Improved Polynomial Secret-Sharing Schemes

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

Oriol Farràs
Universitat Rovira i Virgili,
Tarragona, Spain
oriol.farras@urv.cat

Or Lasri[†]
Ben-Gurion University of the Negev
Be'er-Sheva, Israel
orshlomo@post.bgu.ac.il

November 25, 2023

Abstract

Despite active research on secret-sharing schemes for arbitrary access structures for more than 35 years, we do not understand their share size – the best known upper bound for an arbitrary n-party access structure is $2^{O(n)}$, while the best known lower bound is $\Omega(n/\log(n))$. Consistent with our knowledge, the share size can be anywhere between these bounds. To better understand this question, one can study specific families of secret-sharing schemes. For example, linear secret-sharing schemes, in which the sharing and reconstruction are computed by linear mappings, have been studied in many papers, e.g., it is known that they require shares of size at least $2^{0.5n}$. Secret-sharing schemes in which the sharing and/or reconstruction are computed by low-degree polynomials have been recently studied by Paskin-Cherniavsky and Radune [ITC 2020] and by Beimel, Othman, and Peter [CRYPTO 2021]. It was shown that secret-sharing schemes with sharing and reconstruction computed by polynomials of degree 2 are more efficient than linear schemes (i.e., schemes in which the sharing and reconstruction are computed by polynomials of degree one).

Prior to our work, it was not known if using polynomials of higher degree can reduce the share size. We show that this is indeed the case, i.e., we construct secret-sharing schemes for arbitrary access structures with reconstruction by degree-d polynomials, where as the reconstruction degree d increases, the share size decreases. As a step in our construction, we construct conditional disclosure of secrets (CDS) protocols. For example, we construct 2-server CDS protocols for functions $f : [N] \times [N] \to \{0, 1\}$ with reconstruction computed by degree-d polynomials with message size $N^{O(\log\log d/\log d)}$. Combining our results with a lower bound of Beimel et al. [CRYPTO 2021], we show that increasing the degree of the reconstruction function in CDS protocols provably reduces the message size. To construct our schemes, we define *sparse* matching vectors, show constructions of such vectors, and design CDS protocols and secret-sharing schemes with degree-d reconstruction from sparse matching vectors.

[*]Supported by ERC grant 742754 (project NTSC) and ISF grant 391/21.
[†]Supported by ISF grant 391/21 and by the Frankel center for computer science.

1 Introduction

Secret sharing is a method by which a dealer holding a secret distributes shares to parties such that only predefined authorized subsets of parties can reconstruct the secret and unauthorized subsets should not learn any information about the secret. The collection of authorized sets is called the access structure. Originally, secret sharing was motivated by the problem of secure information storage; nowadays secret-sharing schemes have found numerous other applications in cryptography, distributed computing, and complexity theory (see, e.g., [9] for such applications). A major problem with secret-sharing schemes is that the best known schemes for general n-party access structures have shares of size $2^{O(n)}$ [33, 36, 4, 6], making the known constructions for general access structures impractical. On the other hand, the best known lower bound on the total share size of secret-sharing schemes realizing an arbitrary n-party access structure, proved by Csirmaz [18, 19], is $\Omega(\frac{n^2}{\log n})$. Despite active research on secret-sharing schemes for more than 35 years, determining the share size for arbitrary access structures is a major open problem.

To better understand this question, one can study specific families of secret-sharing schemes. Such study can shed light on general secret-sharing schemes, e.g., provide new techniques for constructing efficient secret-sharing schemes or proved new lower bound techniques. For example, linear secret-sharing schemes, in which the sharing and reconstruction are computed by linear mappings, have been studied in many papers [34, 16, 8, 41, 36, 4, 6], e.g., it is known that they require shares of size at least $2^{0.5n}$ [8] and every n-party access structure can be realized by a secret-sharing scheme with share size $2^{0.757n}$ [6]. Secret-sharing schemes in which the sharing and/or reconstruction are computed by low-degree polynomials have been recently studied by Paskin-Cherniavsky and Radune [40] and by Beimel, Othman, and Peter [14]. It was shown in [14] that every n-party access structure can be realized by a secret-sharing scheme with sharing and reconstruction computed by polynomials of degree 2 and share size $2^{0.705n}$, that is, secret-sharing schemes with degree-2 sharing and reconstruction are more efficient than the best known linear schemes (i.e., schemes in which the sharing and reconstruction are computed by polynomials of degree one). Prior to this work, it was not known if secret-sharing schemes with constant reconstruction degree $d > 2$ are more efficient than secret-sharing schemes with reconstruction degree 2.

In this paper we continue the study of polynomial secret-sharing schemes, i.e., schemes in which the reconstruction of the secret from the shares of an authorized set is done by polynomials of constant degree. Our main result in this paper is showing that the increasing the degree results in better share size, as described in the next theorem.

Theorem 1.1 (Informal)**.** *Every n-party access structure can be realized by a secret-sharing scheme with reconstruction by polynomials of degree d and share size $2^{(0.585+O\left(\frac{\log\log d}{\log d}\right))n}$.*

In particular, for an arbitrary access structure, we get a secret-sharing scheme with share size $2^{0.6731n+o(n)}$ and reconstruction degree 243. As $\lim_{d\to\infty} \frac{\log\log d}{\log d} = 0$, the share size approaches $2^{0.585n+o(n)}$, which is the share size of the best known secret-sharing scheme [14]. In comparison, Beimel et al. [14] constructed a degree-2 secret-sharing scheme with share size $2^{0.705n+o(n)}$, and Applebaum and Nir [6] constructed a linear secret-sharing scheme with share size $2^{0.7575n+o(n)}$.

Beimel and Farràs [10] proved that most access structures can be realized with secret-sharing schemes that are much more efficient than the best known schemes for the worst access structures. Beimel et al. [14] showed a similar result for schemes with reconstruction of degree 2. We generalize this result to arbitrary reconstruction degrees.

Theorem 1.2 (Informal)**.** *Almost all n-party access structures can be realized by a secret-sharing scheme with reconstruction by polynomials of degree d and 1-bit secrets and with share size $2^{O\left(\frac{\log\log d}{\log d}\right)n}$.*

	Linear	Degree-2	Degree-d	Unrestricted
Lower bound for the worst access structures	$\Omega(2^{n/2-o(n)})$ [8]	$\Omega(2^{n/3-o(n)})$ [14]	$\Omega(2^{n/(d+1)-o(n)})$ [14]	$\Omega(n^2/\log(n))$ [18]
Upper bound for all access structures	$2^{0.7576n+o(n)}$ [6]	$2^{0.705n+o(n)}$ [14]	$2^{(0.585+O\left(\frac{\log\log d}{\log d}\right))n+o(n)}$ (this paper, Corollary 7.6)	$2^{0.585n+o(n)}$ [6]
Upper bound for *almost* all access structures	$2^{n/2+o(n)}$ [10]	$2^{n/3+o(n)}$ [14]	$2^{O\left(\frac{\log\log d}{\log d}\right)n}$ (this paper, Corollary 7.9)	$2^{\tilde{O}(\sqrt{n})}$ [10]

Table 1: Summary of the best upper and lower bounds on the share size for secret-sharing schemes.

The previous results and our results on secret-sharing schemes with polynomial reconstruction are summarized in Table 1.

Conditional disclosure of secrets (CDS) protocols were introduced by Gertner, Ishai, Kushilevitz, and Malkin [29]. These protocols are an important tool in the recent constructions of secret-sharing schemes for arbitrary access structures [36, 3, 4, 6]. In a k-server CDS protocol for a Boolean function $f : [N]^k \to \{0,1\}$, there are k servers that hold a secret s and have a common random string. In addition, each server holds a private input $x_i \in [N]$. Each server sends one message to a referee such that the referee, who knows the private inputs of the servers but nothing more, learns the secret s if $f(x_1,\ldots,x_k) = 1$ and learns nothing otherwise. CDS protocols have been used recently in [36, 3, 4, 6, 14] to construct the best known secret-sharing schemes for arbitrary access structures. CDS protocols in which the reconstruction is done by polynomials of degree d have been studied in [27, 37] prior to the works on polynomial secret-sharing schemes. Continuing this line of research, we construct k-server CDS protocols that are provably more efficient as the degree of d of the reconstruction grows. We use them to construct secret-sharing schemes for arbitrary access structures with reconstruction by polynomials of degree d; these schemes are more efficient than the *best known* linear secret-sharing schemes. Specifically, we prove the following result.

Theorem 1.3 (Informal)**.** *For every $N > 0$, $d > 0, k > 1$, and function $f : [N]^k \to \{0,1\}$, there is a k-server CDS protocol for f, with degree-d reconstruction and communication complexity $N^{O\left((k-1)\cdot\frac{\log\log d}{\log d}\right)}$.*

For example, we prove that for any function $f : [N]^2 \to \{0,1\}$ there is a 2-server CDS protocol over $\mathbb{F}_7$ with communication complexity $O(N^{1/4})$ and reconstruction degree 243. In comparison, the best previously known 2-server CDS protocol with constant degree reconstruction has degree-2 reconstruction and communication complexity $O(N^{1/3})$ [14].

Theorem 1.3 is proved by constructing a CDS protocol for the function INDEX^k_N, where for every $D \in \{0,1\}^{N^{k-1}}$ (called the database) and every $(i_2,\ldots,i_k) \in [N]^{k-1}$ (called the index), $\mathrm{INDEX}^k_N(D, i_2,\ldots,i_k) = D_{i_2,\ldots,i_k}$. This strategy was used by Liu et al. in [37]. The 2-server CDS protocol of [37] (and our 2-server CDS protocol) uses the ideas of the 2-server private information retrieval (PIR) protocol of Dvir and Gopi [21]. Our techniques imply 2-server PIR protocols over $\mathbb{Z}_m$, for $m = p_1p_2$ where p_1, p_2 are primes

	Linear	Degree-2	Degree-d	Unrestricted
Lower bound for INDEX_N^2 and for the worst function $f : [N]^2 \to \{0,1\}$	$\Omega(N^{1/2})$ [27, 11]	$\Omega(N^{1/3})$ [14]	$\Omega(N^{1/(d+1)})$ [27, 14]	$\Omega(\log N)$ [2, 5, 7]
Upper bound for INDEX_N^2 and for all functions $f : [N]^2 \to \{0,1\}$	$O(N^{1/2})$ [12, 27]	$O(N^{1/3})$ [37]	$N^{O\left(\frac{\log\log d}{\log d}\right)}$ (this paper, Corollary 3.7)	$N^{O(\sqrt{\log\log N/\log N})}$ [37]
Lower bound for the worst function $f : [N]^k \to \{0,1\}$	$\Omega(N^{(k-1)/2})$ [11, 15]	$\Omega(N^{(k-1)/3})$ [14]	$\Omega(N^{(k-1)/(d+1)}/k)$ [14]	$\Omega(\log N)$ [2, 5, 7]
Upper bound for INDEX_N^k and for all functions $f : [N]^k \to \{0,1\}$	$O(N^{(k-1)/2})$ [37, 15]	$O(N^{(k-1)/3})$ [14]	$N^{O\left((k-1)\cdot\frac{\log\log d}{\log d}\right)}$ (this paper, Corollary 5.15)	$N^{O(\sqrt{k/\log N}\log\log N)}$ [39]

Table 2: Summary of previous and our results on the message size of CDS protocols.

and $p_1|p_2-1$, with communication complexity $N^{O_m(\sqrt{\log\log N/\log N})}$[1] (the protocol of [21] only works over $\mathbb{Z}_6$). Furthermore, we can construct 2-server PIR protocols, in which the answers of the servers can be computed by a degree-d polynomial, the reconstruction function is linear, and the communication complexity is $N^{O\left(\frac{\log\log d}{\log d}\right)}$.

By a lower bound of Beimel et al. [14], the message size of CDS protocols with degree-d reconstruction is $\Omega(N^{1/(d+1)})$. Thus, while the message size of our protocols does not match the lower bound, our results show that increasing the degree of the reconstruction in 2-server CDS protocols provably reduces the message size. The known and new results on the message size CDS protocols are described in Table 2.

1.1 Our Techniques

Our main result is a general construction of secret-sharing schemes for arbitrary access structures in which reconstruction is done by low degree polynomials. We construct it using the same steps as the constructions of the most efficient known secret-sharing schemes for arbitrary access structures. We start by constructing 2-server CDS protocols using matching vectors, following the footsteps of Liu, Vaikuntanathan, and Wee [37]. We use this 2-server CDS protocols to construct k-server CDS protocols using decomposable matching vectors, as in Liu et al. [39]. We then transform this CDS protocol into a robust k-server CDS protocol using the transformation of Applebaum, Beimel, Nir, and Peter [4] (with the better analysis of Beimel, Othoman, and Peter [14]), and finally use a transformation of [6] to construct secret-sharing schemes for arbitrary access structures. The technical contribution of this paper is in the first two steps. We show that if the matching vectors are sparse (i.e., the number non-zero of entries in them is small), then the degree of the

[1] The notation $O_m(\cdot)$ allows the constant in the O notation to depend on m.

reconstruction is low. We construct such matching vectors and show how to use them to construct 2-server and k-server CDS protocols with low-degree reconstruction, as explained below.

Matching vectors and CDS protocols. We start by recalling that a family of pairs of vectors $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$, where $\mathbf{u}_i, \mathbf{v}_i \subseteq \mathbb{Z}_m^h$, is a family S-matching vectors over $\mathbb{Z}_m$ if $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0 \pmod m$ for $i \in [N]$ and $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \mod m \in S$ for $i \neq j \in [N]$ (where $m = p_1 \cdot p_2$ is a product of two distinct primes $p_1 < p_2$, $S \subseteq \mathbb{Z}_m \setminus \{0\}$, and $\langle \mathbf{u}_i, \mathbf{v}_j \rangle$ is the inner product modulo m, i.e., $\sum_{\ell=1}^h \mathbf{u}_i[\ell] \cdot \mathbf{v}_j[\ell] \mod m$). Matching vectors were used by Efremenko [23] and Dvir and Gopi [21] to construct 3-server and 2-server private information retrieval (PIR) protocols, respectively. Liu et al. [37] used the ideas in [21] to construct 2-server CDS protocols. In [21, 37], they used matching vectors over $\mathbb{Z}_6$ are used. We generalize these constructions and show that one can use matching vectors over $\mathbb{Z}_{p_1 p_2}$, where p_1 and p_2 are primes such that p_1 divides $p_2 - 1$. Furthermore, we observe that one can use S-matching vectors for sets S that are larger than the ones used in previous constructions on PIR and CDS protocols, namely, one can take $S_{\text{one}} = \{a \in \mathbb{Z}_m : a \equiv 1 \pmod{p_1} \vee a \equiv 1 \pmod{p_2}\}$ instead of $S_{\text{can}} = \{a \in \mathbb{Z}_m : (a \equiv 0, 1 \pmod{p_1}) \wedge (a \equiv 0, 1 \pmod{p_1})\} \setminus \{0\}$, which was used in previous works. [2] E.g., over $\mathbb{Z}_{21}$ we can use $S_{\text{one}} = \{1, 4, 7, 8, 10, 13, 15, 16, 19\}$ instead of $S_{\text{can}} = \{1, 7, 15\}$. We use this observation to construct better CDS protocols with degree-d reconstruction. The construction of S_{one}-matching vectors that are shorter than the known S_{can}-matching vectors may lead to CDS protocols that are better than the currently best ones.

Sparse matching vectors. The most expensive part of computing the reconstruction function of the CDS protocol over $\mathbb{Z}_m$ (when considering the degree of the reconstruction) is computing $a^{\langle \mathbf{v}_i, \mathbf{m} \rangle} \mod p_2$, where a is an element of order p_1 in $\mathbb{F}_{p_2}$, $1 \leq i \leq N$ is an index, and $\mathbf{m}$ is a vector sent to the referee by the second server. Note that

$$a^{\langle \mathbf{v}_i, \mathbf{m} \rangle} \equiv \prod_{\ell=1}^{h} a^{\mathbf{v}_i[\ell] \cdot \mathbf{m}[\ell]} \pmod{p_2}, \tag{1}$$

where $\mathbf{v}_i, \mathbf{m} \in \mathbb{Z}_m^h$, and $\mathbf{v}_i[\ell], \mathbf{m}[\ell]$ are the ℓ-th coordinates of $\mathbf{v}_i$ and $\mathbf{m}$, respectively. If the server sends $a^{b \cdot \mathbf{m}[\ell]} \mod p_2$ for every $1 \leq \ell \leq h$ and every $b \in \mathbb{Z}_{p_1}$ (this only increases the communication complexity by a factor of p_1), then the referee can compute this value with a polynomial of degree h. In the best constructions of matching vectors, the length of the vectors h is $2^{\Theta(\sqrt{\log(N) \log\log(N)})}$ (over $\mathbb{Z}_6$). Thus, we get a CDS protocol with communication complexity and reconstruction degree $2^{\Theta(\sqrt{\log(N) \log\log(N)})}$.

The starting point of the construction with lower reconstruction degree is to recall that the order of a is p_1 and to write the product in (1) as

$$\prod_{\substack{\ell \in \{1,\ldots,h\}, \\ \mathbf{v}_i[\ell] \not\equiv 0 \pmod{p_1}}} a^{\mathbf{v}_i[\ell] \cdot \mathbf{m}[\ell]} \pmod{p_2}.$$

This implies that the degree of reconstruction is the number of coordinates in the matching vectors that are non-zero modulo p_1. To get a 2-server CDS protocol with degree-d reconstruction, we need a family of matching vectors in which each $\mathbf{v}_i$ contains at most d coordinates that are non-zero modulo p_1; we say that such family is a d-sparse family.

[2] In [21], they also have a construction that uses a $\mathbb{Z}_6 \setminus \{0\}$-matching vector family over $\mathbb{Z}_6$. It is unclear how to use this construction to improve the communication complexity of PIR and CDS protocols.

Constructions of sparse matching vectors. Our goal is to construct a family of N matching vectors over $\mathbb{Z}_{p_1 \cdot p_2}$ that are d-sparse with respect to p_1 and their length h is as short as possible. By the lower bound of [14] their length is at least $h = \Omega(N^{1/(d+1)})$ for a constant d. We present 3 constructions in which $h = d^{O(\frac{\log d}{\log \log d})^{4.18}} N^{O(\frac{\log \log d}{\log d})}$. The first construction is due to Efremenko [23]; the construction as described in [23, Appendix A] is sparse. In the second construction, we show how to improve Efremenko's construction. For concrete parameters, our construction achieves the smallest length h compared to the other 2 constructions. The downside of our construction is that they are S_{one}-matching vectors (compared to S_{can} in the other two constructions). S_{one}-matching vectors suffice for constructing 2-server private information retrieval (PIR) protocols [21], k- CDS protocols, and secret-sharing schemes for arbitrary access structures. However, they cannot be used in the 3-server PIR protocols of Efremenko [23]. The third construction we describe is a construction by Kutin [35]; in this case we need to decouple two of the parameters in the construction to achieve sparse matching vectors. The advantage of Kutin's construction compared to the other two constructions is that every m that is a product of two distinct primes (e.g., $m = 6$) can be used to achieve every sparsity d. In contrast, in Efremenko's construction and in our construction, to get smaller sparsity we need to use bigger m's. We remark that we can also use Grolmusz's construction of matching vectors [30] to construct sparse matching vectors (again by decoupling two parameters). This yields to a construction with similar features as Kutin's construction; we do not describe Grolmusz's construction in this paper.

We next describe the ideas of Efremenko's construction [23] and our improvement. Efremenko starts with a family of vectors $(\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_N)$ and $(\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_N)$ that are the characteristic vectors of N subsets in $\binom{[\tilde{h}]}{m-1}$. If $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{v}}_j$ are the characteristic vectors of A_i and A_j, respectively, then $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle = |A_i \cap A_j| \mod m$. Thus, $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle = m - 1$ and $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle \in \{0, \ldots, m-2\}$ for $i \neq j$. By adding a first coordinate that is 1 in all vectors, Efremenko constructs $\mathbb{Z}_m \setminus \{0\}$-matching vectors, where $\binom{\tilde{h}}{m-1} > N$ (since there must be at least N distinct subsets of size $m-1$). The sparsity of these vectors is m. To construct S_{can}-matching vectors, Efremenko uses the tensor product, Fermat's little theorem, and the Chinese reminder theorem (CRT). The length of the resulting vectors is $\tilde{h}^{p_2 - 1}$ and their sparsity with respect to p_1 is m^{p_1}. We modify this construction by starting with characteristic vectors of sets of size p_1^2 (since $p_1 < p_2$, this is smaller than in Efremenko's construction). We use Fermat's little theorem only with respect to p_1 and use a polynomial of degree p_1 to deal with the vectors modulo p_2. The length of the vectors in our construction is $\tilde{h}^{p_1}$, where $\tilde{h}$ is bigger than in Efremenko's construction; however, our construction yields vectors with roughly the same length as Efremenko's construction and smaller sparsity.

***k*-server CDS protocols with polynomial decoding.** We use 2-server CDS protocols to construct a k-server CDS protocols. Following [39], the first server in the k-server CDS protocol will simulate the first server in the 2-server CDS protocol and the last $k-1$ servers in the k-server CDS protocol will simulate the second server in the 2-server CDS protocol. In the simulation, the last $k-1$ servers need to send a message depending on their collective inputs, but each server only sees its input. As in [39], we use decomposable matching vectors to enable the simulation, that is, matching vectors such that every vector $\mathbf{u}_i$ can be computed from $k-1$ vectors $\mathbf{u}_{2,i_2}, \ldots, \mathbf{u}_{k,i_k}$, where each vector $\mathbf{u}_{t,i_t}$ can be computed from the input of the t-th server. To construct k-server CDS protocols with polynomial decoding using this approach, we have two challenges. First, we need to show that the constructions of sparse matching vectors are decomposable. This is done by changing the basic construction; instead of taking characteristic vectors of arbitrary sets of size $m-1$, we partition the universe into $m-1$ parts (i.e., subsets) and take sets of size $m-1$ that contain exactly one party from each part. The second challenge is to implement the simulation of the second server's message in the 2-server CDS protocol using a protocol in which the referee reconstructs the

message using a low-degree polynomial. Liu et al. [39] use a private simultaneous message (PSM) protocol of [32] for this task; however, it is not clear how to reconstruct the message with low-degree polynomials in this protocol. We design a special purpose protocol for this task exploiting the fact that in CDS protocols the referee knows the inputs (but not the secret).

