Electronic Colloquium on Computational Complexity, Report No. 213 (2026)

The Power of Multislices in Monotone Computation

Amos Beimel
Ben-Gurion University
amos.beimel@gmail.com

Oded Nir
Ben-Gurion University
odednir123@gmail.com

September 26, 2026

Abstract

Motivated by recent constructions and barriers in secret sharing, we study multislice functions. These functions, parametrized by a width parameter w, take the value 0 on inputs of Hamming weight below a base value k, 1 on inputs of weight above $k + w$, and are monotone in between. We first investigate formulas over multislice gates, which generalize the formulas over slices model (Applebaum et al., TOCT 2026). We prove that, although multislices compute complicated functions, they do not help in computing the worst-case function when $w \ll n$; that is, there is an explicit monotone function such that for every w, every formula over multislice gates of width w that computes it, regardless of its depth and gate fan-in, has size $2^{\Omega\left(n/((w+1)\log^2 n)\right)}$. Our lower bound is based on a randomized Karchmer-Wigderson protocol for multislices that can be amortized across the formula.

As a complementary result, we show how to realize multislices using slice gates, i.e., multislice gates of width 0; the construction uses a simple peeling recursion. Consequently, every width-w multislice on n variables has a formula over slice gates of size at most $2n^{w+1}$ and depth $(w+1)$. The same construction allows to compute multislices with small monotone real formulas and circuits, two models introduced by Pudlák (J. Symb. Log., 1997) in the context of proof complexity. This result also has an application to secret sharing: it yields, for sufficiently long secrets, multilinear secret-sharing schemes for width-w multislices with maximal information ratio $n^{O(w+1)}$, which is polynomial for every fixed w.

In addition, we prove that an explicit access structure requires shares of size $2^{\Omega(n)}$ in every secret-sharing scheme from a family of schemes that captures all currently known constructions for general access structures with share size $2^{cn+o(n)}$, for $c < 1$. Our proof goes through an exponential lower bound on the size of formulas over so-called CDS gates and ideal linear gates.

1 Introduction

Recent progress on secret sharing for general access structures has relied on representing monotone functions by formulas over richer bases of monotone gates. In these models, gates corresponding to specially structured access structures are treated as low-cost primitives, even though some of them require exponential-size monotone AND/OR circuits. These stronger gate bases have enabled constructions that bypass classical circuit-size barriers and realize every n-party access structure with share size $2^{cn+o(n)}$ for constants $c < 1$. The current best exponent is $c = 0.494$, due to Nir [46]. On the lower-bound side, Applebaum et al. [6] proved that an explicit monotone function requires formulas of size $2^{\Omega(n/\log^2 n)}$ even when arbitrary slice functions (that we will soon define) are available as gates. Thus, despite the strength of this particular family of monotone gates, they cannot realize every monotone function by formulas of subexponential size.

ISSN 1433-8092

In this work, we study the strength of an even stronger set of gates computing *multislice functions*. A $(k : k + w, n)$-multislice is a monotone Boolean function on n variables that is identically 0 on inputs of Hamming weight below k, identically 1 on inputs of Hamming weight above $k + w$, and is otherwise arbitrary, subject only to monotonicity; we refer to w as the width of the multislice. Slice functions, that were extensively studied in the context of secret sharing, e.g., in [2, 12, 3, 11, 6], are the special case $w = 0$, whereas one multislice width-n suffices to compute every monotone function over n bits. Multislices, sometimes also referred to as "fat slices", arise naturally in a variety of contexts, including secret-sharing schemes [42, 4, 7, 11], theories of partially ordered sets [19, 23, 15, 37, 35, 27], and counting monotone Boolean functions [36].

We study the monotone complexity of multislices from two complementary perspectives.

> **Question 1 (Composition).** *How powerful are multislices as gates for computing general monotone functions?*

To formalize this question, we consider formulas that compute n-bit functions whose gates are arbitrary multislice functions of width at most w. We call these formulas over multislices, or FOMS_w. For multislices of width 0, our model is the formulas-over-slices (FOS) model of [6], for which there exists an explicit function that requires formulas of size $2^{\Omega(n/\log^2 n)}$. Already at width zero, there are $2^{\binom{n}{\lfloor n/2 \rfloor}}$ different slice gates whose critical layer is the middle one. As the width increases, the allowed gates become progressively more expressive. The natural question is therefore how the complexity of computing a general monotone function decreases as the multislice width grows.

The second question we study is the following:

> **Question 2 (Realization).** *How cheaply can a multislice itself be realized using simpler monotone gates?*

Since standard formulas and circuits (with AND and OR gates) for multislices have to be large by a counting argument, we work with other monotone computational circuit models that allow for stronger gates. In particular, we ask for the complexity of multislices in the three computational models: formulas over slices and monotone real circuits and formulas. We now elaborate about these models, and their known connections to multislices.

Formulas over slices [6]. There are several known FOS realizations of multislices implicitly based on constructions of secret-sharing schemes. In particular, Liu and Vaikuntanathan [42] implicitly give FOS of size $2^{\tilde{O}(\sqrt{n})}$ for multislices of width $w = O(\sqrt{n})$, and Applebaum et al. [4, 7] realize $(\alpha n : \beta n, n)$-multislices with a FOS of exponential size that depends on α and β. These lead to FOS for every function with size $1.5^{n+o(n)}$ [7]. All these constructions use slice gates with huge fan-in of 2^{n^ε} for a constant $\varepsilon \in (0, 1)$, and have super-polynomial size even for constant w.

Monotone real circuits and formulas. A monotone real circuit (MRC) [48] is a fan-in-two circuit with Boolean inputs and outputs, whose gates compute arbitrary coordinatewise monotone functions $\mathbb{R}^2 \to \mathbb{R}$. That is, while the inputs and output are Boolean, intermediate values may be real. A monotone real formula (MRF) is an MRC whose underlying graph is a tree. By Rosenbloom [49], every m-bit slice function can be computed by a read-twice MRF, which allows to transfer FOS upper bounds to MRC bounds directly, and to MRF bounds with a blowup that depends on the depth of the formula and fan-in of the gates. This results in upper bounds of $1.5^{n+o(n)}$ for general functions in both models [6]. The strongest lower bounds are $2^{\tilde{\Omega}(n^{1/3})}$

for MRCs [52] and $2^{\Omega(n/\log^2 n)}$ for MRFs [6]. We do not know of lower bounds for w-multislices in these models.

The above Questions 1 and 2 give complementary views of the role of multislices in monotone computation: the first treats them as primitive gates and asks what they can compute, while the second treats them as target functions and asks how complicated they are.

Secret-sharing motivation, applications, and other gate models

The above questions are also motivated by secret-sharing schemes, which give rise to further monotone complexity measures. A secret-sharing scheme, introduced by Shamir [50] and Blakley [14] for the threshold case and generalized by Ito, Saito, and Nishizeki [29], allows a dealer to distribute a secret among n parties so that only predefined authorized subsets can reconstruct it, while every unauthorized subset learns nothing. The collection of authorized subsets is called an *access structure* and can be represented by a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$.

Two well-studied complexity measures in secret sharing are the largest share size and, for long secrets, the *information ratio*, defined as the largest share size normalized by the secret length. Following a sequence of works initiated by the breakthrough result of Liu and Vaikuntanathan [42], the best known share size for general access structures is $2^{0.494n+o(n)}$ [46], whereas the best known lower bounds which also apply to the information ratio are only polynomial [16, 55]. Closing this exponential gap remains a central open problem.

Multislices play an important role in this line of work. With the exception of the most recent construction in [46], the constructions in the Liu-Vaikuntanathan line [42, 6, 4, 7, 1] proceed by first realizing multislice access structures and then using them as building blocks for general access structures. This makes formulas over multislices a natural abstraction of an important family of secret-sharing constructions. Hence, bounding the size of formulas over multislices for general functions would imply that any multislice secret-sharing construction that composes with a formula for such multislices must be correspondingly expensive.

For upper bounds, FOS realizations of multislices give rise to secret-sharing schemes whose information ratio depends on the properties of the formula. This follows from the fact that, for long secrets, there exist multilinear schemes for slice functions with polynomial information ratio [2, 12, 3]. However, the FOS constructions mentioned above only imply schemes with information ratio $2^{\tilde{O}(\sqrt{n})}$ even for constant-width multislices.

Additional gate families. There are three additional gate families that are significant for secret-sharing constructions: *conditional disclosure of secrets* (CDS) gates that correspond to CDS protocols [24], their robust version, RCDS gates, that correspond to RCDS protocols [4], and linear secret-sharing gates, corresponding to monotone span programs [32]. These families are described informally in Section 1.2 and defined in Section 5. CDS and RCDS gates were used as smaller building blocks to realize with a formula the multislice gates in the secret-sharing schemes mentioned above, and were then combined with linear gates to obtain the state of the art exponent of [46]. This motivates the following analogue of Question 1:

> **Question 3 (Composition).** *How powerful are CDS gates, robust CDS gates and their combinations with linear gates for computing general monotone functions and for building secret-sharing schemes for general access structures?*

Related work

In addition to FOS and monotone real-circuit models discussed above, many other works have studied the expressive power of "strong" gates. E.g., Krajicek and Oliveira studied monotone circuits with additional input gates that compute restrictions of the original function [39, 40], and Oliveira and Pudlák investigated the power of linear-programming gates [18]. In a work with a closer connection to ours, Jukna studied gates that compute functions that have only small minterms and maxterms [30]. These include $(0 : w, n)$-multislices and $(n-w : n, n)$-multislices, but do not include more central slices that have higher complexity. He also studied the power of monotone real gates of high fan-in.

1.1 Our results

Our first result extends the FOS lower bound of [6] to the FOMS$_w$ model. We prove the following theorem:

Theorem 1.1 (FOMS$_w$ lower bound)**.** *There is an explicit monotone function $F_n : \{0,1\}^n \to \{0,1\}$ such that, for every function $w : \mathbb{N} \to \mathbb{N}$, every FOMS$_w$ (see Definition 2.4) computing F_n has size*

$$S \;\geq\; 2^{\Omega\left(\frac{n}{(w(n)+1)\log^2 n}\right)}. \tag{1}$$

This result holds even for formulas of large depth and fan-in of its gates. The bound degrades as w approaches $n/\log^2 n$; this cannot be avoided (up to logarithmic factors). Indeed, every monotone function on n bits is a $(0 : n, n)$-multislice, and hence has a FOMS$_n$ of size $O(n)$ consisting of one width-n gate and the input gates. Then, for example, replacing w in the denominator in (1) by $w^{1-\varepsilon}$ would give, at $w = n$, a lower bound of $2^{\Omega(n^{\varepsilon}/\log^2 n)}$, contradicting this $O(n)$ upper bound. For constant width, the exponent matches the FOS bound of [6]. In particular, for width 0 our lower bound gives an alternative proof of the main theorem of [6] that avoids the framework of monotone real circuits.

In our second result, we give upper bounds on the size of FOS, MRFs, and MRCs computing multislice functions, as summarized by the following theorem:

Theorem 1.2 (FOS, MRFs and MRCs for multislices)**.** *Every $(k : k + w, n)$-multislice is computed by a depth-$(w + 1)$ FOS of size at most $2n^{w+1}$, by a monotone real formula of size at most $2^{O(w)}n^{w+1}$, and by a monotone real circuit of size $O\left(n\sum_{d=0}^{w}\binom{n-1}{d}\right)$.*

These bounds are polynomial for constant width and quasipolynomial for $w = \operatorname{polylog} n$, while the circuit bound remains subexponential for every $w = o(n)$. The construction is based on a *peeling* idea: restrict away one coordinate, force the old top layer to 1 to lower the width, and use a single slice gate to restore the discarded information. This result indeed has a direct implication for secret sharing.

Amortized secret sharing. Composing long-secret schemes for slices of [2, 12, 3] with the construction of Theorem 1.2 gives the following consequence. Theorem 4.3 gives a sharper parameterized statement.

Theorem 1.3 (Amortized secret sharing for multislices)**.** *Every $(k : k + w, n)$-multislice access structure has, for long secrets, a multilinear secret-sharing scheme of information ratio $n^{O(w+1)}$. In particular, this ratio is polynomial for every fixed width w.*

A secret-sharing scheme over a finite field $\mathbb{F}$ is *multilinear* if the secret is a vector in $\mathbb{F}^{\ell}$ and every share is an $\mathbb{F}$-linear function of the secret vector and the randomness. This gives the first secret-sharing schemes for multislices of constant and polylogarithmic width with information ratios $n^{O(1)}$ and $n^{\log^{O(1)}(n)}$, respectively. In addition, it was proved in [36] that almost all monotone functions are multislices of width 2, so together with Theorem 4.3 this immediately implies the following corollary, improving the $2^{\tilde{O}(\sqrt{n})}$ result of [10].

Corollary 1.4. *Almost all monotone access structures with n parties have, for sufficiently long secrets, a multilinear secret-sharing scheme with information ratio $O(n^5)$.*

Additional formula models and secret sharing. We prove the following bounds on formulas with CDS gates, denoted as FCDS, on formulas with CDS gates and gates that have ideal linear schemes (that is, linear schemes with information ratio 1) over a common finite field $\mathbb{F}$, denoted as FCDSL$_\mathbb{F}$, and on formulas with fully-robust CDS gates with q servers, denoted as FRCDS$_q$. Recall that such formulas were used to realize general functions (with multislices as an intermediate step) in the secret-sharing constructions mentioned above.

Theorem 1.5 (FCDSL$_\mathbb{F}$ lower bound)**.** *There is an explicit monotone function $F_n : \{0,1\}^n \to \{0,1\}$ and a constant $c > 0$ such that, for every finite field $\mathbb{F}$, every FCDSL$_\mathbb{F}$ computing F_n has size at least 2^{cn}. In particular, every FCDS computing F_n has size at least 2^{cn}.*

Theorem 1.6 (FRCDS$_q$ lower bound)**.** *For every $2 \leq q \leq n$, for almost all monotone functions $F : \{0,1\}^n \to \{0,1\}$, every FRCDS$_q$ computing F has size at least $2^{(n-O(\log n))/q}$.*

Theorem 1.5 allows us to prove the following corollary, that holds for every secret-sharing scheme constructed by composing schemes for CDS gates and *general* (not necessarily ideal) linear schemes according to a formula.

