdot \alpha. \tag{8}$$
This follows from the following analysis.
$$\begin{aligned} \langle \mathbf{u}_i, V(\mathbf{m}_B^1) - \mathbf{m}_A \rangle_{p_2} + m_B^2 &= \langle \mathbf{u}_i, V(s\mathbf{u}_i + \mathbf{r}_1) - V(\mathbf{r}_1) - \mathbf{r}_2 \rangle_{p_2} + \langle \mathbf{u}_i, \mathbf{r}_2 \rangle_{p_2} \\ &= \langle \mathbf{u}_i, V(s\mathbf{u}_i + \mathbf{r}_1) - V(\mathbf{r}_1) \rangle_{p_2}. \end{aligned}$$

- If $s = 0$, then Charlie compute $\langle \mathbf{u}_i, V(\mathbf{r}_1) - V(\mathbf{r}_1) \rangle_{p_2} = 0$.
- Otherwise, if $s = 1$, then from Lemma 4.5, Charlie computes
$$\langle \mathbf{u}_i, V(\mathbf{u}_i + \mathbf{r}_1) - V(\mathbf{r}_1) \rangle_{p_2} = D_i \cdot \alpha$$
where $\alpha \not\equiv 0 \pmod{p_2}$.

Charlie computes $s \cdot D_i \cdot \alpha \bmod p_2$ for $\alpha \not\equiv 0 \pmod{p_2}$ and return 1 iff it is not equal to 0.

Now, we can continue to prove the correctness and security.

Correctness. Correctness should hold when $D_i = 1$. In this case by (8), Charlie outputs 1 if $s \cdot \alpha \bmod p_2$ is not equal to 0 for some $\alpha \not\equiv 0 \pmod{p_2}$, thus Charlie returns s.

Security. We prove that when $D_i = 0$, Charlie learns no information on the secret.

- The joint distribution of $\mathbf{m}_A, \mathbf{m}_B^1$ is uniformly distributed since they are masked by $\mathbf{r}_2, \mathbf{r}_1$ respectively.
- By (8),
$$\langle \mathbf{u}_i, V(\mathbf{m}_B^1) - \mathbf{m}_A \rangle_{p_2} - m_B^2 = s \cdot D_i \cdot \alpha = 0.$$
Therefore, m_B^2 is independent of s and can be computed from $\mathbf{m}_A, \mathbf{m}_B^2, i, D$.

From the observations, when $D_i = 0$, Charlie can simulate $\mathbf{m}_A, \mathbf{m}_B^1, m_B^2$ given i, D. □

4.4 The Multi-Server CDS Protocol

In this section we generalize the 2-server CDS protocol in Protocol 4.7 to a $(k+1)$-server CDS protocol (for $k \geq 2$), similarly to the generalization done by Liu et al. in [54] using decomposable matching vectors. We start with the definitions of point-wise product of vectors and decomposable matching vectors.

Definition 4.9 (Pointwise product)**.** *Let $m, h > 0$ be two positive integers and let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_m^h$. The point-wise product (or Hadamard product) of $\mathbf{x}, \mathbf{y}$, denoted by $\mathbf{x} \odot \mathbf{y}$, is a vector in $\mathbb{Z}_m^h$ whose ℓ-th element is the product of the ℓ-th elements of $\mathbf{x}, \mathbf{y}$, i.e. $(\mathbf{x} \odot \mathbf{y})[\ell] = \mathbf{x}[\ell] \cdot \mathbf{y}[\ell] \mod m$.*

Definition 4.10 (k-decomposability [54])**.** *Let $N' = \sqrt[k]{N}$. A family of vectors $(\mathbf{u}_i)_{i=1}^N$ over $\mathbb{Z}_m^h$ is k-decomposable if there exist vector families $(\mathbf{u}_{1,i})_{i=1}^{N'}, \ldots, (\mathbf{u}_{k,i})_{i=1}^{N'}$ over $\mathbb{Z}_m^h$ such that under the natural mapping $i \mapsto (i_1, \ldots, i_k) \in [N']^k$*

$$\mathbf{u}_i = \mathbf{u}_{1,i_1} \odot \cdots \odot \mathbf{u}_{k,i_k} \mod m$$

for all $i \in [N]$. That is, $\mathbf{u}_i$ is the pointwise product of k vectors $\mathbf{u}_{1,i_1}, \ldots, \mathbf{u}_{k,i_k}$, where each $\mathbf{u}_{j,i_j}$ can be computed from i_j.

Definition 4.11 (Decomposable matching vector families)**.** *For integers $N, m, h, k > 0$ and $S, T \subseteq \mathbb{Z}_m$ such that $S \cap T = \emptyset$, a collection of vectors $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ is a k-decomposable (S,T)-matching vector family if it is an (S,T)-matching vector family, and $(\mathbf{u}_i)_{i=1}^N$, $(\mathbf{v}_i)_{i=1}^N$ are k-decomposable.*

Next, we present a claim similar to Claim 3.12 showing the equivalence between the length of k-decomposable $(S_{\text{one}}, 0)$-matching vectors to the length k-decomposable S_{zero}-matching vectors.

Claim 4.12. *If there is a k-decomposable $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length h, then there is a k-decomposable S_{zero}-matching vector family over $\mathbb{Z}_m$ of length $h+1$. If there is an k-decomposable S_{zero}-matching vector family over $\mathbb{Z}_m$ of length h, then there is a k-decomposable $(S_{\text{one}}, 0)$-matching vector family over $\mathbb{Z}_m$ of length $h+1$.*

Proof. Given $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ define $\mathbf{u}'_i = (1, -\mathbf{u}_i), \mathbf{v}'_i = (1, \mathbf{v}_i)$. We have seen in the proof of Claim 3.12 that if $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ is an $(S_{\text{one}}, 0)$-matching vector family then $((\mathbf{u}'_i, \mathbf{v}'_i))_{i\in[N]}$ is an S_{zero}-matching vector family, and if $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ is an S_{zero}-matching vector family then $((\mathbf{u}'_i, \mathbf{v}'_i))_{i\in[N]}$ is an $(S_{\text{one}}, 0)$-matching vector family. It is left to show that if $((\mathbf{u}_i, \mathbf{v}_i))_{i\in[N]}$ is k-decomposable matching vector family then $((\mathbf{u}'_i, \mathbf{v}'_i))_{i\in[N]}$ is a k-decomposable matching vector family. Let $(\mathbf{u}_{1,i})_{i=1}^{N'}, \ldots, (\mathbf{u}_{k,i})_{i=1}^{N'}$ be the decomposition of $(\mathbf{u}_i)_{i\in[N]}$. For every $i_1 \in [N^{1/k}]$, define $\mathbf{u}'_{1,i_1} = (1, -\mathbf{u}_{1,i_1})$, and for every $2 \leq t \leq k$, $i_t \in [N^{1/k}]$, define $\mathbf{u}'_{t,i_t} = (1, \mathbf{u}_{t,i_i})$, then $(\mathbf{u}'_{1,i})_{i=1}^{N'}, \ldots, (\mathbf{u}'_{k,i})_{i=1}^{N'}$ is a decomposition of $(\mathbf{u}'_i)_{i\in[N]}$. Since, for every $i \in [N]$

$$\mathbf{u}'_i[1] = 1 = \Pi_{t=1}^k \mathbf{u}'_{t,i_t}[1]$$

and for every $2 \leq \ell \leq h+1$,

$$\mathbf{u}'_i[\ell] = -\mathbf{u}_i[\ell-1] = -\Pi_{t=1}^k \mathbf{u}_{t,i_t}[\ell-1] = \Pi_{t=1}^k \mathbf{u}'_{t,i_t}[\ell].$$

□

Since $S_{\text{can}} \subset S_{\text{one}}$, a k-decomposable family of $(S_{\text{can}}, 0)$-matching vectors is a k-decomposable family of $(S_{\text{one}}, 0)$-matching vectors. Since Liu et al. [54] showed that the known results for S_{can} matching vectors [41] are k-decomposable, combining those results with Claim 4.12 imply the following corollary.

Corollary 4.13. *For every distinct primes p_1, p_2, there is a k-decomposable S_{zero}-matching vector family over $\mathbb{Z}_m$ for $m = p_1 \cdot p_2$, of size N and length $2^{O(\sqrt{\log N} \log\log N)}$.*

The multi-server CDS protocol that we present is a generalization of Protocol 4.7 to $k+1$ servers. In the protocol, one server, we call Alice holds the database $D \in \{0,1\}^{N^k}$ and sends the exact same message as in Protocol 4.7. The other k server will jointly hold an index $i \in [N^k]$ i.e. the servers $P_1, \ldots, P_k$ will hold $i_1, \ldots, i_k \in [\sqrt{N}]$ respectively. The k parties will simulate Bob from Protocol 4.7 using a PSM protocol whose functionality is Bob's messages in the 2-server CDS protocol (for a formal definition of PSM see [43]).

