Bounded Collusion Protocols, Cylinder-Intersection Extractors and Leakage-Resilient Secret Sharing

Ashutosh Kumar[*], Raghu Meka[†], David Zuckerman[‡]

April 22, 2020

Abstract

In this work we study *bounded collusion protocols* (BCPs) recently introduced in the context of secret sharing by Kumar, Meka, and Sahai (FOCS 2019). These are multi-party communication protocols on n parties where in each round a subset of p-parties (the *collusion* bound) collude together and write a function of their inputs on a public blackboard.

BCPs interpolate elegantly between the well-studied number-in-hand (NIH) model ($p = 1$) and the number-on-forehead (NOF) model ($p = n - 1$). Motivated by questions in communication complexity, secret sharing, and pseudorandomness we investigate BCPs more thoroughly, answering several questions about them.

- We prove a polynomial (in the input-length) lower bound for an explicit function against BCPs where any constant fraction of players can collude. Previously, nontrivial lower bounds were known only when the collusion bound was at most logarithmic in the input-length (owing to bottlenecks in NOF lower bounds).
- For all $t \leq n$, we construct efficient t-out-of-n secret sharing schemes where the secret remains hidden even given the transcript of a BCP with collusion bound $O(t/\log t)$. Prior work could only handle collusions of size $O(\log n)$. Along the way, we construct leakage-resilient schemes against disjoint and adaptive leakage, resolving a question asked by Goyal and Kumar (STOC 2018).
- An explicit n-source *cylinder intersection* extractor whose output is close to uniform even when given the transcript of a BCP with a constant fraction of parties colluding. The min-entropy rate we require is 0.3 (independent of collusion bound $p \ll n$).

Our results rely on a new class of exponential sums that interpolate between the ones considered in additive combinatorics by Bourgain (Geometric and Functional Analysis 2009) and Petridis and Shparlinski (Journal d'Analyse Mathématique 2019).

[*] a@ashutoshk.com, UCLA. Supported by NSF grants CCF-1553605 and 1619348 and US-Israel BSF grant 2012366.

[†] raghum@cs.ucla.edu, UCLA. Supported by NSF Grant CCF-1553605.

[‡] diz@cs.utexas.edu, UT Austin. Supported by NSF Grant CCF-1705028 and a Simons Investigator Award (#409864).

1 Introduction

We begin by motivating our questions in the three focus areas of this work: multi-party communication complexity, leakage-resilient secret sharing, and extractors.

Multi-Party Communication Complexity. In a seminal work, Yao [Yao79] initiated the study of communication complexity: how much communication is needed to compute a function when the input is split between two parties. Since its introduction, communication complexity has blossomed into a central area within complexity with connections to many other fields. Here we focus on the multi-party case — when there are more than two parties.

The first way one would think to define multiparty communication complexity would be to split the input among the n parties, where each party has access to its own input. This is known as the *number-in-hand* (NIH) model ([PVZ12, BEO$^+$13, BO15]). In a seminal work, Chandra, Furst, and Lipton [CFL83] introduced the much more powerful *number-on-forehead* (NOF) model. Here, each party sees all inputs but their own (number written on forehead) and they can communicate by a writing on a public blackboard. The goal is to compute a function of their inputs while minimizing the total amount of communication (number of bits written on the board). This is a very versatile model with many beautiful connections to circuit complexity and data structure lower bounds among others (cf. book of [KN06]). We currently do not have nontrivial lower bounds in the NOF model when the number of parties is more than logarithmic in the input length. It is a longstanding challenge in complexity theory to overcome this logarithmic barrier [BNS92, Raz00, She14].

Motivated by the above question, we consider a model that interpolates nicely between the NIH and NOF models. This is the class of *bounded collusion protocols* (BCPs) introduced recently by Kumar, Meka, and Sahai [KMS19], originally in the context of modeling leakage-resilience in cryptography (see below for a discussion of the original motivation). These protocols are characterized by a parameter $p < n$ [1] called the collusion bound. The communication proceeds in rounds, and in each round a group of p parties get together ("collude") and write a function of their inputs on the blackboard. Call such protocols *p-party collusion protocols*. The goal once again is to compute a function of all their inputs while minimizing the total amount of communication.

Note that p-party collusion protocols for $p = 1$ correspond exactly to the NIH model, and $p = n - 1$ correspond to the NOF model. Thus, the model interpolates between the two classical models. As observed in [KMS19], NOF lower bounds yield nontrivial lower bounds for BCPs when the collusion bound is logarithmic in the input length. This leads to the following natural question:

Question 1. *Can we prove communication complexity lower bounds against protocols that allow a super-logarithmic number of parties to collude in each round?*

The above should be easier than breaking the logarithmic barrier for NOF protocols. Intuitively, allowing for more parties than the collusion bound (even if the collusion bound is itself super-logarithmic) should limit the power of protocols. Indeed, one of our main result is precisely such a bound.

Theorem 1. *There exists an explicit function* $\mathsf{f} : (\{0,1\}^m)^n \to \{0,1\}$ *such that for* $p < n$*, any* p*-party collusion protocol to compute* f *requires* $\Omega(m^{\frac{\log(n/p)}{\log(n/p)+1}})$ *bits of communication.*

[1] Throughout, p denotes the collusion bound, n denotes the number of parties (as is often used in secret sharing literature), and m denotes the input length to each party.

The above in particular implies that as long as $p \leq cn$ for any constant fraction $c < 1$, f requires $m^{\Omega(1)}$ bits of communication. Note that both p, n can be super-logarithmic (and even $m^{\Omega(1)}$).

Remark. We in fact get average-case lower bounds under the notion of *cylinder-intersection extractors* defined below. The results there subsume our communication lower bounds. We decided to present the results separately as the problems are independently interesting and to improve readability. We think this provides a better conceptual progression.

Leakage-Resilient Secret Sharing. The seminal works of Blakley [Bla79] and Shamir [Sha79] introduced the notion of secret sharing. In their notion—*t-out-of-n* schemes—a secret needs to be shared to n parties such that: (1) Recoverability: Any set of t parties can reconstruct the secret from their shares. (2) Secrecy: No set of less than t parties can learn any information about the secret. Secret sharing schemes are ubiquitous as building blocks in cryptography. In this introduction we will focus only on threshold schemes as above; our results apply to more general access structures (see Section 6 for details).

Secret sharing schemes, while originally envisioned with only the goal of secrecy formulated above, have been strengthened in various ways. These include *verifiability* [RBO89], *robustness* [CDF$^+$08], *functionality* [BGI15], and *non-malleability* [GK18a]. Here we focus on the substantially stronger secrecy goal of *leakage resilience*, which has a long history in cryptography (e.g., recent survey of Kalai and Reyzin [KR19]). The goal here is to make cryptosystems robust to adversaries who learn additional "leaked" information (via say passive side-channel attacks). Leakage resilience in the context of secret sharing itself has been of significant recent interest [DP07, DDV10, GK18a, GK18b, BDIR18, KMS19, BS19, ADN$^+$19, SV19, FV19, LCG$^+$19, NS20, BFV19] (see [LCG$^+$19] for overview).

Our starting point is the recent work [KMS19] which focuses on handling *joint* leakage where an *adaptive* adversary can learn information depending on multiple shares at once. They model leakage as an adversary running a multi-party communication protocol on the shares of the n parties and learning the transcript. Different classes of communication protocols will now yield different models of leakage resilient sharing schemes (LRSSs).

Bounded collusion protocols are especially natural in the context of secret sharing. For instance, in t-out-of-n threshold schemes the *best* class of protocols we can hope to protect against are $(t-1)$-party collusion protocols. In addition, modeling leakage resilience as protocols allows the use of methods and techniques from communication complexity. See [KMS19] for a detailed discussion of the model and comparison with recent works.

We now state our formal definition of LRSS (see Section 6 for more details):

Definition 1 (($\mathbf{p}, \mathbf{t}, \mathbf{n}$)**-LRSS**). *Let (*$\mathsf{Share}, \mathsf{Rec}$*) be a t-out-of-n secret sharing scheme that shares k bit secrets into n shares and let $1 \leq p < t$ be a collusion bound. Let μ be any bound on allowed leakage and $\epsilon \in (0,1)$ be the desired error bound. We say (*$\mathsf{Share}, \mathsf{Rec}$*) is a* $(\mathbf{p}, \mathbf{t}, \mathbf{n}, \mu)$***-leakage-resilient secret sharing scheme*** *(or (p, t, n, μ)-LRSS in short*[2]*) if for any p-party collusion protocol* Leak *with at most μ bits of communication, and any pair of distinct secrets $a \neq b \in \{0,1\}^k$, we have*

$$\mathsf{Leak}(\mathsf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathsf{Share}(b))$$

[2] We drop μ for brevity when it is not important.

Kumar, Meka, and Sahai [KMS19] constructed (p, t, n, μ)-schemes where the share length grew as $\approx 2^{O(p)}(\mu \log n) \cdot k$. Thus, the schemes are not efficient when the collusion bound is super-logarithmic $p = \omega(\log n)$.

Question 2. *Can we get efficient (p, t, n)-leakage-resilient secret sharing schemes for $p = \omega(\log n)$?*

Indeed, as observed in [KMS19], efficient $(p, t = p + 1, n)$-LRSS for $p = \omega(\log n)$ would resolve longstanding bottlenecks in complexity theory. But what if the threshold for reconstruction t is noticeably bigger than p? Our second main result resolves this question as long as $t = \Omega(p \log p)$:

Theorem 2. *There is an efficient $(p = O(t/\log t), t, n, \mu)$-LRSS that shares k bit secrets into $poly(\mu, n) \cdot k$ bit shares.*

We in fact obtain an efficient generic compiler that transforms any secret sharing scheme having authorized sets of size at least t to one that is additionally resilient against p-party collusion protocols for $p = O(t/(\log t))$. Instantiating our compiler with various secret sharing schemes [Sha79, KW93, Bei11, KNY14], we can get leakage-resilient secret sharing schemes for corresponding access structures such as t-out-of-n, monotone span programs, monotone P and monotone NP .

Note that the collusion bound can be as big as $\Omega(n/\log n)$ if t is $\Omega(n)$. Obtaining efficient schemes for $p = \omega(\log n)$ even when the protocol is restricted to use only *disjoint* subsets[3] was open and interesting by itself. For this special case we obtain:

Theorem 3. *There is an efficient compiler that converts any secret sharing into one leakage-resilient against arbitrarily bounded communication amongst disjoint unauthorized subsets.* [4]

As a corollary, we obtain the first construction of t-out-of-n scheme that remains leakage-resilient against leakage from disjoint subsets of size up to $t-1$ (which is optimal). This resolves the question posed by Goyal and Kumar [GK18a] who constructed such schemes for the special case of $t = 2$ for designing 'non-malleable' secret sharing schemes. Our results on disjoint LRSS can be used in a black-box way, using the compiler of Brian, Faonio, and Venturi [BFV19] to get "continuous non-malleable" secret sharing schemes that are resilient against disjoint leakage and disjoint tampering of unauthorized subsets of arbitrary size. We refer the reader to [BFV19] for more details.

To the best of our knowledge, the only other works with some form of joint-leakage are Srinivasan and Vasudevan [SV19] and Lin et al. [LCG$^+$19]. Lin et al. consider a non-compartmentalized model where the leakage can be a linear function of *all* the shares. [SV19] designed t-out-of-n LRSS against an adversary who learns any set of $t-2$ shares and then uses these fixed $t-2$ shares to non-adaptively learn information from each of the other $n-t+2$ shares independently. Our results allow for adaptive leakage either from overlapping subsets of size $O(t/\log t)$, or from disjoint subsets of size at most $t-1$.

Cylinder-Intersection Extractors. Randomness extractors are fundamental objects in pseudorandomness. In their more basic form they take impure source(s) of randomness and output almost uniformly random bits. A source is a probability distribution X on $\{0,1\}^m$. As is standard, we quantify the randomness in X by its *min-entropy* $H_\infty(X) = \min_x -\log_2(\Pr[X = x])$. The basic goal is to design functions that take such source(s) and output almost uniformly random bits.

[3] That is the adversary partitions the n parties into groups of size at most p and leaks from each group.

[4] In both of our results, we can additionally leak any unauthorized set of shares at the end of the leakage-protocol.

Unfortunately, it is impossible to do so given a single source with min-entropy as high as $m-1$. Generalizing work of Santha and Vazirani [SV86] and Vazirani [Vaz87], Chor and Goldreich [CG88] initiated the study of extractors from two independent sources of randomness, each with sufficient min-entropy. In a different direction, Nisan and Zuckerman [NZ93] got around this impossibility by constructing a *seeded extractor* that extracts randomness from one impure source with the help of a *short* auxiliary truly random seed. A final way to circumvent the impossibility result is to consider a single source with additional structure. Examples include samplable sources [TV00, Vio14], affine sources [GR05, Bou07], small-space sources [KRVZ06], and sumset-sources [CL16b] (see [CL16b] for a broader overview).

Since their introduction, extractors have been extensively studied with numerous applications and connections to complexity theory, error correcting codes, Ramsey theory, cryptography and more.

Here we focus on *multi-source extractors*, which have been extensively studied in their own right [BIW06, Bou05, Raz05, Rao09, Bou09, Li13b, Li13a, Li15] . Most works assume the multiple sources are *independent* of each other, as some assumption is needed to circumvent the impossibility result. But what if we do not have complete independence? As an example, consider three sources, where every pair may be correlated. Note that the impossibility result no longer applies in this situation. Concurrent and independent works of Chattopadhyay, Goodman, Goyal, and Li [CGGL20] and Ball, Goldreich, and Malkin [BGM20] also studied generalizations of extractors for independent sources to dependent sources. However their models are different from ours.

Motivated by applications and terminology in cryptography, we consider settings where we start with n independent sources $X_1, \ldots, X_n \in \{0,1\}^m$, each with min-entropy at least k, but an *adversary* now correlates them by learning some side-channel information about the sources. Following our discussions from the previous section, we can model this as an adversary running a communication protocol on the sources and learning the transcript of the protocol. As before, different classes of protocols yield different classes of sources.

For instance, if the adversary runs an NIH protocol, then conditioned on the transcript, the sources remain independent. Thus, usual multi-source extractors can still be used. What if we look at NOF protocols or p-party collusion protocols? Observe that the latter models the case where the sources may be p-wise correlated. Can we still extract uniform randomness? This was formalized as *cylinder-intersection extractors* in [KMS19].

Definition 2 (Cylinder intersection extractors [KMS19]). *A (p, n, μ)-cylinder intersection extractor with error ϵ for (m, k)-sources is a function $Ext : (\{0,1\}^m)^n \to \{0,1\}$ such that the following holds. For all distributions of n independent sources $X_1, \ldots, X_n$ with $H_\infty(X_i) \geq k$ and p-party collusion protocol Π with at most μ bits of communication,*

$$(Ext(X_1, \ldots, X_n), \Pi(X_1, \ldots, X_n)) \approx_\epsilon (U_1, \Pi(X_1, \ldots, X_n)).$$

In other words, the definition says that the extractor output $Ext(X_1, \ldots, X_n)$ is close to uniformly random even given the transcript of a p-party collusion protocol $\Pi(X_1, \ldots, X_n)$. [KMS19] raised the question of constructing explicit extractors as above.

The lower bounds of [BNS92] imply cylinder-intersection extractors as above with $p = n-1$ as long as $n \ll \log m$ and the min-entropy $k \geq c_p m$ for $c_p = 1 - \Omega(1/2^p)$. Thus, nothing nontrivial is known when the collusion bound $p \gg \log n$ (say $1.1 \log n$). Also, even for constant p, the entropy rate required is very close to 1. In contrast, when $p = 1$, we can extract even when the entropy k

is poly($\log n$) (for $n = 2$, this was achieved recently in [CZ19]; for $n = 3$ and higher, this was done earlier — [Li15]).

In a different line of work, Petridis and Shparlinski [PS19] used exponential sum bounds (e.g., [Bou09], [BGK06]) to give (in our language) $(2, 3)$ and $(3, 4)$ cylinder-intersection extractors for min-entropy $k \geq 0.4m$ and $k \geq 0.33m$, respectively. Kerr and Macourt [KM19] extended this to the case of $(p, p+1)$-cylinder intersection sources, essentially obtaining extractors for $5 \leq p < \log m$ and min-entropy $k \geq 0.3m$. Note that while these results improve the entropy requirement over [BNS92], they still hit the logarithmic bottleneck that $p < \log m$. This should be expected, as $(p, p + 1)$-cylinder intersection extractors are clearly stronger than proving lower bounds in NOF.

Following our results on communication complexity, we obtain (p, n)-cylinder intersection extractors even for $p = \omega(\log n)$ when $n \gg p$.

Theorem 4. *Fix a prime $q > 2$. Let $m = \log q$. Then for $n \geq 6$, there exists an explicit function* BouExt : $\mathbb{F}_q^n \to \{0, 1\}$ *such that the following holds. For all fraction $\alpha < 1$,* BouExt *is a $(\alpha n, n, \mu)$-cylinder intersection extractor with error ϵ for all $(m, 0.3m)$-sources and $\mu < cm^{c_\alpha}$ and $\epsilon = 2^{-\Omega(m^{c_\alpha})}$, where $c_\alpha = \log(1/\alpha)/(1 + \log(1/\alpha))$ and $c > 0$ is a universal constant.*

We remark that while there has been tremendous amount of progress recently in getting independent source extractors ([Li15, CZ16, Coh16, BADTS17]) with almost optimal min-entropy, those new results and techniques heavily rely on independence. For instance, it seems quite challenging (even with the new techniques) to even get 2-cylinder intersection extractors (i.e., $p = 2$) for entropy rates $k = \delta m$ for arbitrary $\delta \in (0, 1)$. As a flip-side, one could ask[5] what is the least number of sources n needed to extract when collusion bound is 2 for a specific min-entropy k. As an observation, we note that *sumset-extractors* introduced by [CL16a] imply $(2, n, \mu)$-cylinder intersection extractors even for min-entropy $k = \text{poly}(\log n)$ as long as $n > C\mu$ for a fixed constant C.