Corollary 1.7. *There is an explicit monotone access structure F_n such that every secret-sharing scheme realizing F_n by formula composition of linear schemes over a common finite field and schemes for CDS gates has information ratio $2^{\Omega(n)}$.*

1.2 Our techniques

1.2.1 The FOMS lower bound

Our FOMS$_w$ lower bound follows from a reduction to randomized monotone Karchmer-Wigderson protocols. Informally, in such a protocol for a monotone function f, Alice receives a 1-input x of f and Bob receives a 0-input y of f. They need to find a coordinate i with $x_i = 1$ and $y_i = 0$, which exists by monotonicity. A randomized protocol allows the players to use randomness and is required to find such a coordinate with constant success probability on every pair (x, y), and its cost is the number of communicated bits.[1] Given a size-S FOMS$_w$ that computes a function f on n bits, with no restriction on its depth or gate fan-in, we describe a randomized monotone KW protocol for f with complexity $O((w+1)\log S \log n)$. Combining this with the $\Omega(n/\log n)$ lower bound of Göös and Pitassi [26] on the communication for an explicit function in such monotone protocols directly leads to the bound.

Comparison to the FOS lower bound. Before going into more depth, it is instructive to describe the proof of the $2^{\Omega(n/\log^2 n)}$ FOS lower bound by Applebaum et al. [6], and see where and why we deviate from it. In [6], the bound is also obtained by constructing a randomized KW protocol, but the authors do not build it directly; instead they rely on a few intermediate steps. First, they convert the FOS to an MRC via the transformation of Rosenbloom [49]. They then use the fact that the resulting circuit has a structural property called *separability* to turn it into a low-cost real communication protocol for the monotone Karchmer-Wigderson game via a balancing argument [38]. In such protocols, every communication

[1] In contrast, in the non-monotone KW game it suffices to find any coordinate with $x_i \neq y_i$. This game has a randomized protocol with communication $O(\log n)$ for every function [44].

step consists of Alice and Bob sending a real number to an oracle that computes the greater-than function. Lastly, each real comparison can be simulated by a randomized Boolean protocol at an additional $O(\log n)$ communication cost.

We emphasize that using this framework as is for formulas over multislices does not provide the lower bounds we seek. One possible strategy is to reduce our question directly to the FOS lower bound. Given a FOMS, we can replace every multislice gate in it with a formula over slices that computes the same multislice function. Indeed, our second result shows that multislices do admit relatively small FOS realizations (and this was one of the reasons we obtained this result). However, the DAG resulting from such a transformation is not necessarily a formula, because each multislice gate may have high fan-in, and each of its input variables can be read many times. Hence, this approach either produces a circuit over slice gates for which we can obtain much weaker bounds (i.e., via bounds for general MRCs) or requires a depth-dependent blowup to transform the circuit back to a formula.

Another possible strategy is to take every multislice gate in a FOMS and replace it with an MRF that computes the same function. Similarly to the previous case, this would result in a monotone real circuit, but unlike the circuits that are generated from formulas over slices, it is not apparent why such circuits should be separable, and thus the chain of reductions used in [6] stops. Instead, we realize a randomized KW protocol directly, as specified next.

Our construction. The FOMS$_w$ lower bound follows a three-step pipeline. Given a monotone function $f : \{0,1\}^n \to \{0,1\}$ that can be computed by a size-S FOMS$_w$ formula F, our goal is to implement a cheap randomized KW protocol for f based on F.

1. *Balance the formula.* The balancing lemma of [5, Lemma 6.1] transforms a size-S formula over arbitrary monotone gates into an equivalent formula of depth $O(\log S)$ and size $S^{O(1)}$, using only the original gates and additional AND/OR gates. Thus, it is enough to pass through $O(\log S)$ gates along a root-to-leaf path in the protocol. We describe how this can be done in the next step. [2]

2. *Traversing the gates from root to leaf.* Starting at the root, the players traverse the formula gate by gate, each time using a subprotocol selecting a child whose subformula accepts Alice's input and rejects Bob's input. This produces a root-to-leaf path whose final variable solves the KW game for f. This subprotocol is the substantial part of our solution, and we soon explain how it can be done with total communication of $O((w+1)\log S \log n)$.

3. *Direct randomized protocol and the hard function.* We compare the cost of the resulting randomized KW protocol with the $\Omega(n/\log n)$ lower bound of Göös and Pitassi [26], obtaining our FOMS$_w$ formula-size lower bound.

We now describe our protocol for a single gate. Suppose the protocol has reached a gate v in the formula. Evaluating the children of v on Alice's input and on Bob's input gives two Boolean vectors x and y such that the gate v accepts x and rejects y. To continue toward an input variable, the players must find a child u for which $x_u = 1$ and $y_u = 0$.

We first assume that x is larger than y when both are interpreted as binary numbers (we will show how to get rid of this assumption later). The most significant coordinate u on which they differ then satisfies $x_u = 1$ and $y_u = 0$, so finding this coordinate identifies the desired child.

[2] We note that it is possible to "balance the protocol" along the way without first balancing the formula, but we choose to balance the formula since it cleanly separates the balancing argument from the other orthogonal steps of the protocol.

The players can find this coordinate by binary search. At each step, they divide the current interval of coordinates into a left part and a right part. The players check whether their two left parts are equal. Later, standard randomized equality protocols convert each such comparison into a low-communication randomized Boolean protocol. If the two left parts are equal, the most significant difference lies in the right part; otherwise, it lies in the left part. The parties recursively continue the protocol with the appropriate part. This resembles the randomized non-monotone KW game of [44]. If the players partition their intervals to two equal parts, then the cost is the logarithm of the gate fan-in, which could be arbitrarily large, and would later lead to a square-root dependency in w/n that we try to avoid (see Remark 1.8).

To save costs, the players find the desired coordinate with an unbalanced binary search. We use a Gilbert-Moore alphabetic search tree whose leaf weights are the sizes of the child subformulas. The search is biased toward larger subformulas, and if it selects child u of gate v, its cost is $O(\log(|\Phi_v|/|\Phi_u|))$, where Φ_z is the size of the subtree of z in the formula. Writing $v_0, v_1, \ldots, v_T$ for the nodes on a root-to-leaf path, where v_{i+1} is the child selected at v_i, these charges telescope:

$$\sum_{i=0}^{T-1} \log \frac{|\Phi_{v_i}|}{|\Phi_{v_{i+1}}|} = \log \frac{|\Phi_{v_0}|}{|\Phi_{v_T}|} \leq \log S.$$

To reduce the general case to the case where $x > y$ as assumed above, the players sample a random permutation π of the m children and compare the weighted sums representing the two permuted binary strings; For example, if $x = (0, 0, 1)$ and $y = (1, 1, 0)$, the permutation that swaps the first and third bits leads to permuted inputs where $x > y$. We show that for a width-w multislice gate, Alice's permuted number is larger than Bob's with probability $\Omega(1/(w + 1))$. Alice and Bob check, using a randomized protocol for the greater-than function, if the permuted number of Alice is indeed greater than the permuted number of Bob. If not they choose a fresh random permutation. Repeating with fresh permutations therefore succeeds after an expected $O(w + 1)$ trials. This idea can be traced back to the partially ordered sets literature [19, 23, 15, 27].

Thus traversing all gates on the path uses an expected $O((w + 1) \log S)$ real comparisons, with no dependence on the gate fan-ins.

Gilbert-Moore trees were previously used in many contexts, including comparison-based sorting with partial information, distribution-sensitive sorting and comparison-query twenty questions [21, 45, 17], but to our knowledge, not in balancing formulas in protocols as we do here.

The protocol for the entire formula Each comparison of strings is executed by a randomized greater-than protocol. Markov's inequality shows that a global budget of $O((w + 1) \log S)$ random-permutation trials is exceeded with only constant probability, and a union bound controls all comparison errors. This gives a randomized KW protocol of cost $O((w + 1) \log S \log n)$, which is compared directly with the Göös-Pitassi lower bound (Theorem 2.11) to obtain

$$(w + 1) \log S \;\geq\; \Omega\left(\frac{n}{\log^2 n}\right).$$

Rearranging gives the desired lower bound $S \geq 2^{\Omega\left(\frac{n}{(w+1)\log^2 n}\right)}$.

Remark 1.8 (The importance of the unbalanced search)**.** *Assume we did not use the Gilbert-Moore based search, and in each gate of arity m we would perform a binary search with communication $O(\log m)$. The total protocol cost would then be $\log S(w + 1) \log m \log n$. To get a bound that is independent of the fan-in*

m, we would have to assume $m \le S$. Comparing this with the Göös-Pitassi lower bound $\Omega(n/\log n)$ yields only $(w+1)(\log S)^2 \log n \ge \Omega\Big(\frac{n}{\log n}\Big)$, and hence $\log S \ge \Omega\Big(\frac{1}{\log n}\sqrt{\frac{n}{w+1}}\Big) = \widetilde{\Omega}\Big(\sqrt{\frac{n}{w+1}}\Big)$. Thus, this approach gives merely $S \ge 2^{\widetilde{\Omega}(\sqrt{n/(w+1)})}$ instead of the desired $S \ge 2^{\widetilde{\Omega}(n/(w+1))}$.

1.2.2 The peeling construction

On the upper-bound side, we first describe a simple FOS construction for multislices. The construction peels off the top layer of the multislice. That is, we construct width-w multislices using width-$(w-1)$ multislice gates. For each coordinate i, we first restrict f to inputs in which the ith input is 0 and then force the old top layer to 1, obtaining a residual multislice f_i on $n-1$ variables whose width is $w-1$. Taken together, the residuals retain all the information about f on levels $k, \ldots, k+w-1$: for an input $x \neq 1^n$, choosing any i such that $x_i = 0$ gives $f_i(x) = f(x)$. The only information removed is the behavior of f on level $k+w$; we show that this information can be encoded with a single outer slice gate. Specifically, if we let x^i be obtained by setting the ith bit of x to 0 and leaving all other bits unchanged and denote $z_i = f_i(x^i)$, we show that $f(x) = G(z_1, \ldots, z_n)$ for some slice function G of weight $n-k-w$. Iterating this identity reduces the width of the multislice gates by 1 in each iteration, which ultimately gives a depth-$(w+1)$ FOS of size at most $2n^{w+1}$.

To obtain monotone real formulas, we replace every slice gate in this FOS with the Rosenbloom read-twice formula [49]. To go back to a formula structure we duplicate the two reads of each child at every level, producing the $2^{O(w)}n^{w+1}$ bound.

For monotone real circuits, we use dynamic programming, merging the many occurrences of the same subfunction into a single shared subcircuit. This avoids the duplication required in a formula and gives the sharper binomial-sum bound.

We note that for "low" and "high" multislices, i.e., $(0 : w, n)$ and $(n-w : n, n)$-multislices, the secret-sharing schemes in [11] are implicitly based on FOS of size $w^{O(w)}n\log^2 n$. Thus, their construction outperforms ours when $w = O(\log n/\log\log n)$.

1.2.3 Other Formula Models and Implications

Formulas over CDS gates and linear gates. Another class of monotone gates arising in the secret-sharing literature is *CDS gates*, that correspond to conditional disclosure of secrets protocols [24]. These have surprisingly cheap realization by secret-sharing schemes, and they were used in all constructions with share size 2^{cn} for $c < 1$. A CDS gate with N inputs can be represented by q-partite hypergraphs with N vertices. Concretely, its inputs are partitioned into parts $V_1, \ldots, V_q$, and it is specified by a set of hyperedges $E \subseteq V_1 \times \cdots \times V_q$, thinking of each input variable as a vertex. Then, by a standard definition (for more details, see Section 5.1), the gate outputs 1 exactly when the input variables assigned 1 contain either all the vertices of some hyperedge in E or two vertices belonging to the same part.

We observe that two existing bounds on secret-sharing schemes imply a $2^{\Omega(n)}$ lower bound on the size of formulas with CDS gates for an explicit function, thus settling the worst-case power of this model up to the constant in the exponent. The secret-sharing results are the following: Applebaum and Arkis [2] proved that functions computed by CDS gates can be realized by multilinear secret-sharing schemes with constant information ratio, and recently, Khazaei [34] described an explicit access structure F_n that requires exponential information ratio from any multilinear scheme realizing it over any finite field.

Hence, if F_n had an FCDS of size S, balancing it (as we have done so far) would produce a formula of size $S^{O(1)}$ and depth $O(\log S)$. Composing the constant information-ratio schemes for its gates through

the formula structure would then give a multilinear scheme with information ratio $S^{O(1)}$, and comparing the two bounds yields $S = 2^{\Omega(n)}$. Thus, unlike the counting lower bound for FRCDS_q formulas which appears next, the FCDS lower bound is explicit and holds even when the input of the gates may be partitioned to an arbitrary number of parts.

The argument tolerates additional gates, as long as they have multilinear schemes of constant information ratio over the field in question. In particular, we may add *ideal linear gates* over a finite field $\mathbb{F}$, i.e., functions that have a linear scheme over $\mathbb{F}$ in which every input receives a single field element; this gives the $\text{FCDSL}_{\mathbb{F}}$ lower bound of Theorem 1.5.

Corollary: A secret-sharing lower bound. A size lower bound for a formula model generally implies a lower bound on the information ratio of the secret-sharing schemes obtained by composing, according to such a formula, secret-sharing schemes for its gates. This is true since in a secret-sharing scheme, every essential input (that is, an input that the function depends on) receives a share that is at least as long as the secret [33], so every leaf of the formula contributes at least one secret-length to the share of the party labeling it (Lemma 2.8), and the information ratio is at least the number of leaves divided by n. Thus, Theorem 1.5 bounds the information ratio of schemes composed from arbitrary schemes for CDS gates, which may be nonlinear, and ideal linear schemes over $\mathbb{F}$.

In Corollary 1.7, we show that the bound also covers linear gates realized by non-ideal linear schemes, which is how linear gates appear in the known constructions. A linear scheme for a gate g that gives a_i field elements to input i can be seen as an ideal linear scheme for a gate $\widetilde{g}$ on $\sum_i a_i$ inputs, and g is obtained from $\widetilde{g}$ by feeding the subformula of input i to a_i of its inputs. Replacing every linear gate in this way turns the formula underlying such a scheme into an $\text{FCDSL}_{\mathbb{F}}$ computing F_n, whose size is therefore at least $2^{\Omega(n)}$. Additionally, we show that the size of this modified formula can be bounded in terms of the information ratio ρ of the scheme based on the original formula. Then, applying Theorem 1.5 to the larger formula yields $\rho = 2^{\Omega(n)}$.

This model captures all known secret-sharing schemes achieving a share-size exponent below 1. Nir [46] and Applebaum et al. [6] asked about the power of formulas combining linear gates and FOS gates. Although our result does not resolve this question directly (as CDS gates are weaker than slice gates), it establishes the desired secret-sharing corollary: formulas over the secret-sharing primitives we use today remain weak, in the sense that an explicit access structure still requires exponential information ratio when using them in the formula decomposition method.