From k-server CDS protocols with polynomial decoding to secret-sharing with polynomial reconstruction. We transform our k-server CDS protocol into a robust k-server CDS protocol using the transformation of Applebaum, Beimel, Nir, and Peter [4] (using the better analysis of Beimel, Othoman, and Peter [14]). In a robust CDS protocol (abbreviated as RCDS protocol) for a function f, a server can send messages for more than one input using the same randomness. The security of the protocol should hold as long as the messages correspond to zero-inputs (i.e., inputs for which f evaluates to zero). We finally use a transformation of [6] from RCDS protocols to secret-sharing schemes for arbitrary access structures. The details of this transformation are similar to previous papers.

Summary of construction. The main conceptual contribution of this paper is defining sparse matching vectors and showing that they imply CDS protocols with polynomial reconstruction. Towards this good, we generalize the CDS protocol of [37] to work over arbitrary $m = p_1 \cdot p_2$ where p_1 and p_2 are primes such that p_1 divides $p_2 - 1$. We observe that in this case, we can use a more relaxed notion of matching vectors (i.e., S_{one}-matching vectors). Constructing S_{one}-matching vectors that are shorter than the known constructions of S_{can}-matching vectors will lead to better CDS protocols and secret-sharing schemes. Our most important technical contribution is constructing a new family of sparse matching vectors that for concrete parameters are shorter than the matching vectors of Efremenko [22], which are sparse and sparse generalizations of the constructions of Grolmusz [30] and Kutin [35]. Our contribution of secret-sharing schemes with polynomial reconstruction follows the steps of previous constructions [39, 4, 6]; however, in many steps we encountered technical difficulties and needed to change the constructions to enable polynomial reconstruction.

1.2 Previous Works

Secret-sharing schemes. Secret-sharing schemes were introduced by Shamir [42] and Blakley [17] for the threshold case, and by Ito, Saito, and Nishizeki [33] for the general case. Ito et al. presented two secret-sharing schemes with share size 2^n for every access structure. The best currently known secret-sharing schemes for general n-party access structure are highly inefficient with total share size of $2^{0.585n}$ [3, 4, 6, 36]. The best known lower bound for the total share size of a secret-sharing scheme is $\Omega(\frac{n^2}{\log n})$ [18, 19]; there is an exponential gap between the lower bound and the upper bound.

Polynomial secret-sharing schemes. Paskin-Cherniavsky and Radune [40] defined secret-sharing schemes with polynomial sharing; in these schemes the sharing is computed by constant degree polynomials (there are no restrictions on the reconstruction functions). They showed limitations of various sub-classes of secret-sharing schemes with polynomial sharing. Specifically, they showed that the subclass of schemes for which the sharing is linear in the randomness (and the secret can be with arbitrary degree) is equivalent to multi-linear schemes up to a multiplicative factor of $O(n)$ in the share size. This implies that schemes in this subclass cannot significantly reduce the known share size of multi-linear schemes. In addition, they showed that the subclass of schemes over finite fields with odd characteristic such that the degree of the randomness in the sharing function is exactly 2 or 0 in any monomial of the polynomial can efficiently realize only access structures whose all minimal authorized sets are singletons. They also studied the randomness complexity of

schemes with polynomial sharing and showed an exponential upper bound on the randomness complexity (as a function of the share size).[3] Beimel, Othman, and Peter [14] defined and studied secret-sharing schemes and CDS protocols with polynomial reconstruction and secret-sharing schemes with polynomial sharing and reconstruction. They constructed a k-server CDS protocols with degree 2 sharing and reconstruction with message size $O(N^{1/3})$ and proved a lower bound of $\Omega(N^{1/(d+1)})$ for every 2-server CDS protocol with degree-d reconstruction. They also prove that (under plausible assumptions) secret-sharing-schemes with polynomial sharing are more efficient than secret-sharing schemes with polynomial reconstruction.

Conditional disclosure of secrets protocols. CDS protocols were introduced by Gertner et al. [28]. 2-server CDS are equivalent to secret-sharing for forbidden graph access structures [43]. Beimel et al. [12] showed a construction of 2-server CDS protocols with communication complexity of $O(\sqrt{N})$. Later, Gay et al. [27] showed a construction of 2-server CDS protocol for INDEX_N^2 and for every function $f : [N] \times [N] \to \{0,1\}$ with linear reconstruction and communication complexity $O(\sqrt{N})$. They also proved a lower bound of $\Omega(N^{1/d+1})$ on the communication complexity of 2-server CDS protocols for INDEX_N^2 in which the reconstruction is computed by a degree-d polynomial. In particular, they proved that their linear 2-server CDS protocol for INDEX_N^2 with communication complexity $O(\sqrt{N})$ is optimal. Beimel et al. [11] proved a lower bound of $\Omega(\sqrt{N})$ on the communication complexity of CDS protocols with linear reconstruction for almost all functions $f : [N] \times [N] \to \{0,1\}$; Beimel et al. [14] generalized the lower bound on the communication complexity of 2-server CDS protocols with degree-d reconstruction for almost all functions $f : [N] \times [N] \to \{0,1\}$. Constructions and lower bounds for k-server CDS protocols appear in [29, 38, 11, 15, 14]; see Table 2.

Matching vectors. We next discuss the most relevant results on matching vectors. The study of matching vectors families dates back to the study of set systems with restricted intersections modulo an integer m, that is, a system of sets whose size modulo m is some number μ_0 and the sizes of the intersection of any two sets in the system modulo m is in some set L. Such system implies a family of matching vectors by taking the characteristic vectors of the sets in the system. Frankl and Wilson [26] initiated the study of this question and proved upper bounds on the size of such set systems when the moduli is a prime. Using matching vector terminology, they proved that for any prime p if there is an S-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_p^h$, then $N \leq \binom{h}{|S|}$. They asked if the same lower bounds applies to composite numbers. Frankl [25] showed that this is not true; his result implies that for every N there is an S_{one}-matching vectors family over $\mathbb{Z}_6$ with N vectors of length $h = O(N^{1/3})$ (where $N > \binom{h}{3}$). Grolmusz [30] showed that working over composite numbers can drastically reduce the length of the matching vectors, i.e., his result implies that there is an S_{one}-matching vectors family over $\mathbb{Z}_m$, where $m = p_1 p_2$ for two primes $p_1 \neq p_2$, with N vectors and length $h = 2^{O(p_2\sqrt{\log N \log\log N})}$. Kutin [35] showed that for every pair of primes $p_1 \neq p_2$ and for infinitely many values of N there are S_{one}-matching vectors families over $\mathbb{Z}_{p_1 p_2}$ of length $h = 2^{O(\sqrt{\log N \log\log N})}$ (notice that he removed the dependency of p_2 in the exponent). Efremenko [23] used matching vectors to construct locally decodable codes and 3-server private information retrieval protocols. He also provided another construction of S_{one}-matching vectors with length $h = 2^{O(\sqrt{\log N \log\log N})}$. Dvir, Gopalan, and Yekhanin [20] continued the study of matching vector codes, i.e., locally decodable codes based on matching vectors. Dvir and Gopi [21] used matching vectors to construct 2-server private information retrieval protocols and Liu, Vaikuntanathan, and Wee [37, 39] used them to construct CDS protocols.

[3] For linear and multi-linear schemes, there is a tight linear upper bound on the randomness complexity.

2 Preliminaries

In this section, we will present the definitions needed for this work. We will start with some notations, continue by defining secret-sharing schemes for general access structures, in particular secret-sharing with polynomial reconstruction. Afterward, we define conditional disclosure of secrets (CDS) protocols.

2.1 Notations

For a natural number $n \in \mathbb{N}$, we denote $[n] \triangleq \{1, \ldots, n\}$. We denote $\log$ the logarithmic function with base 2. For $\alpha \in [0,1]$, we denote the binary entropy of α by $H_2(\alpha)$, where $H_2(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ for $\alpha \in (0,1)$, and $H_2(0) = H_2(1) = 0$.

If a random variable x is distributed according to a probability distribution $\mathcal{D}$, we write $x \sim \mathcal{D}$. For a finite set R, we denote by $U(R)$ the uniform distribution over the elements of R.

For a set A and a positive integer k, we denote by $\binom{A}{k}$ the family of subsets of A of size k, i.e., $\{B \subseteq A : |B| = k\}$.

For an integer variable x, and some positive integer i, we define the polynomial $\binom{x}{i} = \frac{x \cdot \ldots \cdot (x-i+1)}{i!}$. This is a degree-$i$ polynomial which has coefficients in $\mathbb{Q}$.

If two integers a and b are congruent modulo m, we denote $a \equiv b \pmod{m}$. If a is the reduction of b modulo m, then we denote $a \leftarrow b \mod m$.

We use the $\widetilde{O}$ notation, called *soft-O*, as a variant of big *O* notation that ignores logarithmic factors, that is, $f(n) \in \widetilde{O}(g(n))$ if $f(n) \in O(g(n) \log^k g(n))$ for some constant k.

We next define 3 products of vectors that are used to construct matching vectors. We define the first two over the ring $\mathbb{Z}_m$ and the last product over a field $\mathbb{F}$ as this is the way that they are used in this paper.

Definition 2.1 (Pointise and dot 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. $(\mathbf{x} \odot \mathbf{y})[\ell] = \mathbf{x}[\ell] \cdot \mathbf{y}[\ell] \mod m$. The dot product (or inner product) of $\mathbf{x}$ and $\mathbf{y}$ is $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{\ell \in [h]} \mathbf{x}[\ell] \cdot \mathbf{y}[\ell] \mod m$.*

Definition 2.2 (Tensor product)**.** *Let $\mathbb{F}$ be a field, let N be an integer, and let $\mathbf{x}, \mathbf{y} \in \mathbb{F}^N$. The tensor product of $\mathbf{x}, \mathbf{y}$, denoted by $\mathbf{x} \otimes \mathbf{y} \in \mathbb{F}^{N^2}$, is defined by $(\mathbf{x} \otimes \mathbf{y})[i,j] \coloneqq \mathbf{x}[i] \cdot \mathbf{y}[j]$, (where we identify $[N^2]$ with $[N]^2$). Similarly, we define the ℓ-th tensor power $\mathbf{x}^{\otimes \ell} \in \mathbb{F}^{N^\ell}$ as $\mathbf{x}^{\otimes \ell} = \mathbf{x}^{\otimes \ell - 1} \otimes \mathbf{x}$, i.e.,*

$$\mathbf{x}^{\otimes \ell}[i_1, i_2, \ldots, i_\ell] \coloneqq \prod_{j=1}^{\ell} \mathbf{x}[i_j].$$

We will need the following inequalities in our constructions.

Claim 2.3. *For every $\alpha, x > 0$, if $\alpha = x \log x$, then*

$$\frac{\alpha}{\log \alpha} \leq x \leq 2 \cdot \frac{\alpha}{\log \alpha}.$$

Proof. Since $\alpha = x \log x$, $\log \alpha = \log x + \log \log x$. Therefore,

$$\frac{x}{2} = \frac{x \log x}{2 \log x} \leq \frac{\alpha}{\log \alpha} = \frac{x \log x}{\log x + \log \log x} \leq \frac{x \log x}{\log x} = x.$$

□

Theorem 2.4 (Chinese reminder theorem (CRT)). *Let $n_1, n_2, \ldots, n_k$ be pairwise relatively prime natural numbers, $N = n_1 n_2 \ldots n_k$, and $b_1, b_2, \ldots, b_k \in \mathbb{Z}$. Then there is a unique $x \in \mathbb{Z}_N$ such that $x \equiv b_i \pmod{n_i}$ for all $1 \leq i \leq k$.*

2.2 Access Structures and Secret-Sharing Schemes

The definitions in this section are mainly based on [9].

Definition 2.5 (Access structures). *Let $P = \{p_1, \ldots, p_n\}$ be a finite set of n parties. A collection $\mathcal{A} \subseteq 2^P$ is monotone if for every set $A \in \mathcal{A}$ and for every $C \subseteq P$ such that $A \subseteq C$ it must be that $C \in \mathcal{A}$. An access structure is a monotone collection $\mathcal{A} \subseteq 2^P \backslash \emptyset$. A set of parties is called authorized if it is in $\mathcal{A}$ and unauthorized otherwise.*

Definition 2.6 (Secret-sharing schemes — Syntax). *Let $P = \{p_1, \ldots, p_n\}$ be a set of n parties. A secret-sharing scheme with domain of secrets S, set of random strings R, and domain of shares $S_1, S_2, \cdots, S_n$ for the parties $p_1, p_2, \ldots, p_n$, is a mapping $\Pi : S \times R \to S_1 \times S_2 \times \cdots \times S_n$. We denote the shares by $\mathsf{sh}_1, \ldots, \mathsf{sh}_n$. For a set $A \subseteq P$, we denote $\Pi_A(s; r)$ as the restriction of Π to its A-entries (i.e., $(\mathsf{sh}_j)_{p_j \in A}$, the shares of the parties in A). We define the $size$ of the secret in Π as $\log |S|$, and the $share\ size$ of party p_j as $\log |S_j|$, the maximum share size as $\max_{1 \leq j \leq n} \log |S_j|$, and the total share size as $\sum_{j=1}^{n} \log |S_j|$.*

Informally, we consider a dealer that 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 $\Pi(s; r) = (\mathsf{sh}_1, \mathsf{sh}_2, \ldots, \mathsf{sh}_n)$, and privately communicating each share sh_j to party p_j.

Definition 2.7 (Secret-sharing schemes — Correctness and security). *A secret-sharing scheme Π with finite domain S, where $|S| \geq 2$, realizes an access structure $\mathcal{A}$ if the following two requirements hold:*

CORRECTNESS. *The secret s can be reconstructed by any authorized set of parties, that is, for any set $A \in \mathcal{A}$ (where $A = \{p_{i_1}, \ldots, p_{i_{|A|}}\}$), there exists a reconstruction function $\text{RECON}_A : S_{i_1} \times \cdots \times S_{i_{|A|}} \to S$ such that for every $s \in S$ and every $r \in R$*

$$\text{RECON}_A(\Pi_A(s; r)) = s.$$

SECURITY. *Every unauthorized set cannot learn anything about the secret from its shares (in the information theoretic sense). Formally, for any set $B \notin \mathcal{A}$, for every two secrets $s, s' \in S$, and for every possible vector of shares $(\mathsf{sh}_j)_{p_j \in B}$:*

$$\Pr_{r \sim U(R)}[\Pi_B(s; r) = (\mathsf{sh}_j)_{p_j \in B}] = \Pr_{r \sim U(R)}[\Pi_B(s', r) = (\mathsf{sh}_j)_{p_j \in B}].$$

All the secret-sharing schemes presented in this paper are with the domain of secrets $S = \{0, 1\}$, unless stated otherwise.

2.3 Conditional Disclosure of Secrets

Informally, in a CDS protocol there are k servers $Q_1, \ldots, Q_k$, each holding a private input x_i, the secret s, and a common random string r, and there is a referee holding $x_1, \ldots, x_k$. Each server Q_i sends the message $m_i = \text{ENC}(x_i, s; r)$ to the referee, and the referee can reconstruct s if and only if $f(x_1, \ldots, x_n) = 1$.

Definition 2.8 (Conditional disclosure of secrets (CDS) protocols)**.** *Let $f : X_1 \times \cdots \times X_k \to \{0,1\}$ be a k-input function. A k-*server CDS protocol *$\mathcal{P}$ for f, with domain of secrets S, domain of common random strings R, and finite message domains $M_1, \ldots, M_k$, consists of k encoding functions $\text{ENC}_1, \ldots, \text{ENC}_k$, where $\text{ENC}_i : X_i \times S \times R \to M_i$ for every $i \in [k]$. For an input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$, secret $s \in S$, and randomness $r \in R$, we let $\text{ENC}(x, s; r) = (\text{ENC}_1(x_1, s; r), \ldots, \text{ENC}_k(x_k, s; r))$. We say that $\mathcal{P}$ is a CDS protocol for f if it satisfies the following properties:*

CORRECTNESS. *There is a deterministic reconstruction function $\text{DEC} : X_1 \times \cdots \times X_k \times M_1 \times \cdots \times M_k \to S$ such that for every input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ for which $f(x_1, \ldots, x_k) = 1$, every secret $s \in S$, and every common random string $r \in R$, it holds that $\text{DEC}(x, \text{ENC}(x, s; r)) = s$.*

SECURITY. *For every input $x = (x_1, \ldots, x_k) \in X_1 \times \cdots \times X_k$ satisfying $f(x_1, \ldots, x_k) = 0$ and every pair of secrets $s, s' \in S$, the encodings $\text{ENC}(x, s; r)$ and $\text{ENC}(x, s'; r)$ are equally distributed, i.e., for every messages $m_1, \ldots, m_k$*

$$\Pr_{r \sim U(R)}[\text{ENC}(x, s; r) = (m_1, \ldots, m_k)] = \Pr_{r \sim U(R)}[\text{ENC}(x, s'; r) = (m_1, \ldots, m_k)],$$

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

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

In two-server CDS protocols, we sometimes refer to the servers as Alice and Bob (instead of Q_1 and Q_2, respectively) and to the referee as Charlie.

Similarly to secret-sharing schemes, all the CDS protocols presented in this paper are with domain of secrets $S = \{0,1\}$, unless stated otherwise.

Definition 2.9 (The predicate INDEX_N^k)**.** *We define the k-input function $\text{INDEX}_N^k : \{0,1\}^{N^{k-1}} \times [N]^{k-1} \to \{0,1\}$ where for every $D \in \{0,1\}^{N^{k-1}}$ (a $(k-1)$ dimensional array called the database) and every $(i_2, \ldots, i_k) \in [N]^{k-1}$ (called the index), $\text{INDEX}_N^k(D, i_2, \ldots, i_k) = D_{i_2, \ldots, i_k}$.*

Observation 2.10 ([27])**.** *If there is a k-server CDS protocol for INDEX_N^k with message size M, then for every $f : [N]^k \to \{0,1\}$ there is a k-server CDS protocol with message size M.*

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

2.4 Robust Conditional Disclosure of Secrets

In the definition of CDS protocols (Definition 2.8), if a server sends messages for different inputs with the same randomness, then the security is not guaranteed and the referee can possibly learn information on the secret. In [4], the notion of robust CDS (RCDS) protocols was presented; the motivation for this definition is constructing more efficient secret-sharing schemes for arbitrary access structures. In RCDS protocols, the security is guaranteed even if the referee receives messages of different inputs with the same randomness. Next, we define the notion of t-RCDS protocols.

Definition 2.11 (Zero sets)**.** *Let $f : X_1 \times X_2 \times \cdots \times X_k \to \{0,1\}$ be a k-input function. We say that a set of inputs $Z \subseteq X_1 \times X_2 \times \cdots \times X_k$ is a zero set of f if $f(x) = 0$ for every $x \in Z$. For sets $Z_1, \ldots, Z_k$, we denote $\text{ENC}(Z, s; r) = (\text{ENC}(x, s; r))_{x \in Z_1 \times \cdots \times Z_k}$.*

Definition 2.12 (t-RCDS protocols)**.** *Let $\mathcal{P}$ be a k-server CDS protocol for a k-input function $f : X_1 \times X_2 \times \cdots \times X_k \to \{0,1\}$ and $Z = Z_1 \times Z_2 \times \cdots \times Z_k \subseteq X_1 \times X_2 \times \cdots \times X_k$ be a zero set of f. We say that $\mathcal{P}$ is robust for the set Z if for every pair of secrets $s, s' \in S$, it holds that $\text{ENC}(Z, s; r)$ and $\text{ENC}(Z, s'; r)$ are identically distributed. For an integer t, we say that $\mathcal{P}$ is a t-RCDS protocol if it is robust for every zero set $Z_1 \times Z_2 \times \cdots \times Z_k$ such that $|Z_i| \leq t$ for every $i \in [k]$.*

2.5 Degree-*d* Secret Sharing and Degree-*d* CDS Protocols

We next quote the definition of [14] of secret-sharing with polynomial reconstruction and CDS with polynomial decoding.

Definition 2.13 (Degree of polynomial)**.** *The degree of a multivariate monomial is the sum of the degree of all its variables; the degree of a polynomial is the maximal degree of its monomials.*

Definition 2.14 (Degree-d mapping over $\mathbb{F}$)**.** *A function $f : \mathbb{F}^\ell \to \mathbb{F}^m$ can be computed by degree-d polynomials over $\mathbb{F}$ if there are m polynomials $Q_1, \ldots, Q_m : \mathbb{F}^\ell \to \mathbb{F}$ of degree at most d s.t. $f(x_1, \ldots, x_\ell) = (Q_1(x_1, \ldots, x_\ell), \ldots, Q_m(x_1, \ldots, x_\ell))$.*

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

Definition 2.15 (Secret-sharing schemes with degree-d reconstruction)**.** *We say that the scheme Π with domain of secrets S has a* degree-d reconstruction over a finite field $\mathbb{F}$ *if there are integers $\ell, \ell_1, \ldots, \ell_n$ such that $S \subseteq \mathbb{F}^\ell$ and $S_i = \mathbb{F}^{\ell_i}$ for every $i \in [N]$, and Recon_B, the reconstruction function of the secret, can be computed by degree-d polynomials over $\mathbb{F}$ for every $B \in \mathcal{A}$.*

Notice that in Definition 2.7, the polynomials in the reconstruction can depend on B.

Definition 2.16 (CDS Protocols with Degree-d Decoding)**.** *A CDS protocol $\mathcal{P}$ with domain of secrets S has a degree-d decoding over a finite field $\mathbb{F}$ if there are integers $\ell, \ell_1, \ldots, \ell_k \geq 1$ such that $S \subseteq \mathbb{F}^\ell$ and $M_i = \mathbb{F}^{\ell_i}$ for every $1 \leq \ell \leq k$, and for every inputs $x_1, \ldots, x_k$ the function $\text{DEC}_{x_1,\ldots,x_k} : \mathbb{F}^{\ell_1 + \cdots + \ell_k} \to S$ can be computed by degree-d polynomials over $\mathbb{F}$, where $\text{DEC}_{x_1,\ldots,x_k}(m_1, \ldots, m_k) = \text{DEC}(x_1, \ldots, x_k, m_1, \ldots, m_k)$.*

2.6 Matching Vectors

We next define matching vectors (MV), which are vectors whose inner product lies in a small set $S \cup \{0\}$. These vectors were used in [26, 30] to construct a family of sets whose intersection lies in a small set. They were used in [23] to construct efficient PIR protocols and in [37] to construct efficient CDS protocols.