1.1 Our Techniques

1.1.1 Lower bounds against BCPs

Before giving intuition behind our lower bounds against BCPs, we recall some of the results from number-on-forehead literature. For n-party function $\mathsf{f} : (\{0, 1\}^m)^n \to \{0, 1\}$, the strongest known lower bounds for NOF model are of the form $\Omega(m/2^n)$. Consequently, these lower bounds become trivial as soon the number of parties $n \gg \log m$. As already mentioned earlier, it is a longstanding problem in complexity theory to obtain non-trivial lower bounds for super-logarithmic number of parties [BNS92, Raz00, She14].

For $n \gg \log m$, Podolskii and Sherstov [PS17] prove communication complexity lower bounds that are at most $\log m$. In more detail, as generalized inner-product $\mathsf{GIP} : (\{0, 1\}^m)^n \to \{0, 1\}$ admits a protocol with communication at most $O(\log m)$ [Gro94], they observe that to prove lower bounds one only needs to rule out protocols with communication at most $O(\log m)$. Consequently, *at most* $O(\log m)$ parties can speak in any such protocol, and one can reduce the case of a larger number of parties to the case of $O(\log m)$ parties, for which lower bounds are already known. Unfortunately, for our application to leakage-resilient secret sharing, this logarithmic lower bound would lead to *exponential* sized shares, whereas we require polynomial sized shares for efficiency.

[5] As was investigated in the literature on multi-source extractors, e.g., [Rao06], [Li13b]

Moreover, by restricting to BCPs, we may hope to prove stronger lower bounds against protocols which allow more than logarithmically many subsets to speak.

While in our proofs we rely on exponential sums, for ease of intuition, we continue with generalized inner-product. Let $\mathsf{GIP}_n : (\mathbb{F}^m)^n \to \mathbb{F}$ be the n-party generalized inner-product function over some finite field $\mathbb{F}$ of *large cardinality* given by $\mathsf{GIP}_n(x_1, \ldots, x_n) \leftarrow \sum_{i \in [m]} \prod_{j \in [n]} x_j[i]$, where $x_j[i]$ refers to value of i^{th} coordinate of vector x_j.

Our first idea: use the round bound to find two non-colluding parties. Suppose, in any round of communication, a pair of parties get together and communicate some message based on their two inputs. We wish to argue that a lot of communication will be required in order to compute the output of GIP_n. Towards this end, observe that if the number of rounds of communication is less than $\binom{n}{2}$, then there will exist a pair of parties $i, j \in [n]$ who do not collectively speak in any round. We can then use a lower bound for inner-product IP $= \mathsf{GIP}_2$ to obtain a lower bound for generalized inner-product GIP_n for this communication model. This can be achieved by a simulation argument where the two parties, namely i and j, simulate the n party protocol after fixing the inputs of other $n-2$ parties [6]. A catch here is that, we need to know in advance the two parties who will not collaborate. This becomes an issue for adaptive protocols where which set colludes can depend on the communication so far. We deal with it by working with a stronger notion of discrepancy which circumvents such issues. [7]

This basic idea can be generalized, trading the number of rounds r with the collusion bound p. Specifically, the number of pairs ruled out in each round is $\binom{p}{2}$. Therefore, as long as the number of rounds r is less than $\binom{n}{2}/\binom{p}{2}$, we can find two non-collaborating parties and run the above simulation. This simple idea already implies some lower bounds: if the collusion bound $p = o(\sqrt{n})$, then we can prove lower bounds for super-linear $r = \omega(n)$.

Our second idea: use collusion bound to rely on NOF lower bounds. What if every pair is collectively speaking in some round? Our previous idea no longer works. However, notice that if we take any three parties, and fix the inputs for the other parties, the n party communication can be simulated as a 3-party number-on-forehead communication amongst the 3 chosen parties. Therefore, we can use 3-party NOF lower bounds to obtain lower bounds for this communication model [8]. More generally, lower bounds for p-party collusion protocols can be easily achieved from NOF lower bounds for $p+1$ parties. This allows us to handle p up to $O(\log n)$ [BNS92,Raz00,She14].

Note that, in both of our ideas, we are making non-black-box use of GIP_n. Specifically, we are relying on its *self-reducibility*: fixing any $n-k$ inputs of GIP_n results in an instance of GIP_k.

Our main idea: non-clique of collaborating parties. To go beyond the logarithmic barrier, we mix the two previous ideas. We aim to find a set of k parties such that there is no round in which all k are involved together.

The first idea corresponds to $k = 2$. The second idea corresponds to observing that once we find such a set of k parties, fixing the inputs to the remaining $n-k$ parties yields an NOF protocol

[6] We need a field of large cardinality to avoid GIP_n from being fixed to 0 after randomly fixing $n-2$ inputs. For instance, over binary field GIP_n is most likely 0 for $n = \omega(\log m)$.

[7] Another way to circumvent this is by guessing the pair of parties who will not collaborate, and proceeding with the reduction suffering some loss in the security parameter.

[8] For the moment, we are assuming to have a NOF lower bound of GIP_n over large finite fields. To the best of our knowledge, such a lower bound has not yet explicitly appeared in literature.

on the inputs of the k parties. Observe that the idea of Podolskii and Sherstov can be seen as an extremal case, where at most k parties are allowed to speak in the NOF model ($p = n - 1$).

To illustrate the utility of this simple idea, we claim that for any round bound $r = n^{O(1)}$, we immediately get non-trivial lower bounds against collusion of any constant fraction of parties.

Fix k to be chosen later. There are $\binom{n}{k}$ k-tuples in general. If in each round p parties collude, then this rules out $\binom{p}{k}$ k-tuples for us. So if the total number of rounds $r < \binom{n}{k}/\binom{p}{k}$, there will be a set of k parties who never collude together. If $r \leq (n/p)^k$, this requirement is satisfied. Substituting p to be any fraction of n and a suitable $k \ll \log m$ (as allowed in NOF lower bounds) proves the required claim.

Extensions with no bounds on number of rounds. Observe that our argument only required that at most $(n/p)^k$ subsets are used for communication in the entire protocol (for a suitable choice of $k \ll \log m$). Building on this observation, our simulation based proof can be easily extended to yield communication complexity lower bounds against arbitrary protocols, as long as at most $(n/p)^k$ subsets are used with no restriction on the number of rounds.

Theorem 1 for instance follows from choosing k appropriately and noting that the number of rounds of communication is at most the total communication.

1.1.2 Exponential sums and cylinder-intersection extractors

The basic idea behind our extractor construction and analysis is similar to the proof of Theorem 1. However, instead of using GIP_n, we use an additive character over a large prime field. This allows us to use non-trivial estimates in additive combinatorics and number theory on exponential sums to achieve better min-entropy bounds [9].

The influential work of Barak, Impagliazzo, and Wigderson [BIW06] was the first to use sum-product theorems of Bourgain, Katz, and Tao [BKT04] from additive combinatorics to build independent source extractors. In a breakthrough, Bourgain [Bou05] used sum-product theorems and exponential sums to design two-source extractors for min-entropy rate slightly less than 1/2, improving results of Chor and Goldreich [CG88].

Our starting point is the result of Kamp, Rao, Vadhan, and Zuckerman [KRVZ06], who used exponential sum estimates of Bourgain, Glibichuk, and Konyagin [BGK06] to construct extractors having any constant min-entropy rate with only a constant number of independent sources. While they had a simpler construction relying on finite fields of small characteristic, our techniques require us to rely on prime fields of large characteristic.

To give our extractor construction, we begin with some notation. Let $\mathbb{F}_q$ be the prime field of cardinality q. Inspired by the extractor of Bourgain [Bou05] and Kamp et al. [KRVZ06], our n source extractor BouExt : $\mathbb{F}_q^n \to \{0,1\}$ is defined as:

$$\mathrm{BouExt}(x_1, \ldots, x_n) = \mathsf{sign} \sin\left(\frac{2\pi \prod_{i\in[n]} x_i}{q}\right)$$

where sign is the usual sign function defined as $\mathsf{sign}(x) = 1$ if and only if $x \geq 0$. Bourgain [Bou05] noted that the above is an extractor for rate δ if we can obtain non-trivial upper bounds on the following exponential sum:

[9] GIP_n cannot extract when min-entropy rate is less than $1 - \frac{1}{n}$.

$$\left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} e_q \left(\prod_{i \in [n]} x_i \right) \right| .$$

Here e_q is the exponential function defined as $e_q(x) = \exp\left(\frac{2\pi i x}{q}\right)$ and $X_1, \ldots, X_n$ are arbitrary subsets of $\mathbb{F}_q$ of size q^δ. Bounds with optimal subset sizes have been obtained by Bourgain [Bou09].

Intuitively, notice that the choice of x_2 is independent of the choice of x_1, and thus, the sums as above correspond to independent source extractors. To model cylinder intersections we would need to look for a richer class of exponential sums. For example, to model $(2, 3)$-cylinder intersection sources, we can use three indicator functions $\phi_{1,2}, \phi_{2,3}, \phi_{1,3}$ each of the form $\mathbb{F}_q^2 \to \{0, 1\}$. In more detail, $\phi_{1,2}$ decides whether or not to sum over the input x_1, x_2, modelling the correlation between the pair of sources. This results in the following type of exponential sum:

$$\left| \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} \sum_{x_3 \in X_3} \phi_{1,2}(x_1, x_2)\phi_{2,3}(x_2, x_3)\phi_{1,3}(x_1, x_3) e_q \left(\prod_{i \in [3]} x_i \right) \right|$$

Fortunately, such exponential sums have been recently considered in the literature starting with the work of Petridis and Shparlinski [PS19], who obtained concrete bounds for the special cases of $(2, 3)$ and $(3, 4)$. Very recently, Kerr and Macourt [KM19] generalized the result to $(n - 1, n)$ for $n \ll \log \log q$. Building upon these constructions, we consider general exponential sums of the following form (slightly simplified):

$$\max_{\mathcal{A}, \phi} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \left(\prod_{S \in \mathcal{A}} \phi_S(x_S) \right) e_q \left(\prod_{i \in [n]} x_i \right) \right|$$

where the maximum is over any collections of subsets $\mathcal{A} \subseteq 2^{[n]}$ that avoids some subset of size k and any collection of corresponding indicator functions $\phi = \{\phi_S : S \in \mathcal{A}\}$.

We then follow the approach of Theorem 1 for upper bounding such exponential sums on n variables in terms of upper bounds for the special case of exponential sums corresponding to $(k, k+1)$-cylinder intersection sources. We iteratively use the Cauchy-Schwarz argument as in [BNS92] and reduce the $(k, k+1)$ case to the $(k-1, k)$ case taking into the consideration the smaller subset sizes. Finally, we use the bounds of [PS19], [KM19] for a small constant to recover Theorem 4 and obtain non-trivial upper bounds on sums as above as long as $k < \log \log q$ (q is the field size) [10]. Working in this greater generality actually also simplifies some of our conditioning arguments and is no more difficult.

1.1.3 Leakage-resilient secret sharing schemes

We describe our ideas for Theorem 3 that constructs schemes resilient against adaptive and disjoint leakage, and then further extend it to obtain Theorem 2. We begin by considering a leader-based t-out-of-n scheme, which at first sight looks artificial, but proves instrumental in both our results. Our notion can be seen as a generalization of an idea present in the recent work of Aggarwal et al. [ADN+19], who implicitly designed leader based 2-out-of-n schemes against non-adaptive and individual leakage, while designing general LRSS schemes in the same leakage model.

[10] We remark that handling $k = \omega(\log \log q)$ will imply NOF lower bounds beyond the logarithmic barrier.

Leader-based t-out-of-n schemes. For any "leader" $\ell \in [n]$, we define and construct t-out-of-n schemes for leader ℓ, that allows the leader and any $t-1$ other parties to reconstruct the secret. More importantly, it guarantees that the transcript of any protocol amongst the two unauthorized subsets, namely, $[n] \setminus \{\ell\}$ and $\{\ell\} \cup U$ for any $|U| = t-2$, reveals nothing about the underlying secret.

Use leader-based schemes to get regular SS schemes. The idea would be to share the secret using any regular t-out-of-n scheme to obtain n shares $m_1, \ldots, m_n \leftarrow \mathsf{Share}_t^n(m)$, and make each of the n parties the leader for exactly one of these shares. That is, m_i is shared using a t-out-of-n scheme for leader i. Notice that any set of less than t parties of the final scheme can only have at most $t-1$ leaders and consequently the secret will be hidden. To prove leakage-resilience, we use a hybrid argument to rely on the leakage-resilience ensured by our leader-based scheme for each choice of leader. We generalize our result to general access structures, by appropriately defining a leader based scheme corresponding to the given access structure and building a black-box compiler that efficiently prunes a leader away from the access structure.

Leakage-Resilience against BCPs. We next sketch the proof of Theorem 2. We first describe the basic construction of LRSSs [KMS19] which we will rely on. The construction in [KMS19] can be abstracted as follows:

1. Use a function hard for p-party NOF protocols (in a black-box way) to get a $(p, p+1, p+1)$-LRSS. Note that the threshold equals the number of parties.

2. Use several instantiations of $(p, p+1, p+1)$-LRSS along with *perfect hash families* to build (p, t, n)-LRSS.

Both of these steps hit barriers at $p = \omega(\log n)$ in [KMS19]: The first step blows up the share-length by a 2^p factor owing to the use of NOF lower bounds and the second step incurs another $2^{O(p)}$ factor owing to the use of perfect hash families [FKS84].

As we are interested in the setting where $p = O(t/\log t)$, we can safely assume that $p \ll t$. Thus, Theorem 1 already provides a way to implement step (1) above without losing a 2^p factor, if we use BCP lower bounds as opposed to NOF lower bounds. The main hurdle is now in implementing step (2) efficiently when $p = \omega(\log n)$. But we need a new idea as there are information theoretic lower bounds against perfect hash families [FK84]. We introduce two additional ingredients to circumvent this hurdle: *ramp hash families* and *leader based* threshold secret sharing schemes.

Ramp Hash Families. Inspired by the ramp secret sharing literature [BM84, KOS$^+$93] and *covering* hash families as defined in [ADM$^+$99], we define ramp hash families as weaker analogues of perfect hash families.

Definition 3 (**Ramp hash families**). *A family of hash functions $H = \{h : [n] \to [p]\}$ is called a (p, t, n)-ramp hash function family if for all subsets $T \subseteq [n]$ of cardinality t, there exists a function h in the family such that h is surjective on T — that is, $\{h(i) : i \in T\} = [p]$.*

Perfect hash families correspond to (p, p, n)-ramp hash families and necessarily need to have size at least $2^{\Omega(p)} \log n$ [FK84]. But, owing to a "coupon collector" phenomenon, if $t > Cp \log p$, then there exist (p, t, n)-ramp hash families with size $\mathrm{poly}(p)(\log n)$. Intuitively, if we fix a single set T of size $Cp\log p$, then a random hash function will be surjective with high probability, and one can then use the probabilistic method to argue existence of (p, t, n)-ramp hash functions.

Such a property was first studied as *covering* in the work of Alon et al. [ADM+99], who asked for the stronger requirement that a random hash function from H be surjective on any fixed set T with high probability. We will use explicit efficient construction of such families of size $\mathrm{poly}(\log n, p)$ due to Meka, Reingold, and Zhou [MRZ14].

Given ramp hash families as above, we can implement the second step of [KMS19] (which in turn is based on a classical idea of Kurosawa and Stinson from 90s– [Bla99, Des98]) to get "ramp" secret sharing schemes that are leakage resilient for $p = O(t/\log t)$ but satisfy a weaker secrecy guarantee. Concretely, while any t parties can recover the secret, no p-parties can reconstruct the secret. However, some set of $p+1$ shares may reveal the secret whereas we need to ensure that no $t-1$ parties can learn anything about the secret.

Stronger Leader-based t-out-of-n schemes. To fix the secrecy issue we rely on our notion of leader based scheme, albeit a stronger one. Apart from the reconstruction property as in the disjoint case, now we also require that the transcript of any $(p, n, \mu)-\mathsf{BCP}$, along with all but the leader's shares, reveal nothing about the underlying secret. To achieve this, we need to strengthen our communication complexity lower bounds in our Theorem 1 to also hold when we additionally allow one set of $n-1$ parties to collude, apart from the usual p-party collusion. Fortunately, our techniques easily generalize to this, and we prove this in appendix A. We can then proceed as in the disjoint case, and use these leader-based schemes to get regular t-out-of-n SS schemes, proving leakage-resilience by an appropriately modified hybrid argument.

1.2 Open Problems

Improved cylinder-intersection extractors. There has been a lot of recent progress on independent source extractors. Unfortunately, those techniques do not seem to work in our setting, and we leave it open to construct cylinder-intersection extractors for min-entropy rate below 0.3.

Lifting theorems for number-on-forehead model. Two source extractors have been useful for obtaining query-to-communication lifting theorems for the case of two parties [GLM+16]. It is an interesting research direction to use our new cylinder-intersection extractors to obtain lifting theorems for the multi-party case.

Reduce the gap between p and t for LRSS. We designed (p, t, n)-LRSS for p up to $O(t/\log t)$ owing to barriers coming from ramp hash-families. It would be interesting to reduce this gap, possibly by directly designing threshold schemes that do not rely on such hash families.

Leakage-resilient multi-party computation (MPC). Goyal et al. [GIM+16] design two party leakage-resilient MPC protocols, and leave open the design for higher number of parties. It may be interesting to explore this as a possible application of our LRSS schemes.

Joint leakage in non-malleable schemes. The compiler of [BFV19], when invoked with our LRSS scheme, yields a continuous NMSS scheme assuming setup and strong computational assumptions. While computational assumptions are necessary for continuous non-malleability, they are no longer necessary when the adversary only tampers a bounded many number of times (or say one time). We leave unconditional constructions of non-continuous schemes as an open problem.

2 Exponential Sums

Exponential sums play a central role in number theory and algebra. They also have several applications in complexity theory, especially in the literature on extractors ([BNS92, Bou05, KRVZ06, HH09, Bou09]. We first introduce some notation.

