Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 1 (2025)

The Meta-Complexity of Secret Sharing*

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

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

May 10, 2025

Abstract

A secret-sharing scheme allows the distribution of a secret s among n parties, such that only certain predefined "authorized" sets of parties can reconstruct the secret, while all other "unauthorized" sets learn nothing about s. The collection of authorized/unauthorized sets is defined by a monotone function $f : \{0,1\}^n \to \{0,1\}$. It is known that any monotone function can be realized by a secret-sharing scheme; thus, the smallest achievable *total share size*, $S(f)$, serves as a natural complexity measure.

In this paper, we initiate the study of the following meta-complexity question: Given a monotone function f, is it possible to efficiently distinguish between cases where the secret-sharing complexity of f is small versus large? We examine this question across several computational models, yielding the following main results.

(Hardness for formulas and circuits): Given a monotone formula f of size L, it is coNP-hard to distinguish between "cheap" functions, where the maximum share size is 1 bit and the total share size is $O(L^{0.01})$, and "expensive" functions, where the maximum share size is $\Omega(\sqrt{L})$ and the total share size is $\Omega(L/\log L)$. This latter bound nearly matches known secret-sharing constructions yielding a total share size of L bits. For monotone circuits, we strengthen the bound on the expensive case to a maximum share size of $\Omega(L/\log L)$ and a total share size of $\Omega(L^2/\log L)$. These results rule out the existence of instance-optimal compilers that map a formula f to a secret-sharing scheme with complexity polynomially related to $S(f)$.

(Hardness for truth tables): Under cryptographic assumptions, either (1) every n-bit slice function can be realized by a $\mathrm{poly}(n)$-size secret-sharing scheme, or (2) given a truth-table representation of f of size $N = 2^n$, it is computationally infeasible to distinguish in time $\mathrm{poly}(N)$ between cases where $S(f) = \mathrm{poly}(n)$ and $S(f) = n^{\omega(1)}$. Option (1) would be considered a breakthrough result, as the best-known construction for slices has a sub-exponential complexity of $2^{\tilde{O}(\sqrt{n})}$ (Liu, Vaikuntanathan, and Wee; Eurocrypt 2018). Our proof introduces a new worst-case-to-average-case reduction for slices, which may be of independent interest.

(Hardness for graphs): We examine the simple case where f is given as a 2-DNF, represented by a graph G whose edges correspond to 2-terms, and ask whether it is possible to distinguish between cases where the share size is constant and those where the share size is large, say $\Omega(\log n)$. We establish several connections between this question and questions in communication complexity. For instance, we show that graphs admitting constant-cost secret sharing form a subclass of graphs with constant randomized communication complexity and constant-size adjacency sketches (Harms, Wild, and Zamaraev; STOC

*This research is supported by ISF grant no. 2805/21 and by the European Union (ERC-2022-ADG) under grant agreement no.101097959 NFITSC.

ISSN 1433-8092

2022). We leverage these connections to establish new lower bounds for specific graph families, derive a combinatorial characterization of graphs with constant-size linear secret-sharing schemes, and show that a natural class of myopic algorithms fails to distinguish cheap graphs from expensive ones.

1 Introduction

Secret-sharing schemes were initially introduced by Shamir and Blakley [Sha79, Bla79] in 1979 and have since become a fundamental cryptographic tool with a broad range of applications, including secure multiparty computation protocols [BGW88, CCD88], threshold cryptography [DF91], access control [NW96], attribute-based encryption [GPSW06, Wat11], and oblivious transfer [SSR08, Tas11]. Technically, secret-sharing schemes can be seen as a distributed analog of encryption. The objective is to take a secret message s and "split" it into n shares, $s_1, \ldots, s_n$, each stored on a different device or "party," so that the secret can be reconstructed when "sufficiently many" shares are gathered, while a "small" coalition of parties remains unable to learn anything about the secret in an information-theoretic sense. (See Definition 3.1 for a formal definition and [Bei11] for general background).

In its general form [ISN87], the problem is parameterized by an *access structure* – a monotone function $f : \{0,1\}^n \to \{0,1\}$ which determines which coalitions can recover the secret. Specifically, a coalition A is authorized if its characteristic vector x_A is accepted by f and unauthorized otherwise.[1] For instance, in the common case of *threshold secret-sharing*, f is a threshold function that accepts all strings with a Hamming weight above a given threshold. For this scenario, Shamir's polynomial-based scheme [Sha79] offers a highly efficient solution: to share a single-bit secret, each party only needs to store a single element s_i from a field of size n, resulting in a maximum share complexity of $\log n$ bits per party and total share size of $n \log n$.

The secret-sharing complexity of general functions. Since the work of [ISN87] it is known that every monotone function admits a secret-sharing scheme. However, determining the optimal share size of the "worst monotone function" is a well-known long-standing open problem. Known positive results are based on *compilers* that construct a secret-sharing scheme for a function f based on its representation under some computational model, such as monotone DNF/CNF [ISN87], monotone formula [BL88] and monotone span-program [KW93, BI92]. Consequently, when f is taken to be a worst n-bit monotone function, these constructions lead to an exponential upper-bound of $2^{n(1-o(1))}$. A more recent line of work [ABF$^+$19, ABNP20, AN21], initiated by [LVW18, LV18], provides improved exponential upper bounds, culminating in the upper bound $1.5^{n+o(n)}$ [AN21]. These results "separate" secret-sharing complexity from traditional computational complexity measures that, by counting arguments, assign $2^{n(1-o(1))}$ cost for most functions. Still, as shown in [ABN$^+$22], these constructions can be captured by a compiler whose starting point is a representation of f by a "formula-over-slices" – a computational model that turns out to be powerful enough to break the 2^n barrier, but does not allow for sub-exponential $2^{n^{1-\epsilon}}$ representation [ABN$^+$22].

On the negative side, despite much effort, the best known lower-bound on the total share size of an n-bit monotone function is $\Omega(n^2/\log n)$ due to [Csi96]. Moreover, we have no better lower-bounds even for *non-explicit* functions! Indeed, counting-based methods that lead to non-explicit *computational* lower-bounds, fail miserably in the context of general secret-sharing schemes.[2] This leaves a huge exponential

[1] Monotonicity here means that for any $A \subset B$, it holds that $f(x_A) \leq f(x_B)$. A non-monotone function would not allow for a secret-sharing scheme, making monotonicity a necessary requirement.

[2] Such methods can work for limited cases of secret-sharing schemes where the sharing/reconstruction algorithms have small representation such as in the case of linear secret-sharing (see, e.g., [BFMP17]) or in the case of efficiently-computable secret-

gap between the upper-bound and the lower-bound. It is widely believed that some functions require super-polynomial cost or even exponential cost. (The latter conjecture goes back to Beimel's 1996 thesis [Bei96].) However, proving such lower-bounds is currently out of reach even under complexity-theoretic assumptions (e.g., $\mathsf{NP} \neq \mathsf{P}$). Moreover, unlike the case of, say circuit lower-bounds, we hardly have any explanation for the lack of such provable lower-bounds. (See [AN23] for some results in this direction.)

The Meta-Complexity of Secret Sharing. In this paper, we shift focus from studying general secret-sharing schemes for worst-case functions to exploring functions that admit particularly "cheap" secret-sharing schemes. To understand what makes functions "cheap," we initiate the study of the following "meta-complexity" problem:

> Given a monotone n-bit predicate f, represented within some computational model, is it possible to efficiently distinguish between the cases where f is "cheap" (there exists a secret-sharing scheme for f with a short share size) and where f is "expensive" (any secret-sharing scheme for f requires a large share size)?

This question follows a recent and fruitful line of research on understanding the computational complexity of approximating various complexity measures (see, e.g., [Sim23]) and is driven by several concrete motivations:

1. **Instance-Optimal Compilers**: As previously mentioned, there are efficient compilers that map monotone formulas to secret-sharing schemes with a total share size linear in the formula-size. While this is optimal for certain simple functions, it is provably wasteful for most functions. Indeed, improvements in general secret sharing show that, for typical functions, the cost of secret sharing is at least polynomially smaller than the formula size. Inspired by works on instance optimal algorithms [FLN01], we may hope to achieve *instance-optimal secret-sharing compilers* (IOSS) that, given a monotone formula, efficiently generate a secret-sharing scheme with complexity close to the best-achievable secret-sharing cost of f. The meta-complexity question above can be viewed as a relaxed version of IOSS, serving as a necessary step towards proving or disproving the existence of IOSS.

2. **Better Efficiency via Cheap Functions**: In some applications of secret-sharing (e.g., attribute-based encryption [GPSW06, Wat11]), one may have strict limitations on the resources (say communication) but some flexibility in choosing which predicate f to use. A positive solution to the meta-complexity question would allow us to evaluate different choices for f and select the most efficient one, optimizing trade-offs between efficiency, privacy, and functionality.

3. **Explaining the Lack of Lower Bounds**: A negative answer to the meta-complexity problem would help explain the scarcity of lower bounds in secret sharing. Following the well-known natural-proof framework [RR97], it's often observed that lower bounds give rise to efficient algorithms for distinguishing between easy and hard instances. Therefore, ruling out efficient algorithms for distinguishing cheap functions from expensive ones would act as a barrier against a broad class of lower-bound techniques.

sharing [LS20]. (See also [AN23] for a more general treatment.)

2 Our Results

Before presenting our results, we first establish some notation. We primarily focus on secret-sharing schemes that distribute a *single bit*. For a monotone function $f : \{0,1\}^n \to \{0,1\}$, we define the *total share size* (resp., *maximal share size*) as the smallest integer m such that f can be implemented by a secret-sharing scheme with a total (resp., maximal) share size of m. We denote these complexity measures by $\mathsf{S}_{\mathsf{sum}}(f)$ and $\mathsf{S}_{\mathsf{max}}(f)$, respectively. Occasionally, we also consider *linear secret-sharing schemes over the binary field* (LSS), where the mapping from the secret and randomness to shares is linear over $\mathbb{F}_2$, and denote the corresponding complexity measures by $\mathsf{LS}_{\mathsf{sum}}(f)$ and $\mathsf{LS}_{\mathsf{max}}(f)$.

We denote by $\mathsf{GapSS}[a,b]$ the promise problem where the input consists of a monotone function f, and the goal is to determine whether the secret-sharing complexity of f is at most a or at least b. We will examine the complexity of this problem under various computational models (e.g., monotone formulas/circuits, truth tables, and monotone 2-DNFs) and with respect to different secret-sharing complexity measures. Unless stated otherwise, we use the total share size as the default measure.

Our main results are negative, demonstrating that even extremely cheap functions (e.g., those with constant share size) are difficult to distinguish from expensive functions. We also provide some simple algorithms that yield weak upper bounds on the complexity of $\mathsf{GapSS}[a,b]$. Furthermore, we reveal new connections between this problem and questions in the domain of randomized communication complexity and randomized labeling schemes for graphs. We will now present a detailed account of our results.

2.1 Cheap Formulas Are Hard to Recognize

We begin with the case where f is represented as a formula. The following hardness result rules out the existence of instance-optimal secret-sharing compilers, even those that slightly improve on formula size.

Theorem 2.1 (GapSS is coNP-hard for formulas)**.** *For every constant $\epsilon > 0$, given a monotone formula f of size L it is coNP-hard to distinguish between the following cases:*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size of $O(L^\epsilon)$ bits.*
- *(No) Every secret-sharing scheme (SS) for f has max-share size of $\Omega(\sqrt{L}/\log L)$ and total share size of $\Omega(L/\log L)$.*

The proof of the theorem shows how to combine a SAT-instance, given as a non-monotone formula g, with a formula E that is expensive to realize by any secret-sharing scheme, so that the resulting monotone formula f is either expensive or cheap depending on the satisfiability of g. To get almost-optimal bounds, we have to make sure that the expensive formula E provides an optimal trade-off between its size and its SS-cost. This is done by a combination of Csirmaz's Lower Bound [Csi96] with techniques from [Bei23, ABI$^+$23b].

Under the exponential-time hypothesis (ETH) [IP99] we get poly-time hardness even when the total share size in the Yes-case is polylogarithmic in L or hardness against $2^{o(n)}$-time algorithms where n is the number of variables at the expense of taking $\epsilon = 0.5$ in the case of Yes-instances. (See Theorem 4.3).

Remark 2.2 (Tightness)**.** *The theorem is tight in several aspects:*

1. *(Time) It is known that the task of checking whether an n-bit formula can be realized with 1-bit shares can be solved in time $\tilde{O}(2^{2n})$ [Gol98] (see also [ABI$^+$23a]). Therefore the ETH variant of the theorem that rules out the existence of a sub-exponential time algorithm provides a tight bound on the running time.*

2. *(Gap) The main statement (*coNP *hardness) establishes the largest possible gap. Indeed, the upper bound on the max-share size in the Yes case cannot be improved below 1 since one can efficiently test if the max-share size is zero (just check if the monotone formula f computes the trivial zero function by testing if $F(1^n) = 0$). Also, the lower bound $\Omega(L/\log L)$ on the total share size for No-instances cannot be improved by more than log-factor since any L-size formula can be realized by a secret sharing of total share size of L.*

3. *(Minor improvements = Better Lower-bounds) In fact, even an improvement of log-factor for the No instance (i.e., average-share size of $\Omega(\sqrt{L})$) seems hard since this would imply an explicit construction of (*) a monotone L-size formula f over $n = L/\Omega(\sqrt{L}) = O(\sqrt{L})$ variables with total share size of $\Omega(n^2)$. (To generate f, apply the reduction to some trivial No-instance of a* coNP*-complete problem.) Such a construction would improve the 3-decade-old lower-bound of Csirmaz [Csi96]. We further note that the proof of the theorem employs, as a gadget, a formula E that is known to be expensive for secret sharing. If this gadget is replaced with an explicit formula that satisfies (*), we will get rid of the logarithmic loss in the theorem. (Even a non-explicit proof for the existence of (*) suffices for proving* coNP*-hardness under non-uniform reductions.) Hence, improving the above theorem is equivalent to proving better lower-bounds.*

We note that, together with known results, the theorem establishes the coNP-completeness of the language IDEAL of all monotone formulas g that can be realized by a secret-sharing scheme with 1-bit shares. (See Section 4).

2.2 Hardness for Circuits

We move on to the case where the given function f is represented by a monotone circuit of size L. It is unknown whether such functions can be realized by secret sharing with total share size S of $\mathrm{poly}(L)$, let alone $O(L)$.[3] The following theorem rules out the existence of such a strong compiler for monotone circuits.

Theorem 2.3. *For every constant $\epsilon > 0$, given a monotone circuit f of size L it is* coNP*-hard to distinguish between the following cases:*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size of $O(L^\epsilon)$ bits.*
- *(No) Every SS for f has max-share size (or even average-share size) of $\Omega(L/\log L)$ and total share size of $\Omega(L^2/\log L)$.*

Recall that in our hardness results for formulas (Theorem 2.1) the "No" case corresponds to functions with max-share size of $\Omega(\sqrt{L}/\log L)$ and total share size of $\Omega(L/\log L)$. To get a quadratic improvement for circuits, we design functions with high secret-sharing complexity that can be realized by small circuits.

One could hope to prove hardness even for $\mathsf{GapSS}_{\mathrm{ckt}}[L^{0.1}, L^{\omega(1)}]$, i.e., even when the total share-size of the No-instance is super-polynomial in L. However, the lack of super-polynomial lower-bound prevents us from doing it unconditionally.[4] Instead, we can prove an all-or-nothing result: the problem $\mathsf{GapSS}_{\mathrm{ckt}}[L^{0.1}, L^{\omega(1)}]$ is either coNP-hard or "trivial", i.e., for infinitely many input lengths there are no "No" instances and an algorithm that always outputs "Yes" solves the problem (infinitely often). For the

[3] The best upper-bound that we have is either exponential in the depth (by turning the circuit into a formula) or exponential in the input length (by applying the general construction for arbitrary monotone functions).