Definition 2.17 (Matching vector family [23])**.** *Let $m, h > 0$ be integers, and let $S \subseteq \mathbb{Z}_m \setminus \{0\}$ be a set. The family of vectors $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$, where $\mathbf{u}_i, \mathbf{v}_i \in \mathbb{Z}_m^h$, is called S-matching vectors if:*

1. $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \mod m = 0$ *for $i \in [N]$, and*
2. $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \mod m \in S$ *for $i \neq j \in [N]$.*

Let $m = p_1 p_2$ for some primes p_1, p_2. In previous works, they mainly considered the set

$$S_{\text{can}} = \{a \in \mathbb{Z}_m : (a \bmod p_1 \in \{0,1\}) \wedge (a \bmod p_2 \in \{0,1\})\} \setminus \{0\}.$$

We consider a bigger set

$$S_{\text{one}} = \{a \in \mathbb{Z}_m : a \equiv 1 \pmod{p_1} \vee a \equiv 1 \pmod{p_2}\}.$$

E.g., for $m = 6$, $S_{\text{can}} = \{1, 3, 4\}$ and $S_{\text{one}} = \{1, 3, 4, 5\}$. Note that every S_{can}-matching vectors family is in particular a S_{one}-matching vectors family since $S_{\text{can}} \subseteq S_{\text{one}}$.

3 A Polynomial 2-Server CDS Protocol

In this section, we present a 2-server CDS protocol with degree-d decoding, for the INDEX_N^2 predicate, with $N^{O(\frac{\log\log d}{\log d})}$ communication. This CDS protocol is a generalization of the CDS protocol from [37], which is based on a PIR protocol presented in [21]. In [37], they use matching vector families over $m = 3 \cdot 2$; We generalize this protocol and use matching vector families over $m = p_1 p_2$, for primes p_1, p_2 such that $p_1 | p_2 - 1$. We will first present the protocol and prove its correctness and security. We will then define sparse matching vectors and show that if we use sparse matching vectors in the CDS protocols, then we get a CDS protocol with degree-d decoding. In Section 4, we will show how to construct sparse matching vectors.

3.1 The CDS Protocol over $m = p_1 p_2$

In Figure 1, we present the 2-server CDS protocol; in the protocol we use an element $a \in \mathbb{F}_{p_2}^*$ whose order is p_1, i.e. p_1 is the smallest positive integer such that $a^{p_1} \equiv 1 \pmod{p_2}$. An element of order p_1 exists if and only if $p_1 | p_2 - 1$. This generalizes the CDS protocol of [37], which uses matching vectors over $m = 2 \cdot 3$ and the element $a = -1$.

Theorem 3.1. *Let p_1, p_2 be two primes such that $p_1 | p_2 - 1$, and let $m = p_1 p_2$. Given an S_{one}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$, the protocol in Figure 1 is a 2-server CDS protocol over $\mathbb{F}_{p_2}$ for INDEX_N^2 with message size $h \cdot \log m$.*

Proof. For the correctness and the security, we need to make the following analysis of Charlie's reconstruction function in the protocol in Figure 1 (i.e. (2)):

$$\begin{aligned}
&\langle \mathbf{u}_i, \mathbf{m}_A^2 \rangle - m_A^1 - m_B^2 - C(\mathbf{m}_B^1) + \langle \mathbf{u}_i, V(\mathbf{m}_B^1) \rangle \\
&\quad \equiv \langle \mathbf{u}_i, \mathbf{r}_2 + ((1-a)s - 1)V(\mathbf{r}_1) \rangle - ((1-a)s - 1)C(\mathbf{r}_1) + r_3 - \langle \mathbf{u}_i, \mathbf{r}_2 \rangle - r_3 \\
&\qquad - C(s\mathbf{u}_i + \mathbf{r}_1) + \langle \mathbf{u}_i, V(s\mathbf{u}_i + \mathbf{r}_1) \rangle \\
&\quad \equiv \langle \mathbf{u}_i, \mathbf{r}_2 \rangle + ((1-a)s - 1)\langle \mathbf{u}_i, V(\mathbf{r}_1) \rangle - ((1-a)s - 1)C(\mathbf{r}_1) - \langle \mathbf{u}_i, \mathbf{r}_2 \rangle \\
&\qquad - C(s\mathbf{u}_i + \mathbf{r}_1) + \langle \mathbf{u}_i, V(s\mathbf{u}_i + \mathbf{r}_1) \rangle \\
&\quad \equiv ((1-a)s - 1)(\langle \mathbf{u}_i, V(\mathbf{r}_1) \rangle - C(\mathbf{r}_1)) + \langle \mathbf{u}_i, V(s\mathbf{u}_i + \mathbf{r}_1) \rangle - C(s\mathbf{u}_i + \mathbf{r}_1) \\
&\quad \equiv ((1-a)s - 1) \sum_j \left[(\langle \mathbf{u}_i, \mathbf{v}_j \rangle - 1) D_j a^{\langle \mathbf{r}_1, \mathbf{v}_j \rangle} \right] + \sum_j \left[(\langle \mathbf{u}_i, \mathbf{v}_j \rangle - 1) D_j a^{\langle s\mathbf{u}_i + \mathbf{r}_1, \mathbf{v}_j \rangle} \right] \\
&\quad \equiv \sum_j \left[(\langle \mathbf{u}_i, \mathbf{v}_j \rangle - 1)((1-a)s - 1 + a^{s\langle \mathbf{u}_i, \mathbf{v}_j \rangle}) D_j a^{\langle \mathbf{r}_1, \mathbf{v}_j \rangle} \right] \pmod{p_2}. \qquad (3)
\end{aligned}$$

We next analyze each term in the sum. Recall that for every $i \neq j$, either $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \equiv 1 \pmod{p_1}$ or $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \equiv 1 \pmod{p_2}$ (or both). Thus, for $i \neq j$,

A polynomial 2-server CDS protocol for INDEX_N^2

Public Knowledge: An S_{one}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ for $m = p_1p_2$ s.t. $p_1|p_2 - 1$, and $h \in \mathbb{N}$. An element $a \in \mathbb{F}_{p_2}^*$ of order p_1 in $\mathbb{F}_{p_2}^*$.
Alice's Input: $D \in \{0,1\}^N$.
Bob's Input: $i \in [N]$.
The secret: $s \in \{0,1\}$.
Shared Randomness: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$, $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h$, $r_3 \in \mathbb{F}_{p_2}$.

Define $C : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}$ as $C(\mathbf{b}) = \sum_{j=1}^N D_j a^{\langle \mathbf{b}, \mathbf{v}_j \rangle} \mod p_2$.
Define $V : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}^h$ as $V(\mathbf{b}) = \sum_{j=1}^N D_j \mathbf{v}_j a^{\langle \mathbf{b}, \mathbf{v}_j \rangle} \mod p_2$.

- Alice sends $m_A^1 \leftarrow ((1-a)s - 1)C(\mathbf{r}_1) - r_3 \in \mathbb{F}_{p_2}$ and $\mathbf{m}_A^2 \leftarrow \mathbf{r}_2 + ((1-a)s - 1)V(\mathbf{r}_1) \in \mathbb{F}_{p_2}^h$.
- Bob sends $\mathbf{m}_B^1 \leftarrow (s\mathbf{u}_i + \mathbf{r}_1 \bmod p_1) \in \mathbb{F}_{p_1}^h$ and $m_B^2 \leftarrow (\langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 \bmod p_2) \in \mathbb{F}_{p_2}$.
- Charlie outputs 1 if

$$\langle \mathbf{u}_i, \mathbf{m}_A^2 \rangle - m_A^1 - m_B^2 - C(\mathbf{m}_B^1) + \langle \mathbf{u}_i, V(\mathbf{m}_B^1) \rangle \neq 0, \tag{2}$$

  and 0 otherwise.

Figure 1: A polynomial CDS protocol using a matching vector family over $\mathbb{Z}_m$ where $m = p_1p_2$ for primes p_1, p_2 such that $p_1|p_2 - 1$.

- If $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \equiv 1 \pmod{p_2}$ then $(\langle \mathbf{u}_i, \mathbf{v}_j \rangle - 1) \equiv 0 \pmod{p_2}$, and the term is 0.
- If $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \equiv 1 \pmod{p_1}$, then $(1-a)s - 1 + a^{s\langle \mathbf{u}_i, \mathbf{v}_j \rangle} \equiv (1-a)s - 1 + a^s \pmod{p_2}$ (since the order of a modulo p_2 is p_1). This expression equals 0 by a case analysis: if $s = 0$, then $(1-a)s - 1 + a^s \equiv -1 + a^0 \equiv 0 \pmod{p_2}$, and if $s = 1$, the term $(1-a)s - 1 + a^s \equiv 1 - a - 1 + a \equiv 0 \pmod{p_2}$.

For $i = j$, $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0 \pmod{m}$, and, in particular $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0 \pmod{p_1}$, thus, the i-th term is $-(1-a)sD_i a^{\langle \mathbf{r}_1, \mathbf{v}_i \rangle}$. Therefore, expression (3) equals $-(1-a)D_i a^{\langle \mathbf{r}_1, \mathbf{v}_i \rangle} s$.

Correctness. From the analysis above, if $D_i = 1$, Charlie computes

$$-(1-a)a^{\langle \mathbf{r}_1, \mathbf{v}_j \rangle} s, \tag{4}$$

and outputs 1 if and only if it is not equal to 0 and otherwise 0. Since $-(1-a)a^{\langle \mathbf{r}_1, \mathbf{v}_j \rangle} \not\equiv 0 \pmod{p_2}$, Charlie outputs s.

Security. The security follows similarly to [37] using the following observations:

- The joint distribution of $\mathbf{m}_B^1, m_A^1, \mathbf{m}_A^2$ is uniformly distributed, since we are using $\mathbf{r}_1, \mathbf{r}_2, r_3$ as one-time pads.

- If $D_i = 0$, then from the reconstruction analysis, the sum in (3) is 0, i.e.,

$$\langle \mathbf{u}_i, \mathbf{m}_A^2 \rangle - m_A^1 - m_B^2 - C(\mathbf{m}_B^1) + \langle \mathbf{u}_i, V(\mathbf{m}_B^1) \rangle = 0$$

thus,

$$m_B^2 = \langle \mathbf{u}_i, \mathbf{m}_A^2 \rangle - m_A^1 - C(\mathbf{m}_B^1) + \langle \mathbf{u}_i, V(\mathbf{m}_B^1) \rangle.$$

Therefore, m_B^2 in independent of s and can be computed given $\mathbf{m}_B^1, m_A^1, \mathbf{m}_A^2, D, i$.

From the two observation, we conclude that when $D_i = 0$ we can simulate $m_A^1, \mathbf{m}_A^2, \mathbf{m}_B^1, m_B^2$ given D, i, i.e., the distribution of the messages is the same for the two values of the secret.

Communication complexity. Clearly the message sizes are at most $(h+1)\log p_2$ since $m_A^1 \in \mathbb{F}_{p_2}$, $\mathbf{m}_A^2 \in \mathbb{F}_{p_2}^h$, $\mathbf{m}_B^1 \in \mathbb{F}_{p_1}^h$, $m_B^2 \in \mathbb{F}_{p_2}$ and $p_1 < p_2$. □

3.2 Sparse Matching Vectors

In order to analyze the degree of the reconstruction function we will introduce a new definition regarding matching vector families. This new definition is one of our most important contributions in this paper.

Definition 3.2 ((d,p)-sparse matching vectors)**.** *Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ be an S-matching vector family over $\mathbb{Z}_m^h$ for some $m, h \in \mathbb{N}$. We say that $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ is a d-sparse S-matching vector family if for all $i \in [N]$,*

$$|\{\ell \in [h] : \mathbf{v}_i[\ell] \neq 0\}| \leq d,$$

i.e., the number of non-zero entries in $\mathbf{v}_i$ is at most d.

For a prime p such that $p|m$, we say that $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ is (d,p)-sparse if for all $i \in [N]$,

$$|\{\ell \in [h] : \mathbf{v}_i[\ell] \not\equiv 0 \pmod{p}\}| \leq d.$$

We could have defined the sparsity property to be over $\mathbf{u}_i$ as well, and our constructions in Section 4 would satisfy this stronger requirement. However, for the reconstruction degree, sparsity solely for the $\mathbf{v}_i$'s suffices.

Next, we use the definition above for the reconstruction degree analysis.

Lemma 3.3. *Let p_1, p_2 be two primes such that $p_1 | p_2 - 1$, and let $m = p_1 p_2$. Given a (d, p_1)-sparse S_{one}-matching vector family over $\mathbb{Z}_m^h$, the reconstruction of the CDS protocol in Figure 1 can be computed by a degree-$p_1 d$ polynomial over $\mathbb{F}_{p_2}$.*

Proof. Charlie's reconstruction function as defined in (2) in Figure 1 is linear except for computing $C(\mathbf{m}_B^1)$ and $V(\mathbf{m}_B^1)$, thus we show how to enable Charlie to compute $C(\mathbf{b})$ as a low degree polynomial by changing the message of Bob; the analysis for $V(\mathbf{b})$ is the same.

$$C(\mathbf{b}) = \sum_{j=1}^N D_j a^{\langle \mathbf{b}, \mathbf{v}_j \rangle} = \sum_{j=1}^N D_j a^{\sum_{\ell \in [h]} \mathbf{b}[\ell] \cdot \mathbf{v}_j[\ell]} \underset{(*)}{=} \sum_{j=1}^N D_j \prod_{\ell \in [h] : \mathbf{v}_j[\ell] \not\equiv 0 \pmod{p_1}} \left(a^{\mathbf{b}[\ell]}\right)^{\mathbf{v}_j[\ell]} \mod p_2.$$

The equality $(*)$ is correct since the order of a is p_1. Thus, instead of sending $\mathbf{m}_B^1$, Bob sends

$$\mathbf{m'}_B^1 = (a^{\gamma \cdot \mathbf{m}_B^1[\ell]})_{\ell \in [h], \gamma \in \mathbb{F}_{p_1}} \in \mathbb{F}_{p_2}^{h \cdot p_1}. \tag{5}$$

The function $C(\mathbf{m}_B^1)$ can be computed as a polynomial of degree $\max_j\{|\{\ell \in [h] : \mathbf{v}_j \not\equiv 0 \pmod{p_1}\}|\}$; since $((\mathbf{u}_i, \mathbf{v}_i))$ is (d, p_1)-sparse, the degree is at most d.

In addition, Charlie outputs 1 if the expression in (2) is non-zero. We use Fermat's little theorem to convert any non-zero value to 1. That is when $D_i = 1$, Charlie computes the expression in (2), multiplies it by the constant $-(a-1)^{-1}$ and by (4) and since $s \in \{0, 1\}$, outputs

$$\left((-(1-a))^{-1} \cdot (-(1-a))sa^{\langle \mathbf{r}_1, \mathbf{v}_i \rangle}\right)^{p_1} \equiv s \pmod{p_2}.$$

To conclude, the total construction degree as most $p_1 \cdot d$. Note, that the modification of the protocol changes the communication complexity by a factor of p_1. □

Now, we present two theorems that will be proven in the next section. These theorems state the existence of sparse matching vector families; combining them with the CDS protocol in Figure 1, we get CDS protocols with various trade-offs between the decoding degree and the communication complexity.

Theorem 3.4. *For every $N, d > 0$, there exists primes p_1, p_2 where $p_1 | p_2 - 1$, and $p_1 \leq \frac{2 \log d}{\log \log d}$ for which there is a (d, p_1)-sparse S_{one}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$, where $m = p_1 p_2$, and $h \leq 2d^{1+\frac{2}{\log \log d}} N^{\frac{16 \log \log d}{\log d}}$.*

Theorem 3.5. *For every $N, d > 0$, there is a $(d, 2)$-sparse S_{can}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_6^h$ with $h \leq d^{O(1)} N^{O(\frac{\log \log d}{\log d})}$.*

Combining the construction of the CDS protocol from Figure 1 and Theorem 3.5 or Theorem 3.4, we get the following theorem.

Theorem 3.6. *For every $N, d > 0$, there is a 2-server CDS protocol over $\mathbb{F}_3$ or over $\mathbb{F}_{p_2}$ for some prime $p_2 = \text{polylog}(d)$ for INDEX$_N^2$, with degree-d reconstruction and communication complexity $d^{O(1)} N^{O\left(\frac{\log \log d}{\log d}\right)}$.*

Proof. Let $N, d > 0$. Let $d' = d/2$, from Theorem 3.5, we get a $(d', 2)$-sparse S_{can}-matching vector family over $\mathbb{Z}_6^h$, and $h \leq d'^{O(1)} N^{O(\frac{\log \log d}{\log d})}$. Thus, from Figure 1 we get a 2-server CDS protocol for INDEX$_N^2$ over $\mathbb{F}_3$, with degree-$2d' = d$ reconstruction and communication complexity $d^{O(1)} N^{O\left(\frac{\log \log d}{\log d}\right)}$.

Let d' be the largest integer such that $2d' \cdot \log d' \leq d$. From Theorem 3.4 we get a (d', p_1)-sparse S_{one}-matching vector family over $\mathbb{Z}_m^h$, where $m = p_1 p_2$, $p_1 \leq \frac{2 \log d'}{\log \log d'} \leq 2 \log d'$, and $h \leq 2d'^{1+\frac{2}{\log \log d'}} N^{\frac{16 \log \log d'}{\log \log d'}}$. Let $\alpha = d' \log d'$, from Claim 2.3, $d' \leq \frac{2 \log \alpha}{\log \log \alpha}$, thus $d' \leq \frac{2 \log d - 1}{\log(\log d - 1)} = d^{o(1)}$. Therefore, $h \leq 2d^{1+\frac{2}{\log \log d}} N^{\frac{16 \log \log d^{o(1)}}{\log d^{o(1)}}} = d^{O(1)} N^{O(\frac{\log \log d}{\log d})}$. Thus, from the protocol in Figure 1 we get a 2-server CDS protocol for INDEX$_N^2$ over $\mathbb{F}_{p_2}$, with degree-$p_1 \cdot d'$ reconstruction and communication complexity $d^{O(1)} N^{O(\frac{\log \log d}{\log d})}$. Also $p_1 d' \leq 2d' \log d' \leq d$, thus the CDS protocol has degree-d reconstruction. □

Corollary 3.7. *For every constant $d > 0$, $N > 0$, and function $f : [N] \times [N] \to \{0, 1\}$, there is a 2-server CDS protocol for f, with degree-d reconstruction and communication complexity $d^{O(1)} N^{O\left(\frac{\log \log d}{\log d}\right)}$.*

4 Constructions of *d*-Sparse Matching Vector Families

In this section, we present three different constructions of (d, p_1)-sparse matching vector families over $\mathbb{Z}_m^h$ where $m = p_1 p_2$ and $h = d^{O(1)} N^{O(\frac{\log \log d}{\log d})}$. The main differences between the constructions are the constraints of choosing the primes p_1, p_2 as N grows.

4.1 Basic Tools

In the three constructions of matching vector families, we use a basic construction of an $\tilde{S}$-matching vectors family for a large set $\tilde{S}$. To avoid repetition, we will present it here, and use it in the construction with different choices of $\tilde{S}$.

Claim 4.1. *Let $N, t, w > 0$ be integers, where $0 < w < t$. There is a $(w+1)$-sparse $\tilde{S}$-matching vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_t^{\tilde{h}}$, for $\tilde{S} = \{t-w, \ldots, t-1\}$, and $\tilde{h} = \lceil N^{1/w} \rceil \cdot w + 1$.*

Proof. Partition $[\tilde{h}-1]$ to w sets of size $\frac{\tilde{h}-1}{w} = \lceil N^{1/w} \rceil$. Let $\{A\}_{i=1}^N$ be N subsets of $[\tilde{h}-1]$ of size w s.t. A_i contains exactly one element from each set in the partition (there are $\left(\frac{\tilde{h}-1}{w}\right)^w = \left(\lceil N^{1/w} \rceil\right)^w \geq N$ such sets). For each set A_i, let $\mathbf{u}_{A_i} \in \{0,1\}^{\tilde{h}-1}$ be its characteristic vector. We define the vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_t^{\tilde{h}}$ as $\tilde{\mathbf{u}}_i = (t-w, \mathbf{u}_{A_i})$, $\tilde{\mathbf{v}}_i = (1, \mathbf{u}_{A_i})$, that is, in $\tilde{\mathbf{u}}_i$, and $\tilde{\mathbf{v}}_i$ we add a coordinate to $\mathbf{u}_{A_i}$. We next argue that $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ is an $\tilde{S}$-matching vector where $\tilde{S} = \{t-w, \ldots, t-1\}$.

- For every $i \in [N]$, $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle \equiv t - w + \langle \mathbf{u}_{A_i}, \mathbf{u}_{A_i} \rangle = t - w + |A_i| \equiv 0 \pmod{t}$.
- For every $i \neq j \in [N]$, $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle = t - w + \langle \mathbf{u}_{A_i}, \mathbf{u}_{A_j} \rangle = t - w + |A_i \cap A_j| \in \tilde{S}$ (since $0 \leq |A_i \cap A_j| \leq w - 1$ as $A_i \neq A_j$).

The sparsity of $\mathbf{u}_{A_i}$ for every $i \in [N]$ is w (since $\mathbf{u}_{A_i}$ is a characteristic vector of a set of size w), thus the sparsity of $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{v}}_i$ is $w+1$. □

We will use the following lemma about tensor products (tensor products are defined in Definition 2.2).

Lemma 4.2. *For every $\mathbf{u}_1, \mathbf{u}_2, \mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}$,*

$$\langle \mathbf{u}_1 \otimes \mathbf{u}_2, \mathbf{v}_1 \otimes \mathbf{v}_2 \rangle = \langle \mathbf{u}_1, \mathbf{v}_1 \rangle \cdot \langle \mathbf{u}_2, \mathbf{v}_2 \rangle.$$

In particular, $\langle \mathbf{u}^{\otimes \ell}, \mathbf{v}^{\otimes \ell} \rangle = \langle \mathbf{u}, \mathbf{v} \rangle^\ell$.