2.1 Notation

Let $\mathbb{F}_q$ be a prime field of cardinality q. Let $e_q(x) = \exp(2ix\pi/q)$. Let $[n]$ denote the set of integers $\{1, \ldots, n\}$. Let 2^A denote the power-set of A. We call a set of subsets $\mathcal{A} \in 2^{[n]}$ a collection. We use capital letters to denote distributions and their support, and corresponding small letters to denote a sample from the distribution. For any set $B \subseteq [n]$, let $\otimes_{i \in B} S_i$ denote the Cartesian product $S_{i_1} \times S_{i_2} \times \ldots \times S_{i_{|B|}}$, where $i_1, i_2 \ldots i_{|B|}$ are ordered elements of B, such that $i_j < i_{j+1}$. Let $X_1, \ldots, X_n$ be subsets of $\mathbb{F}_q$. For a subset $S \subseteq [n]$, we use X_S to denote the restriction of $(X_1, \ldots, X_n)$ to indices corresponding to S, namely $X_S \leftarrow \otimes_{i \in S} X_i$. Similar notation is used for elements $x = (x_1, \ldots, x_n) \in (X_1, \ldots, X_n)$, namely $x_S \leftarrow \otimes_{i \in S} x_i$.

To motivate what comes next, we first recall a fundamental estimate — Vinogradov's inequality — from number theory on *bilinear exponential sums.*

Lemma 1. *For $X_1, X_2 \subseteq \mathbb{F}_q$,*

$$\left| \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} e_q(x_1 x_2) \right| \leq \sqrt{q |X_1| |X_2|}$$

The above is an example of an exponential sum estimate, where the goal typically is to bound the final sum in terms of the sizes of the sets. Motivated by our applications, we study a substantial generalization of sums as above. In particular, we will allow multiple sets, weights, as well as *joint* functions. We then compare our definition to prior generalizations.

For brevity, we call functions $\phi : \mathbb{F}_q^n \to \mathbb{C}$ with $|\phi(x)| \leq 1$ *weight functions.* For $S \subseteq [n]$, $\phi_S : \mathbb{F}_q^S \subseteq [n]$ denotes a weight function that depends only on coordinates in S.

Motivated by our applications, we define the following collection of subsets:

Definition 4. *((k, n)-**Subset-Avoiding Collection**) A collection $\mathcal{A} \subseteq 2^{[n]}$ is called a (k, n)-subset-avoiding collection , if there exists a subset $S \subseteq [n]$ of cardinality k such that $S \nsubseteq T$ for all subsets $T \in \mathcal{A}$.*

Definition 5. *(**Exponential sums for** (k, n)**-subset-avoiding collection**) Fix $k \leq n$ and $X = \otimes_{i=1}^{n} X_i \subseteq \mathbb{F}_q^n$. We define the weighted exponential sum for (k, n)-subset-avoiding collection , Δ_k, as follows:*

$$\Delta_k(X) = \max \left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \left(\prod_{S \in \mathcal{A}} \phi_S(x_S) \right) e_q \left(\eta \prod_{i \in [n]} x_i \right) \right|,$$

where the maximum is over all $\eta \in \mathbb{F}_q^*$, (k,n)-subset-avoiding collection $\mathcal{A} \subseteq 2^{[n]}$, and weight functions[11] $\phi_S : \mathbb{F}_q^S \to \mathbb{C}$ *for* $S \in \mathcal{A}$.

A few remarks and some previous work:

- Taking $n = 2$ and identity weight functions gives Vinogradov's bilinear exponential sum of Lemma 1.
- Bourgain and Garaev [BG09] obtained upper bounds for trilinear exponential sums over three subsets ($n = 3$) where the weight functions are identity functions[12].
- Bourgain [Bou09] obtained upper bounds for multilinear exponential sums over n subsets of optimal size when the weight functions are singletons (depend on at most one coordinate).

In this work, we are interested in obtaining upper bounds for Δ_k, the weighted exponential sum for a (k,n)-subset-avoiding collection . To this end, we will be relying on upper bounds for weighted exponential sums for the special case of (k,k)-subset-avoiding collection .

2.2 Bounds for (k,n)-Subset-Avoiding Exponential Sums

In this section, we obtain an upper bound on $\Delta_k(X)$ for $X = \otimes_i X_i$ with a size bound on X_i. We do so in two steps:

- We first reduce the case of looking at general (k,n) to the *NOF* case of (k,k). This step relies on subset avoidance and the symmetry and *self-reducibility* of $e_q(\prod_i x_i)$ (i.e., fixing a few inputs leads to a function of a similar form).
- We then reduce bounding (k,k) to bounding $(k-1,k-1)$ using the *Cauchy-Schwarz trick* used by e.g., [BNS92]. Petridis and Shparlinski [PS19] and Kerr and Macourt [KM19] use a similar inductive argument based on [BNS92], but their bounds are a bit unwieldy to use directly in our context.

To facilitate the inductive arguments, we will use the following notation:

Definition 6. *For $k \leq n$ and $q \leq p$, let $\Delta_{k,n}(K) = \max \Delta_k(X)$ where the maximum is over all $X = \otimes_i X_i \subseteq (\mathbb{F}_q^*)^n$ with $|X_i| \leq K$* [13]. *Similarly, let $\delta_{k,n}(K) = \Delta_{k,n}(X)/K^n$.*

We can now state other previous work.

- Petridis and Shparlinski [PS19] defined the most powerful collections of weight functions $\{\phi_1, \ldots, \phi_n\}$, where each ϕ_i may depend on all inputs except x_i [14]. In our terminology, such sums correspond to $\Delta_{n,n}$. [PS19] obtained upper-bounds for $\Delta_{3,3}$ and $\Delta_{4,4}$.
- Extending Petridis and Shparlinski [PS19], Kerr and Macourt [KM19] obtained bounds on $\Delta_{n,n}$ for any $n \ll \log\log q$.

[11] Note that the weight functions only need to be defined for elements of $\otimes_{i\in S} X_i$. This distinction is not important for us and we hide it for clarity.

[12] There is no dependence on k if weight functions are identity.

[13] To simplify our calculations, we will work with subsets that only contain non-zero field elements. This only changes the final exponential sum by at most nK^{n-1} which is insignificant for our purposes.

[14] Equivalent to the number-on-forehead model from communication complexity literature

Lemma 2. *For $2 \leq k \leq n$, we have $\delta_{k,n}(K) \leq \delta_{k,k}(K)$.*

Proof. Our idea at a high level is to transform weight functions for a (k,n)-subset-avoiding collection to create weight functions for (k,k)-subset-avoiding collection . Details follow.

We wish to upper bound the following:

$$\Delta_k(K) \leftarrow \max_{\eta,\mathcal{A},\phi,|X_i|\leq K} \left| \sum_{x_1\in X_1} \cdots \sum_{x_n\in X_n} \left(\prod_{S\in\mathcal{A}} \phi_S(x_S) \right) e_q\left(\eta \prod_{i\in[n]} x_i \right) \right|$$

Fix any $\eta \in \mathbb{F}_q^*$, any (k,n)-subset-avoiding collection $\mathcal{A}$, weight functions $\{\phi_S : S \in \mathcal{A}\}$, and $X = (X_1, \ldots, X_n)$ with $|X_i| \leq K$.

By definition of (k,n)-subset-avoiding collection , there is a subset $J \subseteq [n]$ of cardinality k such that J is not contained in any element of $\mathcal{A}$. Fix such a subset $J = \{j_1, \ldots, j_k\}$. For each $i \in [k]$, define new collections $\mathcal{A}_i$ as follows:

$$\mathcal{A}_i = \{S \in \mathcal{A} : \min(J \setminus S) = j_i\}.$$

Consider a fixing of all x_ℓ for $\ell \notin J$. We now treat the exponential sum as a function of variables $((x_j : j \in J))$. For each $i \in [k]$, define a new weight function $\bar{\phi}_i$ as follows:

$$\bar{\phi}_i(x_{J\setminus\{j_i\}}) = \prod_{S\in\mathcal{A}_i} \phi_S(x_S).$$

By construction we have

$$\prod_{S\in\mathcal{A}} \phi_S(x_S) \equiv \prod_{i\in[k]} \bar{\phi}_i(x_{J\setminus\{j_i\}})$$

where $\bar{\phi}_i$ are valid weight functions for a (k,k)-subset-avoiding collection . Let $\zeta \leftarrow \prod_{i\in[n]\setminus J} x_i$. We now use such weight functions to upper bound Δ_k.

$$\begin{aligned} \Delta_k(X) &= \left| \sum_{x_1\in X_1} \cdots \sum_{x_n\in X_n} \left(\prod_{S\in\mathcal{A}} \phi_S(x_S) \right) e_q\left(\eta \prod_{i\in[n]} x_i \right) \right| \\ &= \left| \sum_{\{x_i\in X_i : i\in[n]\setminus J\}} \sum_{\{x_i\in X_i : i\in J\}} \left(\prod_{S\in\mathcal{A}} \phi_S(x_S) \right) e_q\left(\eta \prod_{i\in[n]} x_i \right) \right| \end{aligned}$$

By the triangle inequality, we obtain

$$\leq \sum_{\{x_i\in X_i : i\in[n]\setminus J\}} \left| \sum_{\{x_i\in X_i : i\in J\}} \left(\prod_{S\in\mathcal{A}} \phi_S(x_S) \right) e_q\left(\eta \prod_{i\in[n]} x_i \right) \right|$$

Using previous observation and the definition of ζ, we get,

$$= \sum_{\{x_i\in X_i : i\in[n]\setminus J\}} \left| \sum_{\{x_i\in X_i : i\in J\}} \left(\prod_{i\in[k]} \bar{\phi}_i(x_{J\setminus\{j_i\}}) \right) e_q\left(\zeta\eta \prod_{i\in[k]} x_{j_i} \right) \right|$$

Notice the term on the right is upper bounded by $\Delta_{k,k}(K)$, therefore,

$$\begin{aligned}&\leq \sum_{\{x_i \in X_i : i \in [n]\setminus J\}} \Delta_{k,k}(K)\\ &\leq K^{n-k}\Delta_{k,k}(K)\end{aligned}$$

The claim now follows. □

We next bound $\delta_{k+1,k+1}$ in terms of $\delta_{k,k}$ for $k \geq 2$.

Lemma 3. *For any $k \geq 2$, we have $\delta_{k+1,k+1}(K) \leq \frac{1}{\sqrt{K}} + \sqrt{\delta_{k,k}(K)}$.*

Proof. We wish to upper bound the following:

$$\Delta_{k+1,k+1} \leftarrow \max_{\eta,\mathcal{A}_{k+1},\phi,|X_i|\leq K} \left| \sum_{x_1\in X_1} \cdots \sum_{x_{k+1}\in X_{k+1}} \left(\prod_{S\in\mathcal{A}_{k+1}} \phi_S(x_S) \right) e_q\left(\eta \prod_{i\in[k+1]} x_i \right) \right|,$$

where A_{k+1} is $(k+1,k+1)$-subset avoiding. Fix $\eta \in \mathbb{F}_q^*$, a $(k+1,k+1)$-subset-avoiding collection $\mathcal{A}_{k+1}$, weight functions $\{\phi_S : S \in \mathcal{A}_{k+1}\}$, and X_i with $|X_i| \leq K$ that maximize the above. Note that without loss of generality, we can assume that $\mathcal{A}_{k+1}$ only consists of $k+1$ subsets, namely $\{[k+1]\setminus\{i\} : i \in [k+1]\}$, by considering $k+1$ weight functions $(\phi_1,\ldots,\phi_{k+1})$ where ϕ_i does not depend on x_i. For any $x_1,\ldots,x_{k+1} \in X_1,\ldots,X_{k+1}$, we use x to denote $x_1,\ldots,x_{k+1}$. For clarity of presentation, we slightly abuse notation and write $\phi_i(x)$ to denote $\phi_i(x_{[k+1]\setminus\{i\}})$ (even though $\phi_i(x)$ does not depend on x_i).

Our goal would be transform $\Delta_{k+1,k+1}$ into $\Delta_{k,k}$. At a very high level this will be achieved using a Cauchy-Schwarz argument as in [BNS92], while additionally taking care of the smaller subset sizes,

$$(\Delta_{k+1,k+1})^2 = \left(\sum_{x_1\in X_1} \cdots \sum_{x_{k+1}\in X_{k+1}} \left(\prod_{i\in[k+1]} \phi_i(x) \right) e_q\left(\eta \prod_{i\in[k+1]} x_i \right) \right)^2$$

Using the fact that $\phi_{k+1}(x)$ does not depend on x_{k+1}, we get

$$\leq \left(\sum_{x_1\in X_1} \cdots \sum_{x_k\in X_k} \phi_{k+1}(x) \left(\sum_{x_{k+1}\in X_{k+1}} \left(\prod_{i\in[k]} \phi_i(x) \right) e_q\left(\eta \prod_{i\in[k+1]} x_i \right) \right) \right)^2$$

By triangle inequality, we obtain,

$$\leq \left(\sum_{x_1\in X_1} \cdots \sum_{x_k\in X_k} |\phi_{k+1}(x)| \left| \sum_{x_{k+1}\in X_{k+1}} \left(\prod_{i\in[k]} \phi_i(x) \right) e_q\left(\eta \prod_{i\in[k+1]} x_i \right) \right| \right)^2$$

As $|\phi_{k+1}(x)| \leq 1$ (by definition of weight function), we get

$$\leq \left(\sum_{x_1\in X_1} \cdots \sum_{x_k\in X_k} \left| \sum_{x_{k+1}\in X_{k+1}} \left(\prod_{i\in[k]} \phi_i(x) \right) e_q\left(\eta \prod_{i\in[k+1]} x_i \right) \right| \right)^2$$

By Cauchy-Schwarz inequality, the above quantity is at most,

$$\leq \left(\sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} 1 \right) \left(\sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left(\sum_{x_{k+1} \in X_{k+1}} \left(\prod_{i \in [k]} \phi_i(x) \right) e_q \left(\eta \prod_{i \in [k+1]} x_i \right) \right)^2 \right)$$

as each subset X_i has at most K elements, we get

$$\leq K^k \left(\sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left(\sum_{x_{k+1} \in X_{k+1}} \left(\prod_{i \in [k]} \phi_i(x) \right) e_q \left(\eta \prod_{i \in [k+1]} x_i \right) \right)^2 \right)$$

Next, for notational convenience, we let $y = x_1, \ldots, x_k, y_{k+1}$ and expand the square (using the notation that for any function f, $\bar{\mathsf{f}}$ denotes its complex conjugate),

$$\leq K^k \left(\sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \sum_{x_{k+1}, y_{k+1} \in X_{k+1}} \left(\prod_{i \in [k]} \phi_i(x) \bar{\phi}_i(y) \right) e_q \left(x_{k+1} \eta \prod_{i \in [k]} x_i \right) \bar{e}_q \left(y_{k+1} \eta \prod_{i \in [k]} x_i \right) \right)$$

Next, we use the property of an additive character that $e_q(a)\bar{e_q}(b) = e_q(a-b)$. Moreover, for each $i \in [k]$ and $x_{k+1}, y_{k+1} \in X_{k+1}$, we define a new weight function $\phi_i^{x_{k+1}, y_{k+1}}$ that on input $x_{[k] \setminus \{i\}}$ uses the hard-coded x_{k+1}, y_{k+1} to compute and output $\phi_i(x)\bar{\phi}_i(y)$. Note that as defined, they are valid weight functions for a (k,k)-subset-avoiding collection .

$$= K^k \left(\sum_{x_{k+1}, y_{k+1} \in X_{k+1}} \sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left(\prod_{i \in [k]} \phi_i^{x_{k+1}, y_{k+1}}(x_{[k]}) \right) e_q \left((x_{k+1} - y_{k+1}) \eta \prod_{i \in [k]} x_i \right) \right)$$

By the triangle inequality,

$$\leq K^k \left(\sum_{x_{k+1}, y_{k+1} \in X_{k+1}} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left(\prod_{i \in [k]} \phi_i^{x_{k+1}, y_{k+1}}(x_{[k]}) \right) e_q \left((x_{k+1} - y_{k+1}) \eta \prod_{i \in [k]} x_i \right) \right| \right)$$

Notice that for $x_{k+1} \neq y_{k+1}$, we get an instance of $\Delta_{k,k}(K)$, and for $x_{k+1} = y_{k+1}$, we use the fact that $e_q(0) = 1$ to obtain,

$$\leq K^k \left(\sum_{x_{k+1} = y_{k+1} \in X_{k+1}} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left(\prod_{i \in [k]} \phi_i^{x_{k+1}, y_{k+1}}(x_{[k]}) \right) \right| + \sum_{x_{k+1} \neq y_{k+1} \in X_{k+1}} \Delta_{k,k}(K) \right)$$

By the triangle inequality and definition of weight functions, we get

$$\leq K^k \left(\sum_{x_{k+1} = y_{k+1} \in X_{k+1}} \sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} \left| \left(\prod_{i \in [k]} \phi_i^{x_{k+1}, y_{k+1}}(x_{[k]}) \right) \right| + \sum_{x_{k+1} \neq y_{k+1} \in X_{k+1}} \Delta_{k,k}(K) \right)$$

$$\leq K^k \left(\sum_{x_{k+1} = y_{k+1} \in X_{k+1}} \sum_{x_1 \in X_1} \cdots \sum_{x_k \in X_k} 1 + \sum_{x_{k+1} \neq y_{k+1} \in X_{k+1}} \Delta_{k,k}(K) \right)$$

as each subset X_i has at most K elements, we finish the proof,

$$\begin{aligned}&\leq K^k \left(K^{k+1} + K * (K-1) * \Delta_{k,k}(K)\right)\\&\leq K^{k+2}\left(K^{k-1} + \Delta_{k,k}(K)\right)\end{aligned}$$

Therefore,

$$\Delta_{k+1,k+1}(K) \leq \sqrt{K^{2k+1} + K^{k+2}\Delta_{k,k}(K)} \leq K^{k+1/2} + K^{(k+2)/2}\sqrt{\Delta_{k,k}(K)}.$$

Thus, dividing by K^{k+1} on both sides we get

$$\delta_{k+1,k+1}(K) \leq \frac{1}{\sqrt{K}} + \sqrt{\delta_{k,k}(K)}.$$

□

Iterating the above lemma gives the following:

Lemma 4. *For $1 < a < k$, $\delta_{k,k}(K) \leq k/K^{1/2^{k-a}} + \delta_{a,a}(K)^{1/2^{k-a}}$.*

Proof. Use previous lemma iteratively:

$$\begin{aligned}\delta_{k+1,k+1}(K) &\leq \frac{1}{\sqrt{K}} + \sqrt{\delta_{k,k}(K)}\\&\leq \frac{1}{\sqrt{K}} + \left(\frac{1}{\sqrt{K}} + \sqrt{\delta_{k-1,k-1}(K)}\right)^{1/2}\\&\leq \frac{1}{K^{1/2}} + \frac{1}{K^{1/2^2}} + \delta_{k-1,k-1}(K)^{1/2^2}\\\cdots &\leq \frac{1}{K^{1/2}} + \frac{1}{K^{1/2^2}} + \cdots + \frac{1}{K^{1/2^{k+1-a}}} + \delta_{a,a}(K)^{1/2^{k+1-a}}.\end{aligned}$$

The claim now follows. □

Finally, we state the most suitable bounds of Petridis and Shparlinksi [PS19] and Kerr and Macourt [KM19] that we will use[15] for small constants (3,4,6). In our notation, their results translate to:

Theorem 5. *For some constant $C > 0$,*

- *[PS19]: $\Delta_{3,3}(K) \leq Cq^{1/8}K^{43/16}$ and $\Delta_{4,4}(K) \leq Cq^{1/16}K^{61/16}$.*
- *[KM19]: $\Delta_{6,6}(K) \leq Cq^{1/64}K^{3045/512+o(1)}$*

In particular, we get:

Corollary 1. *For some universal constant C, $\delta_{3,3}(q^{0.401}) \leq Cq^{-1/320}$, $\delta_{4,4}(q^{0.34}) \leq Cq^{-1/200}$, and $\delta_{6,6}(q^{0.3}) \leq Cq^{-1/5120}$.*

[15] This will help us get the best min-entropy rate later.