[4] Indeed, even a quadratic lower-bound of $\Omega(L^2)$ would imply a quadratic lower-bound of $\Omega(n^2)$ on the total share-size of an n-variable function (since $n < L$), which is open (even non-constructively).

case of linear secret-sharing schemes (for which explicit sub-exponential lower-bounds are known [PR18]), one can prove an unconditional result; namely, that $\mathsf{GapLSS}_{\mathrm{ckt}}[L^{0.1}, 2^{L^\beta}]$ is coNP-hard for some constant $\beta > 0$. (See Section 4.3).

2.3 Cheap Truth-Tables Are Hard to Recognize

We move on to the case where the predicate $f : \{0,1\}^n \to \{0,1\}$ is represented by a truth-table of size $N = 2^n$. In this case, the distinguishing algorithm is allowed to run in $\mathrm{poly}(N)$ time and so it can efficiently check non-trivial properties of the function (e.g., the number of satisfying assignments). As a result, strategies like the one used in the context of formulas or circuits fail, and we resort to cryptographic assumptions. In particular, we assume the existence of a pseudorandom function with sub-exponential hardness that can be computed by a polynomial-size formula (see Assumption 5.2). This assumption, hereafter referred to as the sPRF (for simple PRF) assumption, follows from most standard cryptographic assumptions.[5]

Our result also relies on the hypothesis that some slice functions are expensive to realize. Formally, we say that a function $f : \{0,1\}^n \to \{0,1\}$ is a *slice* function if, for some *level parameter* k, the function f takes the value 0 (resp., 1) on all inputs of Hamming weight smaller than k (resp., larger than k), and can take arbitrary values over inputs of weight exactly k. Secret sharing for slices have been extensively studied [AA18, BKN18, BP18, LV18, ABF+19, AN21, ABN+22, BFLN24] and currently the best-known construction for arbitrary k's has total share size of $2^{\tilde{O}(\sqrt{\log N})}$ [LV18]. The following theorem shows that, unless this bound can be significantly improved, $\mathsf{GapSS}_{\mathrm{tt}}[\mathrm{poly}(n), n^{\omega(1)}]$ is intractable.

Theorem 2.4 (Hardness of GapSS for truth tables)**.** *Under the sPRF assumption, one of the following holds:*

- *(Slices are cheap) Every slice function f can be realized by a secret-sharing scheme with total share size polynomial in the number of variables.*
- *(Cheap truth-tables are hard to recognize) Given a monotone function $f : \{0,1\}^n \to \{0,1\}$ represented as a truth table of size $N = 2^n$, no $\mathrm{poly}(N)$-time algorithm can distinguish between the case where f has an LSS with total share size of $\mathrm{poly}(n)$ to the case where the total share size is at least $n^{\omega(1)}$.*

Put differently, assuming that sPRF holds and that slices are not cheap, it is infeasible to distinguish truth-tables of functions that can be realized with polynomial cost from functions that require super-polynomial cost. More aggressively, if one assumes that slices require sub-exponential cost of 2^{n^ϵ} for some constant $\epsilon > 0$, then the distinguishing problem remains hard even if the No case is restricted to functions that require sub-exponetial total share size of $2^{\Omega(n^\epsilon)}$. Indeed, we prove a general version of Theorem 2.4 that provides a smooth trade-off between the two items. We also note that for linear secret-sharing schemes, an exponential lower-bound of $\tilde{\Omega}(2^{n/2})$ is known, and so the intractability of $\mathsf{GapLSS}_{\mathrm{tt}}[\mathrm{poly}(n), \tilde{\Omega}(2^{n/2})]$ follows from the sPRF assumption. (See Section 5 for these extensions).

Worst-case to average-case reduction. The proof of the theorem (for the non-linear case) relies on a novel worst-case to average-case reduction for slice functions. Denoting by $E(n)$ the total share size of the *worst* slice function over n bits, we show that at least a constant fraction of all the slice functions in the middle layer $k = n/2$ require complexity of at least $\Omega(E(c \cdot n))$ for some constants c. (In fact, this holds for any layer $k = \alpha n$ for constant $\alpha \in (0,1)$.)

[5] In fact, this cryptographic assumption can be relaxed and replaced with the average-case hardness of the Minimum Formula Size Problem (MFSP) with respect to 1-sided algorithms. See Section 5.

Algorithms. We complement the above lower-bounds by describing several simple algorithms. We show that given a truth-table f of size N, it is possible to decide whether f can be realized by secret-sharing scheme with total complexity of S in deterministic time which is double exponential in S. While this bound is quite poor, it beats the naive bound obtained by enumerating over all possible probability distributions over $\{0,1\}^S$. Our algorithm can be significantly improved if the function is represented as a k-DNF as long as k is not too large. In the extreme case, when k is a constant, we can check if f can be realized by a secret sharing whose maximum share size is constant in single-exponential time of $\exp(\mathrm{poly}(n))$. (See A for details.)

2.4 Recognizing Cheap Graphs and Randomized Communication Complexity

In light of the aforementioned hardness results, we move on to study an extremely simple, yet non-trivial, class of monotone functions known as *graph* functions. Given a graph G over a set V of nodes, we define a monotone function $f_G : 2^V \to \{0,1\}$ that outputs 1 on a set $S \subset V$ if S contains an edge. That is, the edges of G form the minimal authorized sets of f_G, and so f_G can be described as a monotone 2-DNF. Graph secret sharing was studied in many works (e.g., [BSGV92, BSSV95, vD95, Csi05, Csi09, BGP97, CT13, Sti94, BFM16, BFMP17, BF20]) and proved to be a useful scaled-down version of general monotone functions. To make the GapSS problem even simpler, we further restrict our attention to the case where G is a bipartite graph with n left nodes and n right nodes and try to identify the cheapest graphs among this class that can be realized by a secret-sharing scheme with a *constant* maximal share size. The hope is that this class will be simple enough to be recognized efficiently. It turns out that this problem is tightly connected to questions in the domain of communication complexity.

Let us associate with a bipartite graph a two-party communication problem where Alice gets a left vertex, Bob gets a right vertex, and their goal is to determine whether their vertices are connected by an edge. (That is, the biadjacency matrix of the graph is being used as a two-party communication matrix.) We consider the coRP (resp., the RP model) of *randomized communication complexity* in which Alice and Bob share a common random string and are allowed to output the wrong answer on No (resp., Yes) instances with constant probability. We also consider a slightly non-standard model of *Nondeterministic 1-Equality communication protocols* [BBM$^+$20, PSS23] in which the parties are given an advice and a single access to an equality oracle. (See Section 6 for a formal definition). We show that the class of graphs that have low secret-sharing cost is strongly related to the class of graphs that can be computed with constant communication complexity in these models.

Theorem 2.5 (Informal)**.** *The followings hold for every (family of) bipartite graphs G.*

1. $\mathsf{LS}_{\max}(G) = O(1)$ *if and only if G has $O(1)$ communication in the Nondeterministic 1-Equality model.*
2. *If* $\mathsf{S}_{\max}(G) = O(1)$ *then* $\mathsf{coRP}(G) = O(1)$*. Conversely, if* $\mathsf{RP}(G) = O(1)$*, then G can be realized by a secret-sharing scheme that distributes* $\log n$*-bit secrets with* constant normalized share-size*, i.e., the maximal share size is* $O(\log n)$*.*

Here $\mathsf{S}_{\max}(G)$ denotes the maximal share-size of the function f_G and $\mathsf{LS}_{\max}(G)$ denotes its LSS analogue. The proof establishes quantitative lower-bounds and upper-bounds on secret-sharing complexity in terms of communication complexity. These bounds are *dimension-free* [HH24], and do not depend on the size of the graph. (We also derive a dimension-free upper-bound on a closely-related secret-sharing complexity measure in terms of coNP complexity.) Our results continue a recent line of works [GKW15, AV21]

that relate the complexity of secret-sharing schemes to communication complexity measures. In the context of meta-complexity, these results show that the problem of recognizing graphs that admit "cheap" secret-sharing schemes is closely related to the problem of recognizing graphs that have low-communication protocols.

Indeed, Theorem 2.5 allows us to import tools and results from communication complexity to the secret-sharing domain. For example, communication in the Nondeterministic 1-Equality model is characterized by the logarithm of the graph's so-called *P4-free cover number* [BBM$^+$20].[6] Accordingly, we get a combinatorial criteria for determining whether a graph can be realized by an LSS with constant share-size. This allows us to derive a new lower-bound on the LSS complexity of a concrete family of graphs (essentially corresponding to the Disjointedness function) that was studied by [LVW17]. (See Corollary 6.4). Unfortunately, this criteria does not lead directly to an efficient algorithm since this covering number is NP-hard to compute [BBM$^+$20]. Still, one may hope to develop an efficient approximation algorithm, which would enable us to distinguish between the case where $\mathsf{LS}_{\mathsf{max}}(G) = O(1)$ and the case where, say, $\mathsf{LS}_{\mathsf{max}}(G) = \Omega(\log n)$ or even $\mathsf{LS}_{\mathsf{max}}(G) = \Omega(n^{0.5})$.

The second item of the Theorem 2.5 shows that graphs with constant share-size form a sub-class of graphs with constant randomized complexity – a class that has been extensively studied in the last few years (see, e.g., [HHH21, HWZ22, HH24, FHHH24] for motivation and additional references). Interestingly, this class corresponds to the class of graphs that admit *constant-size adjacency sketches* [HWZ22]. That is, one can randomly assign to each vertex of the graph a constant-size label such that a decoder algorithm that is given a pair of labels can tell, with probability 2/3, whether the corresponding vertices are adjacent without seeing the graph. This notion, originally defined by [FK09] can be viewed as a probabilistic version of the classical notion of adjacency labeling schemes [KNR92], and it is known that constant-size adjacency sketches imply deterministic adjacency labeling schemes of logarithmic size [Har20]. The question of understanding which graphs admit such labeling schemes is a well-known open problem. Understanding which graphs admit constant share size can be therefore viewed as a refined version of the problem that addresses an important special case.

Hardness against myopic algorithms. We do not know whether the graph version of $\mathsf{GapSS}[O(1), g(n)]$ is tractable for some non-trivial function $g(n) = \omega(1)$. One natural algorithmic strategy is to check whether the given graph G can be locally-realizable by a cheap secret-sharing scheme. That is, use exhaustive search to verify that every k-size sub-graph G' of G can be realized with constant cost of $O(1)$, and accept the graph if all tests pass. This algorithm accepts Yes instances: Indeed, if the graph G can be realized with constant max-share size of S then so is every sub-graph of G. The hope is that No instances that require high-cost will be rejected. That is, if the graph is sufficiently "hard", then one can find a small obstacle that explains for its hardness. Unfortunately, we show that this *k-myopic algorithm* fails even for $k = \sqrt{n}$.

Theorem 2.6 (obstacle against myopic algorithms)**.** *There exists a family of bipartite graphs G that requires shares of size $\Omega(\log n)$ from any secret-sharing scheme that realizes it, but every induced subgraph of G with $\sqrt{n}$ vertices on each side can be realized with maximal share size 8.*

The proof is obtained by sampling a random sparse graph. This follows the approach used in [HHH22, HH22] to refute similar local-to-global conjectures with respect to adjacency labels/sketches known (i.e., the probabilistic universal graph conjecture and the *implicit graph conjecture*).

[6] This quantity measures the minimum number of P4-free bipartite graphs that are needed to cover the edges of the given graph. A bipartite graph is P4-free if it is a vertex-disjoint union of complete bipartite graphs. Such graphs arise naturally (sometimes under different terminology such as "blocky matrices" or "fat matchings") in computational complexity and communication complexity [PR94, Juk06, AY24, HHH21, HWZ22]. For a survey on the subject see [Xie24].

We note that the $\Omega(\log n)$ bound in Theorem 2.6 matches the best-known lower-bound on the maximal-share size of an n-vertex graph [vD95, Csi09]. (For linear schemes, this can be improved to $\Omega(\sqrt{n})$ by [BGP97], and in both cases, the best known upper-bound [EP97] is $O(n/\log n)$.)

2.5 Discussion and Open Problems

Our paper provides some initial results on the meta-complexity of secret sharing leaving many interesting open questions. We list a few of them.

1. Design better GapSS algorithms for truth tables. Is it possible to distinguish in time $\mathrm{poly}(N)$ or even in time $N^{O(\log N)}$ between functions whose maximal share size is 2 from functions with $2^{\Omega(n)}$ share size (assuming that such functions exist)? Can this be done by a myopic algorithm that checks all k-restrictions of f for some small $k = o(n)$?

2. Efficiently approximate the P4-free covering number of a graph within a constant (or slightly super-constant) approximation factor.

3. Is it possible to extend the worst-case to average-case reductions to DNFs or even 2-DNFs (i.e., graphs)? Since randomly chosen DNFs are currently cheaper to realize [BF20, AN21], such a reduction would lead to improvement in the worst-case complexity of graphs, and, by known results, can even lead to improvements for worst-case general secret-sharing schemes.

4. The question of constructing *computationally efficient* information theoretic secret-sharing schemes for polynomial-size monotone circuits has been wide open for over four decades. That is, we do not know whether there exists a compiler that takes a monotone circuit of size n^c for some constant $c > 0$ and generates an information-theoretic secret-sharing whose sharing and recovery algorithms run in time, say $n^{c'}$ for some $c' > c$. To rule out such a result, it suffices to prove lower-bounds against *efficient* compilers – a task that may be easier than proving information-theoretic lower-bounds against arbitrary (possibly inefficient) secret-sharing schemes. Is it possible to use meta-complexity arguments to establish such lower-bounds? Less ambitiously, show that, under some potentially strong computational assumptions, secret-sharing compilers for monotone circuits must have super-quadratic overhead improving over the quadratic lower-bound established in this paper.

Organization. Following some preliminaries, we study GapSS for formulas and circuits in Section 4, for truth-tables in Section 5, and for graphs in Section 6. Our basic algorithms appear in Appendix A.

3 Preliminaries

Definition 3.1 (Secret-sharing schemes and linear secret-sharing schemes)**.** *A* secret-sharing scheme*, with domain of secrets S, domain of random strings R, and finite domains of shares $S_1, \ldots, S_n$, is a deterministic function $\mathcal{D} : S \times R \to S_1 \times \cdots \times S_n$. A dealer distributes a secret $s \in S$ according to $\mathcal{D}$ by first sampling a random string $r \in R$ with uniform distribution, computing a vector of* shares *$\mathcal{D}(s, r) = (s_1, \ldots, s_n)$, and privately communicating each share s_i to the ith party. For a binary string $x \in \{0,1\}^n$ representing a set $I_x = \{i : x_i = 1\}$, we denote $\mathcal{D}_x(s, r)$ as the restriction of $\mathcal{D}(s, r)$ to the I_x-entries (i.e., the shares of the parties in I_x). When r is omitted $\mathcal{D}(s)$ (resp., $\mathcal{D}_x(s)$) denotes the probability distribution of $\mathcal{D}(s, r)$ (resp., $\mathcal{D}_x(s, r)$) induced by sampling $r \in R$ uniformly at random.*

A secret-sharing scheme $\mathcal{D}$ realizes a monotone predicate $f : \{0,1\}^n \to \{0,1\}$ if the following two requirements hold:

1. ***Perfect Correctness:*** *The secret s can be reconstructed by any authorized set of parties. That is, for every x such that $f(x) = 1$ there exists a* reconstruction function Recon_x *such that for every secret $s \in S$ and every random string $r \in R$, it holds that $\mathrm{Recon}_x(\mathcal{D}_x(s,r)) = s$. Equivalently, for any pair of secrets $s \neq s'$, the supports of $\mathcal{D}_x(s)$ and $\mathcal{D}_x(s')$ are disjoint.*

2. ***Perfect privacy:*** *An unauthorized set of parties learns nothing about the secret from its shares. Formally, for every x such that $f(x) = 0$ and every pair of secrets $s, s' \in S$, the random variables $\mathcal{D}_x(s)$ and $\mathcal{D}_x(s')$, are identically distributed.*

The secret size *of $\mathcal{D}$ is defined as $\log |S|$, the* total share size *is defined as $\sum_{1 \leq i \leq n} \{\log |S_i|\}$ and the* maximal share size *(or simply share size) of the scheme is defined as the largest share size, i.e., $\max_{1 \leq i \leq n} \{\log |S_i|\}$. The* normalized total share size *and (rep.,* normalized maximal share size*) is defined as the ratio between the total share size (resp., maximal share size) and the secret size. The scheme $\mathcal{D}$ is a* linear secret-sharing scheme over a finite field $\mathbb{F}$ *if $S = \mathbb{F}$, $R = \mathbb{F}^\ell$ for some integer $\ell \geq 1$, the sets $S_1, \ldots, S_n$ are vector spaces over $\mathbb{F}$, and the function $\mathcal{D} : \mathbb{F}^{\ell+1} \to S_1 \times \cdots \times S_n$ is a linear mapping over $\mathbb{F}$.*

By default, we assume that the secret size is 1, i.e., $S = \{0,1\}$ and that linear secret-sharing scheme are defined over the binary field $\mathbb{F}_2$. The total share size (resp., maximal share size) of a monotone function f are taken to be the smallest total share size (resp., smallest maximal share size) of a secret-sharing scheme that realizes f.