In a PSM protocol there are k parties each holding an input x_i and a common randomness. There is also a referee we call Charlie who wants to learn $F(x_1, \ldots, x_k)$ for a fixed functionality F. Each server sends a message to Charlie based on their private input and the common randomness. Given these messages Charlie should be able to learn $F(x)$ without learning anything else about $x_1, \ldots, x_k$.

In the $(k+1)$-server CDS protocol each server $P_1, \ldots, P_k$ will send a message to Charlie according to the PSM, and Charlie would be able to reconstruct the messages of Bob from Protocol 4.7 as if he was holding the full index i, without learning any information about the secret. That special purpose PSM which is presented by Liu et al. [54] uses the k-decomposablity of the matching vectors.

Next we generalize the special purpose PSM protocol from [54] to any two distinct primes p_1, p_2.

Theorem 4.14 ([54])**.** *For integers N, h, $k \leq \log N$ and $m = p_1 \cdot p_2$ for distinct primes p_1, p_2, if $(\mathbf{u}_i)_{i=1}^N$ is k-decomposable, then there is a PSM for the functionality*

$$F_{\mathsf{aux}} : [\sqrt[k]{N}] \times \cdots \times [\sqrt[k]{N}] \times \{0,1\} \times \mathbb{F}_{p_1} \times \mathbb{F}_{p_2} \to \mathbb{F}_{p_1}^h \times \mathbb{F}_{p_2}$$

$$\textit{where } F_{\mathsf{aux}}(i_1, \ldots, i_k; s, \mathbf{r}_1, \mathbf{r}_2) \mapsto (s\mathbf{u}_i + \mathbf{r}_1, \langle \mathbf{u}_i, \mathbf{r}_2 \rangle_{p_2})$$

with communication complexity $O(h \cdot k^2 \log m)$.

Finally, we present our $(k+1)$-server CDS protocol.

Protocol 4.15.
Public parameters: *A decomposable S_{zero}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ for $m = p_1 p_2$ and a PSM protocol $(\mathsf{PSM.B}_1, \ldots, \mathsf{PSM.B}_k, \mathsf{PSM.C})$ for F_{aux}.*
Input of P_t $(1 \leq t \leq k)$: $i_t \in [N]$.
Input of P_{k+1} (Alice): $D \in \{0,1\}^{N^k}$.
The secret: $s \in \{0,1\}$.
Shared randomness: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$, $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h$ *and randomness* $\mathbf{r}_{\mathsf{PSM}}$ *of the* PSM.
Notations: *Let $C : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$ be a share conversion, and let $V : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}^h$ where $V(\mathbf{w}) = \sum_{j=1}^N C(\langle \mathbf{w}, \mathbf{v}_j \rangle_{p_1}) \cdot D_j \mathbf{v}_j \bmod p_2$.*

- *For $1 \leq t \leq k$, the t-th party sends $\mathbf{m}_{\mathsf{PSM},t} \leftarrow \mathsf{PSM.B}_t(i_t, s, \mathbf{r}_1, \mathbf{r}_2; \mathbf{r}_{\mathsf{PSM}})$*
- *Alice sends $\mathbf{m}_A \leftarrow V(\mathbf{r}_1) + \mathbf{r}_2 \bmod p_2$.*

- *Charlie computes* $(\mathbf{m}_B^1, m_B^2) \leftarrow \mathsf{PSM.C}(\mathbf{m}_{\mathsf{PSM},1}, \ldots, \mathbf{m}_{\mathsf{PSM},k})$.
- *Charlie outputs 1 if*

$$\langle \mathbf{u}_i, V(\mathbf{m}_B^1) - \mathbf{m}_A \rangle_{p_2} + m_B^2 \not\equiv 0 \pmod{p_2}, \tag{9}$$

and 0 otherwise.

Theorem 4.16. *Let p_1, p_2 be two distinct primes and let $m = p_1 p_2$. Protocol 4.15 is a $(k+1)$-server CDS protocol over $\mathbb{F}_{p_2}$ for INDEX_N^{k+1} with message size $h \cdot k^2 \log m$, where h is the length of the matching vectors used in the protocol.*

Proof. The correctness and security follow from the correctness and security of Protocol 4.7 and of the PSM protocol for F_{aux} in Theorem 4.14. From the correctness of the PSM for F_{aux}, given $\mathbf{m}_{\mathsf{PSM},1}, \ldots, \mathbf{m}_{\mathsf{PSM},k}$, Charlie computes correctly $\mathbf{m}_B^1 \leftarrow s\mathbf{u}_i + \mathbf{r}_1, m_B^2 \leftarrow \langle \mathbf{u}_i, \mathbf{r}_2 \rangle_{p_2}$. Thus, from the proof of Theorem 4.8 the correctness follows. The security follows from the security of Protocol 4.7; the joint distribution $\mathbf{m}_A, \mathbf{m}_B^1, \mathbf{m}_B^2$ is equally distributed for $s = 0$ and $s = 1$. Finally, from the security of the PSM protocol, the joint distribution of $\mathbf{m}_{\mathsf{PSM},1}, \ldots, \mathbf{m}_{\mathsf{PSM},k}$ can be simulated from $(\mathbf{m}_B^1, m_B^2)$.

Communication. Alice sends a vector over $\mathbb{F}_{p_2}^h$ (of length $h \log p_2$), while all the other k servers send a PSM message, thus from Theorem 4.14 their message length is $h \cdot k^2 \log m$. Thus the message size of the protocol is $h \cdot k^2 \log m$. □

4.5 Constructing Share Conversions

In this section, we show simple constructions of share conversion as used in Protocol 4.3 and Protocol 4.7 – the PIR and CDS protocols.

4.5.1 The First Share Conversion

The first share conversion is used in Section 2; it assumes that $p_2 \nmid p_1 - 1$, in particular it can be applied when $p_1 < p_2$.

Claim 4.17. *Let p_1, p_2 be primes such such that $p_2 \nmid p_1 - 1$, and let $C_1 : \mathbb{F}_{p_1} \to \mathbb{F}_{p_2}$, where for every $x \in \mathbb{F}_{p_1}$, $C_1(x) = x \bmod p_2$. Then, C_1 is a share conversion.*

Proof. We need that for every $x \in \mathbb{F}_{p_1}$, $C_1(x + 1 \bmod p_1) - C_1(x) \not\equiv 0 \pmod{p_2}$. Let $x \in \mathbb{F}_{p_1}$,

- If $x \not\equiv p_1 - 1 \pmod{p_1}$, then

$$C_1(x + 1 \bmod p_1) - C_1(x) \equiv C_1(x+1) - C_1(x) \equiv x + 1 - x \equiv 1 \not\equiv 0 \pmod{p_2}.$$

- Otherwise, if $x = p_1 - 1$, then, since $p_2 \nmid p_1 - 1$

$$\begin{aligned} C_1(p_1 - 1 + 1 \bmod p_1) - C_1(p_1 - 1) &\equiv C_1(0) - C_1(p_1 - 1) \\ &\equiv 0 - (p_1 - 1) \equiv 1 - p_1 \not\equiv 0 \pmod{p_2}. \end{aligned}$$

□

In particular, if $p_1 < p_2$ then $C_1(x) = x$ is a share conversion. We note that when $p_2 | p_1 - 1$, $C_1(x) = x \bmod p_2$ is not a share conversion since

$$C_1(p_1 - 1 + 1) - C_1(p_1 - 1) \equiv 0 - p_1 + 1 \equiv 0 \pmod{p_2}.$$

4.5.2 The Second Share Conversion

The second share conversion assumes that $p_1|p_2-1$. This is the share conversion used in [14]. The share conversion uses an element $\gamma \in \mathbb{F}_{p_2}^*$ whose order is p_1, i.e., p_1 is the smallest positive integer s.t. $\gamma^{p_1} \equiv 1 \pmod{p_1}$. Such element exists if and only if $p_1|p_2-1$.

Claim 4.18. *Let p_1, p_2 primes s.t. $p_1|p_2-1$, and let $\gamma \in \mathbb{F}_{p_2}^*$ be an element of order p_1. Then the following function, $C_2(x) = \gamma^x \bmod p_2$ is a share conversion.*

Proof. Since the order of γ is p_1, C_2 is indeed a mapping from $\mathbb{F}_{p_1}$ to $\mathbb{F}_{p_2}$ For $x \in \mathbb{F}_{p_1}$,

$$C_2(x+1) - C_2(x) \equiv \gamma^{x+1} - \gamma^x \equiv \gamma^x(\gamma - 1) \pmod{p_2}.$$

Since $\gamma \in \mathbb{F}_{p_2}^*$, we know that $\gamma \neq 0$, also since its order is $p_1 > 1$, we know that $\gamma \neq 1$, therefore $C_2(x+1) - C_2(x) \not\equiv 0 \pmod{p_2}$. □

4.5.3 The Third Share Conversion

The third share conversion only assumes that $p_2 \neq 2$. The size of the range of this share conversion is 3.