3 Communication lower bounds against BCPs

In this section we prove Theorem 1. We will also prove a few extensions when a) the number of rounds in the BCP is also limited, and b) we also allow one additional round of an arbitrary communication on $n-1$ parties. The former gives better quantitative bounds in some natural cases and the latter is needed for our construction of LRSS.

A technicality is that while Theorem 1 was stated with inputs to each party being elements of $\{0,1\}^m$, we will on the other hand work with inputs to each party being elements of $\mathbb{F}_q$ for prime $q \approx 2^m$. We assume that we have access to such a prime[16]. Concretely, we show the following:

Theorem 6. *Fix a prime $q > 2$ and $1 \leq p < n$. Let $m = \log q$, and* BouExt : $\mathbb{F}_q^n \to \{0,1\}$ *be defined by*

$$\text{BouExt}(x_1, \ldots, x_n) = \mathsf{sign} \sin\left(\frac{2\pi \prod_{i \in [n]} x_i}{q}\right).$$

Then, any p-party BCP computing BouExt *requires $cm^{\frac{\log(n/p)}{\log(n/p)+1}}$ bits of communication where $c > 0$ is a universal constant.*

Theorem 6 is in turn proved using our exponential sums for (k,n)-avoiding collections.

3.1 Cylinder Intersections and Discrepancy

While Theorem 6 can be proved directly without looking at round bounded BCPs, we work in this slightly more general language as it is no more difficult and lends easily to natural extensions.

Definition 7. *((p,r,n)**-collusion protocol**) A (p,r,n)-collusion protocol consists of r round protocol amongst n parties where in each round at most p parties get together to pool their input to compute the next message.*

We are interested in understanding the minimum amount of communication required by any (p,r,n)-collusion protocol in order to compute the output of some function. We formalize this next.

Definition 8. *((p,r,n,ϵ)**-communication complexity**) Suppose an element of $\mathbb{F}$ is given to each of n parties, who wish to compute a n party predicate $\mathsf{f} : \mathbb{F}^n \to \{0,1\}$. The (p,r,n,ϵ)-communication complexity of f, $\mathsf{CC}_{p,r,n,\epsilon}(\mathsf{f})$, refers to the minimum number of bits of communication required to gain ϵ advantage in computing f using any (p,r,n)-collusion protocol .*

We now follow the approach of the seminal work of Babai, Nisan, and Szegedy [BNS92] who gave the first lower bounds for number-on-forehead (NOF) protocols. To this end, they showed equivalence in between NOF protocols and cylinder intersections (which they defined). After which, they obtained upper bounds on the discrepancy in any cylinder intersection, using which they obtained lower bounds on the NOF communication complexity. We will follow a similar approach, generalizing the definitions of [BNS92] as needed. We begin by recalling the definition of s-component of a protocol Π.

[16] We could potentially avoid this technicality by assuming Cramer's conjecture on primes or using part of the input to generate the prime at random (we only need average-case lower bounds). We do not delve into this issue here.

Definition 9. *([BNS92]) (**s-component of Protocol** Π) Let Π be a multiparty protocol on n parties and s be any transcript. The s-component, $X_{\Pi,s}$ is defined to be the set of n-tuples $x \in \mathbb{F}^n$ such that on input x the protocol Π results in exactly s being written on the board.*

Unlike [BNS92], we cannot think of $X_{\Pi,s}$ as cylinder-intersections as defined by [BNS92], since we limit ourselves to BCPs instead of NOF protocol. To overcome this, we consider the natural generalization of cylinder in i^{th} dimension to cylinder for subset S.

Definition 10. *(**Cylinder corresponding to subset**) A subset Y of n-tuples in called a cylinder corresponding to subset $S \subseteq [n]$ if membership in Y only depends on the coordinates in S.* [17]

Definition 11. *((p, r, n)**-cylinder-intersection**) A subset Y of n-tuples in called a (p, r, n)-cylinder-intersection if Y is the intersection of cylinders corresponding to at most r subsets $S \subseteq [n]$ where each subset S has cardinality at most p.*

Lemma 5. *For any (p, r, n)-collusion protocol Π and transcript s, the s-component $X_{\Pi,s}$ is a (p, r, n)-cylinder-intersection .*

Proof. Follows exactly as in the proof of Lemma 2.1 of [BNS92] replacing cylinder in i^{th} dimension with cylinder for subset S. □

The idea behind the lower bound of [BNS92] is that any protocol Π that computes f should be constant on any s-component (or cylinder-intersection by previous lemma). But any large cylinder-intersection for the function they look at is approximately balanced. This is captured standardly using the notion of discrepancy.

Definition 12. *((p, r, n)**-discrepancy**) Let $\mathsf{f} : \mathbb{F}^n \to \{0, 1\}$ be a boolean function. The (p, r, n)-discrepancy of f is*

$$\Gamma_{p,r,n}(f) = \max_{Y} |Pr[\mathsf{f}(x) = 1 \textit{ and } x \in Y] - Pr[\mathsf{f}(x) = 0 \textit{ and } x \in Y]|$$

where Y ranges over (p, r, n)-cylinder-intersection and x is chosen uniformly over $\mathbb{F}^n$.

Lemma 6. *For any function $\mathsf{f} : \mathbb{F}^n \to \{0, 1\}$,*

$$\mathsf{CC}_{p,r,n,\epsilon}(\mathsf{f}) \geq \log_2 \left(\frac{\epsilon}{\Gamma_{p,r,n}(\mathsf{f})} \right)$$

Proof. Follows exactly as in the proof of Lemma 2.2 of [BNS92] replacing cylinder-intersection with a (p, r, n)-cylinder-intersection . □

Next, we recall an observation from Bourgain [Bou05, Remark 3.3], see also [HH09]) [18], which can be used to obtain upper bounds on sums using BouExt in terms of upper bounds on exponential sums.

[17] The reader may notice that a cylinder corresponding to subset $[n-1]$ is equivalent to the cylinder in n^{th} dimension from [BNS92].

[18] Bourgain's $\mathsf{sign}(x)$ was defined to be -1 for $x < 0$. For binary output, we defined, $\mathsf{sign}(x) = 0$ for $x < 0$.

Lemma 7. *[Bou05] Let x denote $(x_1, \ldots, x_n) \in (\mathbb{F}_q)^n$. For any function ϕ that takes x as input, we have,*

$$\left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \phi(x)(-1)^{\text{BouExt}(x)} \right| \leq (C \log q) \max_{\eta \in F_q^*} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \phi(x) e_q \left(\eta \prod_{i \in [n]} x_i \right) \right|$$

for some universal constant C.

We next bound (p, r, n)-discrepancy using our exponential sums.

Lemma 8. *For any number of parties n, any collusion bound $p < n$, any round bound $r \leq (n/p)^k$, we have*

$$\Gamma_{p,r,n}(\text{BouExt}) \leq (C \log q) \delta_{k,n}(q).$$

for some universal constant C.

Proof. We wish to upper bound the following

$$\Gamma_{p,r,n}(\text{BouExt}) = \max_{Y} |Pr[\text{BouExt}(x) = 1 \text{ and } x \in Y] - Pr[\text{BouExt}(x) = 0 \text{ and } x \in Y]|$$

Fix Y to be any (p, r, n)-cylinder-intersection that maximizes the above. Without loss of generality, let Y be the intersection of at most r cylinders $Y_1, \ldots, Y_r$ corresponding to r subsets, namely S_1, $\ldots, S_r$, where each S_i has cardinality at most p.

Now we argue that the collections of sets $S_1, \ldots, S_r$ is a (k, n)-subset-avoiding collection . Total number of subsets of $[n]$ of cardinality k is $\binom{n}{k}$. Total number of subsets of cardinality k contained in any r subsets, each of cardinality at most p is at most $r\binom{p}{k}$. Therefore, if $r < \binom{n}{k}/\binom{p}{k}$, we have that there is some subset of cardinality k not covered by any $S_1, \ldots, S_r$. This is the case since $r \leq (n/p)^k$ which in turn is less than $\binom{n}{k}/\binom{p}{k}$.

Next, for each $1 \leq i \leq r$, and cylinder Y_i, define a corresponding weight function ϕ_i by setting $\phi_i(x_{S_i}) = 1$ if $x \in Y_i$ and 0 otherwise. Finally, note that the weight functions as defined above satisfy the conditions from the definition of $\Delta_{k,n}(q)$.

Now, observe that $|Pr[\text{BouExt}(x) = 1 \text{ and } x \in Y] - Pr[\text{BouExt}(x) = 0 \text{ and } x \in Y]|$ is in fact equal to,

$$(1/q^n) \left| \sum_{x_1 \in \mathbb{F}_q} \cdots \sum_{x_n \in \mathbb{F}_q} \prod_{i \in [r]} \phi_i(x_{S_i})(-1)^{\text{BouExt}(x)} \right|$$

and, by Bourgain's Lemma 7, the above is at most,

$$\leq \frac{C \log q}{q^n} \max_{\eta \in \mathbb{F}_q^*} \left| \sum_{x_1 \in \mathbb{F}_q} \cdots \sum_{x_n \in \mathbb{F}_q} \prod_{i \in [r]} \phi_i(x_{S_i}) e_q \left(\eta \prod_{i \in [n]} x_i \right) \right|$$

for some universal constant C. Moreover, as the exponential sum on the right is upper bounded by $\Delta_{k,n}(q)$, we obtain the following completing the proof.

$$|Pr[\text{BouExt}(x) = 1 \text{ and } x \in Y] - Pr[\text{BouExt}(x) = 0 \text{ and } x \in Y]| \leq \frac{C \log q \Delta_{k,n}(q)}{q^n} = (C \log q) \delta_{k,n}(q).$$

□

Putting things together.

Proof of Theorem 6. Let μ be the communication cost of BouExt by a p-party BCP. Let $k = \lceil (\log \mu)/(\log(n/p)) \rceil + 1$ be such that $\mu < (n/p)^k$. Then, as the number of rounds in any protocol with cost μ is at most μ, we can apply lemmas 6 and 8 (with $\epsilon = 1/3$ say) to get

$$\mu > \log\left(\frac{\epsilon}{(C\log q)\delta_{k,n}(q)}\right) = \log(1/\delta_{k,n}(q)) - O(\log\log q).$$

By Theorem 5, we have $\delta_{3,3}(q) \leq q^{-3/16}$. By Lemmas 3 and 4, we have $\delta_{k,n}(q) = O(q^{-\lambda/2^k})$, where $\lambda = 3/2$. Therefore, we get

$$\mu > \log(1/\delta_{k,n}(q)) - O(\log\log q) > \frac{c\log q}{2^k}$$

for a constant $c > 0$. The above in turn implies that

$$\mu^{1+1/(\log(n/p))} > c'(\log q),$$

for a constant $c' > 0$. The theorem now follows by rearranging for μ.

□

3.2 Extension allowing one set of $n-1$ parties to collude.

Recall that p-party BCP allows any set of p parties to collude in each round of communication. For our applications to leakage-resilient secret sharing, we need to slightly strengthen this communication model and additionally allow any one set of $n-1$ parties to jointly communicate as well. Fortunately, our techniques can be easily generalized to encompass this, and we defer the proof to Appendix A. In particular, our Theorem 6 can be generalized to obtain the following:

Lemma 9. *Fix a prime $q > 2$ and $1 \leq p < n$. Let $m = \log q$, and* BouExt : $\mathbb{F}_q^n \to \{0,1\}$ *be defined by*

$$\text{BouExt}(x_1, \ldots, x_n) = \mathsf{sign}\sin\left(\frac{2\pi\prod_{i\in[n]} x_i}{q}\right).$$

Then, using any p-party protocol that possibly includes collusion of one set of $n-1$ parties, requires $cm^{\frac{\log(n/p)}{\log(n/p)+1}}$ bits of communication to compute BouExt, *where $c > 0$ is an universal constant.*

4 Cylinder Intersection Extractors

In this section, we build up on techniques developed in the preceding sections and prove Theorem 4. We begin by recalling the definition of statistical distance, which is also known as total variation distance.

Definition 13. *(**Statistical distance**) Let D_1 and D_2 be two distributions on a set S. The statistical distance between D_1 and D_2 is defined to be :*

$$|\mathsf{D}_1 - \mathsf{D}_2| = \max_{T\subseteq S}|\mathsf{D}_1(T) - \mathsf{D}_2(T)| = \frac{1}{2}\sum_{s\in S}|Pr_{X\sim\mathsf{D}_1}[X=s] - Pr_{X\sim\mathsf{D}_2}[X=s]|$$

We say D_1 is ϵ-close to D_2 if $|\mathsf{D}_1 - \mathsf{D}_2| \leq \epsilon$. Sometimes we represent the same using $\mathsf{D}_1 \approx_\epsilon \mathsf{D}_2$. We say $\mathsf{D}_1 \not\approx_\epsilon \mathsf{D}_2$ when $|\mathsf{D}_1 - \mathsf{D}_2| > \epsilon$.

Theorem 7. *Fix a prime $q > 2$. Let $m = \log q$, and* $\text{BouExt} : \mathbb{F}_q^n \to \{0,1\}$ *be defined by*

$$\text{BouExt}(x_1, \ldots, x_n) = \mathsf{sign} \sin\left(\frac{2\pi \prod_{i\in[n]} x_i}{q}\right).$$

Then, for all fraction $\alpha < 1$, for all $n \geq 6$, BouExt *is a $(\alpha n, n, \mu)$-cylinder intersection extractor with error ϵ for all $(m, \delta m)$-sources for $\delta = 0.3$, $\mu < cm^{c_\alpha}$ and $\epsilon = 2^{-\Omega(m^{c_\alpha})}$, where $c_\alpha = \log(1/\alpha)/(1 + \log(1/\alpha))$.*

Proof. We begin with the observation of Chor and Goldreich [CG88], that any source X_i distributed on $\mathbb{F}_q$ with min-entropy rate δ is a convex combination of uniform sources on q^δ sized subsets $X_i \subseteq \mathbb{F}_q$. Therefore, we only need to focus on q^δ sized subsets. Fix any $X = \otimes_i X_i \subseteq (\mathbb{F}_q)^n$ such that $|X_i| = q^\delta$ for each $i \in [n]$. We can further assume that $0 \notin X_i$ for all $i \in [n]$. Because, this only adds at most $O(n/q^\delta)$ to the final statistical error.

Let $p = \alpha n$. Fix any (p, n, μ)-BCP Π. Let Γ be the set of transcripts that can be produced by executing Π on some $x = (x_1, \ldots, x_n) \in X$. Recall the notion of a τ-component from Definition 11: $X_{\Pi,\tau}$ denotes the set of $x \in X$ that result in transcript τ when protocol Γ is executed on x.Therefore, by design, for each $\tau \in \Gamma$, we have that $|X_{\Pi,\tau}| \geq 1$. Moreover, by Lemma 5, for each transcript τ, τ-component $X_{\Pi,\tau}$ is a (p, n, μ)-cylinder-intersection . For each $\tau \in \Gamma$, let the corresponding (p, n, μ)-cylinder-intersection be denoted by $\phi^\tau = \{\phi_S^\tau : S \in \mathcal{A}^\tau\}$ for collection $\mathcal{A}^\tau$.