Formulas over fully robust CDS gates. To obtain secret-sharing schemes with smaller shares, Applebaum et al. introduced in [4] robust conditional disclosure of secrets protocols, and their representation as *robust CDS gates*. These later led to even more efficient schemes, e.g., in [7, 11]. Specifically, we will study q-server fully robust gates with N inputs, that, similarly to CDS gates, can be represented by q-partite hypergraphs with N vertices, though under a different representation. Concretely, partition the input variables into parts $V_1, \ldots, V_q$ and choose a set of hyperedges $E \subseteq V_1 \times \cdots \times V_q$. The corresponding gate is

$$g(z) = \bigvee_{(v_1,\ldots,v_q)\in E} \bigwedge_{i=1}^{q} z_{v_i}$$

(i.e., each minterm of g contains exactly one variable from each part V_i). Thus, the gate outputs 1 exactly when the input variables assigned 1 contain all the vertices of some hyperedge, and unlike CDS gates, inputs with many 1's in each part may be rejected.

These can be seen as a stronger version of CDS gates, as every RCDS gate can be turned to a CDS gate by composing it with additional cheap threshold gates; the other direction is currently known only through involved constructions described in [4]. Additionally, unlike multislice gates, which may require exponentially large AND/OR circuits even at width zero (e.g., for the middle slice), these gates can be computed by sub-exponential formulas for all $q = o(N)$. This allows us to prove a formula lower bound analogous to that of multislices by a simple counting argument, and highlights the interest in the multislice lower bound. We note that this lower bound also holds for weaker versions of the gates, such as RCDS gates that are not fully robust, and ordinary CDS gates for which we proved a stronger bound.

The proof idea is the following. We denote formulas over q-part hypergraph gates by FRCDS$_q$. A gate on N inputs involving at most q parts can be expanded into a monotone circuit of size $N^{O(q)}$. Consequently, a formula over such gates of size S can be expanded into a monotone circuit of size $O(S^q)$. Combining this simulation with the standard counting lower bound for monotone circuits shows that almost every monotone function requires FRCDS$_q$ gates of such size

$$2^{(n-O(\log n))/q}.$$

For this model, we also provide a tight upper bound up to polynomial factors, that generalizes a standard construction for the $q = 2$ case (see, e.g., [8]). Every monotone function has an FRCDS$_q$ of size $O(n2^{\lceil n/q \rceil})$, obtained by partitioning the variables into q blocks, computing all conjunctions within each block, and feeding them to one partite-hypergraph gate. The details appear in Section 5.

Remark 1.9. *The FRCDS$_q$ lower bound above may appear weak when q is large. However, even for the weaker CDS gates, the currently best-known q-server protocols have messages of size $2^{\tilde{O}(\sqrt{q \log N})}$ where N is the domain size of each server [43]. Hence, even if we had RCDS gates at CDS cost (which is far from true today), we would still need improvements in CDS to take full advantage of the upper-bound possibilities when q is large.*

Partially ordered sets and MRC for multislices. In Section 6, we study connections between restricted MRF constructions for multislices and a complexity measure of partially-ordered sets. These connections lead to MRC upper bounds that are usually weaker than the ones obtained with other methods, but for $(1 : w, n)$-multislices with constant w they give circuits of size $O(n \log \log n)$, improving on the $O(n \log^2 n)$ bound obtained by converting the FOS implicit in [11].

In particular, we study a single-top-gate MRF architecture: first compute several monotone real-valued functions of the input, called *coordinates*, and then feed them to one high fan-in monotone gate. To turn such an MRF to a standard MRC with gates of fan-in 2, we use a theorem by Hrubeš-Pudlák [28]. To understand how many coordinates are needed to compute a $(k : k + w, n)$ multislice, we view the subsets of Hamming weights between k and $k + w$ as a partially ordered set under inclusion. The Dushnik-Miller dimension of this poset is the minimum number of total orders whose common comparisons recover exactly the inclusion relation [20]. We show in Theorem 6.1 that this dimension is precisely the minimum number of fixed coordinates needed to represent such multislices. Consequently, known upper and lower bounds on poset dimension translate directly into bounds for this universal single-top-gate MRF architecture, and the upper bounds are passed forward to MRCs. Our construction for MRFs discussed before avoids this architecture and achieves smaller MRFs.

FOMS and MRC upper bounds. For every $w = o(n)$, we do not know how to exploit the additional gate width to obtain an upper bound significantly better than the best known FOS bound, corresponding to

$w = 0$. In particular, determining whether our FOMS lower bound is tight is an interesting open problem. We note that a matching upper bound for every w realized by formulas with polynomial gate fan-in would have strong implications for MRC upper bounds. For example, taking $w = \lceil \sqrt{n} \rceil$, every monotone function would then have a FOMS$_w$ of size $2^{\widetilde{O}(\sqrt{n})}$. Replacing each poly$(n)$ fan-in width-w gate with its MRC peeling realization would give an MRC of size $2^{\widetilde{O}(\sqrt{n})}$ for every function over n bits. Recall that the best MRC construction today for all functions has size $1.5^{n+o(n)}$, and that the best-known lower bound is $2^{\widetilde{\Omega}(n^{1/3})}$.

1.3 AI Disclosure

The research questions studied in this paper were posed by the authors, and the results were discovered in conversations prompted by the authors with GPT-5.6 Sol. The authors verified the correctness and originality of all content.

1.4 Organization

In Section 2, we list the required definitions of formulas, secret sharing and KW protocols. In Section 3 we prove the FOMS$_w$ lower bound using a randomized KW protocol, and in Section 4 we describe the peeling construction for FOS, MRFs, and MRCs, and show that it applies to amortized secret sharing. In Section 5, we prove the lower bound for formulas over CDS gates and linear gates, derive the corresponding lower bound for formula-based secret-sharing schemes, and prove matching bounds for formulas over fully robust CDS gates. Finally, in Section 6 we study universal single-top-gate MRF architectures through poset dimension.

2 Preliminaries

In this section, we define the basic notions used in this work. For two strings $y, y' \in \mathbb{R}^n$, we say that $y \leq y'$ if $y_i \leq y'_i$ for every $i \in [n]$. Throughout, logarithms are base 2.

Monotone Boolean functions. Let $a, b \in \{0,1\}^n$ be a pair of equal-length strings. We say that $a \leq b$ if $a_i \leq b_i$ for every $i \in [n]$, and say that $a < b$ if, in addition, $a_j < b_j$ for some $j \in [n]$. A function $f : \{0,1\}^n \to \{0,1\}$ is monotone if for every $a \leq b \in \{0,1\}^n$, it holds that $f(a) \leq f(b)$. A minterm of a monotone function f is an assignment b such that $f(b) = 1$ and $f(a) = 0$ for every $a < b$. Similarly, a maxterm of $f : \{0,1\}^n \to \{0,1\}$ is an assignment b such that $f(b) = 0$ and $f(a) = 1$ for every $a > b$. A monotone function is fully defined by its minterms (resp., maxterms).

We move on and define monotone real formulas and circuits as introduced in [48].

Definition 2.1 (Monotone real circuits and formulas)**.** *A* monotone real function $f : \mathbb{R}^n \to \mathbb{R}$ *is a function such that, for every two inputs $x = (x_1, \ldots, x_n), x' = (x'_1, \ldots, x'_n) \in \mathbb{R}^n$ satisfying $x_i \leq x'_i$ for every $i \in [n]$, it holds that $f(x) \leq f(x')$. A* monotone real gate G *takes as input n values $x_1, \ldots, x_n \in \mathbb{R}$, computes some monotone real function $f : \mathbb{R}^n \to \mathbb{R}$, and returns $f(x_1, \ldots, x_n)$ as an output. A* monotone real circuit (MRC) C *is a circuit in which each gate is a monotone real gate G with fan-in 2 and for every input $x \in \{0,1\}^n$ the output of the circuit C is Boolean. A* monotone real formula (MRF) *is a monotone real circuit whose DAG is a tree.*

We note that in an MRC/MRF the inputs and outputs are Boolean, while the values on internal edges can be any real numbers. We allow AND and OR gates and other monotone Boolean gates in an MRC with the convention that their inputs are always Boolean. Taking monotone real gates with fan-in 2 is the more common definition of MRCs. Furthermore, in our constructions of MRFs the fan-in of all gates is 2.

Remark 2.2. *Although an internal gate of an MRC or MRF may output arbitrary real values, we may assume without loss of generality that its output lies in* $\{1, \ldots, 2^n\}$. *Indeed, on the* 2^n *Boolean inputs to the circuit, each gate attains at most* 2^n *distinct values. We may replace these values by their ranks in increasing order. This relabeling preserves all comparisons, so the subsequent monotone gates can be redefined on the input pairs that actually occur, and extended monotonically elsewhere, without changing the Boolean function computed by the circuit.*

We continue with the definition of multislice gates and formulas over multislice gates. Throughout the paper, we denote the Hamming weight of a string y by $\mathrm{wt}(y)$.

Definition 2.3 (Multislice and slice functions and gates)**.** *Let* $0 \leq k \leq k + w \leq n$. *A monotone function* $g : \{0,1\}^n \to \{0,1\}$ *is a* $(k : k + w, n)$-multislice function *if for every* $y \in \{0,1\}^n$*:*

- *If* $\mathrm{wt}(y) < k$, *then* $g(y) = 0$.
- *If* $k \leq \mathrm{wt}(y) \leq k + w$, *then* $g(y)$ *can be either* 0 *or* 1 *(subject to monotonicity).*
- *If* $\mathrm{wt}(y) > k + w$, *then* $g(y) = 1$.

We refer to w *as the width of the function. A* $(k : k + w, n)$*-multislice gate is a monotone gate computing a* $(k : k + w, n)$*-multislice function. We call multislices with* $w = 0$ *slice functions, and gates that compute such functions slice gates.*

Definition 2.4 (Formulas over multislices)**.** *A* formula over multislices of width w *(FOMS*$_w$*) is a formula in which every internal gate* g *of fan-in* m_g *computes a* $(k_g : k_g + w_g,\ m_g)$*-multislice function (Definition 2.3) for some* $w_g \leq w$. *The base level* k_g*, width* $w_g \leq w$*, and fan-in* m_g *may vary between gates while only the upper bound* w *is fixed. We call a FOMS*$_0$ formula over slices *(FOS), coinciding with the model of [6].*

Formulas over CDS gates, linear gates, and RCDS gates are defined analogously in Section 5. In this paper, we define the size of a circuit/formula as the number of *gates* in the circuit/formula (including input gates). This convention is used both for circuits with monotone real gates and for formulas over slice gates. We note that since monotone real circuits have fan-in 2, our definition of monotone real circuit size is essentially equivalent to the definition that counts the total number of edges in the circuit. Furthermore, the same is true for the formula models.

Finally, we quote the arity-reduction theorem of Hrubeš and Pudlák, used in Section 6.

Theorem 2.5 ([28])**.** *Let* $f : S \to \mathbb{R}$ *be monotone where* $S \subseteq \mathbb{R}^k$ *is finite. Then* f *can be computed by a fan-in-two monotone real circuit of size* $O\big(\log_2(|S|)^{\,k-2}\big)$.

2.1 Secret Sharing Preliminaries

We recall the definition of generalized secret-sharing schemes, as stated in [6].

Definition 2.6. *An* n*-party* secret-sharing scheme*, with domain of secrets* S *such that* $\{0,1\} \subseteq S$*, 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.*

We say that $\mathcal{D}$ realizes a monotone function f over $\{0,1\}^n$ if for every $x \in \{0,1\}^n$ and every pair of secrets $s, s' \in S$ the random variables $(s_1, \ldots, s_n)$ obtained by invoking $\mathcal{D}$ on s, and the random variables $(s'_1, \ldots, s'_n)$ obtained by invoking $\mathcal{D}$ on s' satisfy the following properties:

Correctness. *If $f(x) = 1$ then the random variables $s_x = (s_i)_{i:x_i=1}$ and $s'_x = (s'_i)_{i:x_i=1}$ have disjoint supports, that is, one can recover the secret from the shares s_x.*

Privacy. *If $f(x) = 0$ then the random variable s_x is identically distributed to the random variable s'_x, that is, the shares s_x do not disclose any information on the secret.*

The secret size *in a secret-sharing scheme $\mathcal{D}$ is defined as $\log |S|$, the* share size *of the scheme $\mathcal{D}$ is defined as the size of the largest share, i.e., $\max_{1 \le i \le n} \{\log |S_i|\}$, and the total share size is defined as the sum of the share sizes, i.e., $\sum_{1 \le i \le n} \log |S_i|$. The* maximal information ratio *(resp.,* total information ratio*) of the scheme is defined as the ratio between the maximal share size (resp., total share size) and the secret size.*

A scheme $\mathcal{D}$ is a linear secret-sharing scheme over a finite field $\mathbb{F}$ *if $S = \mathbb{F}$, $R = \mathbb{F}^\ell$ for some integer $\ell \ge 1$, the sets $S_1, \ldots, S_n$ are vector spaces over $\mathbb{F}$, and the function $\mathcal{D} : \mathbb{F}^{\ell+1} \to S_1 \times \cdots \times S_n$ is a linear mapping over $\mathbb{F}$.*

The following composition lemma originates in the work of Benaloh and Leichter [13]. The composed scheme shares the secret with the scheme of the root gate, and then recursively shares every share for the ith input of a gate using the subformula rooted at its ith child. A share that is longer than the secrets of the child's scheme is shared block by block, by parallel invocations of that scheme. When the schemes of a gate and of its child work over different domains, e.g., one produces vectors over a field and the other takes bit strings as secrets, the share is first mapped by an arbitrary injective encoding into the secret domain of the child's scheme. The share of a party is the concatenation of the shares produced at the leaves labeled by its variable.

Lemma 2.7 (Formula composition; see, e.g., [6])**.** *Suppose that a function $f : \{0,1\}^n \to \{0,1\}$ can be implemented by a formula F over some collection of monotone gates G, and assume that every gate $g \in G$ can be realized by a secret-sharing scheme whose maximal share size is α_g. Then f can be realized by a secret-sharing scheme whose maximal share size is at most α_F, where α_F is defined as follows.*

- *The weight $\alpha_F(v)$ of a leaf v in F is the product $\prod_i \alpha_{g_i}$, where g_i is the ith gate in the unique path from the root to v.*
- *For each party $j \in [n]$, let $\alpha_F(j)$ be the sum of $\alpha_F(v)$ over all leaves v labeled by x_j, and define*

$$\alpha_F := \max_{j \in [n]} \alpha_F(j).$$

Similarly, if every gate g can be realized by a secret-sharing scheme whose maximal information ratio is α_g, then f can be realized with maximal information ratio at most α_F. If all the schemes are multilinear over the same field, then the composed scheme is multilinear over that field as well.

We will also use the following lower bound on the information ratio of the composed scheme, which holds for every choice of the encodings above, as it only uses that an injective encoding does not shorten the share. An input of a gate g is *essential* if g depends on it.