Proof.

$$\begin{aligned}
\langle \mathbf{u}_1 \otimes \mathbf{u}_2, \mathbf{v}_1 \otimes \mathbf{v}_2 \rangle &= \sum_{i,j \in [N]} (\mathbf{u}_1 \otimes \mathbf{u}_2)[i,j] \cdot (\mathbf{v}_1 \otimes \mathbf{v}_2)[i,j] \\
&= \sum_{i,j \in [N]} \mathbf{u}_1[i]\mathbf{u}_2[j] \cdot \mathbf{v}_1[i]\mathbf{v}_2[j] \\
&= \left(\sum_{i \in [N]} \mathbf{u}_1[i] \cdot \mathbf{v}_1[i] \right) \left(\sum_{j \in [N]} \mathbf{u}_2[j] \cdot \mathbf{v}_2[j] \right) \\
&= \langle \mathbf{u}_1, \mathbf{v}_1 \rangle \cdot \langle \mathbf{u}_2, \mathbf{v}_2 \rangle.
\end{aligned}$$

□

4.2 Efremenko's Construction

The first matching vector family we present is the Efremenko's [23, Appendix A]. We observe that Efremenko's construction is sparse. This construction takes the basic construction from Claim 4.1 and uses Fermat's little theorem and the Chinese reminder theorem in order to construct an S_{can}-matching vector family.

Construction 4.3. *Let $p_1 < p_2$ be two primes, $m = p_1 p_2$, and $\tilde{S} = \{1, \ldots, m-1\}$. Let $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$, where $\tilde{h} = \lceil N^{1/(m-1)} \rceil \cdot (m-1) + 1$ be the $\tilde{S}$-matching vector family over $\mathbb{Z}_t^{\tilde{h}}$ of Claim 4.1 where $t = m$ (i.e., $w = m-1$).*

We define for every $i \in [N]$:

$$\mathbf{u}_{p_1,i} = \tilde{\mathbf{u}}_i^{\otimes(p_1-1)} \bmod p_1, \mathbf{u}_{p_2,i} = \tilde{\mathbf{u}}_i^{\otimes(p_2-1)} \bmod p_2,$$

$$\mathbf{v}_{p_1,i} = \tilde{\mathbf{v}}_i^{\otimes(p_1-1)} \bmod p_1, \mathbf{v}_{p_2,i} = \tilde{\mathbf{v}}_i^{\otimes(p_2-1)} \bmod p_2.$$

Construct $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$, where $h = \tilde{h}^{p_2}$ using the CRT per entry, where we pad $\mathbf{u}_{p_1,i}$ and $\mathbf{v}_{p_1,i}$ with zeros to be of length $\tilde{h}^{p_2}$, i.e., we define $\mathbf{u}_i[k] \in \mathbb{Z}_m$ for $k \in [\tilde{h}^{p_2}]$ as the unique element s.t.

- $\mathbf{u}_i[k] \equiv \mathbf{u}_{p_1,i}[k] \pmod{p_1}$,
- $\mathbf{u}_i[k] \equiv \mathbf{u}_{p_2,i}[k] \pmod{p_2}$.

that is

$$\mathbf{u}_i[k] = \left(\mathbf{u}_{p_1,i}[k] \cdot p_2(p_2^{-1} \mod p_1) + \mathbf{u}_{p_2,i}[k] \cdot p_1(p_1^{-1} \mod p_2)\right) \mod m.$$

We define $\mathbf{v}_i$ analogously using $\mathbf{v}_{p_1,i}$ and $\mathbf{v}_{p_2,i}$.

Efremenko [23] proves that $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ is an S_{can}-matching vector family (recall that $S_{\text{can}} = \{a \in \mathbb{Z}_m : a \mod p_1, a \mod p_2 \in \{0,1\}\} \setminus \{0\}$). For completeness, provide this proof.

Claim 4.4. *Let p_1, p_2 be two primes and let $m = p_1 p_2$. The family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ as constructed in Construction 4.3 is an S_{can}-matching vector family.*

Proof. Let $i, j \in [N]$. By Lemma 4.2, for $\ell \in \{1, 2\}$,

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle \equiv \langle \mathbf{u}_{p_\ell,i}, \mathbf{v}_{p_\ell,j} \rangle \equiv \langle \tilde{\mathbf{u}}_i^{\otimes p_\ell - 1}, \tilde{\mathbf{v}}_j^{\otimes p_\ell - 1} \rangle \equiv \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle^{p_\ell - 1} \pmod{p_\ell}.$$

Since $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle \equiv 0 \pmod{m}$ by Claim 4.1,

- $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0^{p_1-1} \equiv 0 \pmod{p_1}$,
- $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0^{p_2-1} \equiv 0 \pmod{p_2}$,

thus, $\langle \mathbf{u}_i, \mathbf{v}_i \rangle \equiv 0 \pmod{m}$.
Let $i \neq j \in [N]$. Since $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle \not\equiv 0 \pmod{m}$, then $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle \not\equiv 0 \pmod{p_\ell}$ for at least one $\ell \in \{1, 2\}$. Also $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \mod p_1 \in \{0, 1\}$, and $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \mod p_2 \in \{0, 1\}$, thus $\langle \mathbf{u}_i, \mathbf{v}_j \rangle \mod m \in S_{\text{can}}$. □

We now will analyze the sparsity of the matching vectors family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$. From Claim 4.1 the sparsity of $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ is m. Thus, the family is (d, p_1)-sparse where $d = m^{p_1-1}$, since the number of entries k where $\mathbf{v}_i[k] \not\equiv 0 \pmod{p_1}$, by the CRT, is the number of entries k where $\mathbf{v}_{p_1,i}[k] \not\equiv 0 \pmod{p_1}$, which is m^{p_1-1}. The same applies to $\mathbf{v}_i$.

For the CDS protocol provided in Figure 1, we need that $p_1 | p_2 - 1$. The next result assures that for every prime p_1 there is a fairly small prime p_2 such that $p_1 | p_2 - 1$.

Theorem 4.5 ([44])**.** *There exists a constant c such that for every integer $d \geq 2$ and every $a \in \mathbb{Z}$ relatively prime to d, there exists a prime $p < cd^{5.18}$ such that $p \equiv a \pmod{d}$.*

Using Theorem 4.5 for a prime p_1, we can take the least prime p_2 such that $p_2 \equiv 1 \pmod{p_1}$ and get that $p_2 \le cp_1^{5.18}$, thus $p_1^2 \le m \le cp_1^{6.18}$. Note that

$$\tilde{h} = (m-1)\lceil N^{1/(m-1)} \rceil + 1 \le 2mN^{1/(m-1)} \le 2cp_1^{6.18}N^{2/m}, \text{ and}$$

and

$$h = \tilde{h}^{p_2-1} \le \left(2cp_1^{6.18}N^{2/m}\right)^{p_2} \le \left(2cp_1^{6.18}\right)^{cp_1^{5.18}} N^{2/p_1}.$$

Since $d = m^{p_1}$,

$$\begin{aligned} p_1^{2p_1} = (p_1^2)^{p_1} \le d \le (cp_1^{6.18})^{p_1} \\ \Rightarrow 2p_1 \log p_1 \le \log d \le p_1 \log c + 6.18 p_1 \log p_1. \end{aligned}$$

We take p_1 as the smallest such that $p_1 > \cdot \frac{\log d}{\log\log d}$. From Bertrand's postulate (see, e.g., [1]) that states that for every integer $k > 0$ there is a prime p such that $k \le p \le 2k$, we take such $p_1 < 2 \cdot \frac{\log d}{\log\log d}$. Combining this with the upper bound on h and on p_1, we get

$$\begin{aligned} h &\le \left(2cp_1^{6.18}\right)^{cp_1^{5.18}} N^{2/p_1} \\ &\le \left(2c\left(\frac{2\log d}{\log\log d}\right)^{6.18}\right)^{c\left(2\frac{\log d}{\log\log d}\right)^{5.18}} \cdot N^{\frac{2\log\log d}{\log d}} \\ &\le O\left(\frac{\log d}{\log\log d}\right)^{O\left(\frac{\log d}{\log\log d}\right)^{5.18}} N^{O\left(\frac{\log\log d}{\log d}\right)} \\ &= d^{O\left(\frac{\log d}{\log\log d}\right)^{4.18}} N^{O\left(\frac{\log\log d}{\log d}\right)} \\ &= 2^{\mathrm{polylog}(d)} N^{O(\frac{\log\log d}{\log d})}. \end{aligned}$$

4.3 Our Construction

In this section, we will prove Theorem 3.4 by showing a construction of a matching vector family generalizing the construction in Section 4.2. The matching vector family in this section will be for a larger set S_{one}; in return, we will get a more efficient protocol and more freedom in choosing the pairs of primes p_1, p_2.

Construction 4.6. *Let p_1, p_2 be primes, and let $m = p_1p_2$. Let $0 < w < m$ be a weight that will be chosen later. We start with the basic matching vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_t^{\tilde{h}}$ from Claim 4.1 with $t = m$ and $\tilde{h} = w\lceil N^{1/w} \rceil + 1$. Define $\mathbf{u}_{p_1,i} = \tilde{\mathbf{u}}_i^{\otimes p_1-1}$, and $\mathbf{v}_{p_1,i} = \tilde{\mathbf{v}}_i^{\otimes p_1-1}$, thus for every $i, j \in [N]$*

$$\langle \mathbf{u}_{p_1,i}, \mathbf{v}_{p_1,j} \rangle \equiv \langle \tilde{\mathbf{u}}_i^{\otimes p_1-1}, \tilde{\mathbf{v}}_j^{\otimes p_1-1} \rangle \equiv \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle^{p_1-1} \equiv \mathbb{1}_{\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle \not\equiv 0 \pmod{p_1}} \pmod{p_1}.$$

Next, we define the set $A = \{a \in \{m-w, \ldots, m-1\} : a \equiv 0 \pmod{p_1}\}$. Since $m \equiv 0 \pmod{p_1}$, the size of A is $\lfloor \frac{w}{p_1} \rfloor$. We consider the polynomial $R : \mathbb{F}_{p_2} \to \mathbb{F}_{p_2}$ (of degree at most $\lfloor \frac{w}{p_1} \rfloor$) such that

1. $R(0) \equiv 0 \pmod{p_2}$,
2. $R(a) \equiv 1 \pmod{p_2}$ *for all $a \in A$,*

This polymatroid is equal to

$$R(x) = x\left(\sum_{a\in A}\prod_{b\in A, b\neq a}\frac{x-b}{a-b}\right).$$

Since $a, b \equiv 0 \pmod{p_1}$ and $0 < a, b < m$, then $a \not\equiv b \pmod{p_2}$, therefore the inverse of $a-b$ exists. Note that $\deg(R) = d_R = |A| = \lfloor \frac{w}{p_1} \rfloor$. Let $R(x) \equiv \sum_{k=1}^{d_R} a_k x^k \pmod{p_2}$ be the explicit representation of R (as $R(0) = 0$, its free coefficient is 0). Define $\mathbf{u}_{p_2,i} = \left(a_1\tilde{\mathbf{u}}^{\otimes 1}, \ldots, a_{d_R}\tilde{\mathbf{u}}^{\otimes d_R}\right)$ (that is, $\mathbf{u}_{p_2,i}$ is a concatenation of d_R vectors), and $\mathbf{v}_{p_2,j} = \left(\tilde{\mathbf{v}}^{\otimes 1}, \ldots, \tilde{\mathbf{v}}^{\otimes d_R}\right)$. By Lemma 4.2, for every $i, j \in [N]$

$$\langle \mathbf{u}_{p_2,i}, \mathbf{v}_{p_2,j}\rangle \equiv \sum_{k=1}^{d_R} a_k \langle \tilde{\mathbf{u}}_i^{\otimes k}, \tilde{\mathbf{v}}_j^{\otimes k}\rangle \equiv \sum_{k=1}^{d_R} a_k \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle^k \equiv R(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle) \pmod{p_2}.$$

We pad either $\mathbf{u}_{p_1,i}$ and $\mathbf{v}_{p_1,i}$ or $\mathbf{u}_{p_2,i}$ and $\mathbf{v}_{p_2,i}$ such that they will have the same length. We construct $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$, where $h = \tilde{h}^{\max\{\lfloor \frac{w}{p_1}\rfloor, p_1-1\}}$ using the CRT per entry, i.e., $\mathbf{u}_i[k]$ is the unique element in $\mathbb{Z}_m$ s.t.

- $\mathbf{u}_i[k] \equiv \mathbf{u}_{p_1,i}[k] \pmod{p_1}$,
- $\mathbf{u}_i[k] \equiv \mathbf{u}_{p_2,i}[k] \pmod{p_2}$;

we define $\mathbf{v}_i$ the same way with $\mathbf{v}_{p_1,i}$ and $\mathbf{v}_{p_2,i}$.

Claim 4.7. *Let p_1, p_2 be primes and let $m = p_1p_2$. The family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ as in Construction 4.6 with $\lfloor \frac{w}{p_1}\rfloor = p_1 - 1$ is a (d, p_1)-sparse S_{one}-matching vector family, such that $h \leq 4d \cdot N^{\frac{2\log\log d}{\log d}}$, and $\sqrt{2}p_1 \log\left(p_1/\sqrt{2}\right) \leq \log d \leq 2p_1 \log p_1$.*

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

- $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \equiv \langle \mathbf{u}_{p_1,i}, \mathbf{v}_{1,j}\rangle \equiv \mathbb{1}_{\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \not\equiv 0 \pmod{p_1}} \pmod{p_1}$,
- $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \equiv \langle \mathbf{u}_{p_2,i}, \mathbf{v}_{p_2,j}\rangle \equiv R(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle) \pmod{p_2}$.

Therefore,

- $\langle \mathbf{u}_i, \mathbf{v}_i\rangle \equiv \mathbb{1}_{\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i\rangle \not\equiv 0 \pmod{p_1}} \equiv 0 \pmod{p_1}$,
- $\langle \mathbf{u}_i, \mathbf{v}_i\rangle \equiv R(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i\rangle) \equiv R(0) \equiv 0 \pmod{p_2}$,

thus, $\langle \mathbf{u}_i, \mathbf{v}_i\rangle \equiv 0 \pmod{m}$. For every $i \neq j \in [N]$,

- If $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \not\equiv 0 \pmod{p_1}$, then $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \equiv \mathbb{1}_{\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \neq 1} \equiv 1 \pmod{p_1}$. By Claim 4.1, $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \equiv \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle^{p_1-1} \equiv 1 \pmod{p_1}$, and $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \in S_{\text{one}}$.
- Otherwise $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \equiv 0 \pmod{p_1}$, and, by Claim 4.1, $m - w \leq \langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \leq m-1$. Thus, $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle \in A$ and $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \equiv R(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j\rangle) \equiv 1 \pmod{p_2}$ by the definition of R, thus $\langle \mathbf{u}_i, \mathbf{v}_j\rangle \in S_{\text{one}}$.

Next, we analyze the p_1-sparsity of the matching vectors and their length h (as a function of N and d). Recall that $h = \tilde{h}^{\max\{\lfloor \frac{w}{p_1} \rfloor, p_1 - 1\}}$. We take w such that $\lfloor \frac{w}{p_1} \rfloor = p_1 - 1$, i.e., $p_1^2 - p_1 \leq w \leq p_1^2 - 1$. By Claim 4.1, the sparsity with respect to p_1 of $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ is $w + 1$. By the definition if $\mathbf{u}_i$ (respectively $\mathbf{v}_i$) the p_1-sparsity of $\mathbf{u}_i$ is the sparsity of $\mathbf{u}_{p_1,i}$ (resp. $\mathbf{v}_{p_1,i}$), i.e., $d = (w+1)^{p_1 - 1}$. Therefore,

$$\frac{1}{4} \cdot p_1^{2p_1 - 2} \leq p_1^{2(p_1-1)} \cdot \left(1 - \frac{1}{p_1}\right)^{p_1 - 1} \leq (p_1^2 - p_1)^{p_1 - 1} \leq d = (w+1)^{p_1 - 1} \leq (p_1^2)^{p_1 - 1} \leq p_1^{2p_1}. \quad (6)$$

This implies

$$(2p_1 - 2) \log p_1 - 2 \leq \log d \leq 2p_1 \log p_1.$$

Let $p_1 \log p_1 = \alpha$. Using Claim 2.3, we get

$$p_1 \geq \frac{\alpha}{\log \alpha} \geq \frac{\frac{\log d}{2}}{\log \frac{\log d}{2}} = \frac{\log d}{2 \log \log d - 2} \geq \frac{\log d}{2 \log \log d}. \quad (7)$$

On the other hand, for every $p_1 > 4$,

$$\log d \geq (2p_1 - 2) \log p_1 - 2 \geq p_1 \log p_1.$$

Thus,

$$p_1 \leq \frac{2\alpha}{\log \alpha} \leq \frac{2 \log d}{\log \log d}.$$

We conclude that for every $p_1 > 4$

$$\frac{\log d}{2 \log \log d} \leq p_1 \leq \frac{2 \log d}{\log \log d}. \quad (8)$$

Also,

$$\tilde{h} = \lceil N^{1/w} \rceil \cdot w + 1 \leq 2N^{1/w} \cdot w \leq 2(p_1^2 - 1) N^{\frac{1}{p_1(p_1-1)}}.$$

From the choice of w, $h = \tilde{h}^{p_1 - 1}$, therefore

$$h \leq 2^{p_1 - 1} (p_1^2 - 1)^{p_1 - 1} N^{\frac{1}{p_1}} \leq \frac{2^{2 \log d / \log \log d}}{2} p_1^{2p_1 - 2} N^{\frac{1}{p_1}} \leq 2d^{1 + \frac{2}{\log \log d}} \cdot N^{\frac{1}{p_1}}$$

(where the last inequality follows from (6)). So we get,

$$h \leq 2d^{1 + \frac{2}{\log \log d}} N^{\frac{2 \log \log d}{\log d}}. \qquad \square$$

Remark 4.8. We next consider a specific choice of parameters in Construction 4.6 and analyze the resulting properties of the matching vector family. Take $p_1 = 3$, $p_2 = 7$, and choose $w = 8$. Then the length of the resulting matching vectors from Construction 4.6 is $h = \tilde{h}^{\max\{\lfloor \frac{8}{3} \rfloor, 2\}}$, where $\tilde{h} = O(N^{1/8})$, i.e., the length is $O(N^{1/4})$. The sparsity is $(8+1)^2 = 81$. Using this matching vector family, the protocol in Figure 1 is a 2-server CDS protocol over $\mathbb{F}_7$, with reconstruction degree p_1 times the sparsity of the matching vectors, i.e. 243, and communication complexity $O(N^{1/4})$.

This 2-server CDS protocol has better communication complexity than the quadratic 2-server CDS protocol from [14] (whose communication complexity is $O(N^{1/3})$).

The following lemma proves Theorem 3.4.

Lemma 4.9. *For every $N, d > 0$, there exist primes p_1, p_2 where $p_1 | p_2 - 1$ such that the matching vectors from Construction 4.6 is a $(d, p_1) - sparse$ S_{one}-matching vector family over $\mathbb{Z}_m^h$, where $m = p_1 p_2$, and $h \leq 2d^{1+\frac{2}{\log\log d}} N^{\frac{16\log\log d}{\log d}}$.*

Proof. Let $N, d > 0$ be some natural numbers. Let p_1 be a prime such that

$$\frac{\log d}{4\log\log d} \leq p_1 \leq \frac{\log d}{2\log\log d}, \tag{9}$$

such prime exists from Bertrand's postulate [1]. Let p_2 be the smallest prime such that $p_1 | p_2 - 1$. Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ be that vector family over $\mathbb{Z}_m^h$ of Construction 4.6, where $m = p_1 p_2$. By Claim 4.7, $((\mathbf{u}_i, \mathbf{v}_i))$ is an S_{one}-matching vector family such that $h \leq 2d'^{1+\frac{2}{\log\log d'}} N^{\frac{2\log\log d'}{\log d'}}$ where d' is the p_1-sparsity of the matching vector family. By equation (8), $\frac{\log d'}{2\log\log d'} \leq p_1 \leq \frac{2\log d'}{\log\log d'}$. Therefore, by (8), (9),

$$\frac{\log d'}{2\log\log d'} \leq p_1 \leq \frac{\log d}{2\log\log d} \Rightarrow d' \leq d,$$

thus, in particular the matching vector family is d-sparse with respect to p_1. Also, by (8), (9),

$$\frac{\log d}{4\log\log d} \leq p_1 \leq \frac{2\log d'}{\log\log d'} \Rightarrow \frac{2\log\log d'}{\log d'} \leq \frac{16\log\log d}{\log d},$$

so, $h \leq 2d^{1+\frac{2}{\log\log d}} N^{\frac{16\log\log d}{\log d}}$. □

4.4 Kutin's Construction

In this section, we will prove Theorem 3.5 by presenting a variant of the construction of matching vector family of Kutin [35]. Let $p_1 < p_2$ be two primes, $m = p_1 p_2$, and $t = p_1^{e_1} p_2^{e_2}$ for some $e_1, e_2 > 0$. By Claim 4.1 there is an $\tilde{S}$-matching vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_t^{\tilde{h}}$, where $\tilde{S} = \{1, \ldots, t-1\}$, and $\tilde{h} = \lceil N^{1/(t-1)} \rceil \cdot (t-1) + 1$ (i.e., $w = t - 1$).

Next we define BBR polynomials, which will be used in the construction.

Theorem 4.10 ([35])**.** *Let $p_1 < p_2$ be two primes, $m = p_1 \cdot p_2$, and $t = p_1^{e_1} p_2^{e_2}$ for two positive integers e_1, e_2. There exists a polynomial $Q_{m,t}(x)$ over $\mathbb{Q}$ such that:*

1. *$Q_{m,t}(x) = \sum_{i=1}^{d_Q} b_i \binom{x}{i}$, where $b_i \in \mathbb{Z}_m$.*
2. *$Q_{m,t}(x) \equiv 0 \pmod m$ if and only if $x \equiv 0 \pmod t$.*
3. *$\deg Q_{m,t} = d_Q = \max\{p_1^{e_1}, p_2^{e_2}\} - 1$.*
4. *If $x \not\equiv 0 \pmod t$ then $Q_{m,t}(x) \mod m \in S_{\text{can}}$.*

Note that the coefficients of Q are not necessarily integers, and yet for every input x, it evaluates to an integer when x is an integer.