To show that BouExt is a (p, n, μ)-cylinder intersection extractor with error ϵ, it suffices to upper bound the following

$$|(\text{BouExt}(X), \Pi(X)) - (U_1, \Pi(X))|$$

where $X = (X_1, \ldots, X_n)$, and each X_i is uniformly distributed over some subset size q^δ. By definition of statistical distance, this is equal to,

$$= \frac{1}{2} \sum_{b\in\{0,1\}} \sum_{\tau\in\Gamma} \left| \Pr_X[\Pi(X) = \tau \text{ and } \text{BouExt}(X) = b] - \Pr_X[\Pi(X) = \tau \text{ and } U_1 = b] \right|$$

$$= \frac{1}{2} \sum_{b\in\{0,1\}} \sum_{\tau\in\Gamma} \left| \Pr_X[\Pi(X) = \tau] \Pr_X[\text{BouExt}(X) = b | \Pi(X) = \tau] - \frac{1}{2} \Pr_X[\Pi(X) = \tau] \right|$$

This can be simplified to,

$$= \frac{1}{2} \sum_{\tau\in\Gamma} \Pr_X[\Pi(X) = \tau] \sum_{b\in\{0,1\}} \left| \Pr_X[\text{BouExt}(X) = b | \Pi(X) = \tau] - \frac{1}{2} \right|$$

$$= \frac{1}{2} \sum_{\tau\in\Gamma} \Pr_X[\Pi(X) = \tau] \left| \Pr_X[\text{BouExt}(X) = 1 | \Pi(X) = \tau] - \Pr_X[\text{BouExt}(X) = 0 | \Pi(X) = \tau] \right|$$

Next, note that the condition $\Pi(X) = \tau$ is equivalent to X being in τ-component. More formally, $X \in X_{\Pi,\tau}$. This in turn is equivalent to X being in the corresponding cylinder-intersection ϕ^τ. More formally, $\prod_{S\in\mathcal{A}^\tau} \phi_S^\tau(X) = 1$. Substituting we get,

$$= \frac{1}{2} \sum_{\tau\in\Gamma} \Pr_X[\Pi(X) = \tau] \left| \Pr_X\left[\text{BouExt}(X) = 1 \middle| \prod_{S\in\mathcal{A}^\tau} \phi_S^\tau(X_S) = 1\right] - \Pr_X\left[\text{BouExt}(X) = 0 \middle| \prod_{S\in\mathcal{A}^\tau} \phi_S^\tau(X_S) = 1\right] \right|$$

Using the observation from the proof of Lemma 8, the above is equivalent to the following, where $x = (x_1, \ldots, x_n)$,

$$= \frac{1}{2} \sum_{\tau \in \Gamma} \Pr_X [\Pi(X) = \tau] \frac{1}{|X_{\Pi,\tau}|} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \prod_{S \in \mathcal{A}^\tau} \phi_S^\tau(x_S)(-1)^{\mathrm{BouExt}(x)} \right|$$

Moreover, as we have uniform distribution over X, $\Pr_X [\Pi(X) = \tau]$ is equal to $|X_{\Pi,\tau}|/|X|$. Plugging this, the above simplifies to,

$$= \frac{1}{2} \sum_{\tau \in \Gamma} \frac{1}{|X|} \left| \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \prod_{S \in \mathcal{A}^\tau} \phi_S^\tau(x_S)(-1)^{\mathrm{BouExt}(x)} \right|$$

We can now proceed as in Lemma 8. Let $k = \lceil \log(\mu)/\log(1/\alpha) \rceil + 1$. Then, $\mu < (n/p)^k$ so that A^τ is a (k, n)-subset-avoiding collection . We can then use the connection to exponential sums (Lemma 7) to obtain the following bound on the above quantity:

$$\leq \frac{1}{2} \sum_{\tau \in \Gamma} (C \log q) \cdot \delta_{k,n}(q^\delta).$$

As there can be at most 2^μ transcripts in Γ, we get

$$|(\mathrm{BouExt}(X), \Pi(X)) - (U_1, \Pi(X))| \leq C(\log q) 2^\mu \delta_{k,n}(q^\delta).$$

Now using Lemma 4 and Corollary 1, we get for $k \geq 6$,

$$\delta_{k,k}(q^\delta) \leq q^{-\Omega(1)/2^k} = 2^{-\Omega(\log q)/\mu^{1/\log(1/\alpha)}}.$$

Now, for $c > 0$ sufficiently small, and $\mu < c(\log q)^{\log(1/\alpha)/(1+\log(1/\alpha))}$, the above bound simplifies to

$$|(\mathrm{BouExt}(X), \Pi(X)) - (U_1, \Pi(X))| \leq \exp\left(-\Omega(1)(\log q)^{\frac{\log(1/\alpha)}{1+\log(1/\alpha)}} \right)$$

proving the claim. □

5 Disjoint Leakage-Resilient Secret Sharing

In this section we present our leakage resilient secret sharing scheme and prove Theorem 3. Blakley [Bla79] and Shamir [Sha79] initiated the study of threshold secret sharing schemes which were generalized to access structures by Ito, Saito, and Nishizeki [ISN89]. We begin by recalling the definition of such schemes from the survey [Bei11].

Definition 14. *(**Access structures and sharing function**) A collection $\mathcal{A}$ is called monotone if $B \in \mathcal{A}$ and $B \subseteq C$, then $C \in \mathcal{A}$. Let $[n] = \{1, 2, \ldots, n\}$ be a set of identities of n parties. An **access structure** is a monotone collection $\mathcal{A} \subseteq 2^{\{1,\ldots,n\}}$ of non-empty subsets of $[n]$. Sets in $\mathcal{A}$ are called **authorized**, and sets not in $\mathcal{A}$ are called **unauthorized**.*

Let $\mathcal{M}$ be the domain of secrets. A ***sharing function*** Share *is a randomized mapping from $\mathcal{M}$ to $S_1 \times \ldots \times S_n$, where S_i is called the domain of shares of party with identity i. A dealer distributes a secret $m \in \mathcal{M}$ by computing the vector* $\mathsf{Share}(m) = (s_1, \ldots, s_n)$*, and privately communicating each share s_i to the party i.*

Definition 15. *(**Secret sharing scheme** [Bei11]). Let $\mathcal{M}$ be a finite set of secrets, where $|\mathcal{M}| \geq 2$. A sharing function* Share *with domain of secrets $\mathcal{M}$ is a **Secret Sharing Scheme** realizing an access structure $\mathcal{A}$ if the following two properties hold:*

1. ***Correctness**. The secret can be reconstructed by any authorized set of parties. That is, for any set $T \in \mathcal{A}$, there exists a deterministic* reconstruction function $\mathsf{Rec} : \otimes_{i \in T} S_i \to \mathcal{M}$ *such that for every $m \in \mathcal{M}$,*
$$Pr[\mathsf{Rec}(\mathsf{Share}(m)_T) = m] = 1$$
(over the randomness of the Sharing function)

2. ***Perfect privacy**. Collusion of unauthorized parties should reveal no information about the underlying secret. More formally, for any unauthorized set $T \notin \mathcal{A}$, and for every pair of secrets $a, b \in \mathcal{M}$, the following holds :*
$$\mathsf{Share}(a)_T \equiv \mathsf{Share}(b)_T$$
*If the two distributions are statistically (resp. computationally) indistinguishable, we call it **statistical** (resp. **computational**) privacy.*

All our constructions will be efficient in the sense that the share-length of the new schemes will be at most a polynomial factor more than those of the original schemes. The same holds for the encoding and reconstruction procedures.

Access Structure for t-out-of-n schemes. Perhaps the most well-studied secret sharing scheme is the threshold secret sharing scheme originally studied by Shamir and Blakley. The threshold access structure can be formally represented by $\mathcal{A}_t = \{B \subseteq [n] : |B| \geq t\}$. We use t-out-of-n to denote secret sharing schemes that realize such threshold access structures.

While one can achieve perfect secrecy, in case an adversary is allowed to leak from all the shares, only statistical leakage-resilience is possible.

Leakage-Resilient Secret Sharing. Leakage-resilience in the context of secret sharing was studied by Dziembowski and Pietrzak [DP07], Goyal and Kumar [GK18a] and Benhamouda et al. [BDIR18] against non-adaptive adversaries. Kumar, Meka and Sahai [KMS19] generalized the adversary substantially by modelling *leakage* as an adversary running a communication protocol among the n parties and trying to guess the secret based on the transcript. We recall their definition below.

Definition 16. *(**Leakage-resilient secret sharing schemes**) Let $\mathcal{M}$ be any secret space. Let $\mathcal{L}$ be a family of (possibly randomized) multi-party protocols that output some transcript. We say that a secret sharing scheme (*$\mathsf{Share}, \mathsf{Rec}$*) is **$\epsilon$-leakage-resilient** w.r.t. $\mathcal{L}$ if for every leakage-protocol* $\mathsf{Leak} \in \mathcal{L}$*, and for every pair of secrets $a, b \in \mathcal{M}$, the following holds :*
$$\mathsf{Leak}(\mathsf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathsf{Share}(b)).$$

That is, the distribution of the transcript of the protocol Leak *when input is* $\mathsf{Share}(a)$ *is statistically close to the distribution of the transcript of the protocol when input is* $\mathsf{Share}(b)$*.*

To facilitate our proof of leakage-resilience, we restate the class of p-party collusion protocols as a leakage-family as in [KMS19].

Bounded Collusion Protocols $(p, n, \mu)-\mathsf{BCP}$**.** Let (p, n, μ) denote *collusion bound, number of parties, and leakage bound* respectively. At a very high level, the leakage family $(p, n, \mu)-\mathsf{BCP}$ contains all possible leakage-protocols among n parties such that at most μ bits are leaked and each leaked bit arbitrarily depends on the shares of parties in any subset of size at most p (along with all the leakages obtained in the preceding rounds). We formally model this in the following way:

- Let $share_1, \ldots, share_n$ be the n shares corresponding to n parties. We use τ to denote the transcript of the leakage-protocol. At the beginning of the leakage-protocol τ is empty. The transcript τ is appended with the leakage from any subset of size at most p, at the end of each round of the leakage-protocol. At the end, τ can be at most μ bits long.
- In each round, the Next function is used to determine which subset of parties will collude to jointly leak information about their shares. Formally, Next function takes the current transcript τ as input, and outputs a subset $S \subseteq [n]$ and a description of an arbitrary leakage function f that takes $\otimes_{i \in S} share_i$ as input. Note that f may possibly depend on τ. At the end of each round, the leaked information is appended to the current transcript.
$$\tau \leftarrow \tau \circ \mathsf{f}(\otimes_{i \in S} share_i)$$
- The previous step is repeated until the Next function outputs $\perp$. Output final transcript τ as leakage.

5.1 Pruning a party from Access Structures

To build our general compiler, we need a method to convert a secret sharing scheme to another secret sharing on a smaller number of parties realizing an appropriately chosen access structure as defined below.

Definition 17. ***(Access structure with party ℓ pruned).*** *For any access structure $\mathcal{A}$ on n parties, we define $\mathcal{A}^\ell$ as an access structure with party ℓ pruned. More formally,* [19]

$$\mathcal{A}^\ell \leftarrow \{T \subseteq [n] : T \cup \{\ell\} \in \mathcal{A}\}$$

Observe that the above definition is making the party i redundant. As a concrete example, pruning a party from the threshold t-out-of-n access structure gives a scheme with threshold $t - 1$.

Lemma 10. *For any secret sharing scheme* $(\mathsf{AShare}, \mathsf{ARec})$ *realizing access structure $\mathcal{A}$ that shares secrets of length a bits into n shares, each of length b bits, for any party $\ell \in [n]$, there is a secret sharing scheme that realizes $\mathcal{A}^\ell$, the access structure with party ℓ pruned. The resulting scheme,* $(\mathsf{AShare}^\ell, \mathsf{ARec}^\ell)$*, shares secrets of length a into n shares, each of length $2b$ bits.*

[19] Ideally, we would like to reduce the number of parties from n to $n-1$ as well. But for the ease of notation, we have chosen to retain the number of parties. So we can assume that the share given to party ℓ is an empty string $\perp$, which will never be used.

Proof. The construction of $(\mathsf{AShare}^\ell, \mathsf{ARec}^\ell)$ is given below:

- **Sharing function** AShare^ℓ:
  Encode the secret m using the given secret sharing scheme for access structure $\mathcal{A}$ to obtain $m^1, \ldots, m^n \leftarrow \mathsf{AShare}(m)$. Let $share_\ell \leftarrow \perp$. For each $i \in [n] \setminus \{\ell\}$, construct $share_i$ as (m^i, m^ℓ).

- **Reconstruction function** ARec^ℓ:
  Consider an authorized set $T \in \mathcal{A}^\ell$ such that $\ell \notin T$. On input the shares $\otimes_{i \in T} share_i$, for each $i \in T$, parse $share_i$ as (m^i, m^ℓ). Run ARec on the shares of m, to obtain $m \leftarrow \mathsf{ARec}(\otimes_{i \in T \cup \{\ell\}} m^i)$. Output m.

Correctness and Efficiency: Correctness follows from the observation that shares of any authorized set $T \in \mathcal{A}^\ell$ of our final scheme implies that either $T \in \mathcal{A}$ or $T \cup \{\ell\} \in \mathcal{A}$ of the underlying scheme $(\mathsf{AShare}, \mathsf{ARec})$. Efficiency trivially follows from the construction.

Perfect Privacy: Any unauthorized set U of the final scheme can only have information about $\{m^i : i \in U \cup \{\ell\}\}$ as the only additional information in each share is m^ℓ. But by definition of access structure with party ℓ pruned. $U \notin \mathcal{A}^\ell$ implies that $U \cup \{\ell\} \notin A$. Therefore, secret remains perfectly hidden by the perfect privacy of the underlying scheme $(\mathsf{AShare}, \mathsf{ARec})$.

□

5.2 Leader Based Leakage-Resilient Schemes

In the previous subsection, we saw how to efficiently prune away a party from an access structure. We now use such a compiler to build a leader-based leakage-resilient scheme that we will crucially use in our final construction. In particular, for any a-priori chosen leader $\ell \in [n]$, any access structure $\mathcal{A}$ such that leader ℓ is not authorized ($\{\ell\} \notin \mathcal{A}$), a scheme satisfying the following three properties is called an ϵ-leakage-resilient secret sharing scheme for leader ℓ corresponding to access structure $\mathcal{A}$.

1. Leader ℓ along with any authorized set corresponding to A^ℓ can efficiently reconstruct the secret.

2. Without the leader's share the secret is perfectly hidden. That is for two distinct messages a, b, $((\mathsf{Share}(a)_i : i \neq \ell \in [n])) \equiv ((\mathsf{Share}(b)_i : i \neq \ell \in [n]))$.

3. For any unauthorized $U \notin \mathcal{A}^\ell$, any protocol in between two subsets $[n] \setminus \{\ell\}$ and $U \cup \{\ell\}$ [20] having leakage-transcript of at most μ bits reveals statistically no information about the underlying secret. That is for any such protocol Leak with total communication μ and two distinct messages $a \neq b$, $\mathsf{Leak}(\mathsf{Share}(a)) \approx_\epsilon \mathsf{Leak}(\mathsf{Share}(b))$.

We first design schemes as above and then use them to get our disjoint leakage-resilient secret sharing schemes. This definition generalizes an idea in the recent work of Aggarwal et al. [ADN$^+$19], who implicitly designed leader based 2-out-of-n leakage-resilient schemes, while designing leakage-resilient scheme for general access structures against non-adaptive and individual leakage.

[20] For the purpose of disjoint leakage, appropriately restricting these two subsets to be disjoint would have sufficed as well. Our choice is for notational convenience.

Lemma 11. *Suppose we have the following primitives:*

1. *For any leakage bound μ, any error bound $\epsilon > 0$, an efficient 2-out-of-2 secret sharing scheme $(\mathsf{LRShare}_2^2, \mathsf{LRRec}_2^2)$ that is ϵ-leakage-resilient w.r.t. $(1, 2, \mu)-\mathsf{BCP}$. The scheme shares a secret of bit-length a into 2 shares, each of bit-length b.*

2. *For any leader $\ell \in [n]$, for any access structure $\mathcal{A}$ on n parties such that leader ℓ is not authorized, a secret sharing scheme $(\mathsf{AShare}^\ell, \mathsf{ARec}^\ell)$ that realizes the pruned access structure $\mathcal{A}^\ell$ and shares secrets of length b into n shares, each of length c bits.*

Then there is an efficient ϵ-leakage-resilient secret sharing scheme for leader ℓ corresponding to access structure $\mathcal{A}$. The resulting scheme, $(\mathsf{LDSh}^\ell, \mathsf{LDRec}^\ell)$, shares secrets of length a into n shares, each of length at most c.

Proof. We begin with the description of the scheme.

- (**Sharing function** LDSh^ℓ).
  On input a secret m, share m using the sharing procedure of underlying leakage-resilient scheme to obtain $(s, r) \leftarrow \mathsf{LRShare}_2^2(m)$. Run the sharing function for the pruned access structure on r to obtain $r_1, \ldots, r_n \leftarrow \mathsf{AShare}^\ell(r)$. Construct leader's share as $share_i \leftarrow s$. For everyone else, namely, for each $i \in [n] \setminus \{\ell\}$, construct $share_i$ as r_i.

- **Reconstruction function** (LDRec^ℓ).
  On input a set of shares corresponding to an authorized set T of cardinality t such that leader $\ell \in T$, for each $i \in T \setminus \{\ell\}$, parse $share_i$ as r_i. Parse leader's $share_\ell$ as s. Use the reconstruction procedure of the pruned access structure to obtain $r \leftarrow \mathsf{ARec}^\ell(\otimes_{T\setminus\{\ell\}} r_i)$. Use the reconstruction procedure of the underlying leakage resilient scheme to compute $m \leftarrow \mathsf{LRRec}_2^2(s, r)$. Output m.

Perfect correctness and Efficiency: Any authorized set $T \subseteq [n]$ contains leader ℓ, hence we have the share s of the 2-out-of-2 scheme. Moreover, by design, $T \setminus \{\ell\} \in \mathcal{A}^\ell$, therefore, r can be reconstructed. Hence, correctness follows from the correctness of the underlying 2-out-of-2 scheme.

Perfect Privacy without the leader: Without the leader ℓ, one of the two shares, namely s, of the undelrying 2-out-of-2 scheme will be missing, and therefore the secret will be perfectly hidden.