Lemma 2.8 (Leaves of a formula composition)**.** *In the setting of Lemma 2.7, assume that every input of every gate of F is essential. For a gate g and an input i of g, let $r_{g,i}$ be the ratio between the share size of input i and the secret size in the scheme realizing g. Then $r_{g,i} \geq 1$, and the maximal information ratio of the composed scheme is at least*

$$\max_{j \in [n]} \sum_{v \text{ labeled by } x_j} \prod_{(g,i) \in \mathrm{path}(v)} r_{g,i},$$

where $\mathrm{path}(v)$ is the set of pairs (g, i) such that the path from the root to the leaf v passes through the gate g and continues through its ith input. In particular, the maximal information ratio is at least the maximal number of leaves of F labeled by a single variable.

Proof. In the composed scheme, the secret shared by the scheme of a gate g is a share produced by the scheme of its parent gate (or the original secret, if g is the root), possibly encoded injectively and split into blocks that are shared in parallel. Thus, if a gate g shares a secret of t bits, then its ith child shares a secret of at least $r_{g,i}t$ bits, and by induction along the path from the root to a leaf v, the length of the share produced at v is at least $\prod_{(g,i) \in \mathrm{path}(v)} r_{g,i}$ times the secret size. Since the share of party j is the concatenation of the shares produced at the leaves labeled by x_j, the claimed bound follows. Finally, in a perfect secret-sharing scheme, the share of every essential input is at least as long as the secret [33]. Hence $r_{g,i} \geq 1$ for every gate g and input i, which proves the last statement. □

For more information on secret-sharing schemes, one can refer to, e.g., [9].

2.2 Communication Preliminaries

Definition 2.9 (Karchmer-Wigderson games [31])**.** *Let $F : \{0,1\}^n \to \{0,1\}$ be a Boolean function. In the* Karchmer-Wigderson game *for F, Alice receives an input $x \in F^{-1}(1)$ and Bob receives an input $y \in F^{-1}(0)$. After exchanging bits, they must output an index $i \in [n]$ such that $x_i \neq y_i$.*

When F is monotone, the monotone Karchmer-Wigderson game *has the same inputs but requires an index i such that $x_i = 1$ and $y_i = 0$. Such an index exists by monotonicity: otherwise $x \leq y$ would imply $F(x) \leq F(y)$, a contradiction. In either game, the communication cost of a protocol is the maximum number of bits exchanged on any valid input pair (x, y).*

It is proved in [31] that the minimum depth of a fan-in-two Boolean formula computing F is exactly the deterministic communication complexity of its KW game, and that for monotone F, the minimum depth of a monotone formula is likewise exactly the communication complexity of the monotone KW game. The randomized monotone game defined next will serve a similar purpose for FOMS: we bound the communication of a protocol obtained from a FOMS and then use a communication lower bound to derive a formula-size lower bound.

Definition 2.10 (Randomized monotone KW game)**.** *In the randomized public-coin monotone KW game, Alice and Bob have the same inputs and goal as in the monotone game above, but may use random coins. A public-coin protocol gives both players access to the same random string. It has error at most δ if for every valid input pair (x, y), it outputs an index i satisfying $x_i = 1$ and $y_i = 0$ with probability at least $1 - \delta$. Its communication cost is the maximum number of bits exchanged over all valid inputs and all outcomes of the public randomness. We say that F has a* randomized KW protocol with communication t *if its monotone KW game has a public-coin protocol of cost at most t and error at most $1/3$.*

Theorem 2.11 (Göös-Pitassi [26])**.** *There is a function F^* over n bits in monotone* NP *whose randomized monotone KW game has communication $\Omega(n/\log n)$.*

3 Randomized Karchmer-Wigderson Protocols for Formulas over Multislices

In this section, we prove the FOMS_w lower bound of Theorem 1.1. The proof is organized top-down. In Section 3.1, we state the main technical result of the section: a randomized Boolean protocol of cost $O((w+1)\log S \log n)$ for the monotone KW game of any function computed by a FOMS_w formula of size S. We then derive Theorem 1.1 by comparing it with the Göös-Pitassi bound (Theorem 2.11). In Section 3.2, we discuss the comparison and equality steps from which the protocol is assembled and their randomized implementation. In Section 3.3, we prove the random-permutation trial and weighted-search lemmas, and use them to construct a protocol for a single multislice gate. We then combine these single-gate protocols along a formula to prove Theorem 3.1. In Appendix A, we provide, for completeness, a proof of the Gilbert-Moore theorem used in the search lemma.

3.1 The Protocol Theorem and the Proof of the Lower Bound

The next theorem describes the protocol for formulas over multislices.

Theorem 3.1 (Randomized KW protocol for a FOMS_w formula)**.** *Let Φ be a FOMS_w formula over n variables of size S, computing a monotone function $F : \{0,1\}^n \to \{0,1\}$. Then F has a randomized KW protocol with communication $O((w+1)\log S \log n)$.*

For the rest of the proof, we assume that $w \le n$ and $S \le 2^n$. These restrictions are harmless: otherwise the stated bound exceeds the $O(n)$ cost of the trivial protocol in which Alice sends her input. We first show how the theorem implies the lower bound on the size of FOMS_w for the explicit function of [26].

Proof of Theorem 1.1. Let F_n be the explicit monotone function of Theorem 2.11, whose randomized monotone KW complexity is $\Omega(n/\log n)$, and let Φ be any FOMS_w formula of size S computing F_n. Then, by Theorem 3.1, the monotone KW game of F_n has a bounded-error public-coin protocol with communication complexity $O((w+1)\log S \log n)$. Comparing with Theorem 2.11,

$$(w+1)\,\log S\,\log n \;\ge\; \Omega\!\left(\frac{n}{\log n}\right), \qquad \text{i.e.,} \qquad \log S \;\ge\; \Omega\!\left(\frac{n}{(w+1)\log^2 n}\right). \qquad \square$$

3.2 Comparison and Equality Steps

The protocol of Theorem 3.1 is assembled from two types of primitive steps. In a *comparison step*, Alice computes a real value $A(x)$, Bob computes a real value $B(y)$. The continuation depends only on whether $A(x) > B(y)$. In an *equality step*, Alice computes a binary string a, Bob computes a binary string b of the same length, and the continuation depends only on whether $a = b$. The following standard lemmas show that each comparison step (resp., equality step) can be implemented by a randomized protocol with logarithmic (resp., $O(1)$) communication.

Lemma 3.2 (Implementing one comparison step)**.** *Let $\mathcal{X}, \mathcal{Y}$ be sets of size at most 2^n each, and let $A : \mathcal{X} \to \mathbb{R}$ and $B : \mathcal{Y} \to \mathbb{R}$ be functions known to both players, where Alice holds $x \in \mathcal{X}$ and Bob holds $y \in \mathcal{Y}$. For every $0 < \delta < 1/2$, there is a public-coin Boolean protocol that decides whether $A(x) > B(y)$ with error at most δ and communication $O\big(\log n + \log(1/\delta)\big)$.*

Proof. Replace every value in $A(\mathcal{X}) \cup B(\mathcal{Y})$ by its rank in the sorted union of the two images; both players compute the same ranking, since A and B are known to both, and the replacement preserves the predicate $A(x) > B(y)$. Since the two images together contain at most 2^{n+1} values, the ranks are integers of at most $n+1$ bits. The public-coin randomized greater-than protocol [47, 54] compares them with error at most δ using $O(\log n + \log(1/\delta))$ bits of communication. □

Lemma 3.3 (Implementing one equality step)**.** *Let Alice hold $a \in \{0,1\}^r$ and Bob hold $b \in \{0,1\}^r$. For every $0 < \delta < 1/2$, there is a public-coin Boolean protocol that decides whether $a = b$ with error at most δ and communication $O(\log(1/\delta))$.*

Proof. This is the standard public-coin equality protocol; see [41]. □

Conditioned on the public randomness and on the outcomes of the previous steps, the functions used at the next step are fixed and known to both players. Thus, the appropriate lemma above applies to each step separately. The proof of Theorem 3.1 accordingly analyzes the procedure with every comparison and equality step answered correctly, and then accounts for the implementation errors by a union bound over all steps.

3.3 A KW Protocol For a Single Multislices Gate

The protocol of Theorem 3.1 solves the KW game of a single multislice gate by two steps we call the *trial* and *search* steps, stated as the two lemmas below. We consider a multislice function f and assume that Alice holds an input a of weight at least k such that $f(a) = 1$ and that Bob holds an input b of weight at most $w + k$ such that $f(b) = 0$. The goal of Alice and Bob is to find an index i such that $a_i = 1$ and $b_i = 0$. For the rest of this section, we consider a permutation π of $[m]$, viewed as listing the coordinates in the order $\pi(1), \ldots, \pi(m)$, and for $z = (z_1, \ldots z_m)$, define the weighted sum

$$\eta_\pi(z) \;=\; \sum_{j \in [m]} 2^{\,m-j} z_{\pi(j)},$$

which is the standard integer represented in binary by $z_{\pi(1)} \cdots z_{\pi(m)}$. We write $\eta(z)$ for $\eta_{id}(z)$, i.e., for the identity permutation. Each player can compute this value on its own input, for any public π. Let j^*_π be the smallest position such that $a_{\pi(j)} \neq b_{\pi(j)}$, and call it the *$\pi$-earliest difference* of a and b.

Protocol intuition. A trial compares the two weighted sums $\eta_\pi(a)$ and $\eta_\pi(b)$. We say that such a comparison succeeds if $\eta_\pi(a) > \eta_\pi(b)$, i.e., if and only if $a_{\pi(j^*)} > b_{\pi(j^*)}$ at the π-earliest difference. For a KW instance of a width-w multislice gate, the number of 1's in b exceeds that of a by at most w, so a uniformly random permutation succeeds with probability at least $1/(w+2)$. Once a trial succeeds, the players locate the first difference under that permutation using a binary search. A balanced binary search would cost $O(\log m)$ comparisons, which would make the communication complexity of the protocol too high to prove Theorem 1.1. Instead, the protocol relies on Gilbert-Moore alphabetic trees, whose weight-dependent leaf depths allow the search costs at consecutive gates on a root-to-leaf path to telescope.

Lemma 3.4 (Random-permutation trial)**.** *Let g be a $(k : k + w, m)$-multislice gate, let $a \in g^{-1}(1)$ and $b \in g^{-1}(0)$. Consider the following trial: choose a permutation $\pi : [m] \to [m]$ with uniform distribution. A trial succeeds if $\eta_\pi(a) > \eta_\pi(b)$. Then, the trial succeeds with probability at least $\frac{1}{w+2}$; if, moreover, $b \leq a$ coordinatewise, it succeeds with probability* 1.

Proof. For every position j, it holds that $2^{m-j} > \sum_{j'>j} 2^{m-j'}$. Thus, if j^* is the earliest difference of a and b, then in

$$\eta_\pi(a) - \eta_\pi(b) = \sum_{j\in[m]} 2^{m-j}\big(a_{\pi(j)} - b_{\pi(j)}\big),$$

the term at j^* dominates in absolute value the sum of all later terms. Consequently, the sign of the difference is determined by the π-earliest difference.

For the success probability, write $A = \{i : a_i = 1\}$, $B = \{i : b_i = 1\}$, $D = A \setminus B$, and $E = B \setminus A$. Since g is monotone and $g(a) = 1 > 0 = g(b)$, we have $a \not\leq b$, so $D \neq \emptyset$; write $s = |D| \geq 1$. Since g is a $(k : k + w, m)$-multislice, $g(a) = 1$ forces $\mathrm{wt}(a) \geq k$ and $g(b) = 0$ forces $\mathrm{wt}(b) \leq k + w$; therefore $|B| - |A| \leq w$ and $|E| = |B| - |A \cap B| = (|B| - |A|) + s \leq s + w$.

The coordinates at which a and b differ are exactly those of $A \triangle B = D \dot\cup E$, where $\triangle$ denotes symmetric difference and $\dot\cup$ denotes disjoint union. Moreover, a exceeds b precisely at the coordinates of D, and therefore, the trial succeeds if and only if the π-earliest difference lies in D. If $b \leq a$, then $E = \emptyset$ and $A \triangle B = D$, so the trial succeeds for every π. For a uniformly random π, each of the $|D| + |E|$ elements of $D \cup E$ is equally likely to appear first among $D \cup E$ in the permutation π; hence, this first element lies in D with probability $|D|/(|D| + |E|)$. Therefore,

$$\Pr_\pi\big[\eta_\pi(a) > \eta_\pi(b)\big] = \frac{|D|}{|D| + |E|} \;\geq\; \frac{s}{2s + w} \;\geq \frac{s}{2s + ws} \;= \frac{1}{w + 2}.$$

□

The weighted-search protocol relies on the following classical theorem of Gilbert and Moore.

Theorem 3.5 (Gilbert-Moore alphabetic trees [25])**.** *Let $p_1, \ldots, p_m > 0$ with $\sum_{i\in[m]} p_i = 1$. There is a rooted binary tree determined by these probabilities alone, in which every internal node has exactly two children, whose leaves are $1, \ldots, m$ in this left-to-right order, and in which the depth of leaf i is at most $\lceil \log_2(1/p_i)\rceil + 1$ for every $i \in [m]$.*

For completeness, the proof of Theorem 3.5 is given in Appendix A.

Lemma 3.6 (Weighted search for the earliest difference)**.** *Let $\mu_1, \ldots, \mu_m > 0$, with $\mu = \sum_{i\in[m]} \mu_i$, be public weights. There is a deterministic procedure for two players holding $a', b' \in \{0, 1\}^m$, whose every step is an equality step, with the following two properties. For* every *set of the equality outcomes, the procedure halts at some coordinate $I \in [m]$ within $\log_2(\mu/\mu_I) + 2$ equality steps. If every equality step is answered correctly and $\eta(a') = \sum 2^{m-j} a'_j > \eta(b') = \sum 2^{m-j} b'_j$, then I is the least index such that $a'_j \neq b'_j$.*

Proof. The procedure. Both players construct the tree T of Theorem 3.5 for the probabilities $p_r = \mu_r/\mu$, $r \in [m]$, whose leaf r represents position r. Identify every node of T with the interval J of leaves below it. Denote by a'_J (resp., b'_J) the string a' (resp., b') restricted to this interval. Since the leaves appear in order, this set is an interval of positions, the root's interval is $[m]$, and a node splits its interval into the left interval of its left child and the right interval of its right child. The players walk the tree T from the root towards a leaf: at a node whose left child has the interval J_L, they check if $a'_{J_L} = b'_{J_L}$ (one equality step), move to the right child if they are equal and to the left otherwise. Upon reaching a leaf r, they halt with the output r.

A bound on the number of steps. One equality step is performed per node on the walk. By Theorem 3.5, if the walk ends at leaf r, their number is at most $\lceil \log_2(1/p_r)\rceil + 1 \leq \log_2(\mu/\mu_r) + 2$, where r is the output produced. This is a bound over every leaf of T, so it holds for every set of the equality outcomes.

Correctness. Suppose that every equality step is answered correctly. We claim that in each step, we are at a node that contains a leaf index j such that $a'_j \neq b'_j$.