4 Hardness of GapSS for Formuals and Circuits

4.1 Secret-Sharing Lower-Bound for a Simple Function

Our hardness result will be based on a secret-sharing lower bound for relatively small formulas/circuits. The following theorem is based on a combination of Csirmaz's Lower Bound [Csi96] with techniques from [Bei23, ABI+23b].

Theorem 4.1. *There exists a polynomial-time uniform family of monotone formulas (resp., monotone circuits) $\{E_t\}_{t \in \mathbb{N}}$ such that E_t is a size-t formula over $\Theta(\sqrt{t})$ variables (resp., size-t circuit over $\Theta(t)$ variables) and has average-share size of at least $\Omega(\sqrt{t}/\log t)$ (resp., $\Omega(t/\log t)$).*

Proof. To prove the theorem, we will describe a monotone function G over $O(n)$ variables whose average-share size is $\Omega(n/\log n)$ and later show that it can be realized by a monotone formula of size $O(n^2)$ and monotone circuit of size $S = O(n)$. We begin by defining a somewhat simpler function F and use it to construct G.

Let n be an integer and let $k = \lceil \log n \rceil$. First, consider the function F that takes n variables $x = (x_1, \ldots, x_n)$ and k variables $y = (y_1, \ldots, y_k)$ and outputs

$$F(x,y) = \bigvee_i \left(x_i \wedge \bigwedge_{j \in S_i} y_j \right)$$

where $S_1, \ldots, S_n$ range over all subsets of $[k]$.

Claim 4.2. *In any secret sharing that realizes F, the total share size of the y's is $\Omega(n)$.*

The proof is inspired by an argument of [Bei23].

Proof of claim. Assume that the sets $S_1, \ldots, S_n$ are ordered in a non-increasing order, i.e., if $i < j$ then $S_i \not\subset S_j$. (If this is not the case, rename the x variable so that the above holds.) Now, consider the monotone function $F'(w, y)$ that takes the variables $w = (w_1, \ldots, w_n)$ and the y variables, substitutes $x_i = \bigwedge_{j \le i} w_j$ and outputs $F(x, y)$. Csirmaz [Csi96] proved that in any scheme that realizes F' the total share size of the y's is $\Omega(n)$. We will show that the same bound applies to F via an efficient reduction. Indeed, by the closure properties of secret-sharing [BL88], a secret-sharing for F implies a secret-sharing for F' that preserves the share-size of the y parties. Specifically, we can share the secret according to F among the parties associated with (x, y), and for each $i \in [n]$, re-share the share of the party x_i among the parties $(w_1, \ldots, w_i)$ via an additive secret-sharing. The claim follows. □

Let us define a new function G by taking the function F and substituting each variable y_i with the expression $\bigvee_{j \in \lceil n/k \rceil} y_{i,j}$ where $Y = (y_{i,j})_{i \in [k], j \in [\lceil n/k \rceil]}$ are new formal variables. The resulting function G is now a monotone function over the variables x and Y. By design, for every $j \in \lceil n/k \rceil$ one can derive the function $F(x, y_{1,j}, \ldots, y_{k,j})$ as a sub-function of G (by setting the variable's $(y_{i,\ell})_{i \in [k], \ell \neq j}$ to zero). Hence, the total share size of each "column" j of variables $(y_{1,j}, \ldots, y_{k,j})$ is $\Omega(n)$ and the total share size is at least $\Omega(n^2/k) = \Omega(n^2/\log n)$. Since G is defined over $O(n)$ variables the average-share size is $\Omega(n/\log n)$. (This transformation was previously used in [ABI+23b]).

The function G can be realized by an $O(n^2)$ size formula: Take the $O(n \log n)$-size formula for F and replace each y_i by an OR-tree of size $O(n/\log n)$. To derive an $O(n)$-size circuit for G, it suffices to describe an $O(n)$-size circuit for F. For this, we first construct a circuit C_k of size $O(2^k)$ that given k inputs $y = (y_1, \ldots, y_k)$ outputs all 2^k monotone terms $(z_S = \bigwedge_{j \in S} y_j)_{S \subset [k]}$. The circuit is constructed recursively: Compute $C_{k-1}(y_1, \ldots, y_{k-1}) = (z_S)_{S \subset [k-1]}$ and output $(z_S, z_S \wedge y_k)_{S \subset [k-1]}$. Since $|C_k| \le |C_{k-1}| + 2^k$ the circuit size is $O(2^k)$. Finally, we can construct a circuit for F based on a circuit for C_k with $k = \log n$ plus $O(n)$ additional gates, and so the total circuit size of G will be $O(n)$, as required. □

4.2 Hardness of GapSS for Formulas

We prove the following theorem:

Theorem 4.3 (GapSS is coNP-hard for formulas)**.** *For every constant $\epsilon > 0$, given a monotone formula f of size L it is* coNP*-hard to distinguish between the following cases:*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size of $O(L^\epsilon)$ bits.*
- *(No) Every SS for f has max-share size (or even average-share size) of $\Omega(\sqrt{L}/\log L)$ and total share size of $\Omega(L/\log L)$.*

Furthermore, under the Exponential-Time Hypothesis (ETH), the following extensions hold.

1. *No polynomial-time algorithm can distinguish between the above No case and the case where f has an LSS with max-share size of 1 bit and total share size of $O((\log L)^c)$ bits for any constant $c > 1$.*
2. *Let n denote the number of variables in f, then no $2^{o(n)}$-time algorithm can distinguish between the above No case and the case where f has an LSS with max-share size of 1 bit and total share size of $O(\sqrt{L})$ bits.*

Proof. The proof is via a Karp reduction R from the coNP-hard problem *Tautology*. In this problem, we are given a (non-monotone) formula g over m variables $\mathbf{x} = (x_1, \ldots, x_m)$ and the goal is to determine whether g is a tautology; that is, whether every possible assignment of true/false values to variables yields a true statement. We assume without loss of generality that the formula is of size $M = O(m^a)$ for some constant $a \geq 1$. (In fact, the problem is coNP-hard even for $O(m)$-size formulas, e.g., 3DNF where each variable occurs a constant number of times.)

Let E_t be the t-size monotone formula promised in Theorem 4.1 that has $k(t) = O(\sqrt{t})$ variables and average-share size of at least $S(t) = \Omega(\sqrt{t}/\log t)$. The parameter t will be chosen to be large enough so that t dominates M, i.e., $t = \Omega(M)$, and such that the number of variables of E_t dominates the number of variables of g, i.e., $k(t) = \Omega(m)$. The exact value of t will be determined later.

The reduction. The reduction R runs in time $\mathrm{poly}(t+m)$ and takes g as an input and generates a monotone formula f over $n = 2m + k(t) = O(k(t))$ variables. Assume, without loss of generality, that all the NOT gates of the formula g are placed in the bottom layer. (This can be done by using the standard De-Morgan based transformation), and let g' denote the monotonized version of g where the negations of every variable x_i are replaced with a new variable x'_i. Define h be the following formula over $2m$ variables:

$$h(\mathbf{x}, \mathbf{x}') := \bigwedge_{i \in [m]} (x_i \vee x'_i).$$

Then, the desired formula f is defined as follows:

$$f := h(\mathbf{x}, \mathbf{x}') \wedge \left(g'(\mathbf{x}, \mathbf{x}') \vee E_t(\mathbf{y})\right).$$

The formula f has $2m + k(t) = n$ input variables, $\mathbf{x} = (x_1, \ldots, x_m)$, $\mathbf{x}' = (x'_1, \ldots, x'_m)$ and $\mathbf{y} = (y_1, \ldots y_{k(t)})$ and it is of size $\Theta(M) + t$. We analyze the reduction.

Claim 4.4. *The reduction R runs in time $\mathrm{poly}(t + m)$ and generates a monotone formula f over $n = 2m + k(t) = O(k(t))$ variables whose size is $L = t + \Theta(M) = O(t)$ such that if g is a tautology then f can be realized by an LSS with maximal share size of 1 and total share size of $2m$, and if g is not a tautology f can only be realized with an average-share size of $\Omega(S(t))$ and total share size of at least $\Omega(S(t)n) = \Omega(S(t)k(t))$.*

Proof of claim. Suppose that g is a tautology, i.e., $g(\mathbf{x}) = 1$ for every assignment $\mathbf{x} \in \{0,1\}^m$. We show that $f(\mathbf{x}, \mathbf{x}', \mathbf{y}) = h(\mathbf{x}, \mathbf{x}')$ for every $\mathbf{x}, \mathbf{x}'$ and $\mathbf{y}$. We distinguish between the following cases. (1) Call $(\mathbf{x}, \mathbf{x}')$ *legal* if exactly one of each pair of variables $(x_i, x'_i)_{i \in [m]}$ equals 1 (i.e., $x'_i = \neg x_i$). In this case, h is satisfied and so is f. To see the latter, note that since g is a tautology, $g'(\mathbf{x}, \mathbf{x}') = 1$ by definition. (2) Next, consider the case where for some $i \in [m]$ it holds that $x_i = x'_i = 0$. In this case, $h = 0$ which means that f is also 0. (3) Finally, assume that h is satisfied and the assignment $(\mathbf{x}, \mathbf{x}')$ is strictly larger than some legal assignment $(\mathbf{x}_0, \mathbf{x}'_0)$.[7] In this case, the assignment $(\mathbf{x}, \mathbf{x}', \mathbf{y})$ satisfies f since f is monotone and since $(\mathbf{x}, \mathbf{x}', \mathbf{y})$ is strictly larger than the assignment $(\mathbf{x}_0, \mathbf{x}'_0, \mathbf{y})$ that satisfies f by (1). It follows that $f = h$ in all these cases. It now remains to show that the function computed by h can be realized with share size 1 and total share size of $2m$. This can be done by following its formula representation (as implied by the known connections between formulas and secret sharing [BL88]): Split the secret s with an additive scheme to m shares $s_1, \ldots, s_{m-1}, s_m$, and for every $i \in [n]$ deal s_i to the parties represented by x_i and x'_i.

[7] An assignment z is strictly larger than an assignment w if for each i, it holds that $w_i \leq z_i$ and for some i, $w_i < z_i$.

Next assume that g is not a Tautology, so there exists a legal assignment $\mathbf{x}_0, \mathbf{x}'_0$ such that $g'(\mathbf{x}_0, \mathbf{x}'_0) = 0$. Since the assignment is legal, $h(\mathbf{x}_0, \mathbf{x}'_0) = 1$, so the restricted function $f(\mathbf{x}_0, \mathbf{x}'_0, \mathbf{y})$ simplifies to $E_t(\mathbf{y})$. By Theorem 4.1 the total share size of the residual function is $\Omega(t^2/\log t)$ and the max-share size is $\Omega(t/\log t)$. Since a lower-bound on the total share size of a residual function implies a similar lower-bound on the total share size of the original function, the claim follows. □

To derive the main result, take $t = m^{\max(1/\epsilon, a)}$. This implies that, on one hand, $m = O(L^\epsilon)$, and on the other hand, $\Omega(S(t)) = \Omega(\sqrt{L}/\log L)$ and $\Omega(S(t)k(t)) = \Omega(L/\log L)$, as required.

To prove the "Furthermore" part, recall that ETH asserts that we cannot determine whether g is a tautology in time $2^{o(m)}$ even when $M = O(m)$[8]. To prove the first item, take $t = 2^{m^{1/c}}$. The reduction then runs in $2^{o(m)}$ time, produces an instance of size $L = 2^{o(m)}$ whose max-share size (resp., total share size) is either 1 (resp., $2m = (\log L)^c$) in the Yes case, or $\Omega(L/\log L)$ (resp., $\Omega(L^2/\log L)$) in the No case, as required. To prove the second item, set the parameter t to m^2. The resulting formula f is of size $L = O(m^2)$ and has $n = 2m + O(\sqrt{t}) = O(m)$ variables, therefore a $2^{o(n)}$ algorithm for the secret-sharing problem implies an algorithm that determines tautology in time $2^{o(n)}$, contradicting the ETH. □

Remark 4.5 (Comparison to [BSGV92])**.** *Blundo et al. [BSGV92] identified some concrete "mildly-hard" function h whose normalized max-share size is a non-trivial constant $c > 1$ and proved that it is NP-complete to determine whether a given monotone formula f contains h as a sub-function. Their reduction maps Yes instances of SAT to "mildly-hard" access structures and No instances to functions that do not contain the function h. However, it is not clear whether the latter case is mapped to "easy" access structures, and so their reduction does not imply NP-hardness of GapSS.*

Let us record the following abstract version of Theorem 4.3 whose proof follows from the proof of Theorem 4.3.

Proposition 4.6 (secret-sharing lower-bounds imply computational hardness)**.** *Let $E = \{E_t\}_{t\in\mathbb{N}}$ be a polynomial-time uniform family of t-size monotone formulas (resp., monotone circuits) over $k(t)$ variables such that any secret sharing that realizes E_t has average-share size of at least $S(t)$. Then, for every $\epsilon > 0$ there exists an efficient reduction from Tautology to the following promise problem over L-size monotone formulas f (resp., circuits):*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size of $O(L^\epsilon)$ bits.*
- *(No) Every SS for f has an average-share size of $\Omega(S(L - L^\epsilon))$ and total share size of $\Omega(S(L - L^\epsilon) \cdot k(L - L^\epsilon))$.*

Furthermore, the theorem extends to the case where the family E is non-uniform at the expense of getting a non-uniform reduction. Similarly, the theorem applies to the case where the lower-bound against E holds only against linear secret-sharing schemes at the expense of relaxing the No-case to LSS.

Getting back to Theorem 4.3, we note that, together with known results, it establishes the coNP-completeness of the language IDEAL of all monotone formulas g that can be realized by a secret-sharing scheme with 1-bit shares.

Corollary 4.7. IDEAL *is* coNP*-complete.*

[8] This is originally proved for Satisfiability [IP99] but the same statement holds for tautology since satisfiability reduces to tautology via a (trivial) linear-time Cook reduction.

Proof. The proof of Theorem 4.3 establishes coNP-hardness (even for the case where f is given as a 3DNF formula). To prove that the problem is in coNP we use the combinatorial criteria of [ABI$^+$23a] that asserts that a monotone function f is not in IDEAL if and only if there exists a minimal satisfying assignment (min-term) M and a maximal unsatisfying assignment (max-term) T whose difference $M \setminus T$ is of even size. The pair (M, T) thus forms a witness for being out of IDEAL and verification consists of checking that (1) M is a min-term (i.e., that $f(1_M) = 1$ and that $f(1_{M\setminus\{i\}}) = 0$ for every $i \in M$); (2) that T is a max-term (i.e., that $f(1_T) = 0$ and that $f(1_{T\cup\{i\}}) = 1$ for every $i \notin T$); and (3) that $|M \setminus T|$ is odd. □

The corollary also holds for the language $\text{IDEAL}_{\text{non-monotone}}$ of all (possibly non-monotone) formulas that compute a (monotone) function that can be realized by a secret-sharing scheme with 1-bit shares. The coNP-hardness follows from Theorem 4.3, and membership in coNP holds by either presenting a certificate for the non-monotonicity of f (i.e., a pair of assignments $x \leq y$ for which $f(x) > f(y)$) or by using the certificate that corresponds to the criteria of [ABI$^+$23a].