Claim 4.19. *Let p_1, p_2 primes, such that $p_2 > 2$ and define C_3 as follows:*

$$C_3(x) = \begin{cases} x \bmod 2 & 0 \leq x < p_1 - 1 \\ 2 & x = p_1 - 1 \end{cases}.$$

Then, C_3 is a share conversion.

Proof. Let $x \in \mathbb{F}_{p_1}$,

- $0 \leq x < p_1 - 1$, then $C_3(x+1) - C_3(x) \in \{-1, 1\}$, i.e., the difference is non-zero.
- Otherwise, if $x = p_1 - 1$, then, since $p_2 > 2$,

$$C_3(x+1 \bmod p_1) - C_3(x) \equiv C_3(0) - C_3(p_1 - 1) \equiv 0 - 2 \not\equiv 0 \pmod{p_2}.$$

□

5 Improved Linear Secret-Sharing Schemes for Arbitrary Access Structures

In this section, we prove new upper bounds on the share size of linear secret-sharing schemes for all access structures.

Theorem 5.1. *Any n-party access structure can be realized with a linear secret-sharing scheme by share size $2^{0.7563n+o(n)}$.*

To simplify notations, if a scheme has share size $2^{S \cdot n + o(n)}$, we will say that it has exponent S. As observed in [9], it is enough to prove the bound for all downslices. Recall that a downslice is an access structure where all maximal authorized sets are of the same size. Applebaum and Nir [9] constructed a scheme for a downslice by a bootstrapping algorithm, taking a linear secret-sharing scheme for a downslice and constructing a better scheme. Our improvement is done by constructing a better scheme for downslices than [9] before applying bootstrapping. The next result states the share size for our basis scheme. In Section 5.3 we apply the bootstrapping, which proves Theorem 5.1.

Lemma 5.2. *Let $\alpha_0 > 0.5412$ denote the unique solution to the equation $(2x-3)^2\log(x)+2x^2-8x+7=0$ in the interval $[0,1]$. Then for all $\beta \in (0,1)$ it holds that*

$$\mathbf{D}_\ell(\beta) \leq \begin{cases} \frac{1}{2}+\frac{\beta}{2} & \text{if } \beta \leq 1/2 \\ \frac{2-\beta}{3-2\beta} & \text{if } 1/2 < \beta \leq \alpha_0 \\ h(\beta) - 0.51079 \cdot (1-\beta) & \text{otherwise} \end{cases}.$$

For example, for $\beta = 0.54$ we get $\mathbf{D}_\ell(0.54) \leq 146/192 \approx 0.7604$. On the other hand, before applying the bootstrapping, Applebaum and Nir [9] proved the upper bound of $\mathbf{D}_\ell(0.54) \leq h(0.54) - 0.5 \cdot (1-0.54) \approx 0.7654$.

The bound for $\beta \leq 1/2$ was shown in [9]. The proof for the case where $1/2 < \beta \leq \alpha_0$ is given in Section 5.1, and the proof for the case where $\alpha_0 < \beta \leq 1$ is given in Section 5.2.

5.1 A Better Linear Secret-Sharing Scheme for Downslices With Low Density

We next show an improved linear secret-sharing scheme for all downslices with low density. That is, we prove the following.

Lemma 5.3 (Low density)**.** *Every (b,n)-downslice access structure can be realized by a linear secret-sharing scheme with share size of at most $2^{\frac{n(2n-b)}{3n-2b}+o(n)}$. Consequently, $\mathbf{D}_\ell(\beta) \leq \frac{2-\beta}{3-2\beta}$ for any $\beta \in [0,1]$.*

Towards constructing such a scheme, we first construct a new linear secret-sharing scheme for all multislices. For this, we use the following 2-server linear CDS for an arbitrary function; this CDS protocol is already fully robust for Bob.

Theorem 5.4 ([38, 7])**.** *Let $f : [N_1] \times [N_2] \to \{0,1\}$. Then there exists a linear $(1,N_2)$-robust CDS protocol for f, where the message size of Alice is $N_1 - 1$, and the message of Bob is a single bit.*

By the immunization theorem of [7] we obtain the following.

Corollary 5.5. *Let $f : [N_1] \times [N_2] \to \{0,1\}$ and let $t \leq N_1$ and $\mathcal{Z}_1 \subseteq 2^{[N_1]}$ such that $|Z| \leq t$ for every $Z \in \mathcal{Z}_1$ and $\mathcal{Z}$ contains at most u maximal sets. Then there exists a linear $(\mathcal{Z}_1, N_2)$-robust CDS protocol for f, where the message size of Alice is $O(\frac{N_1}{t}\log^3 t \log N_1 \log u)$ and the message size of Bob is $O(t \log^3 t \log N_2 \log u)$.*

Our construction of a linear secret-sharing scheme for multislice access structures follows a similar approach to previous constructions [7, 9]. That is, we first reduce the construction to realizing somewhat regular access structures. These are partial access structures where we partition the parties into two sets and every authorized and every unauthorized set are partitioned "nicely" between the two sets. Unlike previous constructions, the partition we consider does not split each set into roughly equal sizes. This uneven partition is the main source for the improved share size in our construction. We next formally define somewhat regular access structures.

Definition 5.6 ((I,a,b)-somewhat regular access structures)**.** *Let $1 \leq a \leq b \leq n$ and let $I \subset P$ of size $|I| = \mu n$. A (partial) access structure $\Gamma = (\Gamma_{\text{yes}}, \Gamma_{\text{no}})$ over n parties is called (I,a,b)-somewhat regular if for every $A \in \Gamma_{\text{yes}} \cup \Gamma_{\text{no}}$ it holds that*

$$\mu a \leq |A \cap I| \leq \mu b \quad \text{and} \quad (1-\mu)a \leq |A \cap (P \setminus I)| \leq (1-\mu)b.$$

Note that previous papers considered the case where $\mu = 1/2$. We next construct a linear secret-sharing scheme for any (I, a, b)-somewhat regular access structure using a robust CDS protocol.

Lemma 5.7. *Let $\Gamma = (\Gamma_{\text{yes}}, \Gamma_{\text{no}})$ be an (I, a, b)-somewhat regular access structure and let $\mu = |I|/n$, $t_1 = \sum_{j=\mu a}^{\mu b} \binom{\mu b}{j}$, and $t_2 = \sum_{j=(1-\mu)a}^{(1-\mu)b} \binom{(1-\mu)b}{j}$. Assume that for all functions $f : \{0,1\}^{\mu n} \times \{0,1\}^{(1-\mu)n} \to \{0,1\}$ and every $\mathcal{Z}_1 \subseteq \{0,1\}^{\mu n}$, $\mathcal{Z}_2\{0,1\}^{(1-\mu)n}$ such that every set in $\mathcal{Z}_1$ and $\mathcal{Z}_2$ has size at most t_1 and t_2 respectively and $\mathcal{Z}_1$ and $\mathcal{Z}_2$ contain at most $\binom{\mu n}{\mu b}$ and $\binom{(1-\mu)n}{((1-\mu)b}$ maximal sets respectively there exists a 2-server $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust linear CDS protocol, such that the message sizes of Alice and Bob are s_{A} and s_{B} respectively.*

Then there exists a linear secret-sharing scheme realizing Γ such that the share size of every party $p_i \in I$ is less than $s_{\mathsf{A}} \cdot \sum_{j=\mu a}^{\mu b} \binom{\mu n}{j}$ and the share size of every party $p_i \in P \setminus I$ is less than $s_{\mathsf{B}} \cdot \sum_{j=(1-\mu)a}^{(1-\mu)b} \binom{(1-\mu)n}{j}$.

Proof. In the following, we let $X = \{0,1\}^{\mu n}$, let $Y = \{0,1\}^{(1-\mu)n}$, and let $I = \{p_{i_1} < \cdots < p_{i_{\mu n}}\}$ and $P \setminus I = \{p_{i_{\mu n+1}} < \cdots < p_{i_n}\}$. For a string $x \in X$ we denote $A^1_x = \left\{p_{i_j} : j \in [\mu n], x_j = 1\right\}$, and for a string $y \in Y$ we denote $A^2_y = \left\{p_{i_{j+\mu n}} : j \in [(1-\mu)n], y_j = 1\right\}$. We next present the construction.