Statistical leakage-resilience: The adversary specifies a set $U \subseteq [n] \setminus \{\ell\}$ such that $U \cup \{\ell\} \notin \mathcal{A}$ and a leakage protocol specified by round function Next in between two the subsets $S_1 \leftarrow U \cup \{\ell\}$ and $S_2 \leftarrow [n] \setminus \{\ell\}$ that allows it to distinguish in between shares of m_1 and m_2 under our scheme using at most μ bits of communication. We use such an adversary to construct $\mathsf{Next}_1 \in (1, 2, \mu)-\mathsf{BCP}$ that violates the leakage-resilience of the underlying 2-out-of-2 scheme.

- **Initial setup** : Share 0 (or any arbitrary b bit string) using AShare^ℓ, the sharing procedure of the pruned access structure to obtain $tr_1, \ldots, tr_n \leftarrow \mathsf{AShare}^\ell(0)$. Fix $share_i \leftarrow tr_i$ for all $i \in U$. Fix randomness \$.

- **Reduction** Next_1 : Using the adversarially specified Next and above fixings we give the description of Next_1.

  On input a transcript τ, execute the Next function with τ as input to obtain a subset $S \in \{S_1, S_2\}$ and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. If $S = S_1$, we design leak function g_1 that parses input as s, lets $share_\ell$ be s, and outputs $\mathsf{g}(\otimes_{i \in S_1} share_i)$. Otherwise (if $S = S_2$), we design leak function g_1 that parses the input as r and use randomness \$ to sample shares $\otimes_{i \in S_2 \setminus S_1} r_i$ forming a valid encoding of secret r under the scheme AShare^ℓ. Next it lets $share_i = r_i$ for each $i \in S_2 \setminus S_1$ and outputs $\mathsf{g}(\otimes_{i \in S_2} share_i)$.

Observe that if the adversary for the ϵ-leakage-resilient secret sharing scheme can distinguish in between shares of m_1 and m_2 with advantage greater than ϵ, then the above reduction can distinguish in between the shares corresponding to m_1 and m_2 of the underlying 2-out-of-2 scheme with advantage greater than ϵ. This violates the leakage-resilience of the underlying scheme, completing the proof. □

5.3 Schemes for General Access Structures

Now we are in position to give the main result for disjoint leakage: a generic compiler that converts any secret sharing scheme into one that allows leakage from disjoint unauthorized subsets of shares.

Lemma 12. *For any access structure $\mathcal{A}$ supported on n parties, any message size $a > 0$, any leakage bound μ, suppose we have the following primitives:*

1. *Let* $(\mathsf{AShare}, \mathsf{ARec})$ *be a secret sharing scheme realizing access structure $\mathcal{A}$ that shares secrets of length a bits into n shares, each of length b bits.*

2. *For any error $\epsilon > 0$, for each choice of leader $\ell \in [n]$, let* $(\mathsf{LDSh}^\ell, \mathsf{LDRec}^\ell)$ *be an ϵ-leakage-resilient secret sharing scheme corresponding to access structure $\mathcal{A}$ for leader ℓ. This scheme shares secrets of length b into n shares, each of length at most c.*

Then there is a secret sharing scheme realizing access structure $\mathcal{A}$ that is $n\epsilon$-leakage-resilient given the transcript of μ bits of communication amongst disjoint unauthorized subsets of shares. The resulting scheme, $(\mathsf{LRShare}, \mathsf{LRRec})$*, shares secrets of length a into n shares, each of length cn bits.*

Proof. The construction of $(\mathsf{LRShare}, \mathsf{LRRec})$ is given below:

- **Sharing function** $\mathsf{LRShare}$:
  Encode the secret m using the given secret sharing scheme for access structure $\mathcal{A}$ to obtain $m^1, \ldots, m^n \leftarrow \mathsf{AShare}(m)$. For each choice of leader $\ell \in [n]$, share m^ℓ using LDSh^ℓ to obtain $m^\ell_1, \ldots, m^\ell_n \leftarrow \mathsf{LDSh}^\ell(m^\ell)$. For each $i \in [n]$, construct $share_i$ as $(m^1_i, \ldots, m^n_i)$.

- **Reconstruction function** LRRec:
  On input the shares $\otimes_{i \in T} share_i$, for each $i \in T$, parse $share_i$ as $(m^1_i, \ldots, m^n_i)$. For each choice of leader $\ell \in T$, run LDRec^ℓ on the shares of m^ℓ, to obtain $m^\ell \leftarrow \mathsf{LDRec}^\ell(\otimes_{i \in T} m^\ell_i)$. Run ARec on the shares of m, to obtain $m \leftarrow \mathsf{ARec}(\otimes_{i \in T} m^i)$. Output m.

Correctness and Efficiency: Correctness follows from the observation that shares of any authorized set T of our final scheme can be used to construct an authorized set of shares of the underlying ϵ-leakage-resilient secret sharing scheme corresponding to each choice of leader $\ell \in T$. Efficiency trivially follows from the construction.

Perfect Privacy: Any unauthorized set U of the final scheme can only have information about $\{m^i : i \in U\}$, by the perfect privacy of the leader based ϵ-leakage-resilient secret sharing scheme . Therefore, secret remains perfectly hidden by the perfect privacy of the underlying scheme (AShare, ARec).

Statistical leakage-resilience: Suppose the adversary specifies k disjoint unauthorized subsets $\{S_1, \dots, S_k\}$ for any $k \leq n$, and specifies a leakage protocol Leak amongst these subsets which can be used to distinguish in between the shares of u_1 and u_2. We use such an adversary to give an explicit leakage protocol Leak_1 for one of the underlying leader based ϵ-leakage-resilient secret sharing scheme .

- **Initial setup**: Randomly fix $\ell \in [n]$. For each $i \in [\ell - 1]$, fix $m_1^i, \dots, m_n^i \mathsf{LDSh^i}(m^i)$ where m^i are generated while sharing u_1. For each $i \in [n] \setminus [\ell]$, fix $m_1^i, \dots, m_n^i \leftarrow \mathsf{LDSh^i}(m^i)$ where m^i are generated while sharing u_2.
- **Reduction** Next_1: Using Leak, as specified by its Next function, and shares fixed above, we give the description of protocol Leak_1 by specifying Next_1.

  On input a transcript τ, execute the adversary specified Next function with τ as input to obtain a subset $S \in \{S_1, \dots, S_k\}$ and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. We construct leakage function g_1 that takes $\otimes_{i \in S} m_i$ as input, for each $i \in S$, sets $m_i^\ell \leftarrow m_i$, computes $share_i$ as $(m_i^1, \dots, m_i^n)$ using fixed values and outputs $\mathsf{g}\big(\otimes_{i \in S} (share_i)\big)$. Output S, g_1.

Observe that if the adversary for our secret sharing scheme can distinguish between shares of u_1, u_2 with advantage greater than $n\epsilon$, then the above reduction can distinguish between the shares of the underlying leader based ϵ-leakage-resilient secret sharing scheme with advantage greater than ϵ, violating its leakage-resilience. Thus our proof is complete. □

6 Leakage-Resilient Secret Sharing against BCPs

The starting point of our construction is a (p, n, n)-LRSS where p is a constant fraction of n.

6.1 n-out-of-n Leakage-Resilient schemes

In this section, we use our communication complexity lower bounds to construct n-out-of-n schemes that are leakage-resilient p- party bounded-collusion protocols, where p is any constant fraction of n. Our construction at a high level is similar to the n-out-of-n construction of Kumar, Meka, and Sahai [KMS19] with the following three modifications:

1. We use lower-bounds against p-party BCP instead of NOF. This allows us to handle joint leakage from $p = \omega(\log n)$ parties.
2. We additionally ensure leakage-resilience against complete leakage of any n-1 shares. As mentioned in introduction, this will prove crucial while ensuring secrecy when using leader

based schemes. For this we use the stronger communication complexity lower bounds which allows one set of $n-1$ parties to collude as well apart from p-party BCP.

3. To use our lower bounds, as a technicality, we need to handle non-binary fields.

Lemma 13. *For any $n \geq 1$, any fraction $\alpha < 1$, let $p = \alpha n$, for any leakage-bound $\mu \geq 0$, any $\epsilon > 0$, if there is an efficient n party function $\mathsf{f} : (\mathbb{F}_q)^n \to \{0,1\}$ having that has ϵ-communication complexity at least $\mu + 1$ against p-party protocols along with collusion of possibly one set of $n-1$ parties (see Section A), then there is an efficient n-out-of-n -secret sharing scheme that is ϵ-leakage-resilient w.r.t. $(p, n, \mu)-\mathsf{BCP}$ along with complete leakage of any $n-1$ shares. The resulting scheme, $(\mathsf{Share}^{\mathsf{n}}_{\mathsf{n}}, \mathsf{Rec}^{\mathsf{n}}_{\mathsf{n}})$, shares single bit secrets into n shares, each of bit-length $1 + b$ where $b = \log\lceil|\mathbb{F}_q|\rceil$.* [21]

Combining the above result with our communication complexity lower bounds from Appendix A, we get the following:

Corollary 2. *For all fraction $\alpha \in (0,1)$, there exists a constant C such that the following holds. For all $n \geq 1$, error $\epsilon > 0$, there exists an efficient n-out-of-n secret sharing scheme ϵ-leakage resilient w.r.t. $(\alpha n, n, \mu)-\mathsf{BCP}$ along with complete leakage of any $n-1$ shares. The scheme shares single bit secrets into n shares, each of bit-length $\mu^C + C\log(1/\epsilon)$.*

We rely on additive secret sharing schemes that we recall for completeness.

XOR based Additive Secret Sharing We recall the n-out-of-n additive secret sharing based on $\oplus$ (XOR) operation. For any $a \geq 1$, let the secrets be a bits long.

- (**Sharing function** $\mathsf{XORShare_n}$) : Let $\mathsf{XORShare_n} : \{0,1\}^a \to \otimes_{i\in[n]}\{0,1\}^a$ be a randomized sharing function. On input a secret $s \in \{0,1\}^a$, uniformly sample the first $n-1$ shares, namely $s_1, \ldots, s_{n-1}$, such that each $s_i \in \{0,1\}^a$. Compute the last share using the secret s and the sampled shares as $s_n \leftarrow s \oplus s_1 \oplus \ldots \oplus s_{n-1}$ Output $s_1, \ldots, s_n$ as the n shares.
- (**Reconstruction function** $\mathsf{XORRec_n}$) : Let $\mathsf{XORRec_n} : \otimes_{i\in[n]}\{0,1\}^a \to \{0,1\}^a$ be a deterministic function for reconstruction. On input n shares, namely $s_1, \ldots, s_n$, compute $s \leftarrow s_1 \oplus \ldots \oplus s_n$ and output the result s.

Lemma 14. *([KGH83]) For secret space of $a \geq 1$ bits, $(\mathsf{XORShare_n}, \mathsf{XORRec_n})$ (described above) is an $(n, n, 0)$-secret sharing scheme.*

Additionally this scheme has a useful property that given the secret and all but one shares, the leftover share can be efficiently computed. Formally,

Lemma 15. *Let $(\mathsf{XORShare_2}, \mathsf{XORRec_2})$ be an $(2,2,0)$-secret sharing scheme for single bit secrets. For any $m, sh_1, sh_2 \in \{0,1\}$, if $m \leftarrow \mathsf{XORRec_2}(sh_1, sh_2)$, then $sh_1 \leftarrow \mathsf{XORRec_2}(m, sh_2)$.*

[21] Observe that we are only making black-box use of f and do not need to efficiently sample from it's pre-image.

Proof of Lemma 13

Proof. Let $\mathsf{Bin} : \mathbb{F}_q \to \{0,1\}^b$ be the function that given an element of $\mathbb{F}_q$ outputs its binary representation using $b = \lceil|\mathbb{F}_q|\rceil$ bits. Bin^{-1} is the corresponding inverse function. Let ($\mathsf{XORShare_n}$, $\mathsf{XORRec_n}$) be the (n,n) additive secret sharing scheme for single bit secrets (as in Lemma 14). Similarly, let $(\mathsf{XORShare_2}, \mathsf{XORRec_2})$ be the $(2,2)$ additive secret sharing scheme for single bit secrets. The leakage-resilient scheme is defined as:

1. (**Sharing function** $\mathsf{Share^n_n}$):
   On input a secret bit m, for each $i \in [n]$, uniformly and independently sample $R_i \in \mathbb{F}_q$ and compute the binary representation $r_i \leftarrow \mathsf{Bin}(R_i)$. Execute function f on $r_1, \dots, r_n$ to compute the bit $r \leftarrow \mathsf{f}(r_1, \dots, r_n)$. Compute $s \leftarrow \mathsf{XORRec_2}(m, r)$. Secret share s using $\mathsf{XORShare_n}$ to obtain $s_1, \dots, s_n \leftarrow \mathsf{XORShare_n}(s)$. For each $i \in [n]$, let $share_i \leftarrow (r_i, s_i)$.

2. (**Reconstruction function** $\mathsf{Rec^n_n}$) :
   On input n shares, namely $share_1, \dots, share_n$, for each $i \in [n]$, parse $share_i$ as (r_i, s_i) and compute $R_i \leftarrow \mathsf{Bin}^{-1}(r_i)$. Compute f on $R_1, \dots, R_n$ to obtain the bit $r \leftarrow \mathsf{f}(R_1, \dots, R_n)$. Apply the reconstruction procedure $\mathsf{XORRec_n}$ on $s_1, \dots, s_n$ to obtain $s \leftarrow \mathsf{XORRec_n}(s_1, \dots, s_n)$. Compute $m \leftarrow \mathsf{XORRec_2}(r, s)$. Output m.

Correctness, efficiency and perfect privacy : Similar to [KMS19]. Notice that we only make black-box use of f and do not need to invert f.

Statistical leakage-resilience: We adapt the proof of [KMS19] to our setting. Suppose the adversary specifies a leakage-protocol $\mathsf{Leak} \in (p, n, \mu)-\mathsf{BCP}$ and an any unauthorized subset U of at most $n-1$ parties, such that the transcript of Leak along with shares corresponding to U violate the leakage-resilience of our scheme. We use such an adversary to give a p-party protocol that violates the communication complexity of f.

- **Initial setup** : Randomly fix $s \leftarrow \{0,1\}$. Compute $s_1, \dots, s_n \leftarrow \mathsf{XORShare_n}(s)$ and fix $s_1, \dots, s_n$.

- **Protocol** : For each $i \in [n]$, party i holds $r_i \in \{0,1\}^b$ as input. We use the Next function specified by the adversary for the secret sharing scheme, and the values of s_i fixed above to give a communication protocol, specifically next function Next_1 for f.

  The next function Next_1 on input transcript τ invokes the underlying next function Next with τ as input to obtain a subset $S \subset [n]$ and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. If the output of Next is $\perp$, then as a last round of communication, define the function g_1 that takes $\otimes_{i \in U} R_i$ as input, and for each $i \in U$, converts R_i to binary $r_i \leftarrow \mathsf{Bin}(R_i)$, and uses the fixed value of s_i to create $share_i \leftarrow r_i, s_i$, and fully outputs $(\otimes_{i \in U} share_i)$. Otherwise, define the function g_1 that takes $\otimes_{i \in S} R_i$ as input, and for each $i \in S$, converts R_i to binary $r_i \leftarrow \mathsf{Bin}(R_i)$, and uses the fixed value of s_i to create $share_i \leftarrow r_i, s_i$, then computes and outputs $\mathsf{g}(\otimes_{i \in S} share_i)$. Finally, Next_1 outputs S, g_1.

Observe that if the adversary of the leakage-resilient secret sharing scheme achieves some advantage in distinguishing shares of 0 and 1, then the communication protocol created in the above reduction achieves the same advantage in computing the value of f. Apart from the complete leakage of shares, the total amount of leakage (through $(p, n, \mu)-\mathsf{BCP}$) is equal to the total communication of the p-party protocol given in the reduction. Moreover, from Lemma 7 of [KMS19], at the end of the leakage protocol, we can completely leak a subset of shares at the cost of one additional bit of adaptive leakage. Notice that our function f has communication complexity at least $\mu + 1$ which allows for this additional bit of leakage. This completes the proof. □

6.2 Leakage-resilient ramp secret sharing schemes

Our construction of (p, t, n)-LRSS uses intermediate LRSS that we call *leader based leakage-resilient ramp schemes*. To build up to it, we first introduce *leakage-resilient ramp secret sharing schemes*. These schemes can be thought of leakage-resilient version of ramp secret sharing schemes originally introduced by Blakley and Meadows [BM84].

We call a scheme satisfying the following two properties as a (p, t, n)-ramp secret sharing scheme.

1. Any t-out-of-n parties can efficiently reconstruct the secret.
2. The leakage-transcript of any p-party bounded-collusion protocol reveals statistically no information about the underlying secret (as in Definition 16).

Note that we have not placed the usual secrecy requirement that any $t-1$ shares hide the secret. While the leakage-resilience ensures that secret is statistically hidden given any p shares, it is not an issue if some subset of $p+1$ parties can recover the secret (in fact, in our construction, some subsets can).

We next construct efficient schemes above whenever $p \ll t/(\log t)$. As described in the introduction, we will exploit the idea of reusing shares via ramp hash families. We begin by defining such hash function families.

Definition 18. ***[Ramp hash families]*** *A family consisting of d functions of the form $\{f : [n] \to [q]\}$ is called a (q, t, n)-ramp hash function family of size d, if for all subsets $T \subseteq [n]$ of cardinality t, there exists a function f in the family such that f is surjective on T; $\{f(i) : i \in T\} = [q]$.*

Such a family of functions is called efficient, if we can generate d efficient functions for this hash family, namely $(f_1, \ldots, f_d) \leftarrow \mathsf{RHF}(q, t, n)$ in time $\mathrm{poly}(n, d)$.

We note that Alon et al. [ADM$^+$99], Meka, Reingold and Zhou [MRZ14] studied *covering hash families* which are stronger objects as they guarantee that a randomly chosen hash function from the family is surjective on T with high probability. We choose to give a new definition of ramp hash family, as this weaker definition may enable for better concrete parameters.

The proof uses the idea of Kurosawa and Stinson (see [Bla99]). Note that [KMS19] relied on (q, n)-perfect hash family which in our notation correspond to (q, q, n)-ramp hash family.