Example 4.11. *Let $p_1 = 2$, $p_2 = 3$, $e_1 = 3$, $e_2 = 2$. Then $m = 6, S_{\text{can}} = \{1, 3, 4\}$, and $t = 72$. Then,*

$$Q_{m,t}(x) = x + 5\binom{x}{2} + \binom{x}{3} + 5\binom{x}{4} + \binom{x}{5} + 5\binom{x}{6} + \binom{x}{7} + 2\binom{x}{8}.$$

(where the polynomials $\binom{x}{i}$ are defined in Section 2.1). The degree of $Q_{m,t}$ is 8; it can be checked that $Q_{m,t}(72) \equiv 0 \pmod 6$, for example $Q_{m,t}(2) \equiv 2 + 5\binom{2}{2} \equiv 1 \pmod 6$.

Construction 4.12. *Let $t = p_1^{e_1} p_2^{e_2}$, and let $Q_{m,t}$ be the polynomial from Theorem 4.10. We use Claim 4.1 with t and $w = t-1$. In this case, $\tilde{\mathbf{u}}_i = \tilde{\mathbf{v}}_i$, since by definition its first entry is $t - w = 1$, and $\tilde{\mathbf{u}}_i$ is a binary vector. Let $A_i \subseteq [\tilde{h}]$ be the subset defined by $\tilde{\mathbf{u}}_i$, i.e., $A_i = \{\ell \in [h] : \tilde{\mathbf{u}}_i[\ell] = 1\}$; as the sparsity of $\tilde{\mathbf{u}}_i$ is $w + 1 = t$, $|A_i| = t$. We define vectors $\mathbf{u}_i, \mathbf{v}_i$ of length $\sum_{i=1}^{d_Q} \binom{h}{i}$, where for every $\emptyset \neq S \subseteq [\tilde{h}]$ of size at most d_Q we have the following coordinate in the vectors*

- $\mathbf{u}_i[S] = b_{|S|} \cdot \mathbb{1}_{S \subseteq A_i}$,
- $\mathbf{v}_i[S] = \mathbb{1}_{S \subseteq A_i}$.

The sparsity of this family is the number of non-empty subsets of A_i of size at most d_Q, i.e., at most $\sum_{i=1}^{d_Q} \binom{t}{i}$.

Our construction is based on Kutin's construction [35]. Kutin uses $\tilde{h} = t^{1.5}$ in Claim 4.1, and gets shorter vectors of length $N^{O\left(\sqrt{\frac{\log\log N}{\log N}}\right)}$; however, his vectors are dense. We get longer vectors that are sparser.

Lemma 4.13. *Let $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$, $Q_{m,t}$, and $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ be as defined in Construction 4.12. Then, for all $i, j \in [N]$,*

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle = Q_{m,t}(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle).$$

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

$$\begin{aligned}
\langle \mathbf{u}_i, \mathbf{v}_j \rangle &= \sum_{i=1}^{d_Q} \sum_{S \in \binom{[h]}{i}} \mathbf{u}_i[S] \cdot \mathbf{v}_j[S] \\
&= \sum_{i=1}^{d_Q} \sum_{S \in \binom{[h]}{i}} b_{|S|} \cdot \mathbb{1}_{S \subseteq A_i} \cdot \mathbb{1}_{S \subseteq A_j} \\
&= \sum_{i=1}^{d_Q} b_i \sum_{S \in \binom{[h]}{i}} \mathbb{1}_{S \subseteq A_i \cap A_j} \\
&= \sum_{i=1}^{d_Q} b_i \binom{|A_i \cap A_j|}{i} \\
&= Q_{m,t}(|A_i \cap A_j|) = Q_{m,t}(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_j \rangle)
\end{aligned}$$

□

Claim 4.14. *For every two primes p_1, p_2 and an integer $e_1 > 0$, there exists an integer $e_2 > 0$ such that for $t = p_1^{e_1} p_2^{e_2}$, the family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ as defined in Construction 4.12 is a (d, p_1)-sparse S_{can}-matching vector family over $\mathbb{Z}_m^h$, such that $h \leq d^{O(p_1)} \cdot N^{\frac{2p_1 \log\log d}{\log d}}$, and $\sqrt{t/p_1} \leq \log d \leq \sqrt{p_1 t}$.*

Proof. Let $i \neq j \in [N]$. By Theorem 4.10, and Lemma 4.13

- $\langle \mathbf{u}_i, \mathbf{v}_i \rangle = Q_{m,t}(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle) \equiv 0 \pmod m$, since $\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle \equiv 0 \pmod t$.
- $\langle \mathbf{u}_i, \mathbf{v}_j \rangle = Q_{m,t}(\langle \tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i \rangle) = Q_{m,t}(a)$ for $a \in \{1, \ldots, t-1\}$, thus $Q(a) \in S_{\text{can}}$ (by the definition of Q).

Before, analyzing the p_1-sparsity of $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$, we show how to choose an exponent e_2 given e_1 and primes p_1, p_2 s.t. $p_1^{e_1}, p_2^{e_2} = \Theta(\sqrt{t})$ up to a factor of $\sqrt{p_1}$. For every $e_1 > 0$, let $\tilde{e}_2$ be the minimal integer s.t. ${p_1}^{e_1} \leq {p_2}^{\tilde{e}_2}$,

- If ${p_2}^{\tilde{e}_2} \leq {p_1}^{e_1+1}$, then we choose $e_2 = \tilde{e_2}$ and
  1. $t = {p_1}^{e_1}{p_2}^{e_2} \geq {p_1}^{-1}{p_2}^{e_2} \cdot {p_2}^{e_2} \quad \Rightarrow \quad {p_2}^{2e_2} \leq p_1 t \quad \Rightarrow \quad p_1^{e_1} \leq {p_2}^{e_2} \leq \sqrt{p_1 t}.$
  2. $t = p_1^{e_1}{p_2}^{e_2} \leq p_1^{2e_1+1} \quad \Rightarrow \quad {p_2}^{e_2} \geq p_1^{e_1} \geq \sqrt{\frac{t}{p_1}}.$
- Otherwise, $p_2^{\tilde{e}_2} > p_1^{e_1+1}$ and we choose $\tilde{e}_2 = e_2 - 1$, thus $p_1^{e_1-1} \leq {p_2}^{\tilde{e}_2 - 1} = p_2^{e_2} \leq p_1^{e_1}$ (since $\tilde{e}_2$ is the minimal s.t. $p_1^{e_1} \leq p_2^{\tilde{e}_2}$) and we get
  1. $t = p_1^{e_1}{p_2}^{e_2} \geq p_1^{e_1}p_1^{e_1-1} \quad \Rightarrow \quad p_1^{2e_1} \leq p_1 t \quad \Rightarrow \quad {p_2}^{e_2} \leq p_1^{e_1} \leq \sqrt{p_1 t}.$
  2. $t = p_1^{e_1}{p_2}^{e_2} \leq p_1 p_2^{e_2} \cdot {p_2}^{e_2} \Rightarrow p_1^{e_2} \geq {p_2}^{e_2} \geq \sqrt{\frac{t}{p_1}}.$

In conclusion, for every e_1 we can choose e_2 such that

$$\sqrt{\frac{t}{p_1}} \leq p_1^{e_1}, p_2^{e_2} \leq \sqrt{p_1 t}. \tag{10}$$

Thus, by the definition, the degree of $Q_{m,t}$, namely d_Q, is at most $\sqrt{p_1 t}$.

Then, $\tilde{h} = (t-1)\lceil N^{1/(t-1)} \rceil + 1 \leq 2(t-1)N^{1/(t-1)}$ and

$$\begin{aligned} h &= \sum_{i=1}^{d_Q} \binom{\tilde{h}}{i} \leq \sum_{i=1}^{\sqrt{p_1 t}} \binom{\tilde{h}}{i} \leq \left(\frac{e\tilde{h}}{\sqrt{p_1 t}}\right)^{\sqrt{p_1 t}} \\ &\leq \left(\frac{6t \cdot N^{1/(t-1)}}{\sqrt{p_1 t}}\right)^{\sqrt{p_1 t}} \leq \left(6\sqrt{t/p_1}\right)^{\sqrt{p_1 t}} \cdot N^{\sqrt{\frac{2p_1}{t}}}. \end{aligned}$$

Let d be the sparsity of $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$. We use the inequalities of Claim 2.3 to get,

$$\begin{aligned} d &= \sum_{i=1}^{d_Q} \binom{t}{i} \leq \sum_{i=1}^{\sqrt{p_1 t}} \binom{t}{i} \leq \left(\frac{et}{\sqrt{p_1 t}}\right)^{\sqrt{p_1 t}} \leq \left(e\sqrt{t/p_1}\right)^{\sqrt{p_1 t}}. \\ &\Rightarrow \log d \leq \sqrt{p_1 t}\log(e\sqrt{t/p_1}) \leq \sqrt{p_1 t}\log(\sqrt{p_1 t}) \Rightarrow \sqrt{t} \geq \frac{\log d}{\sqrt{p_1}\log\log d}. \\ d &\geq \left(\frac{t}{\sqrt{t/p_1}}\right)^{\sqrt{t/p_1}} = \left(\sqrt{p_1 t}\right)^{\sqrt{t/p_1}} \\ &\Rightarrow \log d \geq \sqrt{t/p_1}\log\sqrt{p_1 t} \Rightarrow \sqrt{t} \leq \frac{2\sqrt{p_1}\log d}{\log\log d}. \end{aligned} \tag{11}$$

Thus,

$$h \leq \left(\frac{12\log d}{\log\log d}\right)^{\left(\frac{2p_1 \log d}{\log\log d}\right)} \cdot N^{\frac{\sqrt{2}p_1 \log\log d}{\log d}} = d^{O(p_1)} \cdot N^{\frac{\sqrt{2}p_1 \log\log d}{\log d}}.$$

□

The following lemma proves Theorem 3.5.

Lemma 4.15. *For every $N, d > 0$, there exists integers e_1, e_2 such that the matching vectors from Construction 4.12 is a $(d, 2) - sparse$ S_{can}-matching vector family over $\mathbb{Z}_6^h$, and $h \leq d^{O(1)} N^{O(\frac{\log\log d}{\log d})}$.*

Proof. Let $N, d > 0$. We want to prove the existence of an S_{can}-matching vector family with vectors of length $d^{O(1)} N^{O(\frac{\log\log d}{\log d})}$. We choose $p_1 = 2, p_3 = 3$, $e_1 = \lfloor \log\left(\frac{\log d}{\log\log d}\right) \rfloor - 1$, and we choose e_2 such that equation (10) holds for $t = 2^{e_1} 3^{e_2}$ and we get ,$2^{e_1 - \frac{1}{2}} \leq \sqrt{t} \leq 2^{e_1 + \frac{1}{2}}$. Then,

$$\begin{aligned} &\log\left(\frac{\log d}{\log\log d}\right) - 2 \leq e_1 \leq \log\left(\frac{\log d}{\log\log d}\right) - 1 \\ \Rightarrow\ &\frac{\log d}{4\log\log d} \leq 2^{e_1} \leq \frac{\log d}{2\log\log d} \\ \Rightarrow\ &\frac{\log d}{4\sqrt{2}\log\log d} \leq \sqrt{t} \leq \frac{\log d}{\sqrt{2}\log\log d}. \end{aligned}$$

Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ be the vectors in $\mathbb{Z}_m^h$ as defined in Construction 4.12 given t, m. According to Claim 4.14, $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ is an S_{can}-matching vector family such that $h \leq d'^{O(1)} \cdot N^{\frac{2\sqrt{2}\log\log d'}{\log d'}}$, where d' is the p_1-sparsity of the matching vectors family. Also by equation (11), $\frac{\log d'}{\sqrt{2}\log\log d'} \leq \sqrt{t} \leq \frac{2\sqrt{2}\log d'}{\log\log d'}$ therefore

$$\frac{\log d'}{\sqrt{2}\log\log d'} \leq \frac{\log d}{\sqrt{2}\log\log d} \Rightarrow d' \leq d,$$

$$\frac{\log d}{4\sqrt{2}\log\log d} \leq \frac{2\sqrt{2}\log d'}{\log\log d'} \Rightarrow \frac{2\sqrt{2}\log\log d'}{\log d'} \leq \frac{32\cdot\sqrt{2}\log\log d}{\log d}.$$

Thus, $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ is $(d, 2)$-sparse, and has length $h \leq d'^{O(1)} N^{\frac{2\sqrt{2}\log\log d'}{\log d'}} \leq d^{O(1)} \cdot N^{\frac{32\cdot\sqrt{2}\log\log d}{\log d}}$. □

4.5 Comparison of the Three Constructions

We described three constructions of sparse matching vectors. These constructions have the same asymptotic behavior: For every d there is a d-sparse matching vector family with vectors of length $O_d\left(N^{O(\frac{\log\log d}{\log d})}\right)$. The one based on Kutin's construction is the most interesting as the vectors can be over $\mathbb{Z}_m^h$ for $m = p_1 \cdot p_2$ for every two primes p_1, p_2, e.g., we can take $m = 2 \cdot 3$. In the construction we provide and Efremenko's construction, the value of m increases as d increases (e.g., as the length of the vectors decreases). Our construction yields shorter matching vectors that can be used to construct CDS protocols and secret-sharing schemes that are better than the degree-2 construction of [14]. Efremenko's and Kutin's constructions are for the S_{can}-matching vector family, whereas our construction yields only an S_{one}-matching vector family.

For every two primes p_1, p_2, and $m = p_1 p_2$, the length of Efremenko's matching vectors is $m^{p_2} N^{1/p_1}$, and are (m^{p_1}, p_1)-sparse. The length in our matching vector family is $p_1^{2p_1} \cdot N^{1/p_1}$ and they are $(p_1^{2p_1}, p_1)$-sparse. Thus, the sparsity and length in our matching vector family is much better since $p_2 > p_1$ and $p_1^{2p_1} < (p_1 \cdot p_2)^{p_1} = m^{p_1}$, and is independent of the choice of p_2. Recall that as we need that p_1 divides $p_2 - 1$, we only know that $p_2 \leq c \cdot p_1^{5.18}$.

5 A Polynomial k-Server CDS Protocol

In this section, we describe a construction of k-server CDS protocol for INDEX_N^k with polynomial reconstruction. Our protocol is a generalization of the k-server CDS protocol from [39]. It relies on two components. The first is a matching vector family with a special property of k-decomposability (see Definition 5.1). Thus, we need to prove that the constructions of sparse matching vectors are decomposable. The second is simulation of Bob in the 2-server CDS protocol by $k-1$ servers. We need to modify the simulation of [39] such that it can be computed by a linear function. Towards this goal, we describe a selection protocol that will be used as a black box in the k-server CDS protocol.

Recall that the point-wise product of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_m^h$, for some $m, h > 0$, is a vector in $\mathbb{Z}_m^h$ whose ℓ-th element is $\mathbf{x} \odot \mathbf{y}[\ell] = \mathbf{x}[\ell] \cdot \mathbf{y}[\ell] \bmod m$.

Definition 5.1 (k-decomposability)**.** *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 5.2 (Decomposable Matching Vector Families)**.** *For integers $N, m, h, k > 0$ and $S \subseteq \mathbb{Z}_m \setminus \{0\}$, a collection of vectors $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$ over $\mathbb{Z}_m^h$ is a k-decomposable S-matching vector family if it is an S-matching vector family, and $(\mathbf{u}_i)_{i=1}^N$, $(\mathbf{v}_i)_{i=1}^N$ are k-decomposable (as in Definition 5.1).*

We next give an example, in order to demonstrate the notion of decomposable matching vectors.

Example 5.3. *Let $N = 4$, $k = 2$, $w = 1$, and $i = (i_1, i_2) \in \{1, 2\}^2$. Let $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4 \in \mathbb{Z}_m^4$ be the standard unit vectors (i.e., $\mathbf{e}_1 = (1, 0, 0, 0)$). Let $\tilde{\mathbf{u}}_i = \tilde{\mathbf{v}}_i = (1, \mathbf{e}_i)$ for $1 \le i \le 4$ be the basic matching vector family from Claim 4.1. We will decompose $\mathbf{e}_i$ for every $i \in [4]$ as follows:*

$$(\tilde{\mathbf{u}}_{1,i})_{1 \le i \le 2} = ((1, 1, 1, 0, 0), (1, 0, 0, 1, 1)) \text{ and } (\tilde{\mathbf{u}}_{2,i})_{1 \le i \le 2} = ((1, 1, 0, 1, 0), (1, 0, 1, 0, 1)).$$

For example,

$$\tilde{\mathbf{u}}_{(1,1)} = (1, \mathbf{e}_1) = (1, 1, 0, 0, 0) = (1, 1, 1, 0, 0) \odot (1, 1, 0, 1, 0)$$

and

$$\tilde{\mathbf{u}}_{(2,1)} = (1, \mathbf{e}_3) = (1, 0, 0, 1, 0) = (1, 0, 0, 1, 1) \odot (1, 1, 0, 1, 0).$$

5.1 The Selection Protocol

In this section, we will describe an important component of our k-server CDS protocol. In our k-CDS protocol there will be k servers, the first server will simulate Alice in the 2-server CDS protocol described in Section 3, and the other $k-1$ servers will simulate Bob, i.e., each server Q_j for $2 \le j \le k$, holding an index i_{j-1}, sends a message such that the referee can reconstruct the messages of Bob with input $i = (i_1, \ldots, i_{k-1})$ in the 2-server CDS protocol. This should be done in such a way that the referee will not learn any additional information. Furthermore, the referee should reconstruct the message of Bob using a linear function. We will formulate these requirements as a special case of private simultaneous message (PSM) protocols [24, 31].

Definition 5.4 (PSM protocols)**.** *Let $\mathcal{X}_t$ be a t-th input space, and let $\mathcal{Y}$ be the output space. A private simultaneous messages (PSM) protocol $\mathcal{P}$, consists of:*

- *A finite domain $\mathcal{R}$ of common random inputs, and k finite message domains $\mathcal{M}_1, \ldots, \mathcal{M}_k$, denote $\mathcal{M} = \mathcal{M}_1 \times \cdots \times \mathcal{M}_k$.*
- *Message encoding algorithms $\text{ENC}_1, \ldots, \text{ENC}_k$, where $\text{ENC}_t : \mathcal{X}_t \times \mathcal{R} \to \mathcal{M}_t$.*
- *A decoding algorithm $\text{DEC} : \mathcal{M} \to \mathcal{Y}$.*

We say that a PSM protocol computes a k-argument function $f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_k \to \mathcal{Y}$, if it satisfies the following two properties:

Correctness. *For all $x_1 \in \mathcal{X}_1, \ldots, x_k \in \mathcal{X}_k$, and $r \in \mathcal{R}$:*

$$\text{DEC}(\text{ENC}_1(x_1; r), \ldots, \text{ENC}_k(x_k; r)) = f(x_1, \ldots, x_k),$$

that is, the referee always reconstructs the output of f.

Security. *For every $\mathbf{m} = (m_1, \ldots, m_k) \in \mathcal{M}_1 \times \cdots \times \mathcal{M}_k$, $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{x}' = (x'_1, \ldots, x'_k)$ in $\mathcal{X}$ satisfying $f(\mathbf{x}) = f(\mathbf{x}')$ it holds that*

$$\Pr_{r \sim U(R)}[(\text{ENC}_1(x_1; r), \ldots, \text{ENC}_k(x_k; r)) = \mathbf{m}] = \Pr_{r \sim U(R)}[(\text{ENC}_1(x'_1; r), \ldots, \text{ENC}_k(x'_k; r)) = \mathbf{m}],$$

that is, the referee cannot distinguish between two inputs with the same output, i.e., the referee only learns the output of f. The communication complexity of a PSM protocol is defined as $\log |\mathcal{M}|$.

Next, we define a function, simulating Bob's messages, and we design a PSM for it. In the function, we need the following selection function: each server holds an input $x_i \in \mathbb{F}_p$ and all servers hold a vector $\mathbf{s} = (s_0, \ldots, s_{p-1}) \in \mathbb{Z}_q^p$. The inputs of the servers define a selection index $b = \prod_{i=1}^k x_i \mod p$; the referee, which knows $x_1, \ldots, x_k$, should learn s_b without learning any additional information on $\mathbf{s}$.

Definition 5.5 (The selection function)**.** *Let q be a positive integer, p be a prime, and let $\mathbf{s} = (s_0, \ldots, s_{p-1}) \in \mathbb{Z}_q^p$ be a vector of length p. Let $\mathbb{Z}_q^p \times \mathbb{F}_p$ be the input space for each server; each server holds the common input $\mathbf{s}$, and a private input x_t. The* SELECTION *function is defined as follows*

$$f_{\text{SELECTION}}(\mathbf{s}, x_1, \ldots, x_k) = (s_b, x_1, \ldots, x_k)$$

where $b = \prod_{t=1}^k x_t \bmod p$.

In Protocol SELECTION, we assume that the referee knows $x_1, \ldots, x_k$; this is the case when we use it in a CDS protocol. For the purpose of analyzing Protocol SELECTION in Figure 2 as a PSM protocol (where the referee has no input), we assume that each server Q_j also sends x_j and the referee also outputs $x_1, \ldots, x_k$. Furthermore, in the definition of $f_{\text{SELECTION}}$ we assume that all servers have a common input $\mathbf{s}$. We can modify Protocol SELECTION in a way that only Q_k holds $\mathbf{s}$.

Claim 5.6. *Let p be a prime and let k, q be integers. The PSM $(\text{ENC}_1, \ldots, \text{ENC}_k, \text{DEC})$ described in Figure 2 is a PSM protocol for $f_{\text{SELECTION}}$ with communication complexity $(2p - 2) \log p$.*

Proof. Let $x_1, \ldots, x_k \in \mathbb{F}_p$ be the inputs for each server respectively, and denote $b = \prod_{i=1}^k x_i \mod p$.