Scheme 5.8.
Input: *The dealer holds a secret $s \in \{0,1\}$.*

1. *Let $f : X \times Y \to \{0,1\}$ be some monotone function that agrees with Γ, that is, for every $x \in X$ and $y \in Y$ such that $A^1_x \cup A^2_y \in \Gamma_{\text{yes}} \cup \Gamma_{\text{no}}$, it holds that $f(x,y) = 1$ if and only if $A^1_x \cup A^2_y \in \Gamma_{\text{yes}}$,[3] and let*

$$\mathcal{Z}_1 = \{\{x^* \leq x : \text{wt}(x^*) \geq \mu a\} : \text{wt}(x) \leq \mu b\}$$

and

$$\mathcal{Z}_2 = \{\{y^* \leq y : \text{wt}(^*) \geq (1-\mu)a\} : \text{wt}(y) \leq (1-\mu)b\}\,.$$

2. *Sample a random string r for a 2-server $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust linear CDS protocol $\mathcal{P} = (\mathsf{A}, \mathsf{B})$ for f.*

3. *For every $x \in X$ whose Hamming weight w is in the interval $[\mu a, \mu b]$, compute $\mathsf{sh}_{\mathsf{A}}(x, s) = \mathsf{A}(x, s; r)$ and share it between the parties in A^1_x (i.e., the parties that correspond to the string x) using a w-out-of-w additive secret-sharing scheme. We denote the share of $p_i \in A^1_x$ by $\mathsf{sh}_{\mathsf{A}}(x, s, i)$.*

4. *For every $y \in Y$ whose Hamming weight w is in the interval $[(1-\mu)a, (1-\mu)b]$, compute $\mathsf{sh}_{\mathsf{B}}(y, s) = \mathsf{B}(y, s; r)$ and share it between the parties in A^2_y using a w-out-of-w additive secret-sharing scheme. We denote the share of $p_i \in A^2_y$ by $\mathsf{sh}_{\mathsf{B}}(y, s, i)$.*

5. *The share of $p_i \in I$ is $(\mathsf{sh}_{\mathsf{A}}(x, s, i))_{x \in X, p_i \in A^1_x}$ and the share of $p_i \in P \setminus I$ is $(\mathsf{sh}_{\mathsf{B}}(y, s, i))_{y \in Y, p_i \in A^2_y}$.*

[3] Since $\Gamma_{\text{yes}} \cup \Gamma_{\text{no}}$ is a partial access structure, such monotone f exists.

We first show that Scheme 5.8 is correct. Fix $A \in \Gamma_{\text{yes}}$. Since Γ is (I, a, b)-somewhat regular, $\mu a \leq |A \cap I| \leq \mu b$ and $(1-\mu)a \leq |A \cap (P \setminus I)| \leq (1-\mu)b$. Therefore, for the strings $x \in X$ and $y \in Y$ such that $A^1_x = A \cap I$ and $A^2_y = A \cap (P \setminus I)$, it holds that $f(x, y) = 1$ and that their Hamming weights are in the intervals $[\mu a, \mu b]$ and $[(1-\mu)a, (1-\mu)b]$, respectively. Hence, the parties in A^1_x can reconstruct $\mathsf{sh}_{\mathsf{A}}(x, s)$ and the parties in A^2_y can reconstruct $\mathsf{sh}_{\mathsf{B}}(y, s)$. By the correctness of the $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust CDS protocol, they can reconstruct the secret s.

We next show privacy. Fix $A \in \Gamma_{\text{no}}$, let $x \in X$ denote the string such that $A^1_x = A \cap I$, and let $y \in Y$ denote the string such that $A^2_y = A \cap (P \setminus I)$. Consider two secrets s_0 and s_1. To show privacy, we prove that the corresponding shares of the parties in A, denoted $\mathcal{D}(s_0)$ and $\mathcal{D}(s_1)$ respectively, are identically distributed. Since for every $x' \not\leq x$ and every $y' \not\leq y$, the parties in $A^1_x \cup A^2_y$ miss at least one share of $\mathsf{sh}_{\mathsf{A}}(x', s)$ and one share of $\mathsf{sh}_{\mathsf{B}}(y', s)$, by the security of the additive secret-sharing scheme, the shares $\mathsf{sh}_{\mathsf{A}}(x', s)$ and $\mathsf{sh}_{\mathsf{B}}(y', s)$ held by $A^1_x \cup A^2_y$ are uniformly distributed and independent of the secret and the other shares and can be ignored.

The remaining shares are

$$\left\{\mathsf{sh}_{\mathsf{A}}(x^*, s, j) : x^* \leq x, p_j \in A^1_{x^*}\right\} \cup \left\{\mathsf{sh}_{\mathsf{B}}(y^*, s, j) : y^* \leq y, p_j \in A^2_{y^*}\right\},$$

i.e., the shares of the messages corresponding to the inputs

$$Z_1 = \{x^* \leq x : \mathrm{wt}(x^*) \geq \mu a\} \text{ and } Z_2 = \{y^* \leq y : \mathrm{wt}(y^*) \geq (1-\mu)a\}.$$

Since f is monotone and $f(x, y) = 0$, the set $Z_1 \times Z_2$ is a zero set. By the robustness of the $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust CDS scheme, it follows that $\mathcal{D}(s_0) \equiv \mathcal{D}(s_1)$.

Finally, we analyze the share size. The set $\mathcal{Z}_1$ contains $\binom{\mu n}{\mu b}$ maximal sets (one such set for each string of weight μb) and each $Z \in \mathcal{Z}_1$ contains at most $t_1 = \sum_{j=\mu a}^{\mu b} \binom{\mu b}{j}$ inputs (one input for every string $x^* \leq x$ such that $\mathrm{wt}(x^*) \geq \mu a$ and $\mathrm{wt}(x) \leq \mu b$). Similarly, the set $\mathcal{Z}_2$ contains $\binom{(1-\mu)n}{(1-\mu)b}$ maximal sets and each $Z \in \mathcal{Z}_2$ contains at most $t_2 = \sum_{j=(1-\mu)a}^{(1-\mu)b} \binom{(1-\mu)b}{j}$ inputs. Thus, the message sizes of Alice and Bob in the $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust CDS protocol are s_{A} and s_{B} respectively. Every party $p_i \in I$ receives a share of Alice's message in the $(\mathcal{Z}_1, \mathcal{Z}_2)$-robust CDS protocol for every $x \in X$ of Hamming weight in $[\mu a, \mu b]$, where $p_i \in A^1_x$. Since there are $\sum_{j=\mu a}^{\mu b} \binom{\mu n - 1}{j-1}$ such strings $x \in X$ where $p_i \in A^1_x$, its share size is less than $s_{\mathsf{A}} \cdot \sum_{j=\mu a}^{\mu b} \binom{\mu n}{j}$. Similarly, the share size of every party $p_i \in P \setminus I$ is less than $s_{\mathsf{B}} \cdot \sum_{j=(1-\mu)a}^{(1-\mu)b} \binom{(1-\mu)n}{j}$. □

We next construct a linear secret-sharing scheme for a multislice access structure using linear secret-sharing schemes for somewhat regular access structures. Fix $a, b \in [n^{0.8}, n - n^{0.8}]$ such that $a < b$ (we deal with the general case below) and let $\varepsilon = n^{-0.2}$ be a proximity parameter. Let $I \subseteq P$ and let $\mu = |I|/n$ and let $A \subseteq P$. We say that A is *good* for I if

$$\mu a - \varepsilon \leq |A \cap I| \leq \mu b + \varepsilon \quad \text{and} \quad (1-\mu)a - \varepsilon \leq |A \cap (P \setminus I)| \leq (1-\mu)b + \varepsilon.$$

If A is not good then we call it bad. We will use the following lemma.

Lemma 5.9. *For all $a, b \in [n^{0.8}, n - n^{0.8}]$ such that $a < b$ and every constant $\mu \in (0, 1)$, there exists a collection of $\lambda = O(n)$ subsets $I_1, \ldots, I_\lambda \subseteq P$, each of size μn, such that for all $A \subseteq P$ satisfying $a \leq |A| \leq b$, the set A is good for at least 0.7λ of the subsets.*

Proof. We prove the lemma using the probabilistic method. We sample a collection of $\lambda = O(n)$ subsets, each of size μn, uniformly at random, and show that with positive probability all inputs are good for at least 0.7λ of the subsets.

Let us first analyze this probability with respect to a single subset I sampled uniformly at random from all subsets of P of size μn. Fix $A \subseteq P$ such that $a \leq |A| \leq b$. We next show that

$$\mu a - \varepsilon \leq |A \cap I| \leq \mu b + \varepsilon \quad \text{and} \quad (1-\mu)a - \varepsilon \leq |A \cap (P \setminus I)| \leq (1-\mu)b + \varepsilon$$

with overwhelming probability. We only prove the former as the latter can be proved using an analogous argument.