The claim holds at the root. At a node, write $J = J_L \dot\cup J_R$, with every position of J_L preceding every position of J_R. If $a'_{J_L} \neq b'_{J_L}$, then J_L contains the least j, so the walk correctly moves left. Otherwise, $a'_{J_L} = b'_{J_L}$, and the least j must be in the right child. The invariant is therefore preserved. At the leaf r reached, the interval is $\{r\}$, so the output $I = r$ is the least j of a' and b'. □

Repeating trials with fresh permutations until one succeeds and then searching yields a self-contained procedure based on comparisons and equality steps. The following theorem summarizes the properties of an ideal version of this protocol where these steps do not err. The proof of Theorem 3.1 will instead meter the trials globally along the formula, and then will take errors into consideration.

Theorem 3.7 (A single-gate KW protocol)**.** *Let g be a $(k : k+w, m)$-multislice gate and let $\mu_1, \ldots, \mu_m > 0$ be weights with sum μ. Consider the procedure that performs trials with fresh uniformly random permutations until one succeeds, and then runs the search of Lemma 3.6 with $a' = (a_{\pi(1)}, \ldots a_{\pi(m)})$ and $b' = (b_{\pi(1)}, \ldots b_{\pi(m)})$ and the successful permutation π. With correct answers, on every (a, b) with $g(a) = 1$ and $g(b) = 0$, this procedure halts with probability 1, outputs a coordinate I with $a_I = 1 > 0 = b_I$, and performs an expected $w + 2$ comparison steps and $\log_2\big(\mu/\min_{i\in[m]} \mu_i\big) + 2$ equality steps. In particular, with uniform weights $\mu_i = 1$ for all $1 \leq i \leq m$, the number of equality steps is $O(\log m)$.*

Proof. The permutations are independent and uniform, so, by Lemma 3.4, each trial succeeds with probability at least $\frac{1}{w+2}$, independently of the preceding ones. Hence the probability that none of the first t trials succeeds is at most $\left(1 - \frac{1}{w+2}\right)^t$, which tends to 0 as t grows; the procedure therefore halts with probability 1, and the number G of trials satisfies $\mathbb{E}[G] \leq w + 2$. Upon success, $\eta_\pi(a) > \eta_\pi(b)$ for the successful permutation π, so, by Lemma 3.6, the search adds at most $\log_2(\mu/\mu_I) + 2 \leq \log_2\big(\mu/\min_{i\in[m]} \mu_i\big) + 2$ equality steps and outputs the π-earliest difference I, which satisfies $a_I \neq b_I$, since $\eta_\pi(a) > \eta_\pi(b)$ by Lemma 3.4. Summing the two bounds gives the expected total. □

3.4 A KW Protocol For a Formula Over Multislice Gates

We have presented a protocol for a single gate, in which the logarithmic cost of communication of the weighted search step is charged to the child selected by the protocol. In the proof below, each child's weight is the size of its subformula, so this charge equals the logarithmic decrease in subformula size and therefore telescopes along a root-to-leaf path. We state the following standard balancing lemma (whose proof can be found in [5]), that will allow us to work over a balanced formula, and then move on to the proof of Theorem 3.1.

Lemma 3.8 (Balancing monotone formulas over unbounded gates [5, Lemma 6.1])**.** *Let Φ be a monotone formula of size S whose gates may have unbounded fan-in. There is an equivalent monotone formula Φ' of depth $O(\log S)$ and size $S^{O(1)}$, whose gates are gates of Φ together with AND and OR gates.*

Proof of Theorem 3.1. We may assume $S \geq 2$: otherwise F is a variable x_i, and the game can be solved with no communication. By Lemma 3.8, there is a formula Φ' computing F, of size $S' = S^{O(1)}$ and depth $D = O(\log S)$, whose gates are gates of Φ, possibly with some of their inputs fixed to constants, together with AND and OR gates; all of these are multislice gates of width at most w, since fixing inputs of a multislice gate to constants does not increase its width. For a node v of Φ' and an input $z \in \{0, 1\}^n$, write

$v(z) \in \{0,1\}$ for the value computed at v on z, and write $W_v = \text{size}(\Phi'_v)$ for the size of the subformula rooted at v.

The rest of the proof has two parts. First, we construct an intermediate protocol Π in which each real-valued comparison and equality-test is a unit-cost step that returns a correct answer, and bound its number of steps and its abort probability. Second, we replace each comparison/equality with randomized Boolean greater-than/equality protocols and bound the resulting error and communication.

Part I: The correct-answers protocol. Fix the global trial budget $R = \lceil 12(w+2)(D+1) \rceil$. Whenever a trial is performed at a gate of arity m, the players use fresh public randomness to sample an independent uniformly random permutation over $[m]$. Alice holds $x \in F^{-1}(1)$ and Bob holds $y \in F^{-1}(0)$. The players maintain a node v of Φ', initially the root, aiming to preserve the invariant

$$v(x) = 1 > 0 = v(y), \tag{2}$$

which the root satisfies because Φ' computes F. If v is a leaf, labelled by a variable x_i, the protocol halts with output i. Otherwise, let g be the gate at v, let m denote its fan-in (fan-ins may differ between nodes) and let $u_1, \ldots, u_m$ be its children; Alice privately computes $a = (u_1(x), \ldots, u_m(x))$ and Bob privately computes $b = (u_1(y), \ldots, u_m(y))$, and by (2) the pair (a, b) satisfies $g(a) = 1 > 0 = g(b)$. The players repeatedly perform trials (Lemma 3.4) on (a, b) with fresh random permutations of arity m, and every trial at every node consumes one unit of the budget R. If a trial is requested when the budget R on the path from the root to v is exhausted, Π aborts with an arbitrary output. When a trial succeeds, the players run the search of Lemma 3.6 with the successful permutation and the weights $\mu_i = W_{u_i}$, obtain a coordinate I, set $v \leftarrow u_I$, and repeat from the new node. Every search moves to a child, so the walk visits at most $D+1$ nodes, and, since every trial consumes budget, Π always terminates (either by reaching a leaf or exhausting the budget).

We first bound the number of comparison and equality steps. Every run of Π performs at most $R = O\big((w+1)\log S\big)$ comparison steps as set before (since $D = O(\log S)$), and we claim that the number of equality steps is $\log_2 S' + 2(D+1) = O\big(\log S\big)$ for every fixing of the randomness and of the comparison and equality outcomes. This worst-case formulation will be important in Part II: a simulated step may err and send the walk down a different branch, but the bound still applies to the resulting sequence of step outcomes. By Lemma 3.6 whose step bound holds for every fixing of the equality outcomes, the search at a node v that halts at a coordinate I performs at most $\log_2(W_v/W_{u_I}) + 2$ equality steps, and the walk then moves to u_I. Hence, if $v_0, v_1, \ldots, v_r$ (with $r \le D$) are the successive nodes visited, the search steps total at most

$$\sum_{j<r} \left(\log_2 \frac{W_{v_j}}{W_{v_{j+1}}} + 2 \right) \;\le\; \log_2 S' + 2(D+1),$$

by telescoping. Since $D = O(\log S)$, $\log_2 S' = O(\log S)$, $\log S \ge 1$, the claimed cost follows.

We next prove that Π is correct unless it exhausts its trial budget, and that this event has small probability. Since comparison and equality steps are correct in this first-stage protocol, induction using Lemmas 3.4 and 3.6 shows that the invariant (2) holds at every node visited: at a node satisfying it, the pair (a, b) satisfies $g(a) = 1 > 0 = g(b)$; a successful trial certifies that the π-earliest difference I of a and b under the permutation π satisfies $a_I = 1 > 0 = b_I$ (Lemma 3.4), the search outputs exactly this coordinate (Lemma 3.6), and hence $u_I(x) = a_I = 1 > 0 = b_I = u_I(y)$. In particular, if Π does not abort, it halts at a leaf, labelled by a variable x_i with $x_i = 1 > 0 = y_i$, which is a valid output for the monotone KW game of F.

To bound the abort probability, consider the same correct-answers process without a trial budget. For $t \in [D]$, let G_t be the random variable counting the trials performed at the tth internal node visited, and

set $G_t = 0$ if the walk visits fewer than t internal nodes. Conditioned on the history when the tth internal node is reached, each fresh permutation succeeds with probability at least $1/(w+2)$ by Lemma 3.4; hence $\mathbb{E}[G_t \mid \text{the node is reached}] \le w+2$. Since the walk visits at most D internal nodes, the total number $N = \sum_{t=1}^{D} G_t$ of trials in the unbudgeted process satisfies $\mathbb{E}[N] \le D(w+2)$. The budgeted protocol aborts only if $N > R$, so Markov's inequality gives

$$\Pr[\Pi \text{ aborts}] \;\le\; \frac{\mathbb{E}[N]}{R} \;\le\; \frac{D(w+2)}{12(w+2)(D+1)} \;<\; \frac{1}{12}.$$

Part II: The Boolean communication protocol. Let Q denote the sum of the bounds on the number of comparison and equality steps discussed in part I, that satisfies $Q = O((w+1)\log S)$. Then, let $\widetilde{\Pi}$ be the Boolean protocol that runs Π, replacing each comparison step by the protocol of Lemma 3.2 and each equality step by the protocol of Lemma 3.3, both with error parameter $\delta := 1/(12Q)$. Conditioned on the public randomness and on the transcript so far, every comparison step applies fixed functions known to both players to domains of size at most 2^n, while every equality step compares two binary strings, so the corresponding lemmas apply. By the step bound from Part I, every run of $\widetilde{\Pi}$ performs at most Q simulated steps. Run $\widetilde{\Pi}$ and an ideal execution of Π side by side, using the same public random choices for the two executions; the ideal execution answers each step exactly. As long as all simulated steps have been correct, the two executions have the same transcript, so they reach the same next step. Let $\mathcal{E}$ be the event that at some such step the simulated answer differs from the exact answer. Since at most Q simulations are performed and each errs with probability at most δ conditioned on the past, $\Pr[\mathcal{E}] \le Q\delta = 1/12$. If $\mathcal{E}$ does not occur, induction over the steps shows that the two complete transcripts, and hence their outputs, are identical. Therefore, the output of $\widetilde{\Pi}$ is a valid KW index unless $\mathcal{E}$ occurs or Π aborts, an error probability of at most $1/12 + 1/12 = 1/6$.

It remains to bound the communication. Each of the $Q = O((w+1)\log S)$ simulations costs $O(\log n + \log(1/\delta)) = O(\log n + \log Q) = O\big(\log n + \log(w+1) + \log\log(S+2)\big)$ bits. If $w \le n$ and $S \le 2^n$, then each simulation costs $O(\log n)$ bits, and the total communication is $O((w+1)\log S \log n)$ as desired. □

Remark 3.9 (Randomized KW complexity of a single gate)**.** *The single-gate procedure above is nearly optimal in its dependence on the width. With uniform weights, Theorem 3.7 gives a correct-answers procedure using at most $C := w + 4 + \log_2 m$ comparison and equality steps in expectation. If we truncate it after $Q = \lceil 12C \rceil$ steps and simulate each comparison step using Lemma 3.2 and each equality step using Lemma 3.3, both with error $1/(24Q)$, Markov's inequality and a union bound give constant total error, while each simulation costs $O(\log m + \log Q) = O(\log m)$ bits. Therefore, every width-w multislice over m bits has a randomized KW protocol with communication $O((w + \log m)\log m)$.*

On the other hand, Theorem 2.11 gives a monotone function on w bits whose randomized KW complexity is $\Omega(w/\log w)$. This function is itself a $(0 : w, w)$-multislice. Hence, for $m = w$, the upper and lower bounds are $O(w \log w)$ and $\Omega(w/\log w)$, respectively.

4 Realizing Multislices with Slices

In this section, we show that a width-w multislice on n wires reduces, by a simple recursion, to a depth-$(w+1)$ formula over *slice* gates, and hence to small monotone real formulas and circuits. For long secrets, the FOS construction yields an amortized multilinear secret-sharing scheme with polynomial information ratio for every fixed-width multislice access structure (see Section 4.1). Throughout the section, we identify sets with their supports, writing $f(X)$ for $f(\mathbf{1}_X)$ when $X \subseteq [n]$.

Theorem 4.1. *Let f be a $(k : k+w, n)$-multislice function. Then f can be computed in the following computational models:*

(i) A formula over slice gates of depth $w+1$ with at most $\sum_{d=0}^{\min\{w+1,n\}} \frac{n!}{(n-d)!} \le 2n^{w+1}$ gates.

(ii) A monotone real formula of size $2^{O(w)}n^{w+1}$.

(iii) A monotone real circuit of size $O\big(n\sum_{d=0}^{w}\binom{n-1}{d}\big)$.

Note that when $w = \alpha n$ for a constant $0 < \alpha < 1/2$, the standard entropy estimate on the MRC bound gives $n\sum_{d=0}^{\alpha n}\binom{n-1}{d} = 2^{\mathrm{H}_2(\alpha)n+o(n)}$, where H_2 is the binary entropy. Additionally, this bound is subexponential for every $w = o(n)$, since in this case $\sum_{d=0}^{w}\binom{n-1}{d} \le \left(\frac{e(n-1)}{w}\right)^w$, and the circuit size is therefore $O\left(n\left(\frac{en}{w}\right)^w\right) = 2^{O(w\log(n/w))}$. The formula bound in (ii) is subexponential only for $w = o(n/\log n)$.

The proof rests on a simple recursive observation: after restricting away one coordinate, forcing the old top free layer to 1 reduces the width by one, while a single slice gate restores the information lost from that top layer. The following lemma formalizes this step.

Lemma 4.2 (Peeling one layer)**.** *Let f be a $(k : k+w, n)$-multislice with $w \ge 1$ and $k+w < n$. For $i \in [n]$, define f_i on the subsets of $[n]\setminus\{i\}$, where for $Y \subseteq [n]\setminus\{i\}$*

$$f_i(Y) = \begin{cases} 0, & |Y| < k, \\ f(Y), & k \le |Y| \le k+w-1, \\ 1, & |Y| > k+w-1. \end{cases}$$

That is, f_i is defined only on subsets not containing i, $f_i(Y) = f(Y)$ when $|Y| \ne k+w$ and $f_i(Y) = 1$ when $|Y| = k+w$. Therefore, f_i is a $(k : k+w-1,\, n-1)$-multislice. Setting $z_i = f_i(X\setminus\{i\})$ for $i \in [n]$, there is an $(n-k-w, n)$-slice function G that satisfies $f(X) = G(z_1,\dots,z_n)$ for every $X \subseteq [n]$. Writing $Z = \{i \in [n] : z_i = 1\}$, the function G equals $f([n]\setminus Z)$ when $\mathrm{wt}(z) = n-k-w$, 0 when $\mathrm{wt}(z) < n-k-w$, and 1 when $\mathrm{wt}(z) > n-k-w$.