4.3 Hardness for Circuits

We move on to the case where the given function f is represented by a monotone circuit of size L. It is unknown whether such functions can be realized by secret sharing with total share size S of $\text{poly}(L)$, let alone $O(L)$.[9] Hence, unlike the case of formulas, we can hope to prove meaningful results even when the total share size S of the "No case" is larger than L. Indeed, the following theorem establishes such a hardness result with almost quadratic S.

Theorem 4.8. *For every constant $\epsilon > 0$, given a monotone circuit f of size L it is* coNP*-hard to distinguish between the following cases:*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size of $O(L^\epsilon)$ bits.*
- *(No) Every SS for f has max-share size (or even average-share size) of $\Omega(L/\log L)$ and total share size of $\Omega(L^2/\log L)$.*

Proof. By Theorem 4.1, there exists a polynomial-time uniform family of monotone circuits $E = \{E_t\}_{t\in\mathbb{N}}$ such that E_t is a size-t circuit over $\Theta(t)$ variables and has average-share size of at least $\Omega(t/\log t)$. The proof follows by instantiating Proposition 4.6 with E. □

One could hope to prove hardness even when the total share-size of the No-instance is super-quadratic larger than L or even super-polynomial in L. However, the lack of super-quadratic lower-bound prevents us from doing it unconditionally.[10] Instead, we can prove an all-or-nothing result: Either the problem is coNP-hard or the problem is "trivial", i.e., for infinitely many input lengths there are no "No" instances.

Theorem 4.9 (All-or-nothing hardness for circuits)**.** *For constants $c \geq 1$ and $0 < \epsilon < 1$, the input to the promise problem $\Pi_{c,\epsilon}$ is an L-size monotone circuit f and the goal is to distinguish between*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size $O(L^\epsilon)$.*
- *(No) Every SS for f has average-share size of at least $\Omega(L^c)$ bits.*

[9] The best upper-bound that we have is either exponential in the depth (by turning the circuit into a formula) or exponential in the input length (by applying the general construction for arbitrary monotone functions).

[10] Indeed, even a quadratic lower-bound of $\Omega(L^2)$ would imply a quadratic lower-bound of $\Omega(n^2)$ on the total share-size of an n-variable function (since $n < L$), which is open (even non-constructively).

Then, for every ϵ and c, the problem $\Pi_{c,\epsilon}$ is either coNP*-hard under non-uniform reduction or* trivial *in the sense that, for every constant $a > 0$ there are infinitely many integers L, such that all the L-size monotone circuits can be realized with average-share size smaller than $a \cdot L^c$. Moreover, the same holds for the problem $\Pi_{\omega(1),\epsilon}$ where in the No case the lower-bound is $L^{\omega(1)}$.*

Proof. We begin with the case where $c \geq 1$ is a constant. Suppose that the problem is non-trivial. That is, there is an infinite sequence of monotone circuits $E = \{E_t\}_{t\in\mathbb{N}}$ and constant $a > 0$ such that E_t is of size t and for all sufficiently large t's, the average-share size of E_t is at least at^c bits. By Proposition 4.6, it follows that $\Pi_{c,\epsilon}$ is coNP-hard under a non-uniform reduction. Similarly, for super-constant c, assume that the problem is non-trivial and that $E = \{E_t\}_{t\in\mathbb{N}}$ is a family of t-size monotone cicruits with average-share size of $t^{\omega(1)}$, then by Proposition 4.6, $\Pi_{\omega(1),\epsilon}$ is coNP-hard under a non-uniform reduction. □

Finally, for linear schemes, we can prove an unconditional hardness result with sub-exponential gap by exploiting existing LSS lower-bounds.

Theorem 4.10 (Hardness of gapLSS for circuits)**.** *There exists a constant $\beta > 0$ such that for every $\epsilon > 0$, given an L-size monotone circuit f it is* coNP*-hard to distinguish between the following cases:*

- *(Yes) f has an LSS with max-share size of 1 bit and total share size $O(L^\epsilon)$.*
- *(No) Every LSS for f has total share size of $2^{\Omega(L^\beta)}$ bits.*

Proof. In [PR18], Pitassi and Robere describe a poly-time uniform family of polynomial-size monotone functions $E = \{E_t\}_{t\in\mathbb{N}}$ such that E_t is computable by a t-size monotone circuit over $k(t) = \mathrm{poly}(t)$ inputs and where every LSS for E_t has total share size of at least $2^{\Omega(t^\alpha)}$ for some fixed constant $\alpha > 0$.[11] This implies that the average-share size is $2^{\Omega(t^\beta)}$ for some constant $\beta > 0$. By Proposition 4.6, the theorem follows. □

5 Hardness of Truth-Table GapSS

In this section, we prove Theorem 2.4. We begin with some cryptographic definitions.

Definition 5.1. *An ensemble of S-size circuits (resp., formulas) is given by a probabilistic-time algorithm D that on an input 1^n samples an $S(n)$-size circuit (resp., formula) $f : \{0,1\}^n \to \{0,1\}$. We say that the ensemble is T-pseudorandom if for every $T(n)$-time non-uniform adversary $\mathcal{A}$ and every sufficiently large n, it holds that*

$$\left| \Pr_{f\leftarrow D(1^n)} \left[\mathcal{A}^{f(\cdot)}(1^n) = 1\right] - \Pr_{g\leftarrow B_n} \left[\mathcal{A}^{g(\cdot)}(1^n) = 1\right] \right| \leq 1/T(n),$$

where B_n denotes the set of all functions from $\{0,1\}^n$ to $\{0,1\}$. By default, we assume that D is $T(n)$-pseudorandom for some $T(n) \in 2^{\omega(n)}$, and refer to it simply as a strong-PRF *(sPRF in short). In this case, we may assume that the adversary gets the entire 2^n-size truth-table of the function as an input. Measuring the running-time as a function of the input-length $N = 2^n$, we note that pseudorandomness holds against any adversary that runs in time polynomial in N (since $\mathrm{poly}(N) = 2^{O(n)}$).*

[11] The lower-bound applies for LSS over an arbitrary finite field $\mathbb{F}$ and it follows from a similar lower-bound on the size of the monotone span program over $\mathbb{F}$ for E.

The existence of sPRF (computable by $\mathrm{poly}(n)$-size circuits) is a mild assumption that follows from the existence of sub-exponentially hard one-way functions [HILL99, GGM86]. We will need a slightly stronger variant of this assumption in which the sPRF is computable by polynomial-size formulas (or, equivalently NC^1 circuits).

Assumption 5.2. *For some polynomial $S(n)$, there exists an sPRF ensemble of S-size formulas.*

Assumption 5.2 follows from most standard cryptographic assumptions such as the sub-exponential hardness of lattice problems, the sub-exponential hardness of factoring, the sub-exponential hardness of the Decisional-Diffie Hellman assumption, or asymptotic variants of ad-hoc constructions of block-ciphers. (See [BR17] for a survey.)

Recall that a (k, n)-slice function is a function $f : \{0,1\}^n \to \{0,1\}$ that takes the value 0 on all inputs of Hamming weight smaller than k, takes the value 1 on all inputs of Hamming weight larger than k, and can take arbitrary values over inputs of weight exactly k. When k is unspecified we refer to f as a slice function. As mentioned in the introduction, our hardness result is based on the hypothesis that secret sharing is expensive for some slice function.

Hypothesis 5.3 (some slices are E-expensive)**.** *There exists a sequence of slice functions*

$$f = \{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$$

that requires super-polynomial total share size of $E(n)$.

The hypothesis is parameterized by a super-polynomial function E. Assuming that existing constructions [LVW18] are almost optimal, $E(n)$ can be taken to be $2^{\Omega(\sqrt{n})}$. More conservatively, one may assume that E is sub-exponential, i.e., $E(n) = 2^{\Omega(n^\epsilon)}$ for some $\epsilon > 0$, or leave it as an specified super-polynomial function $E(n) = n^{\omega(1)}$.

In Section 5.1 we prove the following lemma.

Lemma 5.4 (worst-case to average-case hardness of slices)**.** *If Hypothesis 5.3 holds with respect to some cost function $E(n)$ then, for every constant $\alpha \in (0,1)$ there exists a constant β, such that at least $1/4$ fraction of $(\lceil \alpha n \rceil, n)$-slices require total share size of $E'(n) > E(\beta n)$.*

In particular, if $E(n) = 2^{\Omega(n^\epsilon)}$ (resp., $E(n) = n^{\omega(1)}$) then so is $E'(n)$. To prove the following theorem, we employ the lemma with the special case where $\alpha = 0.5$ and denote by β the corresponding constant.

Theorem 5.5 (Hardness of GapSS for truth tables, Thm 2.4 restated)**.** *Under Assumption 5.2 and Hypothesis 5.3, there exists a polynomial $S(n)$ for which the following holds. Given a monotone function $f : \{0,1\}^n \to \{0,1\}$ represented as a truth table of size $N = 2^n$, no $\mathrm{poly}(N)$-time algorithm can distinguish between the following cases:*

- *(Yes) f has an LSS with total share size of $S(n) \in \mathrm{polylog}(N)$.*
- *(No) Every SS for f has total share size of at least $E(\beta n) = n^{\omega(1)} = (\log N)^{\omega(1)}$ where E is the hardness parameter assumed in Hypothesis 5.3 and β is some universal constant.*

Assuming the optimality of existing constructions for slices [LVW18], the No case has total share size of $2^{\Omega(\sqrt{n})} = 2^{\Omega(\sqrt{\log N})}$.

Proof. Under Assumption 5.2, there exists an sPRF D computable by formulas of polynomial size $S'(n)$. Assume, towards a contradiction that there exists a $\mathrm{poly}(N)$-time algorithm $\mathcal{A}$ that distinguishes the Yes case with $S(n) = O(S'(n) \cdot n \log n)$ from the No case. We will use $\mathcal{A}$ to break the pseudorandomness of D as follows. Given a truth table $T : \{0,1\}^n \to \{0,1\}$ we define the slice function

$$h(x) = (T(x) \vee \mathrm{Thr}_{\lceil 0.5n \rceil + 1}(x)) \wedge \mathrm{Thr}_{\lceil 0.5n \rceil}(x),$$

where $\mathrm{Thr}_k : \{0,1\}^n \to \{0,1\}$ is the k-threshold function that returns 1 on every input of Hamming weight at least k. Based on T, we generate the truth-table of h in time $O(N)$, invoke the algorithm $\mathcal{A}$ on the resulting truth table, and return its output $\mathcal{A}(h)$.

Analysis. Clearly, if T is chosen at random then h is a random $\lceil 0.5n \rceil$-slice function. By Hypothesis 5.3 and Lemma 5.4, $\mathcal{A}$ outputs "No" with constant probability. On the other, we will show that (*) if T is sampled from $D(1^n)$ then h can be realized with total share-size of $S(n)$ and therefore $\mathcal{A}$ outputs "Yes", and we get a distinguisher that breaks the security of the sPRF.

To prove (*), first observe that, by assumption, T can be computed by an $S'(n)$-size non-monotone formula f. By using De-Morgan law, we can assume that the negation gates are at the bottom layer. Next, by using a reduction of Berkowitz [Ber82], we can turn the formula to a monotone formula f' that agrees with f on all inputs of Hamming weight $k = \lceil 0.5n \rceil$. This is done by replacing each negation gate over a variable x_i by a threshold gate Thr_k over all other $n-1$ inputs $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$. This means that the monotone formula

$$f''(x) = (f'(x) \vee \mathrm{Thr}_{k+1}(x)) \wedge \mathrm{Thr}_k(x),$$

computes h. Finally, by using standard secret-sharing constructions for formulas over monotone gates, we get an LSS for f'' with complexity of $O(S'(n) \cdot n \log n)$. Indeed, f' is obtained by taking an S'-size AND/OR formula over $2n$ variables and substituting n of its leaves by threshold gates and so it can be realized with total share size of at most $S'(n) \cdot n \log n$ (use formula-based sharing and then re-share the shares bit-by-bit via Shamir's scheme over a binary extension field of size at least $n+1$), and f'' can be realized with an additional cost of $O(n \log n)$ bits. □

Remark 5.6 (generalization)**.** *A close inspection of the proof shows that the sPRF assumption can be replaced with the following average-case hardness of MFSP with respect to 1-sided algorithms. This assumption is parameterized by a polynomial $S(\cdot)$, and it asserts that there is no $\mathrm{poly}(N)$-time algorithm that given a truth table of size N outputs 1 if the function has a formula of size $S(\log(N))$ (i.e., polynomial in the number of variables), and outputs 0 with constant probability over a random truth table. This assumption seems weaker than the existence of PRFs. (See [San20] for a discussion on closely related assumptions).*

Next, we prove a similar theorem for linear schemes. In this case, we can prove unconditionally that most slice functions do not have small LSS, and so all we need is the sPRF assumption.

Theorem 5.7 (Hardness of gapLSS for truth tables)**.** *Under Assumption 5.2, there exists a polynomial $S(n)$ for which the following holds. Given a monotone function $f : \{0,1\}^n \to \{0,1\}$ represented as a truth table of size $N = 2^n$, no $\mathrm{poly}(N)$-time algorithm can distinguish between the following cases:*

- *(Yes) f has an LSS with total share size of $O(S(n)) \in \mathrm{polylog}(N)$.*
- *(No) Every LSS for f has total share size of $\tilde{\Omega}(2^{n/2}) = \tilde{\Omega}(\sqrt{N})$ bits.*

Proof. The proof is identical to the proof for Theorem 5.5, except that we replace "slices-are-expensive" hypothesis with an unconditional lower-bound. Specifically, by [ABF+19], all but $o(1)$-fraction of the $n/2$-slice functions cannot be realized by an LSS with total share size smaller than $\tilde{\Omega}(2^{n/2})$. □

5.1 Slices: Worst-Case to Average-Case Reduction

Notation. A (k,n)-slice function f can be naturally represented by a binary string z of length $\binom{n}{k}$. That is, z is indexed by k-weight strings $x \in \{0,1\}^n$ and its xth entry equals to $f(x)$. We will write f_z to denote the (k,n)-slice that is associated with z. When z is uniformly chosen string of length $\binom{n}{k}$, we get the uniform distribution over (k,n)-slices.

For a pair of equal-length strings a, b denote by $a \vee b$, $a \wedge b$ and $a \oplus b$, the bit-wise OR, AND and XOR of a and b. We write $\bar{a}$ to denote the complement of a. The following observation follows easily:

Observation 5.8. *For every pair of (k,n)-slice functions f_a and f_b it holds that*

$$f_{a \vee b} = f_a \vee f_b \quad \textit{and} \quad f_{a \wedge b} = f_a \wedge f_b.$$

Note that $f_{a \oplus b} \neq f_a \oplus f_b$ and $\neg f_a \neq f_{\bar{a}}$.

The reduction. We show that an arbitrary (k,n)-slice function f_z can be expressed as a constant size monotone formula over randomly chosen (k,n)-slices such that the marginal distribution of each function is uniform (but the joint distribution is correlated). We begin with the following simple claim.

Claim 5.9. *For every (k,n)-slice f_z and (k,n)-slice f_r, let $a = r \oplus z$, then*

$$f_z = f_{a \oplus r} = f_{(\bar{a} \wedge r) \vee (a \wedge \bar{r})} = (f_{\bar{a}} \wedge f_r) \vee (f_a \wedge f_{\bar{r}}).$$

Proof. The first equality holds since $a = r \oplus z$, the second one follows from the definition of XOR, and the last one follows from Observation 5.8. □

We sample r uniformly at random by using $\ell = \binom{n}{k}$ independent random bits. Since each of the strings $\bar{r}, a = r \oplus z$ and $\bar{a}$ is obtained from r by applying a fixed permutation over the set $\{0,1\}^\ell$, it holds that each of the random variables $r, \bar{r}, a = r \oplus z$ and $\bar{a}$ is (marginally) uniform. Hence, we have the following lemma.