Lemma 16. *For any number of parties n, any collusion bound p, any threshold t, any message size $a > 0$, suppose we have the following primitives:*

1. *An efficient (p, q, q, μ)-LRSS with error ϵ that shares secrets of length a into shares of length c each.*
2. *An efficient (q, t, n)-ramp hash family* RHF *of size d.*

Then there is an efficient (p, t, n)-ramp secret sharing scheme that is $d\epsilon$-leakage-resilient w.r.t. $(p, n, \mu)-\mathsf{BCP}$. The resulting scheme, $(\mathsf{RampSh}_{\mathsf{n}}^{\mathsf{p,t}}, \mathsf{RampRec}_{\mathsf{n}}^{\mathsf{p,t}})$, shares secrets of length a into n shares, each of length cd. [22]

Combining the above with the construction from Corollary 2 and ramp hash families from [MRZ14] immediately gives the following.

Claim 1 (Ramp hash families [MRZ14])**.** *There exits a constant $C > 0$ such that for all $1 \leq t \leq n$, there exists an efficient $(t/C \log t, t, n)$-ramp hash family of size $d = \mathrm{poly}(\log n, t)$.*

Corollary 3. *There exists a constant $C > 0$ such that the following holds. For all $t < n$ and $p \leq t/C \log t$, and a communication bound μ, there exists an efficient (p, t, n)-ramp secret sharing scheme that is ϵ-leakage-resilient against $(p, n, \mu)-\mathsf{BCP}$. The resulting scheme, $(\mathsf{RampSh}_{\mathsf{n}}^{\mathsf{p,t}}$, $\mathsf{RampRec}_{\mathsf{n}}^{\mathsf{p,t}})$, shares secrets of a bits into n shares each of length $a \cdot \left(\mu^C + C \log(1/\epsilon)\right) \mathrm{poly}(\log n, t)$.*

Proof of Lemma 16. Generate the d hash functions of the ramp hash family. Let $(f_1, \ldots, f_d) \leftarrow \mathsf{RHF}(q, t, n)$. We use these functions in our construction of $(\mathsf{RampSh}_{\mathsf{n}}^{\mathsf{p,t}}, \mathsf{RampRec}_{\mathsf{n}}^{\mathsf{p,t}})$:

- (**Sharing function** $\mathsf{RampSh}_{\mathsf{n}}^{\mathsf{p,t}}$).
  On input a secret m, for each $j \in [d]$, share m using the sharing procedure of the (p, q, q, μ)-LRSS (using independent randomness) to obtain $(m_1^j, \ldots, m_q^j) \leftarrow \mathsf{LRShare}_{\mathsf{q}}^{\mathsf{q}}(m)$. For each $i \in [n]$, using the above hash functions construct $share_i$ as $\left(m_{f_1(i)}^1, \ldots, m_{f_d(i)}^d\right)$.

- **Reconstruction function** ($\mathsf{RampRec}_{\mathsf{n}}^{\mathsf{p,t}}$).
  On input a set of shares corresponding to an authorized set T of cardinality t, for each $i \in T$, parse $share_i$ as $\left(m_{f_1(i)}^1, \ldots, m_{f_d(i)}^d\right)$. Find $j \in [d]$ such that f_j is surjective on T. Use the reconstruction procedure of the underlying leakage resilient scheme to compute $m \leftarrow \mathsf{LRRec}_{\mathsf{q}}^{\mathsf{q}}(m_1^j, \ldots, m_q^j)$. Output m.

Perfect correctness and efficiency: For any authorized set $T \subseteq [n]$ of t parties, by definition of the (q, t, n)-ramp hash family, there will be a function f_j in the family $(f_j : j \in [d])$, such that f_j is surjective on T (see definition 18). Therefore, all the q shares of j^{th} encoding of m will be available. Hence, correctness follows from the correctness of the underlying q-out-of-q scheme. Efficiency follows from the efficiency of the ramp hash family and the underlying leakage-resilient scheme.

Statistical leakage-resilience:
This follows from a hybrid argument. Suppose we have two secrets $u_1 \neq u_2$ and a leakage protocol $\mathsf{Next} \in (p, n, \mu)-\mathsf{BCP}$ that distinguishes the encoding of u_1, u_2 with advantage more than $d\epsilon$. We will use the protocol to violate the leakage resilience of the underlying (p, q, q, μ)-LRSS.

Suppose we are given a set of shares $(sh_1, \ldots, sh_q)$ using the underlying (p, q, q, μ)-LRSS.

[22] In the next subsection, we will modify our definition of ramp schemes and prove a stronger lemma. We have included this lemma as it is independently interesting and helps in exposition.

- **Initial setup**: Randomly fix $j \in [d]$. For $i \leq j-1$, share u_1 using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_q \leftarrow \mathsf{LRShare}^{\mathsf{q}}_{\mathsf{q}}(u_1)$. For each $i > j$, share u_2 using the sharing procedure of the underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_q \leftarrow \mathsf{LRShare}^{\mathsf{q}}_{\mathsf{q}}(u_2)$. Fix all these sampled shares. Finally, set $(m^j_1, \ldots, m^j_q) \leftarrow (sh_1, \ldots, sh_q)$. Let $share_i = (m^i_{f_1(i)}, \ldots, m^i_{f_d(i)})$ for $1 \leq i \leq n$.

- **Reduction**

  Note that since the values $(m^i_1, \ldots, m^i_q)$ are fixed for $i \neq j$, any $(p, n, \mu)-\mathsf{BCP}$ on $share_1, \ldots, share_n$ is in turn a p-party collusion protocol on $sh_1, \ldots, sh_q$.

  Formally, using the adversarially specified Next and above fixings we give the description of Next_1. On input a transcript τ, execute the Next function with τ as input to obtain a subset $S \subset [n]$ of cardinality at most p and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. Construct the underlying set $T \subseteq [q]$ corresponding to hash function f_j, by setting $T \leftarrow \{f_j(i) : i \in S\}$. Next, we construct leakage function $\mathsf{g_1}$ that takes $\otimes_{i \in T} sh_i$ as input, for each $i \in T$ computes $share_i$ as $\big(m^1_{f_1(i)}, \ldots, m^d_{f_d(i)}\big)$ using the fixed values and outputs $\mathsf{g}\big(\otimes_{i \in S} share_i\big)$. Output $T, \mathsf{g_1}$.

Observe that if the adversary for the (p, t, n)-ramp scheme can distinguish in between shares of u_1 and u_2 with advantage greater than $d\epsilon$, then the above reduction can distinguish in between the shares corresponding to u_1 and u_2 of the underlying q-out-of-q scheme with advantage greater than ϵ. This violates the leakage-resilience of the underlying scheme, completing the proof. □

6.3 Leader Based Ramp Leakage-Resilient Schemes

In the previous section, we saw how to efficiently construct $(p = O(t/\log t), t, n)$-ramp secret sharing schemes. While we got leakage-resilience against p-party protocols, secrecy against $t-1$ shares was no longer ensured. As a step to overcome this limitation, we further refine our definition of ramp secret sharing scheme and incorporate the notion of a leader. In particular, for any a-priori chosen leader $\ell \in [n]$, a scheme satisfying the following three properties is called (p, t, n)-ramp secret sharing scheme for leader ℓ.

1. Leader ℓ along with any other $t-1$ parties can efficiently reconstruct the secret.

2. Without the leader's share the secret is perfectly hidden. That is for two distinct messages a, b, $((\mathsf{Share}(a)_i : i \neq \ell \in [n])) \equiv ((\mathsf{Share}(b)_i : i \neq \ell \in [n]))$.

3. The leakage-transcript of any p-party bounded-collusion protocol along with the other $n-1$ shares (all but leader's shares) reveal statistically no information about the underlying secret. That is for any p-party collusion protocol with total communication μ and two distinct messages $a \neq b$, $\mathsf{Leak}(\mathsf{Share}(a)) \circ ((\mathsf{Share}(a)_i : i \neq \ell \in [n])) \approx_\epsilon \mathsf{Leak}(\mathsf{Share}(b)) \circ ((\mathsf{Share}(b)_i : i \neq \ell \in [n]))$

The above definition generalizes the corresponding leader based definition given in preceding section 5.2, which in turn generalizes an idea implicit in the work of Aggarwal et al. [ADN+19].

Lemma 17. *For any number of parties n, any collusion bound p, any threshold t, any message size $a > 0$, suppose we have the following primitives:*

1. *For any leakage bound μ, any error bound $\epsilon > 0$, an efficient q-out-of-q secret sharing scheme $(\mathsf{LRShare}_\mathsf{q}^\mathsf{q}, \mathsf{LRRec}_\mathsf{q}^\mathsf{q})$ that is ϵ-leakage-resilient w.r.t. $(p, q, \mu)-\mathsf{BCP}$ along with complete leakage of any $q-1$ shares. The scheme shares a secret of bit-length a into q shares, each of bit-length c.*

2. *An efficient $(q-1, t-1, n)$-ramp hash family RHF of size d.*

Then for any leader $\ell \in [n]$, there is an efficient (p, t, n)-ramp secret sharing scheme for leader ℓ that is $d\epsilon$-leakage-resilient w.r.t. $(p, n, \mu)-\mathsf{BCP}$ along with complete leakage of all but leader's share. The resulting scheme, $(\mathsf{LRampSh}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell}, \mathsf{LRampRec}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell})$, shares secrets of length a into n shares, each of length cd.

Combining the above with the construction from Corollary 2 and ramp hash families from [MRZ14] immediately gives the following:

Corollary 4. *There exists a constant $C > 0$ such that the following holds. For all $t < n$ and $p \leq t/C \log t$, and a communication bound μ, there exists an efficient (p, t, n)-ramp secret sharing scheme that for leader ℓ is ϵ-leakage-resilient against $(p, n, \mu)-\mathsf{BCP}$ along with complete leakage of all but leader's share. The resulting scheme, $(\mathsf{LRampSh}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell}, \mathsf{LRampRec}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell})$, shares secrets of a bits into n shares each of length $a \cdot \left(\mu^C + C \log(1/\epsilon)\right) \mathrm{poly}(\log n, t)$.*

Proof of Lemma 17. Our construction is similar to the (p, t, n)-ramp secret sharing scheme scheme of the previous section, with a small twist, we do not scatter the last share of the underlying scheme, and only give it to the leader. Consequently, without the leader, the last share of each instance of underlying scheme will be missing and secrecy and leakage-resilience can be ensured. Details follow. Generate the d hash functions of the ramp hash family. Let $(f_1, \ldots, f_d) \leftarrow \mathsf{RHF}(q-1, t-1, n)$. We use these functions in our construction of $(\mathsf{LRampSh}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell}, \mathsf{LRampRec}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell})$ for any fixed leader $\ell \in [n]$:

- (**Sharing function** $\mathsf{LRampSh}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell}$).
  On input a secret m, for each $j \in [d]$, share m using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $(m_1^j, \ldots, m_q^j) \leftarrow \mathsf{LRShare}_\mathsf{q}^\mathsf{q}(m)$. Construct leader's $share_\ell$ as $\left(m_q^1, \ldots, m_q^d\right)$. For everyone else, namely, for each $i \in [n] \setminus \{\ell\}$, using hash functions construct $share_i$ as $\left(m_{f_1(i)}^1, \ldots, m_{f_d(i)}^d\right)$.

- **Reconstruction function** ($\mathsf{LRampRec}_\mathsf{n}^{\mathsf{p},\mathsf{t},\ell}$).
  On input a set of shares corresponding to an authorized set T of cardinality t such that leader $\ell \in T$, for each $i \in T \setminus \{\ell\}$, parse $share_i$ as $\left(m_{f_1(i)}^1, \ldots, m_{f_d(i)}^d\right)$. Parse leader's $share_\ell$ as $\left(m_q^1, \ldots, m_q^d\right)$. Find $j \in [d]$ such that f_j is surjective on $T \setminus \{\ell\}$. Use the reconstruction procedure of the underlying leakage resilient scheme to compute $m \leftarrow \mathsf{LRRec}_\mathsf{q}^\mathsf{q}(m_1^j, \ldots, m_q^j)$. Output m.

Perfect correctness: For any authorized set $T \subseteq [n]$ of t parties such that leader $\ell \in T$, by the properties of the $(q-1, t-1, n)$-ramp hash family, there will be a function f_j in the family

$(f_j : j \in [d])$, such that f_j is surjective on $T \setminus \{\ell\}$ (see definition 18). Therefore, all the first $q-1$ shares of j^{th} encoding of m will be available, and the leader's shares provides with the leftover q^{th} share of the same encoding. Hence, correctness follows from the correctness of the underlying q-out-of-q scheme.

Efficiency: Efficiency follows from the efficiency of the ramp hash family and the underlying leakage-resilient scheme.

Perfect Privacy without the leader: Without the leader ℓ, only at most $q-1$ shares of each of the underlying scheme will be available, and therefore the secret will be perfectly hidden by the perfect privacy of the underlying q-out-of-q scheme.

Statistical leakage-resilience: The proof is almost identical to the proof of resilience in Lemma 16.

The adversary specifies a $\mathsf{Next} \in (p, n, \mu)-\mathsf{BCP}$ and the set $U = [n] \setminus \{\ell\}$ (for complete leakage) that allows it to distinguish in between shares of u_1 and u_2 under our scheme. We use such an adversary to construct $\mathsf{Next}_1 \in (p, q, \mu)-\mathsf{BCP}$ and the set $[q-1]$ (for complete leakage) that violates the leakage-resilience of the underlying q-out-of-q scheme.

- **Initial setup** : Randomly fix $j \in [d]$. For each $i \in [j-1]$, share u_1 using the sharing procedure of underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots, m^i_q \leftarrow \mathsf{LRShare}^{\mathsf{q}}_{\mathsf{q}}(u_1)$. For each $i \in [d] \setminus [j]$, share u_2 using the sharing procedure of the underlying leakage-resilient scheme (using independent randomness) to obtain $m^i_1, \ldots,$ $m^i_q \leftarrow \mathsf{LRShare}^{\mathsf{q}}_{\mathsf{q}}(u_2)$. Fix all these sampled shares.
- **Reduction** Next_1 : Using the adversarially specified Next and above fixings we give the description of Next_1.

  On input a transcript τ, execute the Next function with τ as input to obtain a subset $S \subset [n]$ of cardinality at most p and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. We construct the underlying set $T \subseteq [q]$ corresponding to hash function f_j. If $\ell \in S$, then set $T \leftarrow \{q\} \cup \{f_j(i) : i \in S \setminus \{\ell\}\}$, else set $T \leftarrow \{f_j(i) : i \in S\}$. Next, we construct leakage function g_1 that takes $\otimes_{i \in T} m_i$ as input, for each $i \in T$, sets $m^j_i \leftarrow m_i$. If $l \in S$, it computes leader's $share_\ell$ as $\big(m^1_q, \ldots, m^d_q\big)$. Then, for each $i \in S \setminus \{\ell\}$, computes $share_i$ as $\big(m^1_{f_1(i)},$ $\ldots, m^d_{f_d(i)}\big)$ using the fixed values and outputs $\mathsf{g}\big(\otimes_{i \in S} share_i\big)$. Output T, g_1. In our original protocol, at the end, shares corresponding to the set$[n] \setminus \{\ell\}$ are revealed. In our new protocol, we instead, fully reveal the the underlying shares corresponding to set $[q-1]$.

Observe that if the adversary for the leader based (p, t, n)-ramp secret sharing scheme scheme can distinguish in between shares of u_1 and u_2 with advantage greater than $d\epsilon$, then the above reduction can distinguish in between the shares corresponding to u_1 and u_2 of the underlying q-out-of-q scheme with advantage greater than ϵ. This violates the leakage-resilience of the underlying scheme, completing the proof. □

6.4 Leakage-Resilient Schemes for General Access Structures

Now we are in position to give the main result of this section: a generic compiler that converts any secret sharing scheme into a p-party leakage-resilient one. As remarked in the introduction, efficient schemes like this were not known for any $p = \omega(\log n)$.

Lemma 18. *For any collusion bound $p \geq 1$, any threshold $t > p$, any access structure $\mathcal{A}$ supported on n parties such that each authorized set has cardinality at least t, any message size $a > 0$, any leakage bound μ, suppose we have the following primitives:*

1. *Let $(\mathsf{AShare}, \mathsf{ARec})$ be a secret sharing scheme realizing access structure $\mathcal{A}$ that shares secrets of length a bits into n shares, each of length b bits.* [23]

2. *For any error $\epsilon > 0$, for each choice of leader $\ell \in [n]$, let $(\mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t},\ell}, \mathsf{LRampRec}_{\mathsf{n}}^{\mathsf{p,t},\ell})$ be a (p, t, n)-ramp secret sharing scheme scheme for leader ℓ that is ϵ-leakage-resilient against $(p, n, \mu)-\mathsf{BCP}$ along with complete leakage of all but leader's share. Moreover, the scheme shares secrets of length b bits into n shares each of length c bits.*

Then there is a secret sharing scheme realizing access structure $\mathcal{A}$ that is $n\epsilon$-leakage-resilient given the transcript of any $(p, n, \mu)-\mathsf{BCP}$ along with complete leakage of any unauthorized set of shares. The resulting scheme, $(\mathsf{LRShare}, \mathsf{LRRec})$, shares secrets of length a into n shares, each of length cn bits.

Proof. The construction of $(\mathsf{LRShare}, \mathsf{LRRec})$ is given below:

- **Sharing function** $\mathsf{LRShare}$:
  Encode the secret m using the given secret sharing scheme for access structure $\mathcal{A}$ to obtain $m^1, \ldots, m^n \leftarrow \mathsf{AShare}(m)$. For each choice of leader $\ell \in [n]$, share m^ℓ using $\mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t},\ell}$ to obtain $m_1^\ell, \ldots, m_n^\ell \leftarrow \mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t},\ell}(m^\ell)$. For each $i \in [n]$, construct $share_i$ as $(m_i^1, \ldots, m_i^n)$.