For every $i \in A$ let X_i denote the indicator for the event $i \in A \cap I$. Then these random variables are negatively associated (see Claim A.4). We denote by $X = \sum_{i \in A} X_i$ the random variable that is equal to $|A \cap I|$. By linearity of expectation, $\mathrm{E}[X] = \mu|A|$, hence $\mu a \leq \mathrm{E}[X] \leq \mu b$. Now, since $\varepsilon = n^{-0.2} < \mu b$, by Chernoff's inequality (see Theorem A.2),

$$\begin{aligned}\Pr[X > \mu b + \varepsilon] &\leq \Pr[X > \mu|A| + \varepsilon] = \Pr\left[X > \mu|A| \cdot \left(1 + \varepsilon \cdot \frac{1}{\mu|A|}\right)\right] \\ &\leq e^{-\varepsilon^2 \cdot \frac{\mu|A|}{3}} \leq e^{-n^{-0.4} \cdot \frac{\mu n^{0.8}}{3}} = e^{-\frac{\mu n^{0.4}}{3}}.\end{aligned}$$

Similarly,

$$\Pr[X < \mu a - \varepsilon] \leq e^{-\frac{\mu n^{0.4}}{3}}.$$

Therefore, by the union bound,

$$\mu a - \varepsilon \leq |A \cap I| \leq \mu b + \varepsilon \quad \text{and} \quad (1-\mu)a - \varepsilon \leq |A \cap (P \setminus I)| \leq (1-\mu)b + \varepsilon$$

except with probability $o(1)$.

Finally, if we independently sample λ subsets, the probability that A is bad for at least 0.3λ of them is, by a Chernoff bound, at most $2^{-\Omega(\lambda)}$. By taking $\lambda = Cn$ for sufficiently large constant C, the latter probability is smaller than 2^{-n}. Applying the union bound over all possible subsets A, the probability that every A is good for at least 0.7λ of the sets is strictly greater than 0. Therefore, there exists λ subsets such that A is good for at least 0.7λ of them. □

We can now realize $(a : b, n)$-multislice access structures. We first realize them for all $n^{0.8n} \leq a < b \leq n - n^{0.8}$.

Lemma 5.10. *Let $n^{0.8} \leq a < b \leq n - n^{0.8}$ and let Γ be an $(a : b, n)$-multislice access structure. Let $c = a - n^{-0.2}$, let $d = b + n^{-0.2}$, and let $\mu \in (0, 1)$. Assume that for all $I \subset P$, where $|I| = \mu n$, any (I, c, d)-somewhat regular access structure can be realized with a linear secret-sharing scheme where the share size is at most m. Then Γ can be realized with a linear secret-sharing scheme with share size at most $O(mn \log n)$.*

Proof. We start by considering an $(a : b, n)$ partial multislice access structure, namely, the access structure is defined only over the inputs whose Hamming weight is in the interval $[a, b]$. The scheme is as follows.

1. Let $I_1, \ldots, I_\lambda$ be the collection of $\lambda = O(n)$ subsets of P guaranteed by Lemma 5.9.

2. Share the secret s using a $\lambda/2$-out-of-λ Shamir's secret-sharing scheme. Let $\sigma_1, \ldots, \sigma_\lambda$ denote the shares.

3. For every $j \in [\lambda]$, share σ_j with fresh randomness using the secret-sharing scheme realizing the (I_j, c, d)-somewhat regular access structure $\Gamma_{I_j,c,d}$ that agrees with Γ.

We now analyze the construction. We start with showing correctness. Let $A \in \Gamma_{\text{yes}}$. By Lemma 5.9, the set $J = \{j : A \text{ is good for } I_j\}$ is of size $|J| \geq 0.7\lambda$. Therefore, at least 0.7λ of the shares σ_j can be reconstructed by the parties in A. Thus, they can reconstruct s.

We next show privacy. Let $A \in \Gamma_{\text{no}}$. Since A is good for at least 0.7λ of the subsets, by the privacy of each $\Gamma_{I_j,c,d}$, it holds that at least 0.7λ of the shares σ_j remain perfectly hidden. Therefore, the secret s remains perfectly hidden.

As for the share size, first, note that each σ_j is of size $O(\log \lambda)$. Therefore, the share size in the above scheme is at most $O(m\lambda \log \lambda) = O(mn \log n)$.

We now handle the fully defined access structure, that is, we need to consider the "big" and "small" sets. Recall that for all $A \subseteq P$ where $|A| < a$ it holds that $A \notin \Gamma$, and for all $B \subseteq P$ where $|B| > b$ it holds that $B \in \Gamma$. The scheme is as follows.

1. Share s using a $(b+1)$-out-of-n Shamir's secret-sharing scheme, and give the i^{th} share to p_i.

2. Share s into s_0 and s_1 using a 2-out-of-2 secret-sharing scheme.

3. Share s_0 using an a-out-of-n Shamir's secret-sharing scheme and give the i^{th} share to the p_i.

4. Share s_1 using the secret-sharing scheme for the partial access structure shown above, and give the i^{th} share to the p_i.

For correctness, note that if $A \in \Gamma$ is of size at least $b+1$ then the parties can reconstruct s using the Shamir shares of the first scheme. Otherwise, it must be the case where $a \leq |A| \leq b$, hence the parties can reconstruct s_0 and s_1, and thus can reconstruct s.

For privacy, observe that if $A \notin \Gamma$ is of size $|A| < a$, the parties cannot reconstruct s_0, hence s is perfectly hidden. Otherwise, it must be the case where $a \leq |A| \leq b$, hence the Shamir shares of Step 1 reveal no information to the parties, and they cannot reconstruct s_1 due to the privacy of the underlying scheme.

Finally, note that the share size is only $O(\log n)$ additively longer than the share size for the partial access structure. Thus, the share is at most $O(mn \log n)$. □

We can now put everything together and construct a secret-sharing scheme for all multislice access structures. As a corollary to this theorem, we obtain Lemma 5.3. Note that in the bootstrapping step, we need to realize multislices (see Lemma 5.15). We note that the result is not optimal for all parameters, and for some parameters better upper bounds can be obtained by either using the cover lemma (see Lemma 5.14 below) or the results of [9]. We use the following results due to Liu and Vaikuntanathan [52], which decomposes an access structure.

Proposition 5.11 ([52])**.** *Let Γ be an access structure and let $a < b \in [n]$. Define three access structures $\Gamma_{\text{bot}}(a)$, $\Gamma_{\text{mid}}(a, b)$, and $\Gamma_{\text{top}}(b)$ as follows*

$$\Gamma_{\text{bot}}(a) : A \in \Gamma_{\text{bot}}(a) \iff \exists A' \in \Gamma \text{ s.t. } A' \subseteq A \text{ and } |A'| < a$$
$$\Gamma_{\text{mid}}(a, b) : A \in \Gamma_{\text{mid}}(a, b) \iff A \in \Gamma \text{ and } a \leq |A| \leq b, \text{ or } |A| > b$$
$$\Gamma_{\text{top}}(b) : A \notin \Gamma_{\text{top}}(b) \iff \exists A' \notin \Gamma \text{ s.t. } A \subseteq A' \text{ and } |A'| > b.$$

Then $\Gamma = (\Gamma_{\text{bot}}(a) \cup \Gamma_{\text{mid}}(a,b)) \cap \Gamma_{\text{top}}(b)$. *Consequently, if* $\Gamma_{\text{bot}}(a)$, $\Gamma_{\text{mid}}(a,b)$, *and* $\Gamma_{\text{top}}(b)$ *can be realized with linear secret-sharing schemes with exponent* S, *then so is* Γ.

Theorem 5.12 (Multislice theorem)**.** *Let* $1 \le a < b \le n$. *Then every* $(a : b, n)$*-multislice access structure can be realized by a linear secret-sharing scheme with share size at most* $2^{\frac{n(2n-h(a/b)\cdot b)}{3n-2h(a/b)\cdot b}+o(n)}$ *if* $a > b/2$, *and share size at most* $2^{\frac{n(2n-b)}{3n-2b}+o(n)}$ *otherwise. Consequently, if* $a = \alpha n$ *and* $b = \beta n$, *then the exponent* $\mathbf{M}_\ell(\alpha : \beta)$ *satisfies.*

$$\mathbf{M}_\ell(\alpha : \beta) \le \begin{cases} \frac{2-h(\alpha/\beta)\cdot\beta}{3-2h(\alpha/\beta)\cdot\beta} & \text{if } \alpha > \beta/2 \\ \frac{2-\beta}{3-2\beta} & \text{otherwise.} \end{cases}$$

Proof. Let Γ be an $(a : b, n)$-multislice access structure. Let $a' = \max\{a, n^{0.8}\}$ and let $b' = \min\{b, n - n^{0.8}\}$. Consider the three access structures $\Gamma_{\text{bot}}(a')$, $\Gamma_{\text{mid}}(a', b')$, and $\Gamma_{\text{top}}(b')$. Then by Proposition 5.11 we can write $\Gamma = (\Gamma_{\text{bot}}(a') \cup \Gamma_{\text{mid}}(a', b')) \cap \Gamma_{\text{top}}(b')$. We realize $\Gamma_{\text{bot}}(a')$ using a DNF scheme, and realize $\Gamma_{\text{top}}(b')$ using a CNF scheme, respectively. The cost of these schemes is at most $n\binom{n}{n^{0.8}} = 2^{o(n)}$.