Protocol SELECTION

Private input: The input of server Q_i is $x_i \in \mathbb{F}_p$.
Common input: A vector $\mathbf{s} = (s_0, \ldots, s_{p-1}) \in \mathbb{Z}_q^p$.
Shared Randomness: $\mathbf{r} = (r_{j,a})_{j \in [k-1], a \in \mathbb{F}_p^*}$ where $r_{j,a} \in \mathbb{Z}_q$. Let $r_{k,a} = s_a$ for $a \in \mathbb{F}_p^*$.

The message of server Q_1:
If $x_1 = 0$ sends $m_1 \leftarrow s_0$, otherwise sends $m_1 \leftarrow r_{1,x_1}$.

The message of server Q_j, for $2 \leq j \leq k$:
If $x_j = 0$ sends $m_j \leftarrow s_0$ otherwise sends $(m_{j,1}, \ldots, m_{j,p-1})$, where

$$m_{j,a} = (r_{j,a} - r_{j-1,a \cdot x_j^{-1} \mod p}) \mod q.$$

Referee:
Denote $b_1 = x_1, b_2 = x_1 \cdot x_2 \mod p, \ldots, b_k = \prod_{i=1}^k x_i \mod p = b$. The referee computes:

- If there exists j for which $x_j = 0$, then the referee outputs m_j.
- Otherwise, outputs $m_1 + \sum_{j=2}^k m_{j,b_j}$.

Figure 2: A PSM protocol for the SELECTION function.

Correctness. Since, for every $j \in [k]$, ENC_j sends x_i, and DEC outputs $x_1, \ldots, x_k$. It is left to show that the referee in Figure 2 outputs s_b. If there is $j \in [k]$ such that $x_j = 0$, then $b = 0$. In this case, server Q_j sends $m_j = s_0$, and the referee outputs m_j. Otherwise, denote $b_1 = x_1, b_2 = x_1 \cdot x_2 \mod p, \ldots, b_k = \prod_{i=1}^k x_i \mod p$; in particular $b_k = b$. We observe that for every $2 \leq j \leq k$, the messages m_{j,b_j} satisfy $m_{j,b_j} = r_{j,b_j} - r_{j-1,b_j \cdot x_j^{-1}} = r_{j,b_j} - r_{j-1,b_{j-1}}$, thus the referee outputs

$$m_1 + \sum_{j=2}^{k} m_{j,b_j} \equiv r_{1,x_1} + \sum_{j=2}^{k} (r_{j,b_j} - r_{j-1,b_{j-1}}) \equiv r_{k,b_k} \equiv s_b \pmod q.$$

Security. Let $\mathbf{s} = (s_1, \ldots, s_p), \mathbf{s}' = (s'_1, \ldots, s'_p) \in \mathbb{Z}_q^p$, and let $\mathbf{x}, \mathbf{x}' \in \mathbb{F}_p^k$ such that

$$f_{\text{SELECTION}}(\mathbf{s}, \mathbf{x}) = f_{\text{SELECTION}}(\mathbf{s}', \mathbf{x}').$$

From the definition of $f_{\text{SELECTION}}$, $\mathbf{x}' = \mathbf{x} = (x_1, \ldots, x_k)$, and $s_b = s'_b$ where $b = \prod_{t=1}^k x_t$. We denote $\mathbf{r} = (r_{j,a})_{j \in [k-1], a \in \mathbb{F}_p^*}$, and $\mathbf{r}' = (r'_{j,a})_{j \in [k-1], a \in \mathbb{F}_p^*}$ as random strings in the execution of $f_{\text{SELECTION}}$ with inputs x and x' respectively. We will describe a bijection between $\mathbf{r}$, and $\mathbf{r}'$ such that for every $t \in [k]$, $\text{ENC}_t(\mathbf{s}, x_t; \mathbf{r}) = \text{ENC}_t(\mathbf{s}', x_t; \mathbf{r}')$. Let $\mathbf{r} = (r_{j,a})_{j \in [k], a \in \mathbb{F}_p^*}$, where $r_{k,a} = s_a$, for every $a \in \mathbb{F}_p^*$. We define $\mathbf{r}'$ as follows. We define $r'_{k,a} = s'_a$ for every $a \in \mathbb{F}_p^*$. There are two cases:

$x_j = 0$ for some $j \in [k]$. In this case, $b = 0$. Let j_0 be the maximal index such that $x_{j_0} = 0$. We define recursively for every $a \in \mathbb{F}_p^*$, $j_0 \leq j \leq k-1$

$$r'_{j,a} = r_{j,a} + r_{j+1,a \cdot x_{j+1}} - r'_{j+1,a \cdot x_{j+1}},$$

and for every $1 \leq j < j_0$, we define $r'_{j,a} = r_{j,a}$. We show that the messages sent by the servers are the same with $(\mathbf{s}, \mathbf{r})$ and $(\mathbf{s}', \mathbf{r}')$. Then, for every $j \leq j_0$, such that $x_j = 0$,

$$\text{ENC}_j(\mathbf{s}, x_j; \mathbf{r}) = (s_0, x_j) = (s'_0, x_j) = \text{ENC}_j(\mathbf{s}', x_j; \mathbf{r}').$$

For $j = 1$, if $x_1 \neq 0$ then $j_0 > 1$ and

$$\text{ENC}_1(\mathbf{s}, x_1; \mathbf{r}) = (r_{1,x_1}, x_1) = (r'_{1,x_1}, x_1) = \text{ENC}_1(\mathbf{s}', x_1; \mathbf{r}').$$

For every $j < j_0$ such that $x_j \neq 0$, consider the message of Q_j,

1. $\text{ENC}_j(\mathbf{s}, x_j; \mathbf{r}) = (m_{j,1}, \ldots, m_{j,p-1}, x_j)$
2. $\text{ENC}_j(\mathbf{s}', x_j; \mathbf{r}') = (m'_{j,1}, \ldots, m'_{j,p-1}, x_j)$

We need to show that $m_{j,a} = m'_{j,a}$ for every $a \in \mathbb{F}^*$.

$$m'_{j,a} = r'_{j,a} - r'_{j-1,a \cdot x_j^{-1}} = r_{j,a} - r_{j-1,a \cdot x_j^{-1}} = m_{j,a}.$$

For $j_0 < j \leq k$, we again need to show that $m'_{j,a} = m_{j,a}$ for every $a \in \mathbb{F}_p^*$.

$$\begin{aligned} m'_{j,a} &= r'_{j,a} - r'_{j-1,a \cdot x_j^{-1}} \\ &= r'_{j,a} - (r_{j-1,a \cdot x_j^{-1}} + r_{j,a} - r'_{j,a}) \\ &= r_{j,a} - r_{j-1,a \cdot x_j^{-1}} = m_{k,a}. \end{aligned} \tag{12}$$

For every j, $x_j \neq 0$. For every $a \in \mathbb{F}_p^*$, $1 \leq j \leq k-1$ we define recursively,

$$r'_{j,a} = r_{j,a} + r_{j+1,a \cdot x_{j+1}} - r'_{j+1,a \cdot x_{j+1}}.$$

For servers $2 \leq j \leq k$, as before, we need to show that $m'_{j,a} = m_{j,a}$ for every $a \in \mathbb{F}_p^*$, which follows exactly as in equation (12). Thus, it is left to show that $r'_{1,x_1} = r_{1,x_1}$, and we will get that

$$\text{ENC}_1(\mathbf{s}, x_1; \mathbf{r}) = (r_{1,x_1}, x_1) = (r'_{1,x_1}, x_1) = \text{ENC}_1(\mathbf{s}', x_1; \mathbf{r}').$$

For every $j \in [k]$, denote $b_j = \prod_{i=1}^{j} x_i$. We prove, by induction on $1 \leq j \leq k$, that for every $a \in \mathbb{F}_p^*$, $r'_{j,b_j} = r_{j,b_j}$, and in particular $r'_{1,x_1} = r_{1,x_1}$.
For the base case, $j = k$, and $r'_{k,b_k} = s'_b = s_b = r_{k,b_k}$. For $j < k$, using an induction hypothesis that $r'_{j+1,b_{j+1}} = r_{j+1,b_{j+1}}$, we get

$$r'_{j,b_j} = r_{j,b_j} + r_{j+1,b_{j+1}} - r'_{j+1,b_{j+1}} = r_{j,b_j}. \qquad \square$$

5.2 Protocols for the Simulation of Bob's Messages

In this section, we present the polynomial k-server CDS protocol for INDEX_N^k (hence for every function $f : [N]^k \to \{0,1\}$). In this protocol, the first server Q_1 holds the database and the $k-1$ servers $Q_2, \ldots, Q_k$ collectively hold the index. In this protocol, the servers that hold the index will simulate Bob in the 2-server CDS protocol described in Figure 1, using the PSM protocol SELECTION from Section 5.1. Server Q_1 will simulate Alice.

In the CDS protocol of Figure 1, Bob sends $\mathbf{m}_B^1 = s\mathbf{u}_i + \mathbf{r}_1$. In the implementation of the protocol as a polynomial protocol, Bob sends $\mathbf{m'}_B^1 = (a^{\mathbf{m}_B^1[1]} \mod p_2, \ldots, a^{\mathbf{m}_B^1[h]} \mod p_2)$ where $a^{\mathbf{m}_B^1[\ell]} \equiv a^{s\mathbf{u}_i[\ell]+\mathbf{r}_1[\ell]} \pmod{p_2}$ (see (5)). Recall that we use decomposable matching vectors, so for $i = (i_1, \ldots, i_{k-1})$ we have $\mathbf{u}_i[\ell] \equiv \prod_{t\in[k-1]} \mathbf{u}_{t,i_t}[\ell] \pmod{m}$. In particular, $\mathbf{u}_i[\ell] = \prod_{t\in[k-1]} \mathbf{u}_{t,i_t}[\ell] \pmod{p_1}$. Thus, the ℓ-th coordinate of $\mathbf{m'}_B^1$ is

$$a^{\mathbf{m}_B^1[\ell]} \equiv a^{\left(s\prod_{t\in[k-1]} \mathbf{u}_{t,i_t}[\ell]\right)+\mathbf{r}_1[\ell] \mod p_1} \equiv a^{s\cdot b+\mathbf{r}_1[\ell]} \pmod{p_2}, \tag{13}$$

where $b = \prod_{t\in[k-1]} \mathbf{u}_{t,i_t}[\ell] \mod p_1$. Consider the vector $(a^{\mathbf{r}_1[\ell]}, a^{s+\mathbf{r}_1[\ell]}, \ldots, a^{(p_1-1)s+\mathbf{r}_1[\ell]})$; the referee should learn the $(b+1)$-th coordinate of this vector without learning any other information. Therefore, each coordinate of the vector can be sent by $Q_2, \ldots, Q_k$ using the selection protocol. A formal description of a protocol for this task appears in Figure 3.

In addition, Bob sends

$$m_B^2 \equiv \langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 \equiv \left(\sum_{\ell\in[h]} \mathbf{u}_i[\ell] \cdot \mathbf{r}_2[\ell]\right) + r_3 \equiv \left(\sum_{\ell\in[h]} \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \cdot r_2[\ell]\right) + r_3 \pmod{p_2}. \tag{14}$$

This can be done by executing ℓ copies of Protocol SELECTION and summing the results. As we only want to disclose the sum of the executions, we mask each with a random element such that the sum of the masks is zero. A formal description of a protocol for this task appears in Figure 4. We next describe the functionality computed by these protocols and prove their correctness and security.

Definition 5.7 (The function SEND_1)**.** *Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N^{k-1}}$ be a decomposable matching vector family. For $i = (i_1, \ldots, i_{k-1})$, where for every $t \in [k-1]$, $i_t \in [N^{1/(k-1)}]$ for every $t \in [k-1]$. Let $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$ be the server's common input and let $i \in [N]$. We define the PSM functionality* SEND_1 *as*

$$f_{\text{SEND}_1}(s, i, \mathbf{r}_1) = (i, (a^{s\mathbf{u}_i[\ell]+\mathbf{r}_1[\ell]} \mod p_2)_{\ell\in[h]}).$$

Notice that $\mathbf{r}_1$ is an input of f_{SEND_1}, thus a PSM protocol for this function should hide it (i.e., the referee should not distinguish between $s = 1, i, \mathbf{r}_1$ and $s = 0, i, \mathbf{r}'_1 = \mathbf{u}_i + \mathbf{r}_1$).

Protocol Send $\mathbf{m}_B^1$

Common input: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^n$.
Private input of Q_{t+1} for $1 \le t \le k-1$: $i_t \in [N^{1/(k-1)}]$.

- For $\ell = 1$ to h:
  - $Q_2, \ldots, Q_k$ execute protocol SELECTION, where the vector is $\mathbf{s}^\ell = (a^{s\cdot b+r_1[\ell]} \mod p_2)_{b\in\mathbb{F}_{p_1}}$ and the input of the server Q_{t+1} for $1 \le t \le k-1$ is $x_t^\ell = \mathbf{u}_{t,i_t}[\ell] \mod p_1$.
- The referee reconstructs:
$$\mathbf{m}_B^1 \leftarrow (s_{b^1}^1, \ldots, s_{b^h}^h)$$
where $b^\ell = \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \mod p_1$ for every $\ell \in [h]$.

Figure 3: A protocol simulating Bob's message $\mathbf{m}_B^1$.

Lemma 5.8. *Protocol **Send** $\mathbf{m}_B^1$ described in Figure 3 is a PSM protocol for the function f_{SEND_1}.*

Proof. We next prove the correctness and security of the protocol.

Correctness. Let $i = (i_1, \ldots, i_{k-1}) \in [N]$ and $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$. From the correctness of Protocol SELECTION (described in Figure 2), for every $\ell \in [h]$, the referee reconstructs $s_{b^\ell}^\ell \equiv a^{s \cdot b^\ell + r_1[\ell]} \pmod{p_2}$ where $b^\ell = \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \mod p_1$. Therefore, by (13),

$$\mathbf{m}_B^1 = (a^{s\mathbf{u}_i[\ell] + \mathbf{r}_1[\ell]} \mod p_2)_{\ell \in [h]} = (a^{\mathbf{m}_B^1[\ell]})_{\ell \in [h]}.$$

Security. Let $s, s', i, i' \in [N]$, and $\mathbf{r}_1, \mathbf{r}_1' \in \mathbb{F}_{p_1}^h$ such that $f_{\text{SEND}_1}(s, i, \mathbf{r}_1) = f_{\text{SEND}_1}(s', i', \mathbf{r}_1')$. By the definition of $f_{\text{SELECTION}_1}$, $i = i' = (i_1, \ldots, i_{k-1})$. Furthermore, $a^{s\mathbf{u}_i[\ell] + \mathbf{r}_1[\ell]} \equiv a^{s'\mathbf{u}_i[\ell] + \mathbf{r}_1'[\ell]} \pmod{p_2}$ for every $\ell \in [h]$. From the security of the protocol SELECTION, the messages sent in every iteration are equally distributed, and since in every iteration the SELECTION protocol is executed independently, the joint distribution of all the messages in all iterations is the same for $s, \mathbf{r}_1$ and $s', \mathbf{r}_1'$. □

Definition 5.9 (The function SEND_2)**.** *Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N^{k-1}}$ be a decomposable matching vector family. Let $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h, r_3 \in \mathbb{F}_{p_2}$ be the servers' common input, and let $i = (i_1, \ldots, i_{k-1})$, where $i_t \in [N^{1/(k-1)}]$ for every $t \in [k-1]$. We define function* SEND_2 *as*

$$f_{\text{SEND}_2}(i, \mathbf{r}_2, \mathbf{r}_3) = (i, \langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 \bmod p_2).$$

Protocol Send $\mathbf{m}_B^2$

Common input: $\mathbf{r_2} \in \mathbb{F}_{p_2}^h, r_3 \in \mathbb{F}_{p_2}$.
Private input of Q_{t+1} for $1 \le t \le k-1$: $i_t \in [N^{1/(k-1)}]$.
Shared randomness: $(r_3^\ell)_{\ell \in [h-1]}$. Define $r_3^h = r_3 - \sum_{\ell \in [h-1]} r_3^\ell \bmod p_2$.

- For $\ell = 1$ to h:
  - $Q_2, \ldots, Q_k$ execute Protocol SELECTION, where the vector is $\mathbf{s}^\ell = (b \cdot r_2[\ell] + r_3^\ell \mod p_2)_{b \in \mathbb{F}_{p_2}}$, and the input of Q_{t+1} for $1 \le t \le k-1$ is $x_t^\ell = \mathbf{u}_{t,i_t}[\ell] \mod p_2$.
- The referee reconstructs:
$$m_B^2 \leftarrow \sum_{\ell \in [h]} (b^\ell \mathbf{r}_2[\ell] + r_3^\ell) \bmod p_2$$
where for every $\ell \in [h]$, $b^\ell = \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \mod p_2$.

Figure 4: A protocol simulating Bob's message m_B^2.

Lemma 5.10. *Protocol **Send** $\mathbf{m_B^2}$ described in Figure 4 is a PSM protocol for the function f_{SEND_2}.*

Proof. We next prove the correctness and security of the protocol and analyze its communication complexity and reconstruction degree.

Correctness. Let $i = (i_1, \ldots, i_{k-1}), \mathbf{r}_2, r_3$ be an input, and let $(r_3^\ell)_{\ell \in [h]}$ be the randomness. By the correctness of the protocol SELECTION (described in Figure 2), for every iteration $\ell \in [h]$, the referee

reconstructs $s^{\ell}_{b^{\ell}} \equiv b^{\ell} \cdot r_2[\ell] + r_3^{\ell} \pmod{p_2}$, where $b^{\ell} = \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \mod p_2$. Therefore, from (14),

$$\begin{aligned} m_B^2 &\equiv \sum_{\ell \in [h]} (b^{\ell} \mathbf{r}_2[\ell] + r_3^{\ell}) \equiv \sum_{\ell \in [h]} \left(\prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] \cdot r_2[\ell] + r_3^{\ell} \right) \\ &\equiv \left(\sum_{\ell \in [h]} \mathbf{u}_i[\ell] \cdot r_2[\ell] \right) + r_3 \equiv \langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 \pmod{p_2} \end{aligned}$$

(since $r_3^h \equiv r_3 - \sum_{\ell \in [h-1]} r_3^{\ell} \pmod{p_2}$).

Security. Let $i, i' \in [N], \mathbf{r}_2, \mathbf{r}'_2 \in \mathbb{F}_{p_2}^h, r_3, r'_3 \in \mathbb{F}_{p_2}$ such that $f_{\text{SEND}_2}(i, \mathbf{r}_2, r_3) = f_{\text{SEND}_1}(i', \mathbf{r}'_2, r'_3)$. By the definition of $f_{\text{SELECTION}_2}$, $i = i' = (i_1, \ldots, i_{k-1})$, and $\langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 = \langle \mathbf{u}_i, \mathbf{r}'_2 \rangle + r'_3$. We first show that the outputs of the selection protocols are equally distributed for $i, \mathbf{r}_2, r_3$ and $i, \mathbf{r}'_2, r'_3$. Let $(r_3^{\ell})_{\ell \in [h-1]}$, and $r_3^h = r_3 - \sum_{\ell \in [h-1]} r_3^{\ell}$ be the randomness used in the execution of Protocol Send m_{B^2} with $(i, \mathbf{r}_2, r_3)$. We define, $r_3'^{\ell} = b^{\ell}(r_2[\ell] - r'_2[\ell]) + r_3^{\ell}$, for every $\ell \in [h-1]$ where $b^{\ell} = \prod_{t=1}^{k-1} \mathbf{u}_{t,i_t}[\ell] = \mathbf{u}_i[\ell]$. We also define $r_3'^h = r'_3 - \sum_{\ell \in [h-1]} r_3'^{\ell}$. Thus, we get for every $\ell \in [h-1]$

$$\begin{aligned} s'^{\ell}_{b^{\ell}} &= b^{\ell} \cdot r'_2[\ell] + r_3'^{\ell} \\ &= b^{\ell} \cdot r'_2[\ell] + b^{\ell}(r_2[\ell] - r'_2[\ell]) + r_3^{\ell} \\ &= b^{\ell} \cdot r_2[\ell] + r_3^{\ell} = s^{\ell}_{b^{\ell}}. \end{aligned}$$

Also, from the correctness

$$s'^h_{b_h} = \langle \mathbf{u}_i, \mathbf{r}'_2 \rangle + r'_3 - \sum_{\ell \in [h-1]} s'^{\ell}_{b^{\ell}} = \langle \mathbf{u}_i, \mathbf{r}_2 \rangle + r_3 - \sum_{\ell \in [h-1]} s^{\ell}_{b^{\ell}} = s^h_{b_h}.$$

Recall, that $s^{\ell}_{b^{\ell}}, s'^{\ell}_{b^{\ell}}$ are the outputs for the protocol SELECTION in iteration ℓ, for the inputs $\mathbf{r}_2, r_3$, and $\mathbf{r}'_2, r'_3$ respectively. From the security of the protocol SELECTION, the messages sent by the servers given the choice for $(r_3^{\ell})_{\ell \in [h]}$, $(r_3'^{\ell})_{\ell \in [h]}$ as above are distributed the same where the distribution is over the randomness of the SELECTION protocol. Since we have seen a bijection from $(r_3^{\ell})_{\ell \in [h]}$ to $(r_3'^{\ell})_{\ell \in [h]}$, the messages in the protocol are distributed the same for the inputs $\mathbf{r}_2, r_3$, and $\mathbf{r}'_2, r'_3$, where the randomness is over $(r_3^{\ell})_{\ell \in [h-1]}$, and the randomness of the SELECTION protocol. □

5.3 The *k*-Server CDS Protocol

In Figure 5, we describe the k-server CDS protocol for INDEX_N^k. This is an implementation of the 2-server CDS protocol from Figure 1, where the index i is distributed between $Q_2, \ldots, Q_k$ and they send Bob's messages using protocols **Send** $\mathbf{m}_B^1$ and **Send** m_B^2.

Theorem 5.11. *Let p_1, p_2 be primes such that $p_1 | p_2 - 1$, $m = p_1 \cdot p_2$ and $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N^{k-1}}$ be a decomposable (d, p_1)-sparse S_{one}-matching vector family over $\mathbb{Z}_m^h$. The protocol in Figure 5 is a k-server CDS protocol over $\mathbb{F}_{p_2}$ for INDEX_N^k with message size $h \cdot 2m \log m$ and reconstruction by polynomial of degree $d \cdot p_1$.*

Proof. We next prove the correctness and security of the protocol and analyze its communication complexity and reconstruction degree.