Lemma 5.10. *Assume the existence of a (k,n)-slice function f_z whose total share-size is at least $T(n)$, then, with probability at least 0.25, a randomly chosen (k,n)-slice has total share size of at least $T(n)/4$.*

Proof. Fix f_z, and sample f_r and define $f_{\bar{a}}$, f_a and $f_{\bar{r}}$ as in the claim above. Let H (for high-cost) denote the set of (k,n)-slice functions that cannot be realized by a secret sharing scheme with total share size smaller than $T(n)/4$. Assume, towards a contradiction, that a random (k,n)-slice falls in H with probability p smaller than 0.25. By Claim 5.9 and standard closure properties of secret-sharing, it suffices to show that, with positive probability, the slice functions $f_r, f_{\bar{a}}, f_a, f_{\bar{r}}$ all fall out of H. Indeed, by a union bound, this happens with probability $1 - 4p > 0$. The lemma follows. □

To prove Lemma 5.4 it remains to prove the following simple "padding" claim.

Claim 5.11 (moving between slices)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a (k,n)-slice function that requires a total share size of $E(n)$ for some increasing function E. Then, for every constant $\alpha \in (0,1)$ there exists an integer $n' \in [n, n + \Theta(n)]$ and a $(\lceil \alpha n' \rceil, n')$-slice function g that requires a total share size of $E(\beta(n'))$ for $\beta = \min(\alpha, (1-\alpha)/2)$.*

Proof. We begin with the case where $k \geq \lceil \alpha n \rceil$. Choose $n' > n$ such that $k = \lceil \alpha n' \rceil$ and choose an arbitrary (k, n')-slice function $g : \{0,1\}^{n'} \to \{0,1\}$ that satisfies the following condition: For every assignment of the form $x \circ 0^{n'-n}$ where $x \in \{0,1\}^n$ of weight k, set $g(x \circ 0^{n'-n}) = f(x)$. (For example, g can take zeroes on all other k-weight assignments). It is not hard to see that any secret-sharing for g with total share size smaller than $E(\alpha n')$, implies a secret sharing for f with a total share size smaller than $E(\alpha n') < E(n)$, contradicting our assumption. (The inequality follows since $\alpha n' < k \leq n$).

We move on to the case where $k < \lceil \alpha n \rceil$. Let $\Delta = \left\lceil \frac{\alpha n - k}{1-\alpha} \right\rceil$, and take $k' = k + \Delta$, $n' = n + \Delta$. It can be verified that $k' = \lceil \alpha n' \rceil$. Choose an arbitrary (k', n')-slice g that satisfies the following requirement: For every n-bit string of weight k, it holds that $g(x \circ 1^{\Delta}) = f(x)$. Given a secret sharing scheme D for g we define a secret sharing scheme for f as follows. Fix the shares to the last Δ parties to some arbitrary values $v = (v_{n+1}, \ldots, v_{n+\Delta})$ in the support of D. To share a secret s, sample an s-sharing $(s_1, \ldots, s_{n'})$ according to D conditioned on $(s_{n+1}, \ldots, s_{n+\Delta}) = (v_{n+1}, \ldots, v_{n+\Delta})$, and give the first n shares to the n parties of f. (Since $\Delta < k'$ the last Δ parties form an unauthorized set of g and so the above distribution is well-defined.) Since the shares of the last Δ parties are fixed, we can think about them as available to all parties. This means that the view of any coalition x is exactly the view of a coalition $x \circ 1^{\Delta}$ in g. Privacy and correctness now follows by noting that for any x it holds that $g(x \circ 1^{\Delta}) = f(x)$. For strings of weight k this holds by design, and for strings of weight smaller than k (resp., larger than k) this follows since g is a (k', n')-slice.

We conclude that if g can be realized with total share size smaller than $E(n' \cdot (1-\alpha)/2) < E(n)$, then so is f, contradicting our assumption. (The inequality follows since $n' \leq 2\left(n + \frac{\alpha n - k}{1-\alpha}\right) \leq 2\left(\frac{n-k}{1-\alpha}\right) < \frac{2n}{1-\alpha}$.) □

6 GapSS for Graphs via Communication Complexity

In the previous sections we proved hardness results for GapSS in a few regimes. A regime that remains elusive is the one where we want to determine whether relatively "simple" access structures can be realized with shares of constant size vs. super-constant size. We devote this section to drawing connections between this problem and some well-studied two-party communication problems. We start by presenting necessary definitions in Section 6.1, and then discuss connections between the GapSS problem and different types of communication protocols in Section 6.2, Section 6.3, Section 6.4. Lower-bounds against myopic algorithms are given in Section 6.5.

6.1 Graph Access Structures and CDS Access Structures

A *graph access structure* is a monotone function f that has minterms of size exactly 2, and thus it can be represented by a graph G: Every party is assigned a vertex, and a pair of parties is authorized iff an edge connects their corresponding vertices. We will call an access structure a *bipartite access structure* if it can be represented by a bipartite graph. By abuse of notation, we sometimes identify the graph G with the corresponding function f_G. For example, we write $\mathsf{S}_{\mathsf{max}}(G)$ for $\mathsf{S}_{\mathsf{max}}(f_G)$. It is known that $\mathsf{S}_{\mathsf{max}}(G) = O(n/\log n)$ for every n-vertex graph G [EP97] and the best-known lower-bound is $\Omega(\log n)$ [vD95, Csi09]. For linear schemes, this can be improved to $\Omega(\sqrt{n})$ [BGP97].

We will also consider *CDS access structures*. These are partial access structures that are also defined according to bipartite graphs with two parts X, Y, but their correctness and privacy constraints only apply to coalitions $e = (x, y)$ such that $x \in X$ and $y \in Y$, and we put no constraints on larger coalitions. Secret sharing schemes for CDS access structures are also known as a 2-party *conditional disclosure of*

secrets protocols [GIKM00] and are closely related to secret sharing for forbidden-graphs [SS97]. We let $\mathsf{CDS}_{\mathsf{max}}(G)$ (resp., $\mathsf{LCDS}_{\mathsf{max}}$) denote the smallest integer k such that G can be realized by a CDS (resp., linear CDS over the binary field) with shares of size k. Since every secret sharing scheme that realizes a graph G is also a CDS for G it holds that

$$\mathsf{CDS}_{\mathsf{max}}(G) \leq \mathsf{S}_{\mathsf{max}}(G) \leq \mathsf{LS}_{\mathsf{max}}(G), \quad \text{and} \quad \mathsf{CDS}_{\mathsf{max}}(G) \leq \mathsf{LCDS}_{\mathsf{max}}(G) \leq \mathsf{LS}_{\mathsf{max}}(G),$$

for every bipartite graph G. In fact, CDS seem to be significantly cheaper than graphs. In particular, for every n-vertex graph G, it is known that $\mathsf{CDS}_{\mathsf{max}}(G) = n^{o(1)}$ [LVW18], and that $\mathsf{LCDS}_{\mathsf{max}}(G) = O(n^{1/2})$ [BFMP17]. The latter bound is known to be tight [BFMP17].

6.2 Secret Sharing and Nondeterministic 1-Equality Protocols

In this part, we describe the connection between secret sharing and Nondeterministic 1-Equality communication protocols. In such protocols, the parties are given a nondeterministic advice and then have 1-query access to an Equality oracle. The complexity of the protocol is measured by the length of the nondeterministic advice.

Definition 6.1 (Nondeterministic 1-Equality communication [BBM+20])**.** *A bipartite graph $G = ((X, Y), E)$ has a k-bit non-deterministic 1-Equality protocol if there exist a pair of functions $A : X \times \{0,1\}^k \to \{0,1\}^*$ and $B : Y \times \{0,1\}^k \to \{0,1\}^*$ for which the followings hold:*

- *If $(x, y) \in E$, then there exists an advice $a \in \{0,1\}^k$ such that $A(x, a) = B(y, a)$.*
- *If $(x, y) \notin E$, then for every $a \in \{0,1\}^k$ it holds that $A(x, a) \neq B(y, a)$.*

Let $\exists\mathsf{EQ}(G)$ denote the smallest integer k such that G has k-bit non-deterministic 1-Equality protocol.

We prove the following theorem:

Theorem 6.2 (LSS vs nondeterministic 1-Equality protocols)**.** *Every family of bipartite graphs G can be realized by a linear secret sharing scheme with shares of constant size if and only if G can be computed with constant communication by a protocol with 1-query access to an Equality oracle. In particular,*

$$\exists\mathsf{EQ}(G)/2 \leq \mathsf{LCDS}_{\mathsf{max}}(G) \leq \mathsf{LS}_{\mathsf{max}}(G) \leq 2^{\exists\mathsf{EQ}(G)}.$$

In order to prove the theorem we will use the following graph-theoretic characterization of nondeterministic 1-Equality protocols due to [BBM+20].

Fact 6.3 ([BBM+20])**.** *A bipartite graph G has a nondeterministic 1-Equality protocol with complexity t if and only if G can be covered by at most 2^t bipartite graphs that do not contain a 4-path as an* induced subgraph *(aka P4-free bipartite graphs).*

A bipartite graph is P4-free if and only if its edges form a vertex-disjoint set of bicliques. That is, each connected component of a P4-free bipartite graph is a complete bipartite graph. Accordingly, the minimal number of P4-free graphs needed to cover the edges of a graph G is a natural combinatorial quantity that refines the well-studied *biclique cover number* of a graph. Thus Theorem 6.2 essentially says that a bipartite graph has an LSS with constant share size iff its *P4-free cover number* is constant.

Proof of Theorem 6.2. The inequality $\mathsf{LCDS}_{\mathsf{max}}(G) \leq \mathsf{LS}_{\mathsf{max}}(G)$ follows by definition. We move on to prove the rightmost inequality. Let G be a bipartite graph over n vertices and let $k := 2^{\exists\mathsf{EQ}(G)}$. We prove that G can be realized by an LSS with maximal share size of k. By Fact 6.3, G can be covered by k P4-free bipartite graphs $G_1, \ldots, G_k$. Viewing each graph as an access structure (whose authorized sets are sets with an induced edge) it holds that $G = \bigvee G_i$, and so to share a secret s under G, it suffices to share s independently for every G_i. The latter task can be done with single-bit shares as follows. Recalling that the edges of G_i form a disjoint union of bicliques, i.e., $G_i = \bigvee B_{i,j}$, we independently share s according to each biclique $B_{i,j}$ by setting the left shares of $B_{i,j}$ to be a random bit $r_{i,j}$ and the right shares to be $r_{i,j} \oplus s$. Since the bicliques covering G_i are vertex-disjoint, each vertex of G_i receives a single bit.

Next, assume that the bipartite graph $G = ((L, R), E)$ can be realized by a Linear CDS D with share size k. By Fact 6.3, in order to prove that $\exists\mathsf{EQ}(G) \leq 2k$, it suffices to show that G can be covered by at most 2^{2k} bipartite graphs that are all P4-free. Denote by ρ the randomness complexity of D. By the standard correspondence between LSS and span programs [Bei96, KW93] (see also [Bei11]), the Linear-CDS D associates with each vertex v a binary $k \times (\rho + 1)$ binary matrix M_v such that (*) the pair $(a, b) \in L \times R$ is an edge if and only if the rows of M_a and M_b span some fixed non-zero target vector e_1 (wlog $e_1 = (1, 0^\rho)$). For every pair of row vectors $(\alpha, \beta) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$, let $G_{\alpha,\beta}$ be the bipartite graph over (L, R) such that $(a, b) \in L \times R$ is an edge if $\alpha M_a + \beta M_b = e_1$. By (*), the 2^{2k} graphs $G_{\alpha,\beta}, (\alpha, \beta) \in \mathbb{F}_2^k \times \mathbb{F}_2^k$ cover the graph G and so it remains to prove that all these graphs are P4-free. Fix some pair (α, β). We show that every set of 4 vertices that are connected by a path of length 3, induce a 4-cycle, and so the graph does not contain an induced 4-path. Fix some vertices a, b, c, d that form a path of length three (a, b, c, d) in the graph $G_{\alpha,\beta}$ and assume that $a, b \in L$ and $b, d \in R$ (the other case is symmetric). Then the edge (a, d) must also belong to $G_{\alpha,\beta}$ since

$$\alpha M_a + \beta M_d = (\alpha M_a + \beta M_b) - (\beta M_b + \alpha M_c) + (\alpha M_c + \beta M_d) = e_1 - e_1 + e_1 = e_1,$$

and the theorem follows. □

The theorem establishes a dimension-free relation between Non-deterministic 1-equality protocols (or the P4-free cover number of the bipartite graph), linear secret sharing, and linear-CDS schemes. The gap between these measures is exponential, and so for very "cheap" graphs we get tighter bounds. In the extreme, known results imply that a graph has 1-bit LSS if and only if it is a P4-free bipartite graph. (See Appendix B for details about this result and about other extensions of the theorem.)

A new lower-bound for disjointness. Let DISJ_N denote the bipartite graph with N left vertices and N right vertices where each vertex represents a subset of $[\lfloor \log N \rfloor]$ and where x and y are connected if the corresponding sets are disjoint. Theorem 6.2 allows us to settle the the linear-CDS complexity of DISJ_N. This quantity was studied by [LVW17] who showed that $\sqrt{\log N} \leq \mathsf{LCDS}_{\mathsf{max}}(\mathrm{DISJ}_N) \leq \log N$ leaving a quadratic gap between the lower and upper bound. In [Juk06, BBM+20] it was shown that $\exists\mathsf{EQ}(G) \geq 0.085 \log N$ and therefore, by Theorem 6.2, we conclude:

Corollary 6.4. $\mathsf{LCDS}_{\mathsf{max}}(\mathrm{DISJ}_N) = \Theta(\log N)$.

We further note that the Lin-CDS complexity of DISJ is of special interest as it is closely related to the complexity of fuzzy-IBE encryption (see [LVW17] for a discussion). We note that the (non-linear) CDS complexity of DISJ (that was suggested as an open problem in [AV21]) remains open.

6.3 Secret Sharing and Randomized Communication Complexity

Public-randomness protocols. A public-randomness protocol is one where the players have access to a shared source of randomness. Equivalently, we think of public-randomness protocols as a distribution over deterministic protocols. (See, e.g., for [KN97] communication complexity background.) Formally, a public-randomness BPP protocol P computes a bipartite graph $G = ((X, Y), E)$ if for every input pair x, y it holds that

$$\Pr_r[P(x, y; r) = G(x, y)] \geq 0.9,$$

where, by abuse of notation, $G(x, y) = 1$ if $(x, y) \in E$. If the protocol never errs on 1-inputs (i.e., edges) it is referred to as a coRP protocol, and if it never errs on 0-inputs (i.e., non-edges) it is referred to as an RP protocol. We let $\mathsf{BPP}(G)$ (resp., $\mathsf{coRP}(G)$, $\mathsf{RP}(G)$) denote the smallest integer k for which G admits a BPP (resp., coRP, RP) protocol in which the parties never communicate more than k bits.

The following theorem upper-bounds the coRP complexity of the graph in terms of its share-complexity. For the converse direction, we relate the RP complexity of the graph to its normalized share size. Formally, for an n-vertex graph G, we let $\mathsf{S}_{\mathsf{max,norm}}(G, \rho)$ denote the smallest integer k such that G can be realized by a secret sharing scheme with secret size ρ and normalized max-share size of k (i.e., max-share size of $k \cdot \rho$).

Theorem 6.5 (Secret Sharing and Randomized Communication Protocols)**.** *For every family G of bipartite graphs, it holds that*

$$\Omega(\log(\mathsf{coRP}(G))) \leq \mathsf{CDS}_{\mathsf{max}}(G) \leq \mathsf{S}_{\mathsf{max}}(G),$$

and (weak converse):

$$\mathsf{S}_{\mathsf{max,norm}}(G, \log n) \leq O\left(2^{\mathsf{RP}(G)}\right),$$

where n denotes the size of G.