Proof. In the proof, we assume $k+w < n$ and set $q := n-k-w \ge 1$. This is without loss of generality: if $k+w = n$, then either $f \equiv 0$, or by monotonicity $f([n]) = 1$, and f is also a $(k : k+w-1, n)$-multislice.

We first note that f_i is indeed a $(k : k+w-1, n-1)$-multislice. It vanishes below weight k, equals 1 above weight $k+w-1$, and is monotone since the restriction of f to subsets of $[n]\setminus\{i\}$ is monotone, and f_i is obtained from this restriction only by forcing the old top free layer $|Y| = k+w$ to equal 1, which preserves monotonicity.

Next, let G be the (q, n)-slice defined in the lemma. It remains to verify $G(z) = f(X)$ for every X. We consider cases according to $r = |X|$.

Case $r < k$. Here $f(X) = 0$. Also, $G(z) = 0$: every z_i vanishes since, for $i \in X$, the deleted set has size $|X\setminus\{i\}| = r-1 < k$, while for $i \notin X$, we have $z_i = f_i(X)$ with $|X| = r < k$. Hence $\mathrm{wt}(z) = 0$, which is strictly below q because $q \ge 1$.

Case $k \le r \le k+w-1$. Here, for $i \notin X$, the weight $|X|$ lies inside the free layers of f_i, so $z_i = f_i(X) = f(X)$. Suppose first that $f(X) = 0$. For $i \in X$, $z_i = f_i(X\setminus\{i\}) \le f(X) = 0$ by the monotonicity of f. Together with $z_i = f(X) = 0$ for $i \notin X$, this gives $\mathrm{wt}(z) = 0 < q$ and $G(z) = 0 = f(X)$. Suppose instead that $f(X) = 1$. Then $z_i = 1$ for all the $n-r$ coordinates $i \notin X$. Then $\mathrm{wt}(z) \ge n-r \ge n-(k+w-1) = q+1 > q$, so $G(z) = 1 = f(X)$.

Case $r = k + w$. This is the top free layer, and the only case in which $\mathrm{wt}(z)$ may equal q, the critical layer of the slice G. For $i \notin X$, the set $X = X \setminus \{i\}$ itself lies above the free layers of f_i, since $|X| = k + w > k + w - 1$, so $z_i = 1$. These coordinates contribute exactly $n - r = q$ ones to z. For $i \in X$, the deleted set has $|X \setminus \{i\}| = k + w - 1$, the top free layer of f_i, so altogether

$$\mathrm{wt}(z) \;=\; q + \big|\{i \in X : f(X \setminus \{i\}) = 1\}\big|.$$

If some immediate predecessor of X is accepted, then $f(X) = 1$ by monotonicity, and at the same time $\mathrm{wt}(z) > q$, so the threshold gives $G(z) = 1 = f(X)$. If $f_i(X \setminus \{i\}) = 0$ for every $i \in X$, then $z = \mathbf{1}_{[n]\setminus X}$ exactly, so $\mathrm{wt}(z) = q$, and by definition we have $G(z) = f\big([n] \setminus ([n] \setminus X)\big) = f(X)$.

Case $r > k + w$. Here $f(X) = 1$. For $i \in [n]$ we have $|X \setminus \{i\}| \geq |X| - 1 > k + w - 1$, so the upper slice cutoff gives $z_i = 1$. Hence $\mathrm{wt}(z) = n > q$, and $G(z) = 1 = f(X)$.

□

Proof of Theorem 4.1(i). We prove the theorem by induction on the width w. For $w = 0$, the function f is itself a slice gate, so together with its n input gates the formula has size $n + 1$. For $w \geq 1$, Lemma 4.2 expresses f as a slice gate G whose n inputs are the width-$(w-1)$ multislices $f_i(X \setminus \{i\})$ on $n-1$ variables. By induction, each f_i has a formula of depth w over slice gates. Placing these formulas below G gives a formula for f of depth $w + 1$.

To prove the size bound, denote by $S_w(n)$ the maximum number of gates required by this construction, then

$$S_0(n) = n + 1, \qquad S_w(n) \leq 1 + nS_{w-1}(n - 1).$$

Consequently,

$$\begin{aligned} S_w(n) &\leq 1 + n \sum_{d=0}^{w} \frac{(n-1)!}{(n-1-d)!} = 1 + \sum_{d=0}^{w} \frac{n!}{(n-1-d)!} = \sum_{d=0}^{w+1} \frac{n!}{(n-d)!} \\ &\leq \sum_{d=0}^{w+1} n^d = \frac{n^{w+2} - 1}{n - 1} \leq 2n^{w+1}. \end{aligned}$$

The first inequality follows by unfolding the recurrence, and observing that the number of nodes at depth d is at most $n(n-1)\cdots(n-d+1) = n!/(n-d)!$. For the last equality on the first line, reindex the sum by $e = d + 1$ and regard the leading 1 as the term $n!/n!$ corresponding to $e = 0$. The inequality that starts the second line holds because each of the summands can be seen, by the same identity, as d factors of size at most n. The following equality is the geometric-series formula, and the final inequality uses $n/(n-1) \leq 2$ for $n \geq 2$. This proves the claimed depth and size bounds. □

Proof of Theorem 4.1(ii). By Rosenbloom [49], every (t, n')-slice function g has a fan-in-two monotone real formula of size $O(n')$ that reads each input variable twice. More precisely, the formula has the form $g(x) = \psi\big(L(x), R(x)\big)$, where

$$L(x) = \sum_{i=1}^{n'} a_i x_i \qquad \text{and} \qquad R(x) = \sum_{i=1}^{n'} b_i x_i$$

for positive coefficients a_i, b_i and ψ that is a monotone real gate. Therefore, each input variable appears once in each linear form, which are both computed by an addition tree of size $O(n')$. In the tree recursion

used in part (i), a node at depth d is a residual function on $n-d$ variables. By Lemma 4.2, its slice gate has fan-in $n-d$, with one child for each remaining coordinate. Replacing this gate by Rosenbloom's formula costs $O(n-d)$ gates and reads each child twice. Since subformulas cannot be shared in a formula, every child subformula needs to be duplicated.

Let $R_w(n)$ be the size of the resulting monotone real formula. Then

$$R_0(n) = O(n), \qquad R_w(n) \le O(n) + 2nR_{w-1}(n-1).$$

Unrolling the recurrence, similarly to the FOS analysis, gives

$$R_w(n) \le O\left(\sum_{d=0}^{w} 2^d n^{d+1}\right) = 2^{O(w)} n^{w+1},$$

as claimed. □

Proof of Theorem 4.1(iii). The formula constructed in part (i) may contain many copies of the same residual multislice. For example, if we unfold two levels of the recursion, we can first omit index 1 and then 2, computing the function restricted to $[n] \setminus \{1,2\}$, and the same function is computed again by first omitting 2 and then 1. To exploit sharing in a circuit, we index each residual function by the *set* of coordinates deleted, rather than by their deletion order. For $I \subseteq [n]$ with $d := |I| \le w$, let $U_I := [n] \setminus I$ and define $f_I : 2^{U_I} \to \{0,1\}$ by

$$f_I(Y) := \begin{cases} 0, & |Y| < k, \\ f(Y), & k \le |Y| \le k+w-d, \\ 1, & |Y| > k+w-d. \end{cases}$$

Thus, $f_\emptyset = f$, and f_I is a $(k : k+w-d,\ n-d)$-multislice with $n-d$ remaining variables and width $w-d$.

Let $d < w$ and $i \in U_I$. Applying Lemma 4.2 to f_I and deleting coordinate i produces exactly $f_{I\cup\{i\}}$: its ground set is $U_I \setminus \{i\} = U_{I\cup\{i\}}$, and its upper free level is $k+w-d-1 = k+w-|I\cup\{i\}|$. Hence the lemma gives a $(q, n-d)$-slice gate G_I such that

$$f_I(X) = G_I\Big(\big(f_{I\cup\{i\}}(X\setminus\{i\})\big)_{i\in U_I}\Big) \qquad \text{for every } X \subseteq U_I. \tag{3}$$

The critical weight of the slice function G_I is independent of I:

$$|U_I| - \big(k + (w-d)\big) = (n-d) - k - (w-d) = n-k-w = q.$$

At level d there is one residual function for each d-element set I, hence $\binom{n}{d}$ distinct nodes. At level w, every f_I is an ordinary $(k, n-w)$-slice.

Use the shared recursion in (3). At level d, each of the $\binom{n}{d}$ nodes is a slice gate of fan-in $n-d$: for $d < w$ it is G_I, while for $d = w$ it is the terminal gate f_I.

Replace every such gate by the Rosenbloom monotone real formula used in part (ii), at cost $O(n-d)$ gates. Although that formula reads each input twice, in a circuit both reads may use the same child subcircuit. The total size is therefore

$$O\Big(\sum_{d=0}^{w}(n-d)\binom{n}{d}\Big) = O\Big(n\sum_{d=0}^{w}\binom{n-1}{d}\Big),$$

using $(n-d)\binom{n}{d} = n\binom{n-1}{d}$. □

4.1 Amortized Secret Sharing for Multislices

We describe a secret-sharing scheme for multislices based on the FOS construction from the previous subsection, existing schemes for slices, and standard composition techniques. Recall the following secret-sharing definitions. For a perfect secret-sharing scheme with secret space S and share spaces $S_1, \ldots, S_n$, denote

$$\rho_{\max} \;=\; \max_{i\in[n]} \frac{\log |S_i|}{\log |S|}.$$

A scheme is *multilinear* over a finite field $\mathbb{F}$ if the secret and the randomness are vectors over $\mathbb{F}$ and every share is a linear function of them. For an access structure f, we write $\rho_{ML}(f)$ for the corresponding asymptotic maximal ratio of multilinear schemes realizing f as the size of the domain of the secret grows. We will prove the following theorem using a standard composition lemma originating in the work of Benaloh and Leichter [13], stated in Lemma 2.7.

Theorem 4.3 (Amortized multilinear secret sharing for multislices)**.** *Let f be a $(k : k+w, n)$-multislice access structure with $1 \le k \le k+w < n$, and let $q = n-k-w$. For $d \ge 1$, define $\lambda(d, n') := \min\{d^2, n'\}$. Then there exists an absolute constant $C \ge 1$ such that, for multilinear schemes over a suitable finite field and for sufficiently long secrets,*

$$\rho_{ML}(f) \;\le\; C^{w+1} \cdot (n-1)(n-2)\cdots(n-w)\; \lambda(k, n-w) \prod_{r=0}^{w-1} \lambda(q, n-r). \tag{4}$$

Consequently, $\rho_{ML}(f) \le C^{w+1} n^{2w+1}$, and fixed-width multislices have polynomial asymptotic multilinear information ratio.

Proof. We apply Lemma 2.7 to the depth-$(w+1)$ peeling formula from Theorem 4.1(i). We first realize each slice gate by a multilinear secret-sharing scheme and then compute the weight contributed to each party by the leaves of the formula.

Realizing the gates. By [3, Cor. 4.8], for secrets of size at least $\Omega\big(d \log n' \cdot 2^{(n'+1)^{d^2}}\big)$, every (d, n')-slice function has a perfect multilinear scheme with $\rho_{\max} = O(d^2)$. Alternatively, combining the multilinear CDS protocol of Applebaum and Arkis [2] with the CDS-to-slices reduction of Beimel et al. [12] gives for secrets of size at least $\Omega\big(n'^2 \cdot 2^{2^{n'}}\big)$ an information ratio of $O(n')$. Thus, for an absolute constant C and sufficiently long secrets, we may use

$$\rho_{\max} \;\le\; C \cdot \lambda(d, n'), \qquad d \ge 1. \tag{5}$$

The case $d = 0$ is elementary: a $(0, n')$-slice is an OR gate or the degenerate constant-one gate, and has ratio 1. Since the peeling formula contains only finitely many parameter pairs, we may choose a common field and a sufficiently large secret dimension for all its gate schemes. Moreover, as stated in Lemma 2.7, the Benaloh-Leichter scheme for formulas preserves multilinearity.

Composing and computing the share size. A root-to-leaf path passes, at deletion depths $r = 0, \ldots, w-1$, through a $(q, n-r)$-slice gate and then through a terminal $(k, n-w)$-slice gate. Hence the weight of every leaf is at most

$$C^{w+1} \cdot \lambda(k, n-w) \prod_{r=0}^{w-1} \lambda(q, n-r).$$

A leaf is determined by an ordered sequence $(i_1, \ldots, i_w, j)$ of $w+1$ distinct parties: the first w entries are the deleted coordinates, and j labels the final input leaf. For a fixed party j, there are exactly $(n-1)(n-$

$2) \cdots (n-w)$ such sequences. The composition lemma therefore bounds the information ratio of every party by

$$C^{w+1} \cdot (n-1)(n-2)\cdots(n-w)\,\lambda(k,n-w)\prod_{r=0}^{w-1}\lambda(q,n-r),$$

which proves (4). Finally, using $(n-1)\cdots(n-w) \le n^w$ and $\lambda(d,n') \le n' \le n$ gives $\rho_{ML}(f) \le C^{w+1}n^{2w+1}$. □

For example, this allows to obtain the following corollary, which gives the bound needed for Corollary 1.4:

Corollary 4.4 (Width two)**.** *If f is a $(k : k+2, n)$-multislice access structure with $k \ge 1$ and $q = n-k-2 \ge 1$, then*

$$\rho_{ML}(f) \;\le\; C^3 \cdot (n-1)(n-2)\,\lambda(k,n-2)\,\lambda(q,n-1)\,\lambda(q,n) \;=\; O(n^5),$$

and the expression is smaller for very "low" or "high" multislices, e.g., $O(n^4)$ when k is constant and $O(n^3)$ when $n-k$ is constant.

We note that these ideas do not directly improve secret sharing for one-bit secrets. For example, for $(n/2-w/2 : n/2+w/2, n)$-multislices (that are multislices of width w centered around the "middle"), the constructions of [42] for one-bit secrets and constant w have share size $2^{\tilde{O}(\sqrt{n})}$, and this is also the share size implied by our construction if we plug into it the state of the art schemes for slices for one-bit secrets. Thus, for one-bit secrets, the gain of our construction is its relative simplicity.

5 Formulas over Standard and Fully Robust CDS Gates

In the first part, we prove the lower bound on formulas with CDS gates and linear gates, and derive from it a lower bound on the information ratio of formula-based secret-sharing schemes; in the second part, we prove matching bounds on formulas with fully-robust CDS gates.