- **Reconstruction function** LRRec:
  On input the shares $\otimes_{i \in T} share_i$, for each $i \in T$, parse $share_i$ as $(m_i^1, \ldots, m_i^n)$. For each choice of leader $\ell \in T$, run $\mathsf{LRampRec}_{\mathsf{n}}^{\mathsf{p,t},\ell}$ on the shares of m^ℓ, to obtain $m^\ell \leftarrow \mathsf{LRampRec}_{\mathsf{n}}^{\mathsf{p,t},\ell}(\otimes_{i \in T} m_i^\ell)$. Run ARec on the shares of m, to obtain $m \leftarrow \mathsf{ARec}(\otimes_{i \in T} m^i)$. Output m.

Correctness and Efficiency: Correctness follows from the observation that shares of any authorized set T of our final scheme can be used to construct all the $|T|$ shares of the underlying (p, t, n)-ramp secret sharing scheme scheme corresponding to each choice of leader $\ell \in T$. Efficiency trivially follows from the construction.

Perfect Privacy: Any unauthorized set U of the final scheme can only have information about $\{m^i : i \in U\}$, by the perfect privacy of the leader based (p, t, n)-ramp secret sharing scheme . Therefore, secret remains perfectly hidden by the perfect privacy of the underlying scheme $(\mathsf{AShare}, \mathsf{ARec})$.

Statistical leakage-resilience: Suppose the adversary specifies a protocol $\mathsf{Leak} \in (p, n, \mu)-\mathsf{BCP}$ and an unauthorized set $U \subseteq [n]$ that distinguishes the shares of u_1, u_2 violating the leakage-resilience of our final scheme. We use such an adversary to give an explicit leakage protocol $\mathsf{Leak}_1 \in (p, n, \mu)-\mathsf{BCP}$ for the underlying leader based (p, t, n)-ramp secret sharing scheme scheme.

[23] The construction and proof also generalizes to any statistically or computationally secure scheme. We have only dealt with perfectly secure schemes for the ease of notation. Of course, for the computational case, we can only get computational leakage-resilience as we allow the adversary to additionally learn any unauthorized set of shares.

- **Initial setup**: For each $i \in U$, fix $m_1^i, \ldots, m_n^i \leftarrow \mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t,i}}(m^i)$ where m^i are generated while sharing u_1. Fix $\ell \in [n] \setminus U$. For each $i \in [\ell - 1] \setminus U$, fix $m_1^i, \ldots, m_n^i \leftarrow \mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t,i}}(m^i)$ where m^i are generated while sharing u_1. For each $i \in [n] \setminus [\ell] \cup U$, fix $m_1^i, \ldots, m_n^i \leftarrow \mathsf{LRampSh}_{\mathsf{n}}^{\mathsf{p,t,i}}(m^i)$ where m^i are generated while sharing u_2.
- **Reduction** Next_1: Using Leak, as specified by its Next function and fixed shares of shares of l we give the description of protocol Leak_1 by specifying Next_1.

  On input a transcript τ, execute the adversary specified Next function with τ as input to obtain a subset $S \subseteq [n]$ and a leakage function g that takes $\otimes_{i \in S} share_i$ as input. We construct leakage function g_1 that takes $\otimes_{i \in S} m_i$ as input, for each $i \in S$, sets $m_i^j \leftarrow m_i$, computes $share_i$ as $(m_i^1, \ldots, m_i^n)$ using fixed values and outputs $\mathsf{g}\big(\otimes_{i \in S} (share_i)\big)$. Output S, g_1. At the end, when Next outputs $\perp$, we output the shares corresponding to unauthorized set U, on input $\otimes_{i \in U} m_i$ as input, for each $i \in U$, set $m_i^j \leftarrow m_i$, compute $share_i$ as $(m_i^1, \ldots, m_i^n)$ using fixed values and output $(\otimes_{i \in S}(share_i))$.

Observe that if the adversary for our secret sharing scheme can distinguish between shares of u_1, u_2 with advantage greater than $n\epsilon$, then the above reduction can distinguish between the shares of the underlying leader based (p, t, n)-ramp secret sharing scheme scheme violating its leakage-resilience. Thus our proof is complete. □

From Single-bit Secrets to Multi-bit Secrets We recall a simple lemma from [KMS19], which can be used to convert any leakage-resilient scheme for single bit secrets into one for multi-bit secrets.

Lemma 19. *([KMS19]) Let $\mathcal{L}$ be any class of leakage family. For any $\epsilon_1 \geq 0$, any $\epsilon_2 > 0$, suppose* $(\mathsf{SBShare}, \mathsf{SBRec})$ *is a (n, ϵ_1)-secret sharing scheme (resp. computational) that is ϵ_2-leakage-resilient w.r.t. $\mathcal{L}$ that shares single bit secrets into n shares, each of length a. Then, for any secret space of $b > 0$ bits, there is an efficient $(n, b\epsilon_1)$-secret sharing scheme (resp. computational) realizing the same access structure that is $b\epsilon_2$-leakage-resilient w.r.t. $\mathcal{L}$. The resulting scheme,* $(\mathsf{MBShare}, \mathsf{MBRec})$*, shares secrets of bit-length b into n shares, each of bit-length ab.*

6.5 Instantiations

Corollary 5. *There exists a constant $C > 0$ such that for any access structure $\mathcal{A}$ supported on n parties such that each authorized set has cardinality at least t, any message size $a > 0$, any leakage bound μ, any error $\epsilon > 0$, suppose there is an efficient secret sharing scheme realizing access structure $\mathcal{A}$ that shares secrets of length a bits into n shares, each of length b bits. Then, for any collusion bound $p \leq \frac{t}{C \log t}$, there is an efficient secret sharing scheme realizing the same access structure $\mathcal{A}$ that is ϵ-leakage-resilient w.r.t. $(p, n, \mu)-\mathsf{BCP}$. The resulting scheme shares secret of length a bits into n shares, each of length $b \cdot n \cdot \mathrm{poly}(\log n, t) \cdot \big(\mu^C + C \log(1/\epsilon)\big)$.*

Proof. We iteratively instantiate the primitives required for Lemma 18 with the given secret sharing scheme along with the leader based (p, t, n)-ramp secret sharing scheme constructed in the preceding section from Corollary 4. □

Corollary 6. *There exists a constant $C > 0$, such that, for any number of parties $n \geq 2$, any threshold $t \leq n$, any collusion bound $p \leq \frac{t}{C \log t}$, any leakage-bound μ, any error $\epsilon > 0$, there is* *a efficient t-out-of-n secret sharing scheme that is ϵ-leakage-resilient w.r.t. $(p, n, \mu)-\mathsf{BCP}$. The resulting scheme shares a bit secrets into $a \cdot n \cdot \mathrm{poly}(\log n, t) \cdot \big(\mu^C + C\log(1/\epsilon)\big)$ bits shares.*

Proof. Use t-out-of-n secret sharing scheme of Shamir [Sha79] in Corollary 5. □

It is straightforward to use secret sharing schemes of [KW93, Bei11, KNY14] to obtain corresponding corollaries mentioned in the introduction, and consequently we omit these details.

Acknowledgements

Ashutosh Kumar thanks Eyal Kushilevitz, Rafail Ostrovsky, Aishwarya Sivaraman, Terence Tao, and Vinod Vaikuntanathan for useful discussions.

References

[ADM+99] Noga Alon, Martin Dietzfelbinger, Peter Bro Miltersen, Erez Petrank, and Gábor Tardos. Linear hash functions. *Journal of the ACM (JACM)*, 46(5):667–683, 1999.

[ADN+19] Divesh Aggarwal, Ivan Damgard, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, Joao Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In *Annual International Cryptology Conference*, pages 510–539. Springer, 2019.

[BADTS17] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to non-malleable extractors: achieving near-logarithmic min-entropy. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1185–1194. ACM, 2017.

[BDIR18] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. In *CRYPTO*, pages 531–561. Springer, 2018.

[Bei11] Amos Beimel. Secret-sharing schemes: a survey. In *International Conference on Coding and Cryptology*, pages 11–46. Springer Berlin Heidelberg, 2011.

[BEO+13] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. A tight bound for set disjointness in the message-passing model. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 668–677. IEEE, 2013.

[BFV19] Gianluca Brian, Antonio Faonio, and Daniele Venturi. Continuously non-malleable secret sharing for general access structures. In *Theory of Cryptography Conference*, pages 211–232. Springer, 2019.

[BG09] Jean Bourgain and MZ Garaev. On a variant of sum-product estimates and explicit exponential sum bounds in prime fields. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 146, pages 1–21. Cambridge University Press, 2009.

[BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 337–367. Springer, 2015.

[BGK06] J. Bourgain, A. A. Glibichuk, and S. V. Konyagin. Estimates for the number of sums and products and for exponential sums in fields of prime order. *Journal of the London Mathematical Society*, 73:380–398, 4 2006.

[BGM20] Marshall Ball, Oded Goldreich, and Tal Malkin. Randomness extraction from somewhat dependent sources. 2020.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM J. Comput.*, 36(4):1095–1118, December 2006.

[BKT04] Jean Bourgain, Nets Katz, and Terence Tao. A sum-product estimate in finite fields, and applications. *Geometric and Functional Analysis GAFA*, 14(1):27–57, 2004.

[Bla79] G. R. Blakley. Safeguarding cryptographic keys. In *AFIPS National Computer Conference (NCC '79)*, pages 313–317, Los Alamitos, CA, USA, 1979. IEEE Computer Society.

[Bla99] SR Blackburn. Combinatorics and threshold cryptography. *Research Notes in Mathematics*, 403:44–70, 1999.

[BM84] George Robert Blakley and Catherine Meadows. Security of ramp schemes. In *Crypto*, pages 242–268. Springer, 1984.

[BNS92] Laszlo Babai, Noam Nisan, and Mario Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[BO15] Mark Braverman and Rotem Oshman. On information complexity in the broadcast model. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing*, pages 355–364. ACM, 2015.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 01(01):1–32, 2005.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[Bou09] Jean Bourgain. Multilinear exponential sums in prime fields under optimal entropy condition on the sources. *Geometric and Functional Analysis*, 18(5):1477–1502, 2009.

[BS19] Saikrishna Badrinarayanan and Akshayaram Srinivasan. Revisiting non-malleable secret sharing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 593–622. Springer, 2019.

[CDF$^+$08] Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT*, pages 471–488, 2008.

[CFL83] Ashok K Chandra, Merrick L Furst, and Richard J Lipton. Multi-party protocols. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 94–99. ACM, 1983.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. 2020.

[CL16a] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors and almost optimal privacy amplification protocols. *FOCS*, 2016.

[CL16b] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 299–311. ACM, 2016.

[Coh16] Gil Cohen. Non-malleable extractors–new tools and improved constructions. In *31st Conference on Computational Complexity*, 2016.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 670–683, 2016.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *International Conference on Security and Cryptography for Networks*, pages 121–137. Springer, 2010.

[Des98] Yvo Desmedt. Some recent research aspects of threshold cryptography. In Eiji Okamoto, George Davida, and Masahiro Mambo, editors, *Information Security*, pages 158–173, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 227–237. IEEE, 2007.

[FK84] Michael L Fredman and János Komlós. On the size of separating systems and families of perfect hash functions. *SIAM Journal on Algebraic Discrete Methods*, 5(1):61–68, 1984.

[FKS84] Michael L Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with 0 (1) worst case access time. *Journal of the ACM (JACM)*, 31(3):538–544, 1984.

[FV19] Antonio Faonio and Daniele Venturi. Non-malleable secret sharing in the computational setting: Adaptive tampering, noisy-leakage resilience, and improved rate. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 448–479, Cham, 2019. Springer International Publishing.

[GIM$^+$16] Vipul Goyal, Yuval Ishai, Hemanta K Maji, Amit Sahai, and Alexander A Sherstov. Bounded-communication leakage resilience via parity-resilient circuits. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 1–10. IEEE, 2016.

[GK18a] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 685–698. ACM, 2018.

[GK18b] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In *CRYPTO*, pages 501–530. Springer, 2018.

[GLM$^+$16] Mika Goos, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016.

[GR05] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 407–418. IEEE Computer Society, 2005.

[Gro94] Vince Grolmusz. The bns lower bound for multi-party protocols ls nearly optimal. *Information and computation*, 112:51–54, 1994.

[HH09] Norbert Hegyvári and François Hennecart. Explicit constructions of extractors and expanders. *Acta Arithmetica*, 140(3):233–249, 2009.

[ISN89] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64, 1989.

[KGH83] Ehud Karnin, Jonathan Greene, and Martin Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, 1983.

[KM19] Bryce Kerr and Simon Macourt. Multilinear exponential sums with a general class of weights. *arXiv preprint arXiv:1901.00975*, 2019.

[KMS19] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing against colluding parties. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 636–660. IEEE, 2019.

[KN06] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 2006.

[KNY14] Ilan Komargodski, Moni Naor, and Eylon Yogev. Secret-sharing for np. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 254–273. Springer, 2014.

[KOS$^+$93] Kaoru Kurosawa, Koji Okada, Keiichi Sakano, Wakaha Ogata, and Shigeo Tsujii. Nonperfect secret sharing schemes and matroids. In *Eurocrypt*, pages 126–141. Springer, 1993.

[KR19] Yael Tauman Kalai and Leonid Reyzin. A survey of leakage-resilient cryptography. *IACR Cryptology ePrint Archive*, 2019:302, 2019.

[KRVZ06] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 691–700. ACM, 2006.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Structure in Complexity Theory Conference, 1993., Proceedings of the Eighth Annual*, pages 102–111. IEEE, 1993.

[LCG$^+$19] Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Leakage-resilient non-malleable secret sharing in non-compartmentalized models. *CoRR*, abs/1902.06195, 2019.

[Li13a] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 100–109, 2013.

[Li13b] Xin Li. New independent source extractors with exponential improvement. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 783–792. ACM, 2013.

[Li15] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 863–882. IEEE, 2015.

[MRZ14] Raghu Meka, Omer Reingold, and Yuan Zhou. Deterministic coupon collection and better strong dispersers. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[NS20] Jesper Buus Nielsen and Mark Simkin. Lower bounds for leakage-resilient secret sharing. In *To appear at Eurocrypt 2020*, 2020.

[NZ93] Noam Nisan and David Zuckerman. More deterministic simulation in logspace. In *STOC*, pages 235–244, 1993.

[PS17] Vladimir V Podolskii and Alexander A Sherstov. Inner product and set disjointness: Beyond logarithmically many parties. *arXiv preprint arXiv:1711.10661*, 2017.

[PS19] Giorgis Petridis and Igor E Shparlinski. Bounds of trilinear and quadrilinear exponential sums. *Journal d'Analyse Mathématique*, 138(2):613–641, 2019.

[PVZ12] Jeff M Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 486–501. SIAM, 2012.

[Rao06] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. In *STOC*, pages 497–506, 2006.

[Rao09] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM Journal on Computing*, 39(1):168–194, 2009.

[Raz00] Ran Raz. The bns-chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[Raz05] Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RBO89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, STOC '89, pages 73–85, New York, NY, USA, 1989. ACM.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[She14] Alexander A. Sherstov. Communication lower bounds using directional derivatives. *J. ACM*, 61(6):34:1–34:71, December 2014.

[SV86] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[SV19] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In *Annual International Cryptology Conference*, pages 480–509. Springer, 2019.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 32–42. IEEE, 2000.

[Vaz87] U. V. Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7:375–392, 1987.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 209–213. ACM, 1979.

A Collusion of one set of $n-1$ parties.

In this section, we extend the results of Section 3 and prove Lemma 9. This lemma is immediately proved once we prove the following Lemma, using the same argument that we used to prove Theorem 6 from Lemma 8.

Lemma 20. *For any number of parties n, any collusion bound $p < n$, any round bound $r \leq (k/p)(n-1/p-1)^{k-1}$, we have*

$$\max_{Y,Z} |Pr[\mathsf{f}(x) = 1 \text{ and } x \in Y \cap Z] - Pr[\mathsf{f}(x) = 0 \text{ and } x \in Y \cap Z]| \leq \frac{\Delta_k}{q^n}$$

where the maximum is over any (p, r, n)-cylinder-intersection Y and any cylinder Z for subset of cardinality $n-1$ and the Δ_k corresponds to sums over $(X_1, \ldots, X_n) = (\mathbb{F}_q)^n$ with q prime.

Proof. We proceed as in lemma 8 and wish to upper bound the following

$$\max_{Y,Z} |Pr[\mathsf{f}(x) = 1 \text{ and } x \in Y \cap Z] - Pr[\mathsf{f}(x) = 0 \text{ and } x \in Y \cap Z]|$$

Fix Y to be any (p, r, n)-cylinder-intersection and Z be any cylinder for subset U of cardinality $n-1$ that maximizes the above. Without loss of generality, let Y be the intersection of at most r cylinders $Y_1, \ldots, Y_r$ corresponding to r subsets, namely $S_1, \ldots, S_r$, where each S_i has cardinality at most p.

Now we argue that the collections of sets $U, S_1, \ldots, S_r$ is a (k, n)-subset-avoiding collection . Total number of subsets of $[n]$ of cardinality k is $\binom{n}{k}$. Total number of subsets of cardinality k contained in any subset of cardinality at most $n-1$ is at most $\binom{n-1}{k}$. Total number of subsets of cardinality k contained in any subset of cardinality at most p is at most $\binom{p}{k}$. . Total number of subsets of cardinality k contained in any r subsets, each of cardinality at most p is at most $r\binom{p}{k}$. We want to show that $\binom{n-1}{k} + r\binom{p}{k} < \binom{n}{k}$ to show that there is some subset of cardinality k not covered by any $U, S_1, \ldots, S_r$. This is equivalent to showing that $r\binom{p}{k} < \binom{n}{k} - \binom{n-1}{k} = \binom{n-1}{k-1}$ or that $r < \binom{n-1}{k-1}/\binom{p}{k} = (k/p)\binom{n-1}{k-1}/\binom{p-1}{k-1}$. This is the case since $r < (k/p)(n-1/p-1)^{k-1}$ which in turn is less than $\binom{n-1}{k-1}/\binom{p}{k}$.

Rest of the proof is identical to lemma 8.

□