Proof. Let $G = ((L, R), E)$ be a bipartite graph. We first prove that a CDS scheme for G with k-bit shares implies a randomized communication protocol for G with complexity of $2^{O(k)}$. We begin with the following basic protocol that is closely related to the protocol from [AV21, Thm. 7]. For every $b \in \{0, 1\}$, Alice and Bob use the public randomness of the protocol to invoke the CDS scheme and locally generate a vector of shares $(s_v^b)_{v \in L \cup R}$. Given an input $x \in L$ (resp., $y \in R$) Alice announces whether $s_x^0 = s_x^1$ (resp., Bob announces whether $s_y^0 = s_y^1$). If both equalities hold the parties reject; and otherwise they accept. This basic protocol is repeated $T = O(2^{2k})$ times and the parties reject if at least one of the copies rejects. Otherwise, the parties accept.

We analyze the basic protocol. Fix $(x, y) \in L \times R$. If the set $\{x, y\}$ is authorized (i.e., the edge (x, y) exists), the pair (s_x^0, s_y^0) and the pair (s_x^1, s_y^1) are distributed over disjoint supports, and therefore the parties accept with probability 1. On the other hand, if $\{x, y\}$ is an unauthorized set, the random variable (s_x^0, s_y^0) is identically distributed to the random variable (s_x^1, s_y^1), and since these random variables are supported over a set of size at most 2^{2k}, we get a collision (and the parties accept) with probability at least 2^{-2k}. After repeating the protocols $T = C2^{2k}$ times for a constant C, we get a one-sided error protocol with a constant error of $(1 - 2^{-2k})^T < e^{-C}$, as required.

We now prove the converse part. Suppose that there exists a randomized RP protocol P for G with communication complexity k such that for every edge (x, y) in G it holds that

$$\Pr_r[P(x, y; r) = 1] \geq 0.9,$$

and for every non-edge of G the protocol always outputs 0 with probability 1. We will need the following claim whose proof follows via a standard use of the probabilistic method.

Claim 6.6. *For $T = O(\log n)$, there exists a sequence of deterministic protocols $P_1, \ldots, P_T$ with communication complexity of k bits such that for every edge $(x, y) \in E$*

$$\Pr_{i \in [T]}[P_i(x, y) = 1] > 0.8,$$

and for every non-edge $(x, y) \in L \times R \setminus E$, $P_i(x, y) = 0$ for all $i \in [T]$.

Proof of claim. For a constant C, sample $T = C \log n$ independent protocols from P (viewed as a probability distribution over deterministic protocols). We show that such a tuple satisfies the claim with positive probability. By a Chernoff bound, for any fixed $(x, y) \in E$ the first equation holds except with probability $\exp(-T)$ which is at most $0.5/n^2$ for sufficiently large constant C. Hence, by applying a union-bound over all the edges, the sampled protocols satisfy the equality for each $(x, y) \in E$ with probability at least 0.5. Finally, the second equality holds (with probability 1) since P has zero-error on No instances. □

For each protocol P_i define a bipartite graph $G_i = (L, R, E_i)$ whose edges correspond to the one-inputs of P_i, i.e., $(x, y) \in E_i$ iff $P_i(x, y) = 1$. Let $\rho : \mathbb{F}_2^K \to \mathbb{F}_2^T$ be a linear error correcting code whose rate is constant, i.e., $K = \Theta(T)$ and whose distance is larger than $0.2T$. (Such a code exists, e.g., via the probabilistic method [Var57]). Given a K-bit secret s we map it to T bits $z = \rho(s)$ and then, for each $i \in [T]$ share the bit z_i to the parties $L \cup R$ by using a secret sharing for G_i. In Claim 6.7 below we will show that this can be done by a linear scheme with maximal-share size of at most 2^k bits. Hence, each party holds a share of bit-length at most $T \cdot 2^k$ bits, and since the secret is of bit-length $\Theta(T)$ the normalized share size is $O(2^k)$. It is not hard to see that the scheme realizes G. Indeed, correctness holds since every edge (x, y) in G appears in at least $0.8T$ of the graphs and so the pair (x, y) can recover at least $0.8T$ of the z-shares, and then uniquely recover s (by exploiting the large distance of the code). On the other hand, for privacy, note that any independent set S in G is also an independent set of $G_i, \forall i \in [T]$. Therefore the coalition S learns nothing on the codeword z, and the secret s remains perfectly hidden. (Formally, the view of such a coalition is distributed independently of the secret.)

It remains to show that G_i can be realized by secret-sharing scheme with max-share size of 2^k. Indeed, we prove the following claim.

Claim 6.7 (Deterministic protocols and secret sharing)**.** *If a bipartite graph H admits a k-bit deterministic protocol with communication of k, then it can be realized by an LSS (over an arbitrary field) with maximal share size of 2^k.*

Proof. It is a basic fact in communication complexity that every deterministic protocol P with communication k for a bipartite graph G induces a partition of the one-inputs (edges) to at most 2^k bicliques. We can thus share the secret s to each of the bicliques independently with share-size 1 by giving the left parties in the biclique a random field element r and the right parties $s - r$. Overall, each vertex gets a share of size at most 2^k. □

We conclude that $\mathsf{S}_{\mathsf{max},\mathsf{norm}}(G, C \log n)) \leq O\left(2^{\mathsf{RP}(G)}\right)$ for some constant C. Note that we can take $C = 1$, by (possibly) padding the secret and increasing the constant in the big-O notation, and so the second part of the theorem follows. □

From secret-sharing to adjacency sketches. Recall that a family of graphs $\mathcal{G}$ has k-size adjacency sketches if there exists a probabilistic labeling algorithm $L_G(v)$ that, given $G \in \mathcal{G}$ and vertex v, assigns a k-bit label ℓ_v to v such that a universal decoder algorithm D that has only access to labels but not to G can determine with probability $2/3$ if two vertices are adjacent. That is, for every $G \in \mathcal{G}$ and vertices u, v

$$\Pr[D(L_G(v), L_G(u)) = G(v,u)] \geq 2/3.$$

As mentioned in the introduction, graphs with a constant randomized communication complexity can be realized by constant-size sketches, and vice-versa [HWZ22]. By Theorem 6.5, graphs with constant share-size give rise to constant-size sketches. Below, we sketch a direct proof for this fact without going through communication complexity.

Let $\mathcal{G}$ be the family of graphs that can be realized by secret sharing schemes with max-share size of $k = O(1)$ bits. We construct a constant-size adjacency sketch for $\mathcal{G}$ with 1-sided error as follows. Independently share the secret zero and the secret one into $(s_v)_{v \in V}$ and $(s'_v)_{v \in V}$, and label each vertex v with its shares (s_v, s'_v). Given a pair of labels (s_v, s'_v) and (s_u, s'_u) declare an "edge" if $(s_v, s_u) \neq (s'_v, s'_u)$ and "no-edge" otherwise. The privacy and correctness of the secret sharing scheme imply that (s_v, s_u) and (s'_v, s'_u) are identically distributed (resp., have disjoint support) if v and u are not connected (resp., connected). In the former case, collision happens with constant probability of at least 2^{-k} since the shares are of length k. The error can be amplified by sampling $O(2^k)$ independent labels.

6.4 CDS Complexity and coNP Protocols

A co-nondeterministic protocol allows the party to determine the output based on an advice a whose length serves as the communication complexity of the protocol. Formally, a *k-bit co-nondeterministic protocol* for a bipartite graph $G = ((X,Y),E)$ is defined by a pair of functions $A : X \times \{0,1\}^k \to \{0,1\}$ and $B : Y \times \{0,1\}^k \to \{0,1\}$ that satisfy the following properties:

- If $(x,y) \notin E$, then there exists $a \in \{0,1\}^k$ such that $A(x,a) = B(y,a)) = 1$.
- If $(x,y) \in E$, then for every $a \in \{0,1\}^k$ it holds that $A(x,a) = 0$ or $B(y,a) = 0$.

Let $\mathsf{coNP}(G)$ denote the minimal integer k for which G can be realized by a k-bit co-nondeterministic protocol. We prove the following dimension-free bound.

Theorem 6.8 (coNP complexity and LCDS)**.** *For every family G of bipartite graphs it holds that*

$$\mathsf{LCDS}_{\mathsf{max}}(G) \leq 3 \cdot 2^{\mathsf{coNP}(G)}.$$

We do not know whether the theorem extends to the case of $\mathsf{LS}_{\mathsf{max}}(G)$, but we note that Theorem 6.2 implies that $\mathsf{LS}_{\mathsf{max}}(G) \leq 2^{\mathsf{NP}(G)}$.

Proof. Let $G = ((L,R),E)$ be a bipartite graph with n vertices in each side with coNP complexity k. Then, the bipartite non-edges of G can be covered by $K = 2^k$ independent sets $I_1, \ldots, I_K$. That is, each non-edge in any of the independent sets $I_1, \ldots, I_K$ is a non-edge in G and every non-edge of G is a non-edge in at least on I_j. For each $i \in [K]$, define a bipartite graph $G_i = (L, R, E_i)$ where two vertices are connected by an edge iff at least one of them is not in I_i. That is, for every $x \in L$ and $y \in R$, $(x,y) \in E_i$ iff $\{x,y\} \not\subset I_i$.

We can now use additive secret sharing to get a secret-sharing scheme for G based on secret-sharing schemes for the graphs G_i: Given a secret s generate K uniform bits $s_1, \ldots, s_K$ subject to $\bigoplus_{i=1}^{K} s_i = s$, and share each s_i according to the graph G_i. The scheme is correct since an edge in G must appear as an

edge in all G_j's (since every non-edge of G_j is a non-edge in G), and the scheme is private since a non-edge in G must appear as a non-edge in some G_j.

It remains to realize the bipartite graphs G_i. For this, observe that in every G_i the edges can be partitioned to three disjoint bicliques: (1) edges (x, y) where $x \in I_i$ and $y \notin I_i$; (2) edges (x, y) where $x \notin I_i$ and $y \in I_i$; and (3) edges (x, y) where $x \notin I_i$ and $y \notin I_i$. Since we can share a secret s' to a biclique by dealing every left party a random bit r and every right one the bit $r \oplus s'$, we can realize G_i with a maximal share size of 3. It follows that the maximal share size for G is $3 \cdot 2^k$, as required. □

The co-nondeterministic complexity is known to lower-bound CDS complexity as proved in [AV21]:

Theorem 6.9 ([AV21])**.** *For every family G of bipartite graphs it holds that*

$$\Omega(\mathsf{coNP}(G)) - O(\log\log(n)) \leq \mathsf{CDS}_{\mathsf{max}}(G),$$

where n is the size of the graph.

Together with Theorem 6.8, we get that for every n-vertex bipartite graph G,

$$\Omega(\mathsf{coNP}(G)) - O(\log\log(n)) \leq \mathsf{CDS}_{\mathsf{max}}(G) \leq \mathsf{LCDS}_{\mathsf{max}}(G) \leq 3 \cdot 2^{\mathsf{coNP}(G)}.$$

6.5 Failure of Myopic Algorithms

We prove that myopic algorithms cannot distinguish the case where $\mathsf{LS}_{\mathsf{max}}(G) = O(1)$ from the case where $\mathsf{S}_{\mathsf{max}}(G) = \Omega(\log n)$.

Theorem 6.10. *There exists a family of bipartite graphs G that requires shares of size $\Omega(\log n)$ from any secret sharing scheme that realizes it, but every induced subgraph of G with $\sqrt{n}$ vertices in each side can be realized by a linear scheme with maximal share size 8.*

The proof is based on the probabilistic method. We denote by $M_{n,p}$ the distribution over $n \times n$ Boolean matrices where each entry is independently 1 with probability $p = p(n)$ and 0 with probability $\overline{p} = 1 - p$, and by $BG(n, p)$ the distribution over bipartite graphs that corresponds to $M_{n,p}$. In our proof, we will use the following fact regarding covers in random matrices, by which matrices sampled from $M_{n,p}$ with p that is not "too large" do not have "huge" combinatorial rectangles of 1s:

Fact 6.11 ([PT17], Corollary 4.3)**.** *There exists a constant $\lambda_0 > 0$ such that for every parameter $\lambda_0 < \lambda(n) < o(n)$ and $p(n) = 1 - \lambda(n)/n$ the following holds. With probability $1 - o(1)$, the largest combinatorial rectangle of 1s in $M_{n,p(n)}$ is of size $\frac{n^2}{e\lambda} + O(n)$.*

Proof of Theorem 6.10. We show that when a bipartite graph G is sampled from $BG(n, p)$ with $p = n^{-0.99}$ it satisfies both the upper and lower bounds required in the theorem. To prove the lower bound we first notice that Fact 6.11 implies the following lemma:

Lemma 6.12. *Let $0 < \alpha < 1$ be a constant and let G be a bipartite graph sampled from $BG(n, p)$ with $p = \frac{n^\alpha}{n}$. Then, with probability $1 - o(1)$ it holds that $\mathsf{coNP}(G) = \Omega(\log n)$.*

Proof. By Fact 6.11, the largest 1-monochromatic rectangle in a bipartite graph sampled from $BG(n, 1-p)$ is of size $O(n^{2-\alpha})$ with probability $1 - o(1)$. Switching the roles of zeroes and ones, this means that the largest 0-monochromatic rectangle in a bipartite graph sampled from $BG(n, p)$ is of size $O(n^{2-\alpha})$ with probability $1 - o(1)$. By a Chernoff bound, such a graph has at least $\Omega(n^2)$ with probability $1 - o(1)$.

For graphs that satisfy both properties, the number of zero rectangles that needed to cover all the zero entries is at least $\Omega(n^2/n^{2-\alpha}) = \Omega(n^\alpha)$ and so the non-deterministic communication complexity is at least $\alpha \log n - O(1)$, as required. □

Now, by setting $\alpha = 0.01$ and combining Lemma 6.12 and the inequality $\mathsf{CDS}_{\mathsf{max}}(G) \geq \Omega(\mathsf{coNP}(G)) - O(\log\log(n))$ from Theorem 6.9 we get that with $1 - o(1)$ probability, the CDS complexity of a graph that is sampled from $BG(n,p)$ with $p = n^{-0.99}$ is $\Omega(\log n)$.

We move on to prove that if we sample a matrix M from $M_{n,n^{-0.99}}$, then with high probability every $a \times b$ submatrix of M with $a, b \leq \sqrt{n}$ corresponds to a graph that can be realized with shares of size 8. Call a matrix M *good* if for every $a, b \leq \sqrt{n}$ every $a \times b$ submatrix of M contains a row or a column with at most four 1s. We prove the following lemma.

Lemma 6.13. *A matrix M sampled from $M_{n,n^{-0.99}}$ is good with probability $1 - o(1)$*

Proof. This statement is trivial when $min(a,b) \leq 4$ and so we fix $a, b > 4$. Since $b \leq \sqrt{n}$, the probability that a single row in the matrix has at least five 1s is bounded by

$$\binom{\sqrt{n}}{5} \cdot \left(\frac{1}{n^{0.99}}\right)^5 \leq \frac{1}{n^{2.45}}$$

Hence, if $a \geq b$, the probability that there is an $a \times b$ submatrix such that each of its a rows contains at least five 1's is bounded by

$$\binom{n}{a} \cdot \binom{n}{b} \cdot \left(\frac{1}{n^{2.45}}\right)^a \leq n^{2a} \cdot \frac{1}{n^{2.45a}} = o\left(\frac{1}{n}\right),$$

where the last equality holds since $a > 4$. Similarly, if $a < b$, the probability that there exists an $a \times b$ submatrix where each of its b columns contains at least five 1's is bounded by $o\left(\frac{1}{n}\right)$. Thus, by a union bound over the n choices of $a, b \leq \sqrt{n}$, the probability that there are $a, b \in [\sqrt{n}]$ and an $a \times b$ submatrix where every column or row contains at least five 1's is bounded by $o(1)$. □

Fix a good matrix M and let M' be some $\sqrt{n} \times \sqrt{n}$-submatrix of M. It remains to show that M' can be realized by a secret sharing scheme with max-share size of 8. Observe that every submatrix M'' of M' contains a row or a column with at most four 1's. By an argument of [HHH22], it follows that M' corresponds to the biadjacency matrix of a disjoint union of four bipartite graphs F_1, F_2, F_3, F_4 that are all forests. We need the following simple claim.