5.1 Formulas over CDS gates and linear gates

We first consider formulas over CDS gates. Let the inputs of a gate g be partitioned into q_g parts $V_i = \{z_{i,a} : a \in [N_i]\}$ (corresponding to server inputs in the description of CDS as a protocol), and let $R_g \subseteq [N_1] \times \cdots \times [N_{q_g}]$ be a relation. By its definition in [24], a CDS gate computes a partial function: on inputs having exactly one active variable z_{i,a_i} in each part, it outputs 1 precisely when $(a_1,\ldots,a_{q_g}) \in R_g$. To use such gates inside ordinary Boolean formulas, we apply the standard transformation (used, e.g., in [42]), which extends this partial function to the total monotone function

$$g(z)=1 \quad\Longleftrightarrow\quad \begin{cases} z_{i,a}=z_{i,b}=1 & \text{for some } i \text{ and distinct } a,b\in[N_i],\\ \text{or} & \\ z_{i,a_i}=1 \text{ for every } i\in[q_g] & \text{for some } (a_1,\ldots,a_{q_g})\in R_g. \end{cases}$$

Thus, the extension agrees with the CDS predicate on its promised domain and additionally accepts whenever two inputs in the same part are active. As shown in the proof below, the transformation increases the information ratio by at most 2. A formula over CDS gates, denoted as *FCDS*, is a formula whose internal gates are these totalized CDS gates, with no restriction on their number of inputs or parts. Its size is its number of gates, including input gates.

We next define linear gates. Recall that a secret-sharing scheme over a finite field $\mathbb{F}$ is *linear* if the secret is an element of $\mathbb{F}$, the randomness is a vector over $\mathbb{F}$, and every share is a vector of $\mathbb{F}$-linear functions of the secret and the randomness (such schemes are equivalent to monotone span programs [32]). A monotone function g on m inputs is an *ideal $\mathbb{F}$-linear gate* if it is realized by a linear scheme over $\mathbb{F}$ in which every input receives a single field element. For example, AND and OR gates and, when $|\mathbb{F}| \geq m$, every threshold function on m inputs are ideal $\mathbb{F}$-linear gates. A formula over CDS gates and $\mathbb{F}$-linear gates, denoted as *FCDSL*$_{\mathbb{F}}$, is a formula whose internal gates are totalized CDS gates or ideal $\mathbb{F}$-linear gates. Its size is its number of gates, including input gates.

The balancing construction of Lemma 3.8 substitutes constants for some subformulas, so a gate of the balanced formula may be a gate of the original formula with some of its inputs fixed to constants. The next lemma, whose proof is deferred to Appendix B, shows that such gates remain within the model, up to OR gates.

Lemma 5.1 (Gates with constant inputs)**.** *Let g' be a non-constant function obtained from a gate g by fixing some of its inputs to constants. If g is an ideal $\mathbb{F}$-linear gate, then g' is an ideal $\mathbb{F}$-linear gate. If g is a totalized CDS gate, then $g' = h \vee \bigvee_{z \in Z} z$, where Z is a set of inputs of g' and h is a totalized CDS gate on the inputs of g' not in Z.*

Theorem 5.2 (Exponential lower bound for formulas over CDS gates and linear gates)**.** *There is an explicit monotone function $F_n : \{0,1\}^n \to \{0,1\}$ and a constant $c > 0$ such that, for every finite field $\mathbb{F}$, every FCDSL$_{\mathbb{F}}$ that computes F_n has size at least 2^{cn}. In particular, every FCDS that computes F_n has size at least 2^{cn}, even when the CDS gates may have arbitrarily many parts.*

Proof. Khazaei [34, Theorem 1.2] recently showed how to construct an explicit access structure F_n and a constant $\alpha > 0$ such that, for every finite field $\mathbb{F}$ and every sufficiently large n, every multilinear scheme realizing F_n over $\mathbb{F}$ has information ratio at least $2^{\alpha n}/n$; in particular, $\rho_{ML}(F_n) = 2^{\Omega(n)}$ over every finite field, with a constant independent of the field.

Fix a finite field $\mathbb{F}$ and suppose that an FCDSL$_{\mathbb{F}}$ Φ of size S computes F_n. By Lemma 3.8, Φ can be transformed into an equivalent formula Φ' of size $S' = S^{O(1)}$ and depth $D = O(\log S)$, whose gates are AND and OR gates and gates of Φ, possibly with some of their inputs fixed to constants. A gate that became constant can be eliminated by fixing the corresponding input of its parent, so by Lemma 5.1 every gate of Φ' is an AND gate, an OR gate, an ideal $\mathbb{F}$-linear gate, or the OR of a totalized CDS gate with some of its input variables. We stress that Lemma 5.1 is used only to describe the function computed by each gate of Φ'; the gates of Φ', and hence its size and depth, are unchanged.

We claim that every gate in Φ' has a multilinear secret-sharing scheme of information ratio at most 6 over $\mathbb{F}$. Indeed, Applebaum and Arkis [2] present a multilinear CDS protocol (corresponding to the partial-function version described above) with information ratio at most 4 for every number of parties and parts over any fixed finite field. To turn this protocol into a scheme for a totalized CDS gate g, we first share the secret as $s = s_0 + \cdots + s_{q_g}$ and give every input in part V_i the piece s_i, which increases the information ratio by 1. Then, we use the CDS protocol to disclose s_0 precisely on the hyperedges in R_g. A hyperedge in R_g then recovers all the pieces, whereas a set that misses a part or selects a hyperedge outside R_g learns nothing. Independently, within every part V_i, we use an ideal Shamir two-out-of-N_i threshold scheme for s over the same field. (If $N_i > |\mathbb{F}|$, we use a standard extension field $\mathbb{K}/\mathbb{F}$ with $|\mathbb{K}| \geq N_i$. Viewing $\mathbb{K}$ as a vector space over $\mathbb{F}$, the Shamir scheme remains $\mathbb{F}$-linear and ideal, since both its secret and each share consist of one element of $\mathbb{K}$.) This authorizes every pair in a part and increases the information ratio of every party by another 1. Thus every totalized CDS gate has a multilinear scheme of information ratio at most 6, and so does the OR of such a gate h with a set Z of its input variables: realize h on the inputs not in Z as

above and give the secret itself to every input in Z, which yields a multilinear scheme of information ratio at most 6 for the gate. Ideal $\mathbb{F}$-linear gates have linear schemes over $\mathbb{F}$ of information ratio 1 by definition, and so do binary AND and OR gates.

We then apply Lemma 2.7 to Φ'. Every root-to-leaf path contains at most D gates, so every leaf has weight at most 6^D. Since Φ' has at most S' leaves, the resulting multilinear scheme satisfies

$$\rho_{ML}(F_n) \leq S' 6^D = S^{O(1)}.$$

Combining this upper bound with $\rho_{ML}(F_n) = 2^{\Omega(n)}$ gives $S \geq 2^{cn}$ for a constant $c > 0$ that depends only on the constant α and on the constants in the balancing lemma, as required. □

Lower bounds for formula-based secret-sharing schemes. Theorem 5.2 also yields a lower bound on the *information ratio* of secret-sharing schemes obtained by composing, according to a formula as in Lemma 2.7, linear schemes over a common field and schemes for CDS gates. The schemes used for the CDS gates need not be linear or multilinear, and shares may be encoded arbitrarily when they are passed from a gate of one kind to a gate of the other; the only property of the composition that we use is Lemma 2.8.

Corollary 5.3. *Let F_n be the explicit access structure from Theorem 5.2. Every secret-sharing scheme realizing F_n by formula composition of linear schemes over a common finite field $\mathbb{F}$ and schemes for CDS gates has maximal information ratio $2^{\Omega(n)}$.*

Proof. Let Φ be a formula underlying such a construction, and let ρ be the maximal information ratio of the resulting scheme. Without loss of generality, every input of every gate is essential, and Φ has no unary gates: an inessential input can be deleted together with its subformula, while an essential unary monotone gate computes the identity and can be suppressed; neither operation changes the function computed or increases the information ratio of the composed scheme.

We use the standard fact that, for every set of inputs in a linear scheme, its shares either linearly reconstruct the secret or are distributed independently of the secret. Consider a linear gate g whose scheme gives a_i field elements to input i. Give each of these field elements to a separate new input. The resulting scheme realizes an ideal $\mathbb{F}$-linear gate $\widetilde{g}$ on $\sum_i a_i$ inputs, and

$$g(z_1, \ldots, z_m) = \widetilde{g}(\underbrace{z_1, \ldots, z_1}_{a_1}, \ldots, \underbrace{z_m, \ldots, z_m}_{a_m}). \tag{6}$$

We therefore replace g by $\widetilde{g}$ and attach a separate copy of the subformula of input i to each of the a_i corresponding inputs of $\widetilde{g}$. Applying this replacement to all linear gates gives an $\mathrm{FCDSL}_{\mathbb{F}}$ Φ' computing F_n, so by Theorem 5.2 the size of Φ' is at least $2^{\Omega(n)}$.

We next bound the size of Φ' in terms of ρ. For a leaf v of Φ, let $w(v)$ be the product of the numbers a_i on the linear-gate input wires along its root-to-leaf path. By construction, v gives rise to exactly $w(v)$ leaves in Φ'. Hence, if L_j denotes the number of leaves of Φ' labeled by x_j, then

$$L_j = \sum_{\substack{v \text{ a leaf of } \Phi \\ v \text{ labeled by } x_j}} w(v).$$

We now compare these leaf counts with the information ratio of the original composition. In the notation of Lemma 2.8, a linear gate g gives a_i field elements to input i while its secret is one field element, and

therefore $r_{g,i} = a_i$. Every CDS-gate input is essential, so its ratio satisfies $r_{g,i} \geq 1$. Consequently, for every leaf v of Φ,

$$\prod_{(g,i)\in\mathrm{path}(v)} r_{g,i} \geq \prod_{\substack{(g,i)\in\mathrm{path}(v)\\ g \text{ linear}}} a_i = w(v).$$

Applying Lemma 2.8 and then the identity defining L_j gives

$$\rho \geq \max_{j\in[n]} \sum_{\substack{v \text{ a leaf of } \Phi\\ v \text{ labeled by } x_j}} \prod_{(g,i)\in\mathrm{path}(v)} r_{g,i} \geq \max_{j\in[n]} \sum_{\substack{v \text{ a leaf of } \Phi\\ v \text{ labeled by } x_j}} w(v) = \max_{j\in[n]} L_j.$$

Every leaf of Φ' is labeled by some variable, and hence the number of leaves of Φ' is

$$n\rho \geq n \max_{j\in[n]} L_j \geq \sum_{j=1}^{n} L_j.$$

Since every internal gate of Φ' has fan-in at least two, a tree with L leaves has at most $L-1$ internal gates. Thus $|\Phi'| \leq 2L - 1 \leq 2n\rho$. Combining this with $|\Phi'| = 2^{\Omega(n)}$ yields $\rho \geq 2^{\Omega(n)}/(2n) = 2^{\Omega(n)}$. □

Remark 5.4. *Corollary 5.3 concerns linear schemes, in which the secret is a single field element. In a multilinear scheme with secret $(s_1, \ldots, s_\ell)$ that gives a_i field elements to input i, the shares determine the secret if and only if they determine every s_k, and treating the other coordinates of the secret as randomness they form a linear scheme with secret s_k; hence the gate is the AND of ℓ linear gates with the same shares. Replacing such a gate by ideal linear gates thus requires ℓa_i copies of the subformula of input i, whereas the per-input information ratio is only $r_{g,i} = a_i/\ell$. Hence, the argument above does not extend to multilinear gate schemes as is.*

5.2 Formulas over Fully Robust CDS Gates

In this section, we prove matching upper and lower bounds for the formulas over RCDS model. Before the proofs, we fix some notation for the model. Consider an internal gate g whose inputs are partitioned among $q_g \leq q$ parts. Let the block sizes be $N_{g,1}, \ldots, N_{g,q_g}$, and write $N_g := \sum_{i=1}^{q_g} N_{g,i}$. As a Boolean function, g is the monotone q_g-partite DNF $g(z) = \bigvee_{(a_1,\ldots,a_{q_g})\in R_g} \bigwedge_{i=1}^{q_g} z_{i,a_i}$, for some relation $R_g \subseteq [N_{g,1}] \times \cdots \times [N_{g,q_g}]$ [4]. Thus, fully robust gates are precisely the access structures generated by partite hypergraphs. We note that these gates are also $(q : N, N)$-multislices, which are defined by a set of minterms, such that each minterm contains exactly one variable from each part, and that RCDS versions that are not fully robust are comparable multislices of smaller width. A formula over RCDS gates with q parts, denoted as ***FRCDS***$_q$ is then a formula whose internal gates are fully robust CDS gates involving at most q parts, with no restriction on their number of inputs. Its size S is its number of gates, including input gates.

Theorem 5.5 (FRCDS$_q$ lower bound)**.** *For almost all monotone functions $F : \{0,1\}^n \to \{0,1\}$, every **FRCDS**$_q$ of size S computing F satisfies*

$$S \geq 2^{(n - O(\log n))/q}.$$

Proof. We begin by recalling the standard counting lower bound for monotone circuits. Considering the slice functions whose critical layer is layer $n/2$, i.e., assigning arbitrary values on the middle layer and extending by zero below it and one above it, gives $2^{\binom{n}{\lfloor n/2 \rfloor}}$ monotone functions. Since there are at most

$2^{O(T\log(T+n))}$ fan-in-two monotone circuits of size T, almost every monotone function requires monotone circuit size $2^{n-O(\log n)}$.

Next, we expand each fully robust gate into a fan-in-two monotone circuit. The gate g above has at most $\prod_{i=1}^{q_g} N_{g,i}$ terms, each of width q_g, and therefore has a monotone circuit of size

$$O\left(q_g \prod_{i=1}^{q_g} N_{g,i}\right) \le O\left(q_g \left(\frac{N_g}{q_g}\right)^{q_g}\right) \le O\big(N_g^{q_g}\big)\,,$$

where the first inequality follows from the arithmetic-geometric mean inequality. Now let T be the size of the monotone circuit obtained by making this substitution at every gate. Since the original computation is a formula, every child occurrence contributes one edge connected to a unique gate, and hence $\sum_g N_g \le S$. As $q_g \le q$, we obtain

$$T \le O\left(S + \sum_g N_g^q\right) \le O\left(S + \left(\sum_g N_g\right)^q\right) \le O(S^q).$$

As laid down earlier, for almost all monotone functions, every monotone circuit computing F has size $T \ge 2^{n-O(\log n)}$. Combining the two inequalities gives $S^q \ge 2^{n-O(\log n)}$; taking q'th roots proves the theorem. □

We next prove a matching upper bound.

Theorem 5.6 (FRCDS$_q$ upper bound)**.** *For every $2 \le q \le n$, every monotone function $F : \{0,1\}^n \to \{0,1\}$ has an FRCDS$_q$ of size $O\left(n2^{\lceil n/q\rceil}\right)$.*

Proof. Partition $[n]$ into q blocks $P_1, \dots, P_q$, each of size at most $\lceil n/q\rceil$. For every $i \in [q]$ and $A \subseteq P_i$, compute

$$z_{i,A} := \bigwedge_{j\in A} x_j,$$

with $z_{i,\emptyset} = 1$. Each conjunction can be computed by a tree of fan-in-two AND gates, which are themselves fully robust CDS gates with two parts.