It is left to realize $\Gamma_{\text{mid}}(a', b')$. Observe that $\Gamma_{\text{mid}}(a', b')$ is an $(a' : b', n)$-multislice access structure where $n^{0.8} \le a' < b' \le n - n^{0.8}$. We realize Γ_{mid} using the scheme given by Lemma 5.10, and using the scheme from Lemma 5.7 to realize the somewhat regular access structures. The total share size is $s \cdot O(n \log n)$, where

$$s = \max\left\{ s_{\mathsf{A}} \cdot \sum_{j=\mu a}^{\mu b} \binom{\mu n}{j},\ s_{\mathsf{B}} \cdot \sum_{j=(1-\mu)a}^{(1-\mu)b} \binom{(1-\mu)n}{j} \right\},$$

where s_{A} and s_{B} are the message sizes of Alice and Bob in the $(\mathcal{Z}_1, 2^{(1-\mu)n})$-robust linear CDS protocol guaranteed by Corollary 5.5, where $t = \sum_{j=\mu a}^{\mu b}\binom{\mu b}{j}$ and $u = \binom{\mu n}{\mu b}$. Then $s_{\mathsf{A}} = O(\frac{2^{\mu n}}{t}\log^3 t \cdot \mu n \cdot \log\binom{\mu n}{\mu b})$, and $s_{\mathsf{B}} = O(t \log^3 t \cdot (1-\mu)n \cdot \log\binom{\mu n}{\mu b})$. Since $\sum_{j=\mu a}^{\mu b}\binom{\mu n}{j} < 2^{\mu n}$ and $\sum_{j=(1-\mu)a}^{(1-\mu)b}\binom{(1-\mu)n}{j} < 2^{(1-\mu)n}$, the share size is

$$s = \max\left\{ O\left(\frac{2^{\mu n}}{t}\log^3 t \cdot \mu n \cdot \log\binom{\mu n}{\mu b}\right)\cdot 2^{\mu n},\right.$$
$$\left. O\left(t \log^3 t \cdot (1-\mu)n \cdot \log\binom{\mu n}{\mu b}\right)\cdot 2^{(1-\mu)n} \right\},$$

In order to optimize the share complexity (up to polynomial factors in n), we choose μ such that

$$\frac{2^{2\mu n}}{t} = t \cdot 2^{(1-\mu)n}. \tag{10}$$

Observe that

$$t = \begin{cases} O(2^{h(a/b)\cdot\mu b}) & \text{if } a > b/2 \\ O(2^{\mu b}) & \text{otherwise.} \end{cases}$$

Therefore, to ensure that (10) holds we take

$$\mu = \begin{cases} \frac{n}{3n-2h(a/b)\cdot b} & \text{if } a > b/2 \\ \frac{n}{3n-2b} & \text{otherwise.} \end{cases}$$

Observe that in both cases it holds that $\mu \in (0,1)$. Thus, if $a > b/2$ then, up to polynomial factors, the share size is $\frac{2^{2\mu n}}{t} = 2^{2\mu n - h(a/b)\cdot \mu b} = 2^{\frac{n(2n-h(a/b)\cdot b)}{3n-2h(a/b)\cdot b}}$, and if $a \leq b/2$ then, up to polynomial factors, the share size is $\frac{2^{2\mu n}}{t} = 2^{2\mu n - \mu b} = 2^{\frac{n(2n-b)}{3n-2b}}$. □

As a corollary, we obtain Lemma 5.3.

Proof of Lemma 5.3. Observe that for every $b \in [n]$, every $(0 : b, n)$-multislice access structure is also a (b, n)-downslice access structure. Therefore any (b, n)-downslice can be realized with a linear secret-sharing scheme with share size at most $2^{\frac{n(2n-b)}{3n-2b}+o(n)}$. □

5.2 Constructing a Linear Secret-Sharing for Downslices With High Density

Similarly to [9], for downslices with high density, we reduce this to the case of low density via the covering lemma.

Lemma 5.13 (Cover reduction lemma [6, 9])**.** *Let $a < b \leq n$ be positive integers. If $(b-a, n-a)$-downslices can be linearly realized with share size $z(b-a, n-a)$ then (b, n)-downslices can be linearly realized with share size of*

$$\left[\binom{n}{n-b} / \binom{n-a}{n-b}\right] \cdot \left[1 + \log \binom{n-a}{n-b}\right] \cdot z(b-a, n-a).$$

Consequently, for all constants $0 \leq \alpha < \beta \leq 1$ it holds that if $(\alpha m, m)$-downslices can be linearly realized with exponent $\tilde{z}(\alpha)$, then $(\beta n, n)$-downslices can be linearly realized with exponent at most

$$h(\beta) - (1-\beta) \cdot \frac{h(\alpha) - \tilde{z}(\alpha)}{1-\alpha}.$$

Combined with Lemma 5.3, we obtain the following result for downslices with high density. We note that in [9], the reduction of a downlice with density β was to a downslice with density $1/2$. Since we improve the results for slices with low density, we reduce to a different value.

Lemma 5.14 (High density)**.** *Let $\alpha_0 > 0.5412$ denote the unique solution to the equation $(2x-3)^2 \log(x) + 2x^2 - 8x + 7 = 0$ in the interval $[0,1]$. For every integers n and $b \in (\alpha_0 n, n]$, every (b, n)-downslice can be linearly realized with share size at most*

$$\left[\binom{n}{n-b} / \binom{\frac{n-b}{1-\alpha_0}}{n-b}\right] \cdot \left[1 + \log \binom{\frac{n-b}{1-\alpha_0}}{n-b}\right] \cdot 2^{\frac{2-\alpha_0}{(3-2\alpha_0)(1-\alpha_0)}\cdot(n-b)}.$$

Consequently, for all $\beta \in (\alpha_0, 1]$,

$$\mathbf{D}_\ell(\beta) \leq h(\beta) - (1-\beta) \cdot \frac{h(\alpha_0) - \frac{2-\alpha_0}{3-2\alpha_0}}{1-\alpha_0} < h(\beta) - 0.51079 \cdot (1-\beta).$$

Proof. By Lemmas 5.3 and 5.13, for every $a \in (0, b)$, every (b, n)-downslices can be linearly realized with share size of

$$\begin{aligned}
&\left[\binom{n}{n-b} / \binom{n-a}{n-b}\right] \cdot \left[1 + \log \binom{n-a}{n-b}\right] \cdot 2^{\frac{2n-b-a}{3n-2b-a}\cdot(n-a)+o(n)} \\
&\leq 2^{h(b)\cdot n - h\left(\frac{n-b}{n-a}\right)\cdot(n-a) + \frac{2n-b-a}{3n-2b-a}\cdot(n-a)+o(n)} \\
&= 2^{h(b)\cdot n - \left(h\left(\frac{n-b}{n-a}\right) - \frac{2n-b-a}{3n-2b-a}\right)\cdot(n-a)+o(n)}.
\end{aligned}$$

Let $a' = \frac{b-a}{n-a} \cdot n \in (0,b)$ and let $\alpha' = a'/n$. Then $a = \frac{b-a'}{n-a'} \cdot n$ and $n - a = \frac{n-b}{n-a'} \cdot n$, hence the share size is at most

$$\begin{aligned} 2^{h(b)\cdot n - \left(h\left(\frac{n-b}{n-a}\right) - \frac{2n-b-a}{3n-2b-a}\right)\cdot(n-a)+o(n)} &= 2^{h(b)\cdot n - \left(h(\alpha') - \frac{n-b+\frac{n-b}{n-a'}\cdot n}{2(n-b)+\frac{n-b}{n-a'}\cdot n}\right)\cdot\frac{n-b}{n-a'}\cdot n + o(n)} \\ &= 2^{\left(h(b) - (n-b)\cdot\frac{h(\alpha') - \frac{2n-a'}{3n-2a'}}{n-a'}\right)\cdot n + o(n)} \\ &= 2^{\left(h(b) - (n-b)\cdot\frac{h(\alpha') - \frac{2-\alpha'}{3-2\alpha'}}{(1-\alpha')n}\right)\cdot n + o(n)}. \end{aligned}$$

The share size is minimized when $\frac{h(\alpha') - \frac{2-\alpha'}{3-2\alpha'}}{1-\alpha'}$ is maximized. Taking the derivative, we obtain that the maximum occurs when $(2\alpha'-3)^2 \log(\alpha') + 2(\alpha')^2 - 8\alpha' + 7 = 0$. By definition, this holds for $\alpha' = \alpha_0$. Thus, the claim follows. □

5.3 Applying the Bootstrapping of Applebaum and Nir

Applebaum and Nir [9] showed a bootstrapping algorithm for linear secret-sharing schemes by exploiting duality. They proved the following.