Claim 6.14. *A forest F can be realized with max-share size of 2.*

Proof. It suffices to prove the claim for a tree T and apply the tree secret sharing independently for each connected component of F. Indeed, given a tree graph with n vertices choose one of them to be the root, and sample n uniform bits $r_1, \ldots r_n$. Scan the tree, starting from the root, set the root's share to be r_1, and set the share of the ith vertex to be $(r_i, r_j \oplus s)$ where j is the father of the ith vertex. Correctness and privacy are immediate. □

We conclude that F_i can be realized with max-share size of 2 and therefore the graph G' associated with M' can be realized with max-share size of at most 8. The theorem follows. □

Acknowledgment. We thank Amos Beimel for his comments and for pointing out an error in an earlier version of this manuscript.

References

[AA18] Benny Applebaum and Barak Arkis. On the power of amortization in secret sharing: d-uniform secret sharing and CDS with constant information rate. In Amos Beimel and S. Dziembowski, editors, *TCC 2018*, volume 11239 of *LNCS*, pages 317–344. Springer-Verlag, 2018.

[ABF+19] Benny Applebaum, Amos Beimel, Oriol Farràs, Oded Nir, and Naty Peter. Secret-sharing schemes for general and uniform access structures. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 441–471. Springer, 2019.

[ABI+23a] Damiano Abram, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Varun Narayanan. Cryptography from planted graphs: security with logarithmic-size messages. In *Theory of Cryptography Conference*, pages 286–315. Springer, 2023.

[ABI+23b] Benny Applebaum, Amos Beimel, Yuval Ishai, Eyal Kushilevitz, Tianren Liu, and Vinod Vaikuntanathan. Succinct computational secret sharing. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1553–1566. ACM, 2023.

[ABN+22] Benny Applebaum, Amos Beimel, Oded Nir, Naty Peter, and Toniann Pitassi. Secret sharing, slice formulas, and monotone real circuits. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[ABNP20] Benny Applebaum, Amos Beimel, Oded Nir, and Naty Peter. Better secret sharing via robust conditional disclosure of secrets. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 280–293. ACM, 2020.

[AN21] Benny Applebaum and Oded Nir. Upslices, downslices, and secret-sharing with complexity of 1.5n. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Part III*, volume 12827 of *Lecture Notes in Computer Science*, pages 627–655. Springer, 2021.

[AN23] Benny Applebaum and Oded Nir. Advisor-verifier-prover games and the hardness of information theoretic cryptography. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 539–555, 2023.

[AV21] Benny Applebaum and Prashant Nalini Vasudevan. Placing conditional disclosure of secrets in the communication complexity universe. *J. Cryptol.*, 34(2):11, 2021.

[AY24] Daniel Avraham and Amir Yehudayoff. On blocky ranks of matrices. *Comput. Complex.*, 33(1):2, 2024.

[BBM+20] Alexander R Block, Simina Brânzei, Hemanta K Maji, Himanshi Mehta, Tamalika Mukherjee, and Hai H Nguyen. p_4-free partition and cover numbers and application. *Cryptology ePrint Archive*, 2020.

[Bei96] Amos Beimel. *Secure Schemes for Secret Sharing and Key Distribution*. PhD thesis, Technion, 1996.

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011.

[Bei23] Amos Beimel. Lower bounds for secret-sharing schemes for k-hypergraphs. In Kai-Min Chung, editor, *4th Conference on Information-Theoretic Cryptography, ITC 2023, June 6-8, 2023, Aarhus University, Aarhus, Denmark*, volume 267 of *LIPIcs*, pages 16:1–16:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[Ber82] S Berkowitz. On some relationships between monotone and nonmonotone circuit complexity. Technical report, Technical report, Department of Computer Science, University of Toronto . . . , 1982.

[BF20] Amos Beimel and Oriol Farràs. The share size of secret-sharing schemes for almost all access structures and graphs. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020*, volume 12552 of *Lecture Notes in Computer Science*, pages 499–529. Springer, 2020.

[BFLN24] Amos Beimel, Oriol Farràs, Or Lasri, and Oded Nir. Secret-sharing schemes for high slices. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography - 22nd International Conference, TCC 2024, Milan, Italy, December 2-6, 2024, Proceedings, Part IV*, volume 15367 of *Lecture Notes in Computer Science*, pages 581–613. Springer, 2024.

[BFM16] Amos Beimel, Oriol Farràs, and Yuval Mintz. Secret-sharing schemes for very dense graphs. *J. Cryptol.*, 29(2):336–362, 2016.

[BFMP17] Amos Beimel, Oriol Farràs, Yuval Mintz, and Naty Peter. Linear secret-sharing schemes for forbidden graph access structures. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, volume 10678 of *Lecture Notes in Computer Science*, pages 394–423. Springer, 2017.

[BGP97] Amos Beimel, Anna Gál, and Mike Paterson. Lower bounds for monotone span programs. *Comput. Complex.*, 6(1):29–45, 1997.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM, 1988.

[BI92] Michael Bertilsson and Ingemar Ingemarsson. A construction of practical secret sharing schemes using linear block codes. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques*, volume 718 of *Lecture Notes in Computer Science*, pages 67–79. Springer, 1992.

[BKN18] Amos Beimel, Eyal Kushilevitz, and Pnina Nissim. The complexity of multiparty PSM protocols and related models. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 287–318. Springer, 2018.

[BL88] Josh Cohen Benaloh and Jerry Leichter. Generalized secret sharing and monotone functions. In Shafi Goldwasser, editor, *Advances in Cryptology – CRYPTO '88, 8th Annual International Cryptology Conference*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35. Springer, 1988.

[Bla79] George R. Blakley. Safeguarding cryptographic keys. In Richard E. Merwin, Jacqueline T. Zanca, and Merlin Smith, editors, *Proceedings of the 1979 AFIPS National Computer Conference*, volume 48 of *AFIPS Conference proceedings*, pages 313–317. AFIPS Press, 1979.

[Bog23] Andrej Bogdanov. Csirmaz's duality conjecture and threshold secret sharing. In *4th Conference on Information-Theoretic Cryptography, ITC 2023, June 6-8, 2023, Aarhus University, Aarhus, Denmark*, pages 3:1–3:6, 2023.

[BP18] Amos Beimel and Naty Peter. Optimal linear multiparty conditional disclosure of secrets protocols. In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 332–362. Springer, 2018.

[BR17] Andrej Bogdanov and Alon Rosen. Pseudorandom functions: Three decades later. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 79–158. Springer International Publishing, 2017.

[BSGV92] Carlo Blundo, Alfredo De Santis, Luisa Gargano, and Ugo Vaccaro. On the information rate of secret sharing schemes (extended abstract). In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference*, pages 148–167, 1992.

[BSSV95] Carlo Blundo, Alfredo De Santis, Douglas R. Stinson, and Ugo Vaccaro. Graph decompositions and secret sharing schemes. *J. Cryptol.*, 8(1):39–64, 1995.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 11–19. ACM, 1988.

[Csi96] László Csirmaz. The dealer's random bits in perfect secret sharing schemes. *Studia Sci. Math. Hungar.*, 32(3–4):429–437, 1996.

[Csi05] László Csirmaz. Secret sharing schemes on graphs. *IACR Cryptol. ePrint Arch.*, page 59, 2005.

[Csi09] László Csirmaz. An impossibility result on graph secret sharing. *Des. Codes Cryptogr.*, 53(3):195–209, 2009.

[CT13] László Csirmaz and Gábor Tardos. Optimal information rate of secret sharing schemes on trees. *IEEE Trans. Inf. Theory*, 59(4):2527–2530, 2013.

[DF91] Yvo Desmedt and Yair Frankel. Shared generation of authenticators and signatures (extended abstract). In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91, 11th Annual International Cryptology Conference*, volume 576 of *Lecture Notes in Computer Science*, pages 457–469. Springer, 1991.

[EP97] Paul Erdös and László Pyber. Covering a graph by complete bipartite graphs. *Discret. Math.*, 170(1-3):249–251, 1997.

[FHHH24] Yuting Fang, Lianna Hambardzumyan, Nathaniel Harms, and Pooya Hatami. No complete problem for constant-cost randomized communication. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1287–1298, 2024.

[FK09] Pierre Fraigniaud and Amos Korman. On randomized representations of graphs using short labels. In Friedhelm Meyer auf der Heide and Michael A. Bender, editors, *SPAA 2009: Proceedings of the 21st Annual ACM Symposium on Parallelism in Algorithms and Architectures, Calgary, Alberta, Canada, August 11-13, 2009*, pages 131–137. ACM, 2009.

[FLN01] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. In Peter Buneman, editor, *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 21-23, 2001, Santa Barbara, California, USA*. ACM, 2001.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GIKM00] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting data privacy in private information retrieval schemes. *J. Comput. Syst. Sci.*, 60(3):592–629, 2000.

[GKW15] Romain Gay, Iordanis Kerenidis, and Hoeteck Wee. Communication complexity of conditional disclosure of secrets and attribute-based encryption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, volume 9216 of *Lecture Notes in Computer Science*, pages 485–502. Springer, 2015.

[Gol98] Jovan Dj. Golic. On matroid characterization of ideal secret sharing schemes. *J. Cryptol.*, 11(2):75–86, 1998.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006*, pages 89–98. ACM, 2006.

[Har20] Nathaniel Harms. Universal communication, universal graphs, and graph labeling. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 33:1–33:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[HH22] Hamed Hatami and Pooya Hatami. The implicit graph conjecture is false. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1134–1137. IEEE, 2022.

[HH24] Hamed Hatami and Pooya Hatami. Guest column: Structure in communication complexity and constant-cost complexity classes. *SIGACT News*, 55(1):67–93, 2024.

[HHH21] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Electron. Colloquium Comput. Complex.*, TR21-066, 2021.

[HHH22] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. A counter-example to the probabilistic universal graph conjecture via randomized communication complexity. *Discret. Appl. Math.*, 322:117–122, 2022.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[HWZ22] Nathaniel Harms, Sebastian Wild, and Viktor Zamaraev. Randomized communication and implicit graph representations. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1220–1233. ACM, 2022.

[IP99] Russell Impagliazzo and Ramamohan Paturi. Complexity of k-sat. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, pages 237–240, 1999.

[ISN87] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing schemes realizing general access structure. In *Globecom 87*, pages 99–102. IEEE, 1987. Journal version: Multiple assignment scheme for sharing secret. *J. Cryptol.*, 6(1):15-20, 1993.

[Juk06] Stasys Jukna. On graph complexity. *Comb. Probab. Comput.*, 15(6):855–876, 2006.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

[KNR92] Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM J. Discret. Math.*, 5(4):596–603, 1992.

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

[LS20] Kasper Green Larsen and Mark Simkin. Secret sharing lower bound: Either reconstruction is hard or shares are long. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks - 12th International Conference, SCN 2020, Amalfi, Italy, September 14-16, 2020, Proceedings*, volume 12238 of *Lecture Notes in Computer Science*, pages 566–578. Springer, 2020.

[LV18] Tianren Liu and Vinod Vaikuntanathan. Breaking the circuit-size barrier in secret sharing. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th*

Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, pages 699–708. ACM, 2018.

[LVW17] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 758–790. Springer, 2017.

[LVW18] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018 – 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Part I*, volume 10820 of *Lecture Notes in Computer Science*, pages 567–596. Springer, 2018.

[NW96] Moni Naor and Avishai Wool. Access control and signatures via quorum secret sharing. In Li Gong and Jacques Stearn, editors, *CCS '96, Proceedings of the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, March 14-16, 1996*, pages 157–168. ACM, 1996.

[PR94] Pavel Pudlák and Vojtech Rödl. Some combinatorial-algebraic problems from complexity theory. *Discret. Math.*, 136(1-3):253–279, 1994.

[PR18] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1207–1219. ACM, 2018.

[PSS23] Toniann Pitassi, Morgan Shirley, and Adi Shraibman. The strength of equality oracles in communication. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 89:1–89:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[PT17] Mozhgan Pourmoradnasseri and Dirk Oliver Theis. The rectangle covering number of random boolean matrices. *Electron. J. Comb.*, 24(2):2, 2017.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[San20] Rahul Santhanam. Pseudorandomness and the minimum circuit size problem. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*, volume 151 of *LIPIcs*, pages 68:1–68:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

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

[Sim23] Meta-complexity workshop at simons institute for the theory of computing. `https://simons.berkeley.edu/programs/Meta-Complexity2023`, 2023.

[SS97] Hung-Min Sun and Shiuh-Pyng Shieh. Secret sharing in graph-based prohibited structures. In *Proceedings IEEE INFOCOM '97, The Conference on Computer Communications, Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution, Kobe, Japan, April 7-12, 1997*, pages 718–724, 1997.

[SSR08] Bhavani Shankar, Kannan Srinathan, and C. Pandu Rangan. Alternative protocols for generalized oblivious transfer. In Shrisha Rao, Mainak Chatterjee, Prasad Jayanti, C. Siva Ram Murthy, and Sanjoy Kumar Saha, editors, *Distributed Computing and Networking, 9th International Conference, ICDCN 2008*, volume 4904 of *Lecture Notes in Computer Science*, pages 304–309. Springer, 2008.

[Sti94] Douglas R. Stinson. Decomposition constructions for secret-sharing schemes. *IEEE Trans. Inf. Theory*, 40(1):118–125, 1994.

[Tas11] Tamir Tassa. Generalized oblivious transfer by secret sharing. *Des. Codes Cryptogr.*, 58(1):11–21, 2011.

[Var57] Rom R. Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Akad. Nauk SSSR*, 117:739—-741, 1957.

[vD95] Marten van Dijk. On the information rate of perfect secret sharing schemes. *Des. Codes Cryptogr.*, 6(2):143–169, 1995.

[Wat11] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography*, volume 6571 of *Lecture Notes in Computer Science*, pages 53–70. Springer, 2011.

[Xie24] Daniel Yu-Long Xie. *Breaking Graphs into Independent Rectangles*. PhD thesis, Purdue University Graduate School, 2024.

A Algorithms for Recognizing Cheap Truth Tables

We present two simple algorithms for the MinSS problem that given a monotone function $f : \{0,1\}^n \to \{0,1\}$ and a complexity bound S (either on the total share size or on the max-share size) determines whether f can be realized with complexity at most S.

We begin with a simple *non-deterministic* algorithm for the case where f is given as a truth table and where S upper-bounds the total share size. Naively, the witness consists of a full description of a pair of probability distributions over S bits, and so, on the face of it, the length of the witness may be long (e.g., if the distribution assigns weights of very small magnitude). However, we note that it suffices to describe only the support of the distributions leading to a non-deterministic algorithm of complexity $2^{O(S)}$.[12]

Theorem A.1 (Algorithm for MinSS)**.** *Given a truth-table of a monotone function $f : \{0,1\}^n \to \{0,1\}$ of size $N = 2^n$ and a bound $S > n$ on the total share-size, there is an algorithm that determines if f has a secret-sharing of total share size of S in non-deterministic time of $2^{O(S)}$ where the witness size is $2^{S+1} + n \log S < 2^{S+1} + S \log S$.*

[12]The argument also shows that one can always find a distribution whose weights can be succinctly described - see Remark A.2.

When $S = O(n)$ we get that the problem is solvable in *non-deterministic* time of $\mathrm{poly}(N)$, i.e., in NP. Also, by trying all witnesses we get a *deterministic* algorithm whose complexity is $\tilde{O}(2^{2^S})$. Unfortunately, this is at least exponential in N and, may even be double-exponential in N (say, $\tilde{O}(2^{2^{N^{0.5}}})$ if one wants to check if f has secret sharing of complexity of $N^{0.5}$.)

Proof. We assume that the function is non-trivial, i.e., $f \neq 0$ (otherwise we can simply accept the input). We begin with some notation. Suppose that f can be realized by a secret-sharing scheme $\mathcal{D}$ whose total share size is at most S where the ith share is of length S_i. By definition, $\sum_i S_i \leq S$ and by using padding we can assume that equality holds. Recall that for a secret $b \in \{0,1\}$, the distribution $\mathcal{D}(b)$ samples a vector of shares $z = (z[1], \ldots, z[n])$ where $z[i] \in \{0,1\}^{S_i}$. It will be convenient to pack z into an S-bit string via concatenation. Conversely, given an S-bit string z in the support of $\mathcal{D}(b)$, we write $z[i]$ to denote the ith share of z, i.e., the restriction of z to the indices $(1 + S_1 + \cdots + S_{i-1}), \ldots, (S_1 + \cdots + S_i)$ where $S_0 = 0$. For a set $I \subset [n]$, we let $z[I]$ denote $(z[i])_{i \in I}$.