The Polynomial CDS Protocol for INDEX_N^k

Parameters: A decomposable S_{one}-matching vector family $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N^{k-1}}$ over $\mathbb{Z}_m^h$ for $m = p_1 p_2$ s.t. $p_1 | p_2 - 1$, and $h \in \mathbb{N}$, where for every $i \in [N^{k-1}]$ the decomposition of $\mathbf{u}_i$ is $\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_{k-1}}$, and an element $a \in \mathbb{F}_{p_2}^*$ of order p_1 in $\mathbb{F}_{p_2}^*$.
Input of Q_1**:** $D \in \{0,1\}^{N^{k-1}}$.
Inputs of $Q_2, \ldots, Q_k$**:** $i_1, \ldots, i_{k-1} \in [N]$.
The secret: $s \in \{0,1\}$.
Shared Randomness: $\mathbf{r}_1 \in \mathbb{F}_{p_1}^h$, $\mathbf{r}_2 \in \mathbb{F}_{p_2}^h$, $r_3 \in \mathbb{F}_{p_2}$, $\mathbf{r}_1^\ell \in \mathbb{F}_{p_1}^{(k-1)\cdot(p_1-1)}$, $\mathbf{r}_2^\ell \in \mathbb{F}_{p_2}^{(k-1)\cdot(p_2-1)}$ for every $\ell \in [h]$, and the randomness of the protocols **Send** $\mathbf{m}_B^1$ and **Send** m_B^2.

Define $C : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}$, as $C(\mathbf{b}) = \sum_{j=1}^N D_j a^{\langle \mathbf{b}, \mathbf{v}_j \rangle} \mod p_2$.
Define $V : \mathbb{F}_{p_1}^h \to \mathbb{F}_{p_2}^h$, as $V(\mathbf{b}) = \sum_{j=1}^N D_j \mathbf{v}_j a^{\langle \mathbf{b}, \mathbf{v}_j \rangle} \mod p_2$.

- Q_1 sends $m_A^1 \leftarrow ((1-a)s - 1)C(\mathbf{r}_1) - r_3 \in \mathbb{F}_{p_2}$ and $\mathbf{m}_A^2 \leftarrow \mathbf{r}_2 + ((1-a)s - 1)V(\mathbf{r}_1) \in \mathbb{F}_{p_2}^h$.
- Charlie and $Q_2, \ldots, Q_k$:
  - Execute protocol **Send** $\mathbf{m}_B^1$ described in Figure 3 with inputs $(s, i, \mathbf{r}_1)$.
  - Execute protocol **Send** m_B^2 described in Figure 4 with inputs $(i, \mathbf{r}_2, r_3)$.
- Charlie outputs 1 if

$$\langle \mathbf{u}_i, \mathbf{m}_A^2 \rangle - m_A^1 - m_B^2 - C(\mathbf{m}_B^1) + \langle \mathbf{u}_i, V(\mathbf{m}_B^1) \rangle \neq 0, \tag{15}$$

  and 0 otherwise.

Figure 5: A polynomial k-server CDS protocol using a decomposable matching vector family over $\mathbb{Z}_m$, where $m = p_1 p_2$ for primes p_1, p_2 such that $p_1 | p_2 - 1$.

Correctness. The correctness follows immediately from Lemma 5.8, Lemma 5.10, and from the correctness of the 2-server CDS protocol in Figure 1 (proven in Theorem 3.1).

Security. From the security of the protocols **Send** $\mathbf{m}_B^1$ and **Send** m_B^2 described in Figure 3, and Figure 4 respectively, Charlie can reconstruct $\mathbf{m}_B^1, m_B^2$, without learning any additional information about $s, \mathbf{r}_1, \mathbf{r}_2, r_3$. From the security of the 2-server CDS protocol described in Figure 1 (as probed in Theorem 3.1), Charlie does not learn any information about the secret s if $D_i = 0$, when he does not have any information about $\mathbf{r}_1, \mathbf{r}_2, r_3$.

Communication complexity. The message of Q_1 has length $O(h \log m)$. For $2 \leq t \leq k$, the message of Q_t is the sequence of the messages resulting from $2h$ executions of Protocol SELECTION. By Claim 5.6, each such message has length at most $m \log m$. Therefore, the total communication complexity of each server is at $O(h \cdot m \log m)$.

Reconstruction degree: The reconstruction function of the PSM protocol SELECTION described in Figure 2 is linear (i.e., has degree 1). Furthermore, the reconstruction function of Charlie as a function of

$m_A^1, \mathbf{m}_A^2, \mathbf{m}_B^1, m_B^2$ has degree $d \cdot p_1$ (by Lemma 3.3). Therefore, the degree of the reconstruction function in the k-server CDS protocol is $d \cdot p_1$. □

In Section 6, we will see how to decompose the matching vector families we have seen in Section 4, yielding a decomposable matching vector families as summarized in the next two theorems.

Theorem 5.12. *For every $N, d > 0$, there exist primes p_1, p_2 where $p_1 | p_2 - 1$, and $p_1 \leq \frac{2 \log d}{\log \log d}$ such that there is a decomposable (d, p_1)-sparse S_{one}-matching vector family over $\mathbb{Z}_m^h$ where $m = p_1 p_2$ and $h \leq 2d^{1+\frac{2}{\log \log d}} N^{\frac{2 \log \log d}{\log d}}$.*

Theorem 5.13. *For every $N, d > 0$, there is a decomposable $(d, 2)$-sparse S_{can}-matching vector family over $\mathbb{Z}_6^h$, where $h \leq d^{O(1)} N^{O(\frac{\log \log d}{\log d})}$.*

Combining Theorems 5.12 and 5.13 with Theorem 5.11, we get the following theorem, which can be proved similarly as Theorem 3.6

Theorem 5.14. *For every $N, d > 0$, and $k > 1$, there is a k-server CDS protocol over $\mathbb{F}_3$ or over $\mathbb{F}_{p_2}$ for some prime $p_2 = polylog(d)$ for INDEX$_N^k$, with degree-d reconstruction and communication complexity $d^{O(1)} N^{O\left((k-1) \cdot \frac{\log \log d}{\log d}\right)}$.*

Corollary 5.15. *For every $N, d > 0$, $k > 1$, and function $f : [N]^k \to \{0, 1\}$, there is a k-server CDS protocol for f, with degree-d reconstruction and communication complexity $d^{O(1)} N^{O\left((k-1) \cdot \frac{\log \log d}{\log d}\right)}$.*

Remark 5.16. Using the construction of the matching vector family over $\mathbb{Z}_{21}$ from Remark 4.8 (in the next section we will show that it is decomposable), we get a k-server CDS protocol over $\mathbb{F}_7$, with reconstruction degree 243, and communication complexity $O(N^{(k-1)/4})$. Previously, the best known k-server CDS protocol with polynomial reconstruction had communication complexity $O(N^{(k-1)/3})$ and degree 2 [14].

6 Construction of Decomposable Matching Vector Families

In this section, we show that the three construction we have seen in Section 4 are decomposable.

6.1 Decomposability of the Basic MV

First, we show a decomposition of the basic matching vector family in Claim 4.1. In Example 5.3, we have shown the decomposition for vectors of length 5. We first generalize Example 5.3 and show that the standard basis is decomposable.

Claim 6.1. *Let $N', m, \alpha, h' > 0$, and let $\mathbf{u} = (\mathbf{e}_{i_1, \ldots, i_\alpha})_{i_1, \ldots, i_\alpha \in [N']}$ be the standard basis of $\mathbb{Z}_m^{h'}$. Then there is a decomposition $\mathbf{u}^1, \ldots, \mathbf{u}^\alpha$ of $\mathbf{u}$.*

Proof. We define the following decomposition of standard basis vectors.

For every $j \in [\alpha]$, we define $\mathbf{u}^j = (\mathbf{y}_{j, \ell_1, \ldots, \ell_\alpha})_{\ell_1, \ldots, \ell_\alpha \in [N']}$:

$$\mathbf{y}_{j, i_1, \ldots, i_\alpha}[\ell_1, \ldots, \ell_\alpha] = \mathbb{1}_{\ell_j = i_j}.$$

Note that $\mathbf{y}_{j,i_1,\ldots,i_\alpha} \in \{0,1\}^{N'^\alpha}$. For every $i = (i_1,\ldots,i_\alpha)$ and a vector $\mathbf{e}_i$, we have that $\mathbf{e}_i[\ell] = 1$ if and only if $\ell = (\ell_1,\ldots,\ell_\alpha) = (i_1,\ldots,i_\alpha)$, i.e., $i_1 = \ell_1,\ldots,i_\alpha = \ell_\alpha$. Thus, for every $\ell_1,\ldots,\ell_\alpha \in [N']$:

$$\begin{aligned}\mathbf{y}_{1,i_1,\ldots,i_\alpha} \odot \cdots \odot \mathbf{y}_{\alpha,i_1,\ldots,i_\alpha}[\ell_1,\ldots,\ell_\alpha] &= \prod_{j\in[\alpha]} \mathbf{y}_{j,i_1,\ldots,i_\alpha}[\ell_1,\ldots,\ell_\alpha] \\ &= \prod_{j\in[\alpha]} \mathbb{1}_{\ell_j=i_j} = \mathbb{1}_{(\ell_1,\ldots,\ell_\alpha)=(i_1,\ldots,i_\alpha)} \\ &= \mathbf{e}_{i_1,\ldots,i_\alpha}[\ell_1,\ldots,\ell_\alpha].\end{aligned}$$

□

We next show how to decompose the basic construction of matching vectors assuming that w divides k. In Claim 6.3 we will remove this assumption.

Claim 6.2. *Let $N, m, w, k \geq 0$ where $0 \leq w \leq m$ and $k = \alpha w$ for some integer α. There is a decomposable $(w+1)$-sparse $\tilde{S}$-matching vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_m^{\tilde{h}}$ for $\tilde{S} = \{m-w,\ldots,m-1\}$, where $\tilde{h} = \lceil N^{1/w} \rceil \cdot w + 1$.*

Proof. We show that the basic matching vector from Claim 4.1 is decomposable. Denote $N' = N^{1/k}$. For $i \in [N]$, we denote $i = (i_{b,j})_{b\in[w],j\in[\alpha]}$, where $i_{b,j} \in [N']$. We define:

$$\mathbf{u}'_i = (\mathbf{e}_{i_{1,1},\ldots,i_{1,\alpha}}, \ldots, \mathbf{e}_{i_{w,1},\ldots,i_{w,\alpha}}),$$

i.e., $\mathbf{u}'_i$ is a concatenation of w standard basis vectors, each of length $N^{1/w} = N'^\alpha$. The vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ where $\tilde{\mathbf{u}}_i = (w, \mathbf{u}'_i), \tilde{\mathbf{v}}_i = (1, \mathbf{u}'_i)$ is the vector family from Claim 4.1; it is a $(w+1)$-sparse $\tilde{S}$-matching vector family. We need to prove that this matching vector family is decomposable. For every $\mathbf{e}_{i_1,\ldots,i_\alpha}$, let $\mathbf{y}_{1,i_1,\ldots,i_\alpha},\ldots,\mathbf{y}_{\alpha,i_1,\ldots,i_\alpha}$ be its decomposition from Claim 6.1.

Now, for every $t \in [k]$ we define $\mathbf{u}'_{t,i_t} = (\mathbf{u}'_{t,i_t,1},\ldots,\mathbf{u}'_{t,i_t,w})$ where for every $i = (i_1,\ldots,i_k)$, and $b \in [w]$:

$$\mathbf{u}'_{t,i_t,b} = \begin{cases} \mathbf{1}_{N^{1/w}} & \text{if } \lceil \frac{t}{\alpha} \rceil \neq b, \\ \mathbf{y}_{(t-1) \bmod \alpha+1,i_{b,1},\ldots,i_{b,\alpha}} & \text{if } \lceil \frac{t}{\alpha} \rceil = b. \end{cases}$$

That is, $\mathbf{u}'_{t,i_t}$ is a concatenation of w blocks, where the $\lceil \frac{t}{w} \rceil$'s block is a decomposition of the appropriate standard basis vector and the other blocks are all ones vectors. Note that for every $t \in [k]$, $b \in [w]$, $\mathbf{u}'_{t,i_t,b} \in \{0,1\}^{N^{1/w}}$, thus $\mathbf{u}'_{t,i_t} \in \{0,1\}^{\tilde{h}}$. Next we prove that for $i = (i_1,\ldots,i_k)$

$$\mathbf{u}'_i = \mathbf{u}'_{1,i_t} \odot \cdots \odot \mathbf{u}'_{k,i_t}. \tag{16}$$

Let $\ell \in [\tilde{h}]$, denote $b = \lceil \frac{\ell}{N^{1/w}} \rceil$, and $\ell' = \ell \mod N^{1/w}$. Then,

$$\begin{aligned}\prod_{t\in[k]} \mathbf{u}'_{t,i_t}[\ell] = \prod_{t\in[k]} \mathbf{u}'_{t,i_t,b}[\ell'] &= \prod_{t\in[k]:\lceil \frac{t}{\alpha} \rceil \neq b} \mathbf{u}'_{t,i_t,b}[\ell'] \cdot \prod_{t\in[k]:\lceil \frac{t}{\alpha} \rceil = b} \mathbf{u}'_{t,i_t,b}[\ell'] \\ &= \prod_{t\in[k]:\lceil \frac{t}{\alpha} \rceil \neq b} 1 \cdot \prod_{t\in[k]:\lceil \frac{t}{\alpha} \rceil = b} \mathbf{y}_{(t-1) \bmod \alpha+1,i_{b,1},\ldots,i_{b,\alpha}}[\ell'] \\ &= \prod_{j\in[\alpha]} \mathbf{y}_{j,i_{b,1},\ldots,i_{b,\alpha}}[\ell'] = \mathbf{e}_{i_{b,1},\ldots,i_{b,\alpha}}[\ell'] = \mathbf{u}'_i[\ell].\end{aligned}$$

Thus, by (16) for every $i \in [N]$

$$\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_{1,i_1} \odot \cdots \odot \tilde{\mathbf{u}}_{k,i_k}, \qquad \tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_{1,i_1} \odot \cdots \odot \tilde{\mathbf{v}}_{k,i_k},$$

where for every $t \in [k]$,

$$\tilde{\mathbf{v}}_{t,i_t} = (1, \mathbf{v}'_{t,i_t}), \quad \textit{and} \quad \tilde{\mathbf{u}}_{t,i_t} = \begin{cases} (1, \mathbf{u}'_{t,i_t}) & \textit{if } t < k, \\ (w, \mathbf{u}'_{t,i_t}) & \textit{if } t = k. \end{cases}$$

which concludes the proof. □

If w does not divide k, we decompose it into kw vectors, and get a decomposition to k vectors by composing w vectors into one vector.

Claim 6.3. *Let $N, m, w, k \geq 0$ where $0 \leq w \leq m$. There is a decomposable $(w+1)$-sparse $\tilde{S}$-matching vector family $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ over $\mathbb{Z}_m^{\tilde{h}}$ for $\tilde{S} = \{m-w, \ldots, m-1\}$, where $\tilde{h} = \lceil N^{1/w} \rceil \cdot w + 1$.*

Proof. Let $k' = kw$, $\tilde{h} = \lceil N^{1/w} \rceil \cdot w + 1$, and let $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ be a k'-decomposable, $(w+1)$-sparse, $\tilde{S}$-matching vector family over $\mathbb{Z}_m^{\tilde{h}}$ for $\tilde{S} = \{m-w, \ldots, m-1\}$ from Claim 6.2, where for every $i \in [N]$

$$\tilde{\mathbf{u}}_i = \tilde{\mathbf{u}}_{1,i_1} \odot \cdots \odot \tilde{\mathbf{u}}_{k',i_{k'}}, \qquad \tilde{\mathbf{v}}_i = \tilde{\mathbf{v}}_{1,i_1} \odot \cdots \odot \tilde{\mathbf{v}}_{k',i_{k'}}.$$

For every $t \in [k']$, we denote $(t_1, t_2) \in [k] \times [w]$, and for every $i \in [N], t_1 \in [k]$, we define

$$\tilde{\mathbf{u}}'_{t_1,i_{t_1}} = \tilde{\mathbf{u}}_{(t_1,1),i_{(t_1,1)}} \odot \cdots \odot \tilde{\mathbf{u}}_{(t_1,w),i_{(t_1,w)}},$$

and we get

$$\begin{aligned} \tilde{\mathbf{u}}_i &= \tilde{\mathbf{u}}_{1,i_1} \odot \cdots \odot \tilde{\mathbf{u}}_{k',i_{k'}} \\ &= \tilde{\mathbf{u}}_{(1,1),i_{(1,1)}} \odot \cdots \odot \tilde{\mathbf{u}}_{(1,w),i_{(1,w)}} \odot \cdots \odot \tilde{\mathbf{u}}_{(k,1),i_{(k,1)}} \odot \cdots \odot \tilde{\mathbf{u}}_{(k,w)i_{(k,w)}} \\ &= \tilde{\mathbf{u}}'_{1,i_1} \odot \cdots \odot \tilde{\mathbf{u}}'_{k,i_k}. \end{aligned}$$

Thus, we get that $(\tilde{\mathbf{u}}'_{1,i})_{i=1}^{\sqrt[k]{N}}, \ldots, (\tilde{\mathbf{u}}'_{k,i})_{i=1}^{\sqrt[k]{N}}$ is a k-decomposition of $(\tilde{\mathbf{u}}_i)_{i \in [N]}$. We k-decompose $(\tilde{\mathbf{v}}_i)_{i \in [N]}$ similarly, and we get that $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i \in [N]}$ is k-decomposable. □

6.2 Decomposability of Efremenko's and Our MVs

Now, we will show a decomposition for our matching vector family from Construction 4.6; since our construction uses the same techniques as in Construction 4.3, using polynomials and CRT per entry, the decomposition of our matching vector family will yield a decomposition of Efremenko's construction.

Construction 6.4. *Let $N, m, w, k \geq 0$ where $0 \leq w \leq m$, and $m = p_1 p_2$, for two primes $p_1 < p_2$. Let $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^N$ be the basic matching vector family over $\mathbb{Z}_m^{\tilde{h}}$ from Claim 4.1 For every $i = (i_1, \ldots, i_k) \in [N]$, let $(\tilde{\mathbf{u}}_{1,i_1}, \ldots, \tilde{\mathbf{u}}_{k,i_k})$, and $(\tilde{\mathbf{v}}_{1,i_1}, \ldots, \tilde{\mathbf{v}}_{k,i_k})$ be the decompositions for $\tilde{\mathbf{u}}_i$, and $\tilde{\mathbf{v}}_i$ respectively from Claim 6.3.*

Recall, in Construction 4.6, we defined $\mathbf{u}_{p_1,i} = \tilde{\mathbf{u}}_i^{\otimes p_1 - 1}$, and $\mathbf{u}_{p_2,i} = (a_1 \tilde{\mathbf{u}}_i^{\otimes 1}, \ldots, a_{d_R} \tilde{\mathbf{u}}_i^{\otimes d_R})$ for some polynomial $R(x) = \sum_{j=0}^{d_R} a_j x^j$; we defined $\mathbf{v}_{p_1,i}, \mathbf{v}_{p_2,i}$ similarly.
For every $t \in [k]$, we define

$$\mathbf{u}_{p_1,t,i_t} = \tilde{\mathbf{u}}_{t,i_t}^{\otimes p_1 - 1}, \qquad \mathbf{u}_{p_2,t,i_t} = \begin{cases} (\tilde{\mathbf{u}}_{t,i_t}^{\otimes 1}, \ldots, \tilde{\mathbf{u}}_{t,i_t}^{\otimes d_R}) & \textit{if } t < k, \\ (a_1 \tilde{\mathbf{u}}_{t,i_t}^{\otimes 1}, \ldots, a_{d_R} \tilde{\mathbf{u}}_{t,i_t}^{\otimes d_R}) & \textit{if } t = k. \end{cases}$$

Similarly, we define $\mathbf{v}_{p_1,t,i_t}, \mathbf{v}_{p_2,t,i_t}$.
Now, for every $i \in [N]$, $t \in [k]$, *we define* $\mathbf{u}_{t,i_t}$ *using the CRT per entry similarly to Construction 4.6, and as in the construction in Construction 4.6, (we pad with zeros all vectors of length less than* h*). For every* $\ell \in [h]$ *for* $h = \tilde{h}^{\max\{\lfloor \frac{w}{p_1} \rfloor, p_1 - 1\}}$, $\mathbf{u}_{t,i_t}[\ell]$ *is the unique element in* $\mathbb{Z}_m$ *satisfying*

- $\mathbf{u}_{t,i_t}[\ell] \equiv \mathbf{u}_{p_1,t,i_t}[\ell] \pmod{p_1}$, *and*
- $\mathbf{u}_{t,i_t}[\ell] \equiv \mathbf{u}_{p_2,t,i_2}[\ell] \pmod{p_2}$.

We define $\mathbf{v}_{t,i_t}$ *in the same way.*

Claim 6.5. *Let* $N, k, m, h > 0$, *and let* $(\mathbf{u}_i)_{i=1}^N$ *be a decomposable vector family over* $\mathbb{Z}_m^h$, *where for every* $i = (i_1, \ldots, i_k) \in [N]$ *the decomposition of* $\mathbf{u}_i$ *is* $(\mathbf{u}_{1,i_1}, \ldots, \mathbf{u}_{k,i_k})$. *Then the* r*-th tensor power operation preserves decomposability, i.e.*

$$\mathbf{u}_i^{\otimes r} = (\mathbf{u}_{1,i_1}^{\otimes r} \odot \cdots \odot \mathbf{u}_{k,i_k}^{\otimes r}).$$

Proof. Let $\ell = (\ell_1, \ldots, \ell_r) \in [h]$, then

$$\prod_{t \in [k]} \mathbf{u}_{t,i_t}^{\otimes r}[\ell] = \prod_{t \in [k]} \prod_{j \in [r]} \mathbf{u}_{t,i_t}[\ell_j] = \prod_{j \in [r]} \prod_{t \in [k]} \mathbf{u}_{t,i_t}[\ell_j] = \prod_{j \in [r]} \mathbf{u}_i[\ell_j] = \mathbf{u}_i^{\otimes r}[\ell].$$

□

The next proves Theorem 5.12.