Lemma 5.15 (Bootstrapping)**.** *Given an integer n and a target slice $b \in [n]$ let $\mathbf{D}_\ell(b,n)[0]$ denote the share size given by Lemma 5.2. For every $i \geq 0$ define*

$$\mathbf{D}_\ell(b,n)[i+1] = \min_{a \leq b}\left(\max\left(\mathbf{M}_\ell(a:b,n), \max_{c \leq a}\left(h(c/n) - (c/n)\cdot\frac{h(b/n) - \mathbf{D}_\ell(b,n)[i]}{1-b/n}\right)\right)\right)$$

Then for every $i \geq 0$, every (b,n)-downslice can be realized with a linear secret-sharing scheme with share size at most $\mathbf{D}_\ell(b,n)[i]$.

We next define the function that captures the exponent of the construction. For every $i \geq 0$ we define a sequence of functions $d_i : [0,1] \to [0,1]$ as follows. Let

$$d_1(\beta) = \begin{cases} \frac{1}{2} + \frac{\beta}{2} & \text{if } \beta \leq 1/2 \\ \frac{2-\beta}{3-2\beta} & \text{if } 1/2 < \beta \leq 0.5412 \\ h(\beta) - 0.51079\cdot(1-\beta) & \text{otherwise} \end{cases}$$

be the exponent given by Lemma 5.2. Next, for every $i \geq 0$ let

$$d_{i+1}(\beta) = \min_{\alpha \leq \beta}\left(\max\left(m(\alpha,\beta), \max_{\gamma \leq \alpha}(u_i(\gamma,\beta))\right)\right), \tag{11}$$

where $u_i, m : [0,1]^2 \to [0,1]$ are defined as

$$u_i(\gamma,\beta) = h(\gamma) - \gamma\cdot\frac{h(\beta) - d_i(\beta)}{1-\beta}$$

and

$$m(\alpha,\beta)=\begin{cases}\frac{2-h(\alpha/\beta)\cdot\beta}{3-2h(\alpha/\beta)\cdot\beta} & \text{if } \alpha>\beta/2\\ \frac{2-\beta}{3-2\beta} & \text{otherwise}\end{cases}$$

is the exponent given by Theorem 5.12. By Lemma 5.15, we obtain the following.

Lemma 5.16. *For every $i\geq 0$ and every $\beta\in[0,1]$, it holds that $\mathbf{D}_{\ell}(\beta)\leq d_i(\beta)$.*

We next show how to simplify Equation (11). Since

$$\frac{\partial u_i(\gamma,\beta)}{\partial\gamma}=\log\left(\frac{1}{\gamma}-1\right)-\frac{h(\beta)-d_i(\beta)}{1-\beta},$$

it follows that for every β, the function $u_i(\gamma,\beta)$ is maximized at

$$\gamma_i:=\left(1+2^{\frac{h(\beta)-d_i(\beta)}{1-\beta}}\right)^{-1}.$$

Let $\widetilde{u}_i:[0,1]^2\to[0,1]$ be defined as $\widetilde{u}_i(\alpha,\beta)=\max_{\gamma\leq\alpha}(u_1(\gamma,\beta))$. Then

$$\widetilde{u}_i(\alpha,\beta)=\begin{cases}u_i(\gamma_i,\beta) & \text{if } \gamma_i\leq\alpha\leq\beta\\ u_i(\alpha,\beta) & \text{otherwise}\end{cases}.$$

Therefore,

$$d_{i+1}(\beta)=\min_{\alpha\leq\beta}\left(\max\left(m(\alpha,\beta),\ \widetilde{u}_i(\alpha,\beta)\right)\right).$$

Finally, observe that $m(\alpha,\beta)$ is a decreasing function of α, while $\widetilde{u}_i(\alpha,\beta)$ is an increasing function of α. Since both are continuous, they intersect at exactly one value $\widetilde{\alpha}_i$. This value also minimizes $\max\left(m(\alpha,\beta),\ \widetilde{u}_i(\alpha,\beta)\right)$. Therefore,

$$d_{i+1}(\beta)=m(\widetilde{\alpha}_i,\beta).$$

5.4 A Linear Secret-Sharing Scheme for All Access Structures

We can now show that every access structure can be realized with a linear secret-sharing scheme with an exponent at most 0.7563, thus proving Theorem 5.1. The idea for the construction is roughly as follows. First, as observed by [9] it suffices to consider downslice accesss structures. Similarly to [9], we realize each $(\beta n,n)$-downslice, for $\beta\in[0,1]$, by partitioning $[0,1]$ into 3 segments and deal with each segment using a different scheme. We partition $[0,1]$ differently than [9]. Specifically, we consider the segments $[0,1/2]$, $[1/2,0.554]$, and $[0.554,1]$ ([9] used 0.535 instead of 0.554). Now, for every $\beta<1/2$, we use the scheme of Applebaum and Nir [9] as stated in Lemma 5.2, for every $\beta\in[1/2,0.554]$ we reduce it to $(n/2,n)$-downslice, and for every $\beta\in[0.554,1]$ we reduce it to $(0.554\cdot n,n)$-downslice. We then apply the bootstrapping construction (Lemma 5.15) for the two values $n/2$ and $0.554\cdot n$ for 7 iterations. We stress that the choice of our parameters was done using a computer to approximate the optimal choice.

We now formalize the proof. We use the following claim proved by [9], stating that every n-party access structure is the conjunction of at most n downslices.

Claim 5.17. *Let Γ be an n-party access structure. For every $b \in [n]$ let Γ_b denote the b-downslice of Γ. Then $\Gamma = \bigcap_{b=1}^{n} \Gamma_b$. In particular, if every Γ_b can be realized with a linear secret-sharing scheme with exponent S, then so can Γ.*

Proof of Theorem 5.1. By Claim 5.17, it suffices to construct a scheme for every downslice. The proof is done by applying Lemma 5.15 for $b_1 = n/2$ and $b_2 = 0.554 \cdot n$ for 7 iterations, and then use the cover reduction lemma (Lemma 5.13) to reduce every other downslice to one these two values. Using a computer to compute the exponents, we get that $\mathbf{D}_\ell(0.5) < 0.736$ and $\mathbf{D}_\ell(0.554) < 0.752$. Using the cover reduction lemma, we obtain the following bounds on the share size of linear secret-sharing schemes for downslices, as explained below.

1. If $0 \leq \beta < 1/2$ then by Lemma 5.2 it holds that $\mathbf{D}_\ell(\beta) \leq \frac{1}{2} + \frac{\beta}{2} < 0.75$.
2. If $1/2 \leq \beta < 0.554$ then we use the covering lemma with $\alpha = 1/2$ to obtain
$$\mathbf{D}_\ell(\beta) < h(\beta) - (1-\beta) \cdot \frac{h(1/2) - \mathbf{D}_\ell(0.5)}{1 - 0.5} < 0.7561.$$
3. If $0.554 \leq \beta \leq 1$ then we use the covering lemma with $\alpha = 0.554$ to obtain
$$\mathbf{D}_\ell(\beta) < h(\beta) - (1-\beta) \cdot \frac{h(0.554) - \mathbf{D}_\ell(0.554)}{1 - 0.554} < 0.7563.$$

□

Bibliography

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

[2] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proc. of the 24th International Colloquium on Automata, Languages and Programming*, volume 1256 of *LNCS*, pages 401–407, 1997.

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

[4] Benny Applebaum, Barak Arkis, Pavel Raykov, and Prashant Nalini Vasudevan. Conditional disclosure of secrets: Amplification, closure, amortization, lower-bounds, and separations. In *CRYPTO 2017*, volume 10401 of *LNCS*, pages 727–757, 2017.

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

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

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

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

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

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

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

[12] Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011.

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

[14] Amos Beimel, Oriol Farràs, and Or Lasri. Improved polynomial secret-sharing schemes. Cryptology ePrint Archive, Paper 2023/1158, 2023.

[15] Amos Beimel, Anna Gál, and Mike Paterson. Lower bounds for monotone span programs. Research Series BRICS-RS-94-46, BRICS, Department of Computer Science, University of Aarhus, December 1994.

[16] Amos Beimel and Yuval Ishai. Information-theoretic private information retrieval: A unified construction. In *ICLAP 2001*, volume 2076 of *LNCS*, pages 912–926, 2001. Journal version in [18].

[17] Amos Beimel, Yuval Ishai, Ranjit Kumaresan, and Eyal Kushilevitz. On the cryptographic complexity of the worst functions. In Yehuda Lindell, editor, *TCC 2014*, volume 8349 of *LNCS*, pages 317–342. Springer-Verlag, 2014.

[18] Amos Beimel, Yuval Ishai, and Eyal Kushilevitz. General constructions for information-theoretic private information retrieval. *J. of Computer and System Sciences*, 71(2):213–247, 2005.

[19] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Ilan Orlov. Share conversion and private information retrieval. In *Proceedings of the 27th Conference on Computational Complexity, CCC 2012*, pages 258–268, 2012.

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

[21] Amos Beimel and Naty Peter. Optimal linear multiparty conditional disclosure of secrets protocols. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 332–362. Springer, 2018.

[22] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35, 1988.