Given $\mathcal{D}$, our witness w consists of the following parts: (1) the sequence of indices $S_1, \ldots, S_n \in [S]$ and (2) A pair of disjoint sets $L^0, L^1 \subset \{0,1\}^S$ where the set L^b contains all the strings in the support of $\mathcal{D}(b)$.[13] The total length of the witness is at most $n \log S + 2 \cdot 2^S$.

To verify the validity of a witness w, we first verify that $\sum S_i = S$ and that the sets L^0 and L^1 are disjoint. Next, we verify that one can define a probability distribution $\mathcal{D}(b)$ over L^b such that both privacy and correctness constraints hold. This is done as follows.

Correctness: For every minimal authorized set $I \subseteq [n]$, and every pair of strings $z_0 \in L^0$ and $z_1 \in L^1$, we verify that $z_0[I] \neq z_1[I]$, and reject the witness if the condition does not hold. It is not hard to verify that if $\mathcal{D}$ is a secret-sharing scheme that realizes f, then the witness passes this test. In the other direction, if L^0 and L^1 pass the correctness condition then any pair of distributions defined over L^0 and L^1 resp., meets the correctness requirements induced by f.

Privacy: We define for each string $z \in L := L^0 \cup L^1$ a formal variable p_z, and check whether there exists a pair of probability vectors $(p_z)_{z \in L^0}$ and $(p_z)_{z \in L^1}$ that satisfies the privacy constraints. That is, we check if there exists a *non-negative* real-valued assignment for $(p_z)_{z \in L}$ that satisfies the following system of constraints: For every maximal unauthorized set $I \subset [n]$ and every string $w \in \{z[I] : z \in L\}$

$$\sum_{z \in L^0 : z[I] = w} p_z = \sum_{z \in L^1 : z[I] = w} p_z.$$

In addition, we add the constraints

$$\sum_{z \in L^0} p_z = 1, \quad \text{and} \quad \sum_{z \in L^1} p_z = 1.$$

Since this is a linear system, one can use linear programming to check the existence of a positive solution in time polynomial in the system size. Noting that we have 2^S variables and at most 2^{n+S} equations, the verification takes $\mathrm{poly}(2^{n+S}) = 2^{O(S)}$ time. It is not hard to verify that the test passes if and only if there exists a pair of distributions over L^0 and L^1 resp., that realize the privacy requirements induced by f. □

Remark A.2 (On the magnitude of the distribution weights)**.** *The proof of Theorem A.1 can be used to show that if f can be realized with total share size of S, then it can be realized with "sharing distributions"*

[13] Note that $\mathcal{D}(0)$ and $\mathcal{D}(1)$ have indeed disjoint supports since the coalition that contains all the parties is authorized (as $f \neq 0$) and so if the sets are not disjoint correctness is violated. In fact, it suffices to keep only one of these sets (say $\mathcal{D}^0$) as part of the witness but we avoid this optimization for simplicity.

$\mathcal{D}(0), \mathcal{D}(1)$ *over* S*-bits whose description is not too large. Specifically, since the linear system of inequalities constructed in the proof has constant-size coefficients (plus/minus 1), the system must have a rational solution. Furthermore, the numerator and denominator in these solutions are of size that is polynomial in the number of variables and constraints in the linear program and so* $\mathcal{D}(0)$ *and* $\mathcal{D}(1)$ *can be represented as a distribution vector of length* 2^S *that each of its entries (weights) can be represented by* $\mathrm{poly}(2^S) = 2^{O(S)}$ *bits.*

We note that a closely-related observation regarding the rationality of the distribution weights was made by Bogdanov in [Bog23].

Even when we deal with the simplest case where the maximal share size $\mathsf{S}_{\mathsf{max}}$ is constant, the above theorem yields a deterministic algorithm whose complexity is at least exponential in the size of the truth table $N = 2^n$. We show that this can be improved for the case of (monotone) k-DNF, i.e., a DNF that each of its conjunctions contains at most k variables. Specifically, for the case of constant k we get a deterministic algorithm with complexity of $2^{\mathrm{polylog}(N)}$ improving the $2^{\Omega(N)}$ of the previous theorem. We note that the case of $k = O(1)$-DNF captures well-studied classes of functions such as graph access structures, forbidden graph access structures, and k-hypergraph access structures. Extending the algorithm to larger classes (e.g., polynomial size DNF with large terms) remains an interesting open question.

Theorem A.3 (MinSS algorithm for k-DNFs)**.** *Given a* k*-DNF with at most* M *terms computing a monotone function* $f : \{0,1\}^n \to \{0,1\}$ *and a bound* $\mathsf{S}_{\mathsf{max}}$ *on the maximal share-size, there is an algorithm that determines if* f *has a secret-sharing of maximal share size of* $\mathsf{S}_{\mathsf{max}}$ *in non-deterministic time of* $M2^{O(n\mathsf{S}_{\mathsf{max}})}$ *where the witness size is* $M2^{k\mathsf{S}_{\mathsf{max}}+1}$.

As a result, we get a deterministic algorithm with complexity that is exponential in $O(M2^{k\mathsf{S}_{\mathsf{max}}} + n\mathsf{S}_{\mathsf{max}})$. Focusing on the case where $\mathsf{S}_{\mathsf{max}} = O(1)$ and $k = O(1)$, (and so $M \leq \binom{n}{k} = \mathrm{poly}(n)$), we get a deterministic algorithm whose complexity is exponential in $\mathrm{poly}(n)$. More generally, since Theorem A.1 yields a deterministic algorithm whose complexity is exponential in $2^{n\mathsf{S}_{\mathsf{max}}}$, the current theorem provides an improvement whenever $M2^{k\mathsf{S}_{\mathsf{max}}} \ll 2^{n\mathsf{S}_{\mathsf{max}}}$. (I.e., even in the case where, say, $k = 0.9n$, $M \leq 2^n$ and $\mathsf{S}_{\mathsf{max}} > 10$.)

Proof. Let $I_1, \ldots, I_M$ denote the terms of f, and so we can represent each I_i by a subset of $[n]$ of size at most k. Given a secret-sharing scheme $\mathcal{D}$ that realizes f with max-share size of $\mathsf{S}_{\mathsf{max}}$, we may assume, via padding, that each share is of size exactly $\mathsf{S}_{\mathsf{max}}$ and so $\mathcal{D}(b)$ is a distribution over $(\{0,1\}^{\mathsf{S}_{\mathsf{max}}})^n$. Accordingly, for a string $z = (z[1], \ldots, z[n]) \in (\{0,1\}^{\mathsf{S}_{\mathsf{max}}})^n$ and a set $I \subset [n]$ we let $z[I] = (z[i])_{i \in I}$. As in the proof of Theorem A.1, our witness is essentially some representation of the supports of $\mathcal{D}(0)$ and $\mathcal{D}(1)$. The main difference is that here, we can get a *succinct* representation of these sets. Specifically, for each term $I_i, i \in [M]$ and bit $b \in \{0,1\}$, we define the list L_i^b to be

$$L_i^b := \{z[I_i] \quad | \quad z \in \mathrm{supp}(\mathcal{D}(b))\},$$

and define the lists $(L_1^0, \ldots, L_M^0, L_1^1, \ldots, L_M^1)$ as our witness. Since each list is a subset of $\{0,1\}^{k\mathsf{S}_{\mathsf{max}}}$ it can be represented by $2^{k\mathsf{S}_{\mathsf{max}}}$ bits (using standard characteristic vector representation) and the total length of the witness is $2M \cdot 2^{k \cdot \mathsf{S}_{\mathsf{max}}}$.

Correctness: Given such a witness we first verify that for every $i \in [M]$ the lists L_i^0 and L_i^1 are disjoint. This can be done in time $O(M2^{k\mathsf{S}_{\mathsf{max}}})$. Next, we reconstruct "global" sets L^0 and L^1 that are consistent with the witness while satisfying the correctness constraints. For this, we initialize empty lists L^0 and L^1 and put a string $z \in (\{0,1\}^{\mathsf{S}_{\mathsf{max}}})^n$ in L^b if for every $i \in [M]$ it holds that $z[I_i]$ is in L_i^b. (Some strings may not be added to any list.) Note that (1) L^1 and L^0 are disjoint; (2) Any pair of distributions over L^1 and L^0 satisfy

the correctness constraints; and (3) if the witness corresponds to a valid secret sharing $\mathcal{D} = (\mathcal{D}(0), \mathcal{D}(1))$ then $\mathrm{supp}(\mathcal{D}(b)) \subset L^b$ for every $b \in \{0,1\}$. The total time complexity of this step is $O(M2^{n\mathsf{S}_{\mathsf{max}}})$.

Privacy: Next, we verify that there exists a pair of probability distributions over L^1 and L^0 that satisfy the privacy requirements. This is done just like in the proof of Theorem A.1 by solving a linear program over at most $2^{n\mathsf{S}_{\mathsf{max}}}$ variables (one for each string in $L^0 \cup L^1$) with at most $2^{n+n\mathsf{S}_{\mathsf{max}}} + 2$ constraints (one for each maximal unauthorized set T and for each string in $\{0,1\}^{|T|\mathsf{S}_{\mathsf{max}}}$ and two additional constraints). This can be done with complexity of $2^{O(n\mathsf{S}_{\mathsf{max}})}$. □

We move on to describe an algorithm for MinLSS, that is based on the equivalence between linear secret sharing and *monotone span programs* (from here on abbreviated as MSPs), a computational model introduced in [KW93]. We define MSPs, state the equivalence between them and LSS and an additional useful fact, and then follow with the description of the algorithm for MinLSS.

Definition A.4 (Monotone Span Program [KW93])**.** *A monotone span program is a triplet $\mathcal{M} = (\mathbb{F}, M, \rho)$, where $\mathbb{F}$ is a field, M is an $a \times b$ matrix over $\mathbb{F}$, and $\rho : \{1, \ldots, a\} \to [n]$ labels each row of M by an index $i \in [n]$. The size of $\mathcal{M}$ is the number of rows of M (i.e., a). For any input $x \in \{0,1\}^n$, let M_x denote the sub-matrix obtained by restricting M to the rows labeled by indices i such that $x_i = 1$. We say that $\mathcal{M}$ accepts x if the rows of M_x span the vector $e_1 = (1, 0, \ldots, 0)$. We say that $\mathcal{M}$ computes a monotone function f if $\mathcal{M}$ accepts an input x iff $f(x) = 1$.*

Theorem A.5 (Linear SSS and monotone span programs [KW93])**.** *A monotone function $f : \{0,1\}^n \to \{0,1\}$ can be realized by a linear secret-sharing scheme over a field $\mathbb{F}$ with total share size $M \log |\mathbb{F}|$ iff it can be computed by a monotone span program of size M over $\mathbb{F}$.*

We can always restrict the matrix of a span program to a set of linearly independent columns without changing the function that is computed by the program, which implies the following well-known fact:

Fact A.6. *For every MSP $\mathcal{M}$ represented by an $(a \times b)$ matrix there exists an MSP $\mathcal{M}'$ that computes the same function with a matrix of size $(a \times a)$.*

Theorem A.7 (Algorithm for MinLSS)**.** *Given a truth-table of a monotone function $f : \{0,1\}^n \to \{0,1\}$ of size $N = 2^n$ and a bound $S > n$ on the total share-size, there is an algorithm that determines if f has a linear secret-sharing of total share size of S in non-deterministic time of $2^n \cdot \mathrm{poly}(S)$ where the witness size is S^2.*

Proof. By Definition A.4, Theorem A.5 and Fact A.6, the fact that f has a linear secret-sharing scheme (over $\mathbb{F}_2$) with total share size S can be witnessed by an MSP that computes f with a binary matrix M of dimension $S \times S$. It can be verified that such an MSP computes a function f in time $2^n \cdot \mathrm{poly}(S)$ by enumerating every input x of f and checking via Gaussian elimination whether the linear-algebraic correctness or privacy constraints of the span program for x are satisfied. I.e., if $f(x) = 1$ the rows of M_x should span the unit vector e_1, and if $f(x) = 0$ they should not. □

Deterministically, we can run over all the S^2 possible monotone span programs over $\mathbb{F}_2$ of size S and verify if they compute the function f with total complexity $2^{S^2+n} \cdot \mathrm{poly}(S)$. We note that when $S = O(n)$, this algorithm runs in time $N^{O(\log N)}$, compared to the algorithm for general schemes that has complexity $2^{\Omega(N)}$ in the worst case.

B More on Graphs and Communication Complexity

We partially extend Theorem 6.2 to the case of general (non-bipartite) graphs.

Theorem B.1 (Linear secret sharing for graphs and P4-free covers)**.** *Let G be a graph that can be realized by a LSS with max-share size k. Then, the edges of G can be covered by at most $\log n \cdot 2^{2k}$ P4-free graphs.*

Proof. The theorem can be derived from Theorem 6.2, where it is shown that *bipartite* graphs with LSS with max-share size k can be covered by at most 2^{2k} P4-free graphs. Given a graph G, we can cover its edges by at most $\log n$ bipartite graphs $G_1, \ldots, G_{\log n}$. This can be done by writing the names of the nodes using $\log n$ bits, and letting G_i denote the graph that contains only the edges of G that connect nodes whose ith bit is different. By assumption, we have a LSS with k bits for each G_i. Now, we apply Theorem 6.2 and cover each G_i by 2^{2k} P4-free bipartite graphs. Overall, we cover G by $\log n \cdot 2^{2k}$ P4-free bipartite graphs as required.

□

In [BBM$^+$20, Appendix C] the writers show that for every *bipartite* graph G it holds that $\exists\mathsf{EQ}(G)$=0 iff it is a P4-free graph. By Theorem 6.2, $\exists\mathsf{EQ}(G) = 0$ iff $\mathsf{LS}_{\mathsf{max}}(G) = 1$, and in total we get a covering-characterization of bipartite graphs that can be realized with 1-bit shares. We extend this characterization to general graphs, and prove it without going through communication-complexity measures:

Fact B.2 (Graphs with 1-bit shares)**.** *A graph G has 1-bit shares iff G is a P4-free bipartite graph.*

Proof. First, we show that if a graph G does not have a 1-bit scheme then it is not a P4-free bipartite graph. By [ABI$^+$23a], an access structure G has 1-bit shares iff for every minterm W and maxterm Z it holds that $|W \setminus Z|$ is odd. Let G be a graph access structure and let $W = \{w_1, w_2\}$ and Z be a minterm and a maxterm such that $|W \setminus Z|$ is even (W is of size 2 since G is a graph). Due to monotonicity $W \subsetneq Z$, and so if $|W \setminus Z|$ is even then W and Z are disjoint. This implies that w_1 and w_2 are not in Z, and since Z is a maxterm there exist edges between w_1 and some $z_i \in Z$ and between w_2 and some $z_j \in Z$. If z_i and z_j are the same party the graph contains a triangle $\{w_1, w_2, z_i\}$, and is therefore not bipartite. If z_i and z_j are different parties then the edge (z_i, z_j) cannot exist since Z is unauthorized, and therefore $\{z_i, w_1, w_2, z_j\}$ is a P4 subgraph of G.

In the other direction, if a graph G contains a triangle $\{a, b, c\}$, let W denote the minterm $\{a, b\}$ and Z denote a maxterm that contains c (there must exist such a maxterm since c is unauthorized). It must hold that $|W \setminus Z|$ is even, and therefore G cannot be realized with 1-bit shares. Similarly, if a graph contains a P4 subgraph $\{a, b, c, d\}$ the minterm $W = \{b, c\}$ and the maxterm Z that contains $\{a, d\}$ must be disjoint, and then they also satisfy that $|W \setminus Z|$ is even. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il