Lemma 6.6. *Let* $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^N$, *be the matching vector family over* $\mathbb{Z}_m^h$ *from Construction 4.6. For every* $i \in [N]$, $(\mathbf{u}_{1,i_1}, \ldots, \mathbf{u}_{k,i_k})$ *and* $(\mathbf{v}_{1,i_1}, \ldots, \mathbf{v}_{k,i_k})$ *from Construction 6.4 are decomposition of* $\mathbf{u}_i$, $\mathbf{v}_i$, *respectively.*

Proof. For $i \in [N]$, let $\mathbf{u}_{p_1,i}, \mathbf{u}_{p_2,i}$, $\mathbf{v}_{p_1,i}, \mathbf{v}_{p_2,i}$ be the vectors from Construction 4.6. For every $t \in [k]$, let $\mathbf{u}_{p_1,t,i_t}, \mathbf{u}_{p_2,t,i_2}$, and $\mathbf{v}_{p_1,t,i_t}, \mathbf{v}_{p_2,t,i_2}$ be the vectors from Construction 6.4. Then, by Claim 6.5, $(\mathbf{u}_{p_1,1,i_1}, \ldots, \mathbf{u}_{p_1,k,i_k})$ and $(\mathbf{u}_{p_2,1,i_1}, \ldots, \mathbf{u}_{p_2,k,i_k})$ are decompositions of $\mathbf{u}_{p_1,i}, \mathbf{u}_{p_2,i}$, respectively, over $\mathbb{Z}_{p_1}$, and $\mathbb{Z}_{p_2}$ respectively. The same goes for $\mathbf{v}_{p_1,i}, \mathbf{v}_{p_2,i}$. Now we will see that $(\mathbf{u}_{1,i_1}, \mathbf{u}_{k,i_k})$ is a decomposition of $\mathbf{u}_i$. Let $\ell \in [h]$, then

- $\mathbf{u}_i[\ell] \equiv \mathbf{u}_{p_1,i}[\ell] \equiv \prod_{t \in [k]} \mathbf{u}_{p_1,t,i_t}[\ell] \equiv \prod_{t \in [k]} \mathbf{u}_{t,i_t}[\ell] \pmod{p_1}$.
- $\mathbf{u}_i[\ell] \equiv \mathbf{u}_{p_2,i}[\ell] \equiv \prod_{t \in [k]} \mathbf{u}_{p_2,t,i_t}[\ell] \equiv \prod_{t \in [k]} \mathbf{u}_{t,i_t}[\ell] \pmod{p_2}$.

Therefore, by the CRT, and from the fact that $\mathbf{u}_i \in \mathbb{Z}_m$, where $m = p_1 p_2$, we get

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

□

6.3 Decomposability of Kutin's MVs

In this section, we will show that the techniques used in Construction 4.12 to construct S_{can}-matching vector family from the basic matching vector from Claim 4.1 preserve decomposability, and thus will yield a decomposition of the matching vectors in Construction 4.12.

Construction 6.7. *Let $N, k, m > 0$, such that $m = p_1 p_2$ for two primes $p_1 < p_2$, and let $t = p_1^{e_1}, p_2^{e_2}$ for some integers $e_1, e_2 > 0$. Let $((\tilde{\mathbf{u}}_i, \tilde{\mathbf{v}}_i))_{i=1}^{N}$ $\mathbb{Z}_m^{\tilde{h}}$ be the basic matching vector family over $\mathbb{Z}_m^{\tilde{h}}$ from Claim 4.1, and for every $i = (i_1, \ldots, i_k) \in [N]$, let $(\tilde{\mathbf{u}}_{1,i_1}, \ldots, \tilde{\mathbf{u}}_{k,i_k})$ be the decomposition of $\tilde{\mathbf{u}}_i$ from Claim 6.3. Let $Q_{m,t}(x) = \sum_{i=1}^{d_Q} b_i \binom{x}{i}$ be the BBR polynomial from Theorem 4.10. For every $t \in [k]$, let $A_{t,i_t} \subseteq [\tilde{h}]$ be the subset defined by $\tilde{\mathbf{u}}_{t,i_t}$, i.e., $A_{t,i_t} = \{\ell \in [h] : \tilde{\mathbf{u}}_{t,i_t}[\ell] = 1\}$. We define the vectors $\mathbf{u}_{t,i_t}$, $\mathbf{v}_{t,i_t}$ of length $\sum_{i=1}^{d_Q} \binom{h}{i}$ where for every $\emptyset \neq S \subseteq [\tilde{h}]$ of size at most d_Q we have the following coordinate in the vectors*

- $\mathbf{u}_{t,i_t}[S] = \begin{cases} \mathbb{1}_{S \subseteq A_{t,i_t}} & \text{if } t < k, \\ b_{|S|} \cdot \mathbb{1}_{S \subseteq A_{t,i_t}} & \text{if } t = k. \end{cases}$
- $\mathbf{v}_{t,i_t}[S] = \mathbb{1}_{S \subseteq A_{t,i_t}}$.

Claim 6.8. *Let $((\mathbf{u}_i, \mathbf{v}_i))_{i=1}^{N}$ be the matching vector family over $\mathbb{Z}_m^h$ from Construction 4.3. For every $i = (i_1, \ldots, i_k) \in [N]$, the vectors $(\mathbf{u}_{1,i_1}, \ldots, \mathbf{u}_{k,i_k})$ and $(\mathbf{v}_{1,i_1}, \ldots, \mathbf{v}_{k,i_k})$ from Construction 6.7 are a decomposition of $\mathbf{u}_i$ and $\mathbf{v}_i$, respectively.*

Proof. Let $i = (i_1, \ldots, i_k) \in [N]$, and let $A_i \subseteq [\tilde{h}]$ be the subset defined by $\tilde{\mathbf{u}}_i$, i.e., $A_i = \{\ell \in [h] : \tilde{\mathbf{u}}_i[\ell] = 1\}$, and let A_{t,i_t} as defined in Construction 6.7, for every $t \in [k]$. We make the following observation:

$$\begin{aligned} \bigcap_{t \in [k]} A_{t,i_t} &= \bigcap_{t \in [k]} \{\ell \in [h] : \tilde{\mathbf{u}}_{t,i_t}[\ell] = 1\} \\ &= \{\ell \in [h] : \prod_{t \in [k]} \tilde{\mathbf{u}}_{t,i_t}[\ell] = 1\} \\ &= \{\ell \in [h] : \tilde{\mathbf{u}}_i[\ell] = 1\} = A_i. \end{aligned}$$

Recall the definition of $\mathbf{u}_i, \mathbf{v}_i$ from Construction 4.12, for every $\emptyset \neq S \subseteq [h]$, s.t. $|S| \leq d_Q$:

- $\mathbf{u}_i[S] = b_{|S|} \cdot \mathbb{1}_{S \subseteq A_i}$.
- $\mathbf{v}_i[S] = \mathbb{1}_{S \subseteq A_i}$.

Thus, for every entry S:

- $\prod_{t \in [k]} \mathbf{u}_{t,i_t}[S] = b_{|S|} \cdot \prod_{t \in [k]} \mathbb{1}_{S \subseteq A_{t,i_t}} = b_{|S|} \cdot \mathbb{1}_{S \subseteq \bigcap_{t \in [k]} A_{t,i_t}} = b_{|S|} \cdot \mathbb{1}_{S \subseteq A_i} = \mathbf{u}_i[S]$.
- $\prod_{t \in [k]} \mathbf{v}_{t,i_t}[S] = \prod_{t \in [k]} \mathbb{1}_{S \subseteq A_{t,i_t}} = \mathbb{1}_{S \subseteq \bigcap_{t \in [k]} A_{t,i_t}} = \mathbb{1}_{S \subseteq A_i} = \mathbf{v}_i[S]$.

□

7 A Polynomial Secret Sharing Scheme for General Access Structures

CDS protocols were used in [36, 4, 14, 6] to construct secret-sharing schemes for arbitrary access structures. Similarly to Applebaum et al. [4], we construct a secret-sharing scheme from k-server CDS protocols in two steps, first constructing robust CDS protocols (i.e., RCDS protocols as defined in Definition 2.12), and then constructing secret-sharing scheme for arbitrary access structures, while preserving the reconstruction degree throughout the steps. Specifically, we use an improved analysis of this transformation given in [14].

7.1 A Polynomial t-RCDS Protocol

Beimel et al. [14] show a construction of a quadratic (i.e., degree 2) t-RCDS protocol based on a quadratic k-server CDS protocol. Their construction is actually more general and applies to any reconstruction degree and any communication complexity of the CDS protocol, as stated in the next theorem.

Theorem 7.1 ([14])**.** *Assume that there is a k-server CDS protocol, with polynomial reconstruction function of degree d and with communication complexity $c(k,N,d) = N^{(k-1)/\xi(d)}$, for some function $\xi(d)$. Let $t < \min\{N/2k, 2^{\sqrt{N}/k}\}$. Then there is a k-server t-RCDS protocol with reconstruction degree d and message size*

$$\begin{aligned}&N^{O((k-1)/\xi(d))} \cdot t^{(k-1)\cdot(1-1/\xi(d))+1} \cdot k^{3k\cdot(1-1/\xi(d))} \cdot \log^2 N \cdot \log^{2k(1-1/\xi(d))}(t)\\ &= \tilde{O}(N^{O((k-1)/\xi(d))} \cdot t^{(k-1)\cdot(1-1/\xi(d))+1} \cdot k^{3k\cdot(1-1/\xi(d))}).\end{aligned}$$

Furthermore, the degree of encoding and decoding of this t-RCDS protocol is the degree of encoding and decoding, respectively, of the underlying k-server CDS protocol.

Combining Theorem 7.1 with Theorem 5.14, we get

Corollary 7.2. *Let $t < \min\{N/2k, 2^{\sqrt{N}/k}\}$. Then there is a degree-d k-server t-RCDS protocol with message size*

$$\begin{aligned}&O(N^{(k-1)\cdot O\left(\frac{\log\log d}{\log d}\right)} \cdot t^k \cdot k^{3k} \cdot \log^2 N \cdot \log^{2k}(t))\\ &= \tilde{O}(N^{(k-1)\cdot\frac{\log\log d}{\log d}} \cdot t^k \cdot k^{3k}).\end{aligned}$$

7.2 A Construction for All Access Structures

We present a theorem from [6] that constructs secret-sharing schemes from a k-server t-RCDS protocols.

Theorem 7.3 ([6])**.** *Assume that for every function $f : [N]^k \to \{0,1\}$ there is a k-server t-RCDS protocol with message size $c(k,N,t,d)$, then there is a secret-sharing scheme realizing an arbitrary n-party access structure with share size*

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

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

Combining Theorem 7.1 and, Theorem 7.3 we obtain the following result.

Theorem 7.4. *Assume that there is a k-server CDS protocol, with degree-d reconstruction, with communication complexity $c(k,N,d) = N^{(k-1)/\xi(d)}$, for some function $\xi(d) \ge 2$, then there is a secret-sharing scheme realizing an arbitrary n-party access structure with share size*

$$\max\left\{2^{0.5n(1+1/\xi(d))}, 2^{n(\log(2^{1/\xi(d)}+2)-1)}\right\} \cdot 2^{o(n)}.$$

Proof. Applying Theorem 7.1 we get a polynomial k-server t-RCDS protocol with degree-d reconstruction and message size

$$c(k,N,t,d) = \tilde{O}(N^{(k-1)/\xi(d)} \cdot t^{(k-1)\cdot(1-1/\xi(d))+1} \cdot k^{3k\cdot(1-1/\xi(d))}).$$

We will need to compute

$$\begin{aligned}
&c\left(\sqrt{2n(1-\beta)}, 2^{\sqrt{2n(1-\beta)}}, 2^{\sqrt{n(1-\beta)/2}}, d\right) \\
&= \tilde{O}\left(2^{\sqrt{2n(1-\beta)}\cdot\frac{\sqrt{2n(1-\beta)}-1}{\xi(d)}} \cdot 2^{\sqrt{n(1-\beta)/2}(\sqrt{2n(1-\beta)}-1)(1-\frac{1}{\xi(d)})+1} \cdot \sqrt{2n(1-\beta)}^{3\sqrt{2n(1-\beta)}(1-\frac{1}{\xi(d)})}\right) \\
&= 2^{\frac{2n(1-\beta)}{\xi(d)}} \cdot 2^{n(1-\beta)(1-\frac{1}{\xi(d)})} \cdot 2^{o(n)} \\
&= 2^{n(1-\beta)(\frac{1}{\xi(d)}+1)} \cdot 2^{o(n)}.
\end{aligned}$$

Next, we need to find

$$\begin{aligned}
&\max_{0.5\le\beta\le 1} c\left(\sqrt{2n(1-\beta)}, 2^{\sqrt{2n(1-\beta)}}, 2^{\sqrt{n(1-\beta)/2}}, d\right) \cdot 2^{H_2(\beta)n-2(1-\beta)n} \\
&\quad= \max_{0.5\le\beta\le 1} 2^{n(1-\beta)(\frac{1}{\xi(d)}+1)} \cdot 2^{o(n)} \cdot 2^{H_2(\beta)n-2(1-\beta)n} \\
&\quad= \max_{0.5\le\beta\le 1} 2^{n((1-\beta)(\frac{1}{\xi(d)}-1)+H_2(\beta))} \cdot 2^{o(n)}.
\end{aligned}$$

Thus, we will maximize the function $f(\beta) = (1-\beta)(1/\xi(d)-1) + H_2(\beta)$ by checking when the derivative is equal to zero.

$$\begin{aligned}
f'(\beta) = 1 - \frac{1}{\xi(d)} + \log\left(\frac{1-\beta}{\beta}\right) &= 0 \\
\Rightarrow \log\left(\frac{1-\beta}{\beta}\right) &= \frac{1}{\xi(d)} - 1 \\
\Rightarrow \frac{1-\beta}{\beta} &= 2^{1/\xi(d)-1} \\
\Rightarrow \beta &= \frac{2}{2^{1/\xi(d)}+2}.
\end{aligned}$$

Since that for every $\xi(d) > 1$, $0.5 \le \beta \le 1$,

$$\begin{aligned}
&\max_{0.5\le\beta\le 1} 2^{n((1-\beta)(1/\xi(d)-1)+H_2(\beta))} \\
&\quad= 2^{n\left(\left(1-\frac{2}{2^{1/\xi(d)}+2}\right)\left(\frac{1}{\xi(d)}-1\right)+H_2\left(\frac{2}{2^{1/\xi(d)}+2}\right)\right)} \\
&\quad= 2^{n\left(\left(\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\right)\left(\frac{1}{\xi(d)}-1\right)+\frac{2}{2^{1/\xi(d)}+2}\cdot\log\left(\frac{2^{1/\xi(d)}+2}{2}\right)+\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\cdot\log\left(\frac{2^{1/\xi(d)}+2}{2^{1/\xi(d)}}\right)\right)} \\
&\quad= 2^{n\left(\left(\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\right)\left(\frac{1}{\xi(d)}-1\right)+\frac{2}{2^{1/\xi(d)}+2}\cdot\left(\log\left(2^{1/\xi(d)}+2\right)-1\right)+\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\cdot\left(\log\left(2^{1/\xi(d)}+2\right)-\frac{1}{\xi(d)}\right)\right)} \\
&\quad= 2^{n\left(\left(\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\right)\left(\frac{1}{\xi(d)}-1\right)+\log\left(2^{1/\xi(d)}+2\right)-\frac{2}{2^{1/\xi(d)}+2}-\frac{1}{\xi(d)}\cdot\frac{2^{1/\xi(d)}}{2^{1/\xi(d)}+2}\right)} \\
&\quad= 2^{n\left(\log\left(2^{1/\xi(d)}+2\right)-1\right)}.
\end{aligned}$$

Thus, we obtain a secret-sharing scheme with reconstruction degree d, and the required share size. □

Using Taylor expansion for $\log(2^x + 2)$ we get the following corollary.

Corollary 7.5. *Assume that there is a k-server CDS protocol, with degree-d reconstruction, with communication complexity $c(k, N, d) = N^{(k-1)/\xi(d)}$, for some function $\xi(d) \geq 2$, then there is a secret-sharing scheme realizing an arbitrary n-party access structure with share size*

$$\max\left\{2^{0.5n(1+1/\xi(d))}, 2^{n(0.585+\frac{1}{3\xi(d)}+O\left(\frac{1}{\xi(d)^2}\right))}\right\} \cdot 2^{o(n)}.$$

Proof. Using Taylor expansion, we get $\log(2^x + 2) = \log 3 + \frac{x}{3} + O(x^2)$, thus

$$\log(2^{1/\xi(d)} + 2) - 1 = \log 3 + \frac{1}{3\xi(d)} + O(\frac{1}{\xi^2(d)}) - 1 = 0.585 + \frac{1}{3\xi(d)} + O\left(\frac{1}{\xi(d)^2}\right).$$

Thus, from Theorem 7.4 we get the required share size. □

Combining Theorem 5.14 with Corollary 7.5 we get the following corollary.

Corollary 7.6. *Let $d > 2$. Every n-party access structure can be realized by a secret-sharing scheme with degree-d reconstruction over $\mathbb{F}_3$ or over $\mathbb{F}_{p_2}$ for some prime $p_2 = polylog(d)$ and share size $2^{n(0.585+O\left(\frac{\log\log d}{\log d}\right))}$.*

Remark 7.7. Applying Theorem 7.4 with the k-server CDS protocol from Remark 5.16 with communication complexity $O(N^{(k-1)/4})$ and reconstruction degree 243, we get a secret-sharing scheme for an arbitrary access structure with share size $2^{0.6731n+o(n)}$, and reconstruction degree 243.

In comparison, Beimel et al. [14] constructed a quadratic (i.e., degree-2) secret-sharing scheme with share size $2^{0.705n+o(n)}$, and Applebaum and Nir [6] constructs a linear secret-sharing scheme with share size $2^{0.7575n+o(n)}$ and a general (non-polynomial) secret-sharing scheme with share size $2^{0.585n+o(n)}$. As d increases, the share size in our secret-sharing scheme approaches $2^{0.585n}$, i.e., it approaches the share size of the scheme of Applebaum and Nir [6], the best known secret-sharing scheme for an arbitrary access structure.

7.3 A Construction for Almost All Access Structures

By [10], almost all access structures can be realized by secret-sharing scheme with shares of size $2^{o(n)}$ and by a linear secret-sharing scheme with share size $2^{n/2+o(n)}$. In [14], they showed that almost all access structures can be realized by a quadratic secret-sharing scheme over $\mathbb{F}_2$ with share size $2^{n/3+o(n)}$. We will use the same technique, and construct a secret-sharing scheme with polynomial reconstruction and smaller share size for almost all access structures.

Theorem 7.8 ([10])**.** *Assume that there is a k-server CDS protocol, with reconstruction of degree-d and with communication complexity $c(k, N, d) = N^{(k-1)/\xi(d)}$ for some function $\xi(d) > 0$. Then almost all access structures can be realized by secret-sharing scheme with degree-d reconstruction and share size $2^{n/\xi(d)+o(n)}$.*

Combining Theorem 7.8, with our polynomial k-server CDS protocol with message size $c(k, N, d) = N^{O((k-1)\cdot\frac{\log\log d}{\log d})}$ from Theorem 5.14 we get the following theorem.

Corollary 7.9. *Almost all access structures can be realized by secret-sharing scheme with degree-d reconstruction over $\mathbb{F}_3$ or over $\mathbb{F}_{p_2}$ for some prime $p_2 = polylog(d)$ and share size $2^{O(n\log\log d/\log d)+o(n)}$.*

As d grows, we get share size $2^{\epsilon n}$ for every constant $\epsilon > 0$. If we take $d - O(\log n)$ (or even $d = o(1)$), then the share size is $n^{o(1)}$, however larger than the share size of [10], where the degree of reconstruction is not bounded.

Acknowledgement. The first and the third authors are supported by the ISF grant 391/21. The first author is also supported by the ERC grant 742754 (project NTSC). The third author is also supported by the Frankel center for computer science. The second author is supported by the grant 2021SGR 00115 from the Government of Catalonia, the project ACITHEC PID2021-124928NB-I00 from the Government of Spain, and the project HERMES, funded by INCIBE and NGEU/PRTR.

References

[1] Martin Aigner and Günter M. Ziegler. Bertrand's postulate. In *Proofs from THE BOOK*, pages 7–12. Springer, Berlin, Heidelberg, 2010.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] Amos Beimel, Oriol Farràs, Yuval Mintz, and Naty Peter. Linear secret-sharing schemes for forbidden graph access structures. In *TCC 2017*, volume 10678 of *LNCS*, pages 394–423, 2017.

[12] 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.

[13] 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.

[14] 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.

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

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

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

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

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

[20] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM J. Comput.*, 40(4):1154–1178, 2011.

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

[22] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *STOC'09*, pages 39–44, November 2009.

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

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

[25] Peter Frankl. Constructing finite sets with given intersections. In *Combinatorial Mathematics, Proceedings of the International Colloquium on Graph Theory and Combinatorics 1981*, volume 17 of *Annals of Discrete Mathematics*, pages 289–291, 1983.

[26] Peter Frankl and Richard M. Wilson. Intersection theorems with geometric consequences. *Combinatorica*, 1(4):357–368, 1981.

[27] 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.

[28] 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.

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

[30] V. Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs. *Combinatorica*, 20:71–86, 2000.

[31] 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 [13].

[32] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *Automata, Languages and Programming, 29th International Colloquium, ICALP 2002*, pages 244–256, 2002.

[33] 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.

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

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

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

[37] 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.

[38] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. Announced in NY Crypto Day, Sep 15, `https://nycryptoday.wordpress.com/`., 2017.

[39] 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.

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

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

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

[43] Hung-Min Sun and Shiuh-Pyng Shieh. Secret sharing in graph-based prohibited structures. In *INFOCOM '97*, pages 718–724. IEEE, 1997.

[44] Triantafyllos Xylouris. On the least prime in an arithmetic progression and estimates for the zeros of dirichlet l-functions. *Acta Arithmetica*, 150(1):65–91, 2011.