[23] George Robert Blakley. Safeguarding cryptographic keys. In *Managing requirements knowledge, international workshop on*, pages 313–313. IEEE Computer Society, 1979.

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

[25] Y. M. Chee, T. Feng, S. Ling, H. Wang, and L. F. Zhang. Query-efficient locally decodable codes of subexponential length. *Computational Complexity*, 22(1):159–189, 2013.

[26] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proc. of the 36th IEEE Symp. on Foundations of Computer Science*, pages 41–51, 1995. Journal version: *J. of the ACM*, 45:965–981, 1998.

[27] Ronald Cramer, Ivan Damgård, and Ueli Maurer. General secure multi-party computation from any linear secret-sharing scheme. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 316–334. Springer-Verlag, 2000.

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

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

[30] Devdatt P Dubhashi, Volker Priebe, and Desh Ranjan. Negative dependence through the fkg inequality. *BRICS Report Series*, 1996.

[31] Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

[32] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 705–714. IEEE Computer Society, 2010.

[33] Zeev Dvir and Sivakanth Gopi. 2-server PIR with sub-polynomial communication. In *47th STOC*, pages 577–584, 2015.

[34] Zeev Dvir and Sivakanth Gopi. 2-server PIR with subpolynomial communication. *J. ACM*, 63(4):39:1–39:15, 2016.

[35] Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, 41(6):1694–1703, 2012.

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

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

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

[39] Y. Gertner, Yuval Ishai, Eyal Kushilevitz, and T. Malkin. Protecting data privacy in private information retrieval schemes. In *Proc. of the 30th ACM Symp. on the Theory of Computing*, pages 151–160, 1998. Journal version: *J. of Computer and System Sciences*, 60(3):592–629, 2000.

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

[41] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20(1):71–86, 2000.

[42] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information theoretic private information retrieval. In *Proc. of the 31st ACM Symp. on the Theory of Computing*, pages 79 – 88, 1999. Journal version in [18].

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

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

[45] T. Itoh. Efficient private information retrieval. *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, E82-A(1):11–20, 1999.

[46] Toshiya Itoh and Yasuhiro Suzuki. Improved constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions on Information and Systems*, E93-D(2):263–270, 2010.

[47] Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.

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

[49] K. Kedlaya and S. Yekhanin. Locally decodable codes from nice subsets of finite fields and prime factors of Mersenne numbers. Technical Report TR07-040, Electronic Colloquium on Computational Complexity, www.eccc.uni-trier.de/eccc/, 2007.

[50] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. of Computer and System Sciences*, 69(3):395–420, 2004.

[51] Samuel Kutin. Constructing large set systems with given intersection sizes modulo composite numbers. *Combinatorics, Probability* Computing, Sep 2002.

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

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

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

[55] E. Mann. Private access to distributed information. Master's thesis, Technion – Israel Institute of Technology, Haifa, 1998.

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

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

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

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

[60] Lajos Rónyai, László Babai, and Murali K. Ganapathy. On the number of zero-patterns of a sequence of polynomials. *Journal of the AMS*, 14(3):717–735, 2001.

[61] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

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

[63] S. Wehner and R. de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Proc. of the 32nd International Colloquium on Automata, Languages and Programming*, volume 3580 of *LNCS*, pages 1424–1436, 2005.

[64] David Woodruff and Sergey Yekhanin. A geometric approach to information-theoretic private information retrieval. In *Proc. of the 20th IEEE Conf. on Computational Complexity*, pages 275–284, 2005.

[65] S. Yekhanin. Towards 3-query locally decodable codes of subexponential length. In *Proc. of the 39th ACM Symp. on the Theory of Computing*, pages 266–274, 2007.

A Negatively Associated Random Variables

Definition A.1 (Negative association [47])**.** *Let $X_1, \ldots, X_n$ be random variables. The random variables are negatively associated if for every two disjoint index sets $I, J \subseteq [n]$,*

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

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

It is known that the Chernoff-Hoeffding bounds are applicable to sums of variables that satisfy the negative association [56] (see also [31, Proposition 5]).

Theorem A.2 (Chernoff-Hoeffding bounds)**.** *Let $X_1, \ldots, X_n$ be negatively associated random variables taking values in $\{0, 1\}$, let $X = \sum_{i=1}^{n} X_i$, and let $\mu = \mathrm{E}[X]$. Then for every for every $\delta \in (0, 1)$,*

$$\Pr[X > (1+\delta)\mu] \leq e^{-\delta^2\mu/3} \quad \text{and that} \quad \Pr[X < (1-\delta)\mu] \leq e^{-\delta^2\mu/3}.$$

Finally, it is also known that when sampling a subset of $[n]$ uniformly at random, the random variables that correspond to the indicator of whether an element is in the sampled set, are negatively correlated.

Claim A.3 ([30, Theorem 10])**.** *Let $X_1, \ldots, X_n$ be random variables that take values in $\{0, 1\}$ and are distributed uniformly over m-weight vectors where $m \leq n$. That is, for every $x \in \{0, 1\}^n$ with Hamming weight m,*

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

and $\Pr[X = x] = 0$ for any other $x \in \{0, 1\}^n$. Then the random variables in $X_1, \ldots, X_n$ are negatively correlated.

Since any subset of negatively associated variables are also negatively associated [47, Property 4] we obtain the following.

Claim A.4. *Let I be a random subset of $[n]$ of size μn, sampled uniformly at random, and let $A \subseteq [n]$. For every $i \in A$ let X_i denote the indicator for the event $i \in A \cap I$. Then $(X_i)_{i\in A}$ are negatively correlated.*

B The PIR Protocol of Dvir and Gopi [34]

To be able to compare the PIR/CDS protocols that we present in this paper to previous protocols, we present the PIR protocol of Dvir and Gopi [34]. They use a matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N}$ over $\mathbb{Z}_6^h$ such that for all $i \neq j$ it holds that

$$\langle \mathbf{u}_i, \mathbf{v}_i \rangle \bmod m = 0 \quad \text{and} \quad \langle \mathbf{u}_i, \mathbf{v}_j \rangle \bmod 6 \in \{1, 3, 4\}.$$

(Note that in our construction over $\mathbb{Z}_6$, $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \bmod m = 1$ and $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \bmod 6 \in \{0, 2, 3, 4\}$.) Such a matching vector family with $h = 2^{O(\sqrt{\log N \log\log N})}$ is constructed in [41, 51]. In the following, for a prime p let $\langle \mathbf{u}, \mathbf{v} \rangle_p = \sum_{\ell=1}^{h} \mathbf{u}[\ell]\mathbf{v}[\ell] \bmod p$.

Protocol B.1.
Public parameters: *A matching vector family* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ *over* $\mathbb{Z}_6^h$.
Alice's and Bob's input: $D \in \{0,1\}^N$.
The user's input: $i \in [N]$.

- *The user chooses* $\mathbf{r} \leftarrow \mathbb{Z}_2^h$ *with uniform distribution and sends* $\mathbf{q}_a = \mathbf{r}$ *to Alice and* $\mathbf{q}_b = \mathbf{u}_i + \mathbf{r} \bmod 2$ *to Bob.*

- *Alice computes*

$$\mathbf{m}_A^1 = \sum_{j=1}^{N} (-1)^{\langle \mathbf{q}_A, \mathbf{v}_j \rangle_2} \cdot D_j \bmod 3, \quad \mathbf{m}_A^2 = \sum_{j=1}^{N} ((-1)^{\langle \mathbf{q}_A, \mathbf{v}_j \rangle_2} \cdot D_j)\mathbf{v}_j \bmod 3$$

  and Bob computes

$$\mathbf{m}_B^1 = \sum_{j=1}^{N} (-1)^{\langle \mathbf{q}_B, \mathbf{v}_j \rangle_2} \cdot D_j \bmod 3, \quad \mathbf{m}_B^2 = \sum_{j=1}^{N} ((-1)^{\langle \mathbf{q}_B, \mathbf{v}_j \rangle_2} \cdot D_j)\mathbf{v}_j \bmod 3.$$

  Alice and Bob send m_A^1, m_A^2 *and* m_A^1, m_A^2 *respectively to the user (each answer is a vector in* $\mathbb{Z}_3^h$ *and an element in* $\mathbb{Z}_3$*).*

- *The user outputs* $D_i = 1$ *if*

$$\langle \mathbf{u}_i, \mathbf{m}_B^2 - \mathbf{m}_A^2 \rangle - \mathbf{m}_B^1 + \mathbf{m}_A^1 \not\equiv 0 \pmod 3, \tag{12}$$

  and $D_i = 0$ *otherwise.*

Notice that in Protocol B.1 Alice and Bob need to send an additional element compared to Protocol 4.3 and the reconstruction and the proof of correctness are more involved. Furthermore, writing $(-1)^{\langle \mathbf{q}_A, \mathbf{v}_j \rangle_2}$ instead of $\langle \mathbf{q}_A, \mathbf{v}_j \rangle_2$ further complicated the protocol.