Feed these values into a single q-partite hypergraph gate whose ith part is indexed by the subsets of P_i. Include $(A_1, \dots, A_q)$ as a hyperedge exactly when $F(A_1 \cup \cdots \cup A_q) = 1$, identifying a set with its characteristic vector.

Fix an input x. If the top gate accepts, then there is a hyperedge $(A_1, \dots, A_q)$ such that $z_{i,A_i} = 1$, and hence $A_i \subseteq P_i \cap \mathrm{supp}(x)$, for every i, where $\mathrm{supp}(x) = \{i : x_i = 1\}$. Therefore, $F(A_1 \cup \cdots \cup A_q) = 1$, and monotonicity implies $F(x) = 1$. Conversely, if $F(x) = 1$, choose $A_i = P_i \cap \mathrm{supp}(x)$. Then every $z_{i,A_i} = 1$, and $(A_1, \dots, A_q)$ is a hyperedge, so the top gate accepts. Finally, the total size of all conjunction formulas is

$$O\left(\sum_{i=1}^{q} \sum_{A\subseteq P_i} \max\{1, |A|\}\right) = O\left(\sum_{i=1}^{q} |P_i| 2^{|P_i|}\right) = O\Big(n2^{\lceil n/q\rceil}\Big)\,.$$

Adding the top gate does not change the asymptotic bound. This is the formula analogue of the standard reduction from arbitrary access structures to q-partite q-hypergraph access structures (see, e.g., [8]). □

6 Poset Dimension and Single-Top-Gate Monotone Real Formulas

In this section, we review a restricted blueprint for constructing monotone real circuits for multislices. The construction has two stages:

- The Boolean inputs to the circuit are fed to several monotone real subformulas that compute real-valued summaries of the inputs. Their outputs are then fed into one monotone top gate that decides the function. We call these summaries *monotone coordinates* and refer to the resulting tree-like form as a *single-top-gate MRF architecture*.
- The top gate is initially allowed to have high fan-in, so the architecture is not yet a standard fan-in-two MRF. The Hrubeš-Pudlák arity-reduction theorem [28] replaces this top gate by a fan-in-two monotone real circuit. The result is an MRC rather than an MRF due to the structure of the fan-in reduction transformation.

Thus, the number and complexity of the coordinates control the size of the MRC obtained from it. Our key observation is that the number of coordinates required by this route to compute a $(k : k + w, n)$-multislice is exactly the Dushnik-Miller dimension of the subsets of $[n]$ of sizes between k and $k+w$ ordered by inclusion. We lay down the required definitions and describe the connections below.

Posets and multislices. Fix $0 < k \leq \ell < n$, write $w = \ell - k$, and let $\mathcal{B}_{k,\ell} := \{x \in \{0,1\}^n : k \leq \mathrm{wt}(x) \leq \ell\}$. A *partially ordered set* (poset) is a set equipped with a reflexive, antisymmetric, and transitive relation. Two elements are incomparable if neither lies below the other. Identifying a Boolean vector with its support turns $\{0,1\}^n$ into the Boolean lattice $(2^{[n]}, \subseteq)$, and $\mathcal{Q}^n_{[k,\ell]}$ denotes the induced subposet on the set $\mathcal{B}_{k,\ell}$.

A *linear extension* of a poset P is a total order that preserves every comparison of the partial order. A family $L_1, \ldots, L_d$ of linear extensions is a *realizer* of P if

$$X \leq_P Y \quad \Longleftrightarrow \quad X \leq_{L_j} Y \text{ for every } j \in [d],$$

that is, comparable elements appear in P in the same order in every extension, whereas each *incomparable* pair must not appear in all extensions with the same order. In other words, P is the intersection of $L_1, \ldots, L_d$. The Dushnik-Miller dimension $\dim(P)$ defined in [20] is the minimum size of a realizer (see, e.g., [35]).

It is not necessary to check every incomparable pair separately. An ordered incomparable pair (X, Y) is *critical* if every element below X is also below Y, and every element above Y is also above X. A standard criterion says that a family is a realizer exactly when, for every critical pair (X, Y), some extension reverses it by placing Y before X. In $\mathcal{Q}^n_{[k,\ell]}$ (with $0 < k \leq \ell < n$ through this section), the critical-pair conditions force X to lie on the bottom level k and Y on the top level ℓ. Consequently, the intermediate levels do not increase the dimension:

$$\dim(\mathcal{Q}^n_{[k,\ell]}) = \dim(\mathcal{Q}^n_{\{k,\ell\}})$$

where the right-hand side is the subposet consisting only of the two extreme levels [35].

Coordinates and realizers. Formally, a *monotone coordinate* is a monotone real-valued function $\eta : \{0,1\}^n \to \mathbb{R}$. In the single-top-gate MRF architecture, each coordinate is computed by an MRF. We note that when deriving MRC size upper bounds, we may more generally allow an MRC implementation of each

coordinate, but all the realizations we use from the poset literature have simple MRF forms. Consider d coordinate subformulas and one total monotone top gate $\Gamma : \mathbb{R}^d \to \mathbb{R}$:

$$F(x) = \Gamma\big(\eta_1(x), \ldots, \eta_d(x)\big)$$

for every x with $k \le \mathrm{wt}(x) \le \ell$. We call the coordinates $\eta_1, \ldots, \eta_d$ *universal* if every $(k : \ell, n)$-multislice agrees on $\mathcal{B}_{k,\ell}$ with $\Gamma\big(\eta_1(x), \ldots, \eta_d(x)\big)$ for some monotone Γ while the coordinates remain fixed and only Γ changes.

Theorem 6.1 (Poset-dimension correspondence)**.** *The minimum number of monotone coordinates in a universal single-top-gate MRF architecture for $(k : \ell, n)$-multislices is*

$$\dim(\mathcal{Q}^n_{[k,\ell]}) = \dim(\mathcal{Q}^n_{\{k,\ell\}}).$$

Proof. First, let $L_1, \ldots, L_d$ be a realizer of the poset $\mathcal{Q}^n_{[k,\ell]}$. For each j, list the poset elements in the order L_j and number them from 1 to $|\mathcal{B}_{k,\ell}|$. Define the corresponding *rank coordinate* that tracks this order by

$$\eta_j(X) := 1 + \big|\{Z \in \mathcal{B}_{k,\ell} : Z <_{L_j} X\}\big|.$$

Thus $\eta_j(X)$ is simply the position of X in the ordered list L_j. Each η_j is monotone: if $X \subseteq Y$, then every linear extension places X before Y, so $\eta_j(X) \le \eta_j(Y)$. Conversely, if $\eta_j(X) \le \eta_j(Y)$ for every j, then every L_j places X before Y, and the realizer property implies $X \subseteq Y$. Thus the rank tuples capture inclusion exactly: for poset elements X and Y, we have $X \subseteq Y$ if and only if $\eta_j(X) \le \eta_j(Y)$ for every j. Define the top gate to accept whenever its coordinate tuple dominates the rank tuple of some accepted set. On a poset element Y, this occurs exactly when there is an accepted set $X \subseteq Y$, which, by monotonicity, is equivalent to Y itself being accepted. Thus $\Gamma(\eta_1, \ldots, \eta_d)$ agrees with the multislice on $\mathcal{B}_{k,\ell}$, as required. Hence $\dim(\mathcal{Q}^n_{[k,\ell]})$ coordinates suffice.

For the converse, suppose that fixed monotone coordinates $\eta_1, \ldots, \eta_d$ are universal for the poset. We show that they yield a realizer. We first claim that $\eta_1, \ldots, \eta_d$ are incomparable over every critical pair. Let (X, Y) be a critical pair. Consider the multislice whose restriction to the elements of $\mathcal{B}_{k,\ell}$ is generated solely by X: it accepts an element Z exactly when $X \subseteq Z$. This multislice accepts X but rejects Y, since X and Y are incomparable. If $\eta_i(X) \le \eta_i(Y)$ for every i, then the input to the top gate at X is coordinatewise at most its input at Y. Monotonicity would then imply $F(X) \le F(Y)$, contradicting $F(X) = 1$ and $F(Y) = 0$. Hence some coordinate satisfies $\eta_i(X) > \eta_i(Y)$.

Finally, fix a linear extension L^* of the containment order on $\mathcal{B}_{k,\ell}$, and define a total order L_i by sorting the elements in $\mathcal{B}_{k,\ell}$ in increasing order of η_i, with ties broken according to L^*. We first verify that L_i extends the containment order. If $X \subseteq Y$, then monotonicity gives $\eta_i(X) \le \eta_i(Y)$. If the inequality is strict, the sorting places X before Y, and if equality holds, L^* places X before Y. Thus L_i is a linear extension of the containment order on $\mathcal{B}_{k,\ell}$. Then, we note that the resulting family reverses every critical pair. Indeed, for each critical pair (X, Y), we found an i such that $\eta_i(X) > \eta_i(Y)$, and hence L_i places Y before X. Therefore, $L_1, \ldots, L_d$ form a realizer and $d \ge \dim(\mathcal{Q}^n_{[k,\ell]})$, which proves the claimed equality. □

From the single-top-gate MRF architecture to an MRC. The single-top-gate MRF architecture is an intermediate description rather than a standard fan-in-two circuit. We convert it to an MRC in two steps: the top gate is replaced by a fan-in-two circuit that is correct on $\mathcal{B}_{k,\ell}$, and the values outside $\mathcal{B}_{k,\ell}$ are then restored by two threshold gates.

Let $S \subseteq \mathbb{R}^d$ be the finite set of coordinate tuples attained on $\mathcal{B}_{k,\ell}$, that is, $S = \{(\eta_1(X), \ldots, \eta_d(X)) : X \in \mathcal{B}_{k,\ell}\}$. Since the coordinates are universal, a $(k : \ell, n)$-multislice f induces a well-defined monotone

function $f_S\colon S \to \{0,1\}$ by $f_S(\eta_1(X),\dots,\eta_d(X)) := f(X)$. By the arity-reduction theorem of Hrubeš and Pudlák [28] (from here on HP, Theorem 2.5), f_S is computed by a fan-in-two monotone real circuit C of size $O\big((\log_2|S|)^{d-2}\big)$. Feeding C with the coordinates yields a monotone real circuit that outputs $f(x)$ on every input $x \in \mathcal{B}_{k,\ell}$; on inputs outside $\mathcal{B}_{k,\ell}$ its output is some real value that we do not control.

We fix these values using the Hamming weight of the inputs. Let $u(c) := 1$ if $c \geq 1$ and $u(c) := 0$ otherwise. Then u is a monotone unary gate, and $g := u \circ C$ is a monotone Boolean function on $\{0,1\}^n$ that agrees with f on $\mathcal{B}_{k,\ell}$. Writing Th_t for the t-out-of-n threshold function, we have

$$f(x) \;=\; \mathrm{Th}_{\ell+1}(x) \;\vee\; \big(\mathrm{Th}_k(x) \wedge g(x)\big) \qquad \text{for every } x \in \{0,1\}^n,$$

since for $\mathrm{wt}(x) < k$ both sides equal 0, for $\mathrm{wt}(x) > \ell$ both sides equal 1, and otherwise the right-hand side equals $g(x) = f(x)$. The two thresholds cost $O(n)$ gates, since an addition tree can compute $\mathrm{wt}(x)$, and each threshold is then a single unary gate applied to it. Thus, if every coordinate is computed by a monotone real formula or circuit of size at most σ, then f is computed by a fan-in-two MRC of size

$$d\sigma \;+\; O\Big(\big(\log_2|\mathcal{B}_{k,\ell}|\big)^{\max\{d-2,\,0\}}\Big) \;+\; O(n).$$

Bounds imported from poset dimension. The basic dimension estimates therefore describe the power and limitations of single-top-gate MRF architectures. An upper bound on dimension implies a universal family of coordinates, and when these coordinates have efficient implementations, the HP compilation above gives an MRC upper bound. A dimension lower bound implies that no universal single-top-gate MRF architecture can use fewer coordinates. In all of the following upper bounds, the realizers can be computed by small MRFs, and thus the dominant factor in the circuit size is the number of coordinates going through the HP transformation.

- *General multislices.* By Brightwell et al. [15], for every width $w \geq 1$,

  $$\dim(\mathcal{Q}^n_{[k,k+w]}) = O(w^2 \log n).$$

  This gives an MRC of size $2^{O(w^2\log^2 n)}$. In the other direction, for $k = 1$ the lower bound of Trotter [53] gives an $\Omega(w^2)$ coordinate requirement when the width w is below $\sqrt{n}$, and Dushnik [19] proved a bound of $n - \sqrt{n}$ for width above $\sqrt{n}$.

- *Width-1 multislices.* By Kostochka [37], for width one it holds for every s that

  $$\dim(\mathcal{Q}^n_{[s,s+1]}) = O(\log n/\log\log n).$$

  This gives an MRC of size $2^{O((\log n)^2/\log\log n)}$. Füredi et al. proved in [22] an $\Omega(\log\log n)$ bound on the number of coordinates for $\mathcal{Q}^n_{[1,2]}$.

- *Low multislices.* For multislices beginning at level one, Spencer's [51] construction gives

  $$\dim(\mathcal{Q}^n_{[1,w]}) = O(w2^w \log\log n).$$

  Kierstead [35] proves a dimension lower bound of $\Omega(2^{w-2}\log\log n)$ for this case for $2 \leq w \leq \log\log n - \log\log\log n$. Spencer's construction also gives explicit coordinate formulas of size $O(n)$

each, so the single-top-gate architecture can be converted efficiently into an MRC. Therefore, every $(1 : w, n)$-multislice can be computed by a monotone real circuit of size

$$O(w2^w \cdot n \log\log n) + 2^{O(w2^w \log\log n\,(\log\log n + \log w))}.$$

In particular, for every fixed $\varepsilon > 0$, if $w \leq (1-\varepsilon)\log_2\log_2 n$, the size is $O(w2^w \cdot n\log\log n) = n^{1+o(1)}$.

We note that this is the only case of an MRC created with this architecture that outperforms existing unrestricted MRCs. E.g., when w is constant its size is $O(n\log\log n)$, while the FOS implicit in [11, Lemma 5.9], converted to an MRC, has size $O(n\log^2 n)$.

We conclude by noting that all lower bounds in this section concern the number of coordinates in *universal single-top-gate MRF architectures before the Hrubeš-Pudlák conversion*. They are not lower bounds for computing one fixed multislice.

Acknowledgments

The authors are supported by ISF grant 1614/25. The first author is also supported by ERC project NFITSC(10107959). We thank Or Lasri for discussions that led to Corollary 5.3, and Oriol Farràs for a question that prompted us to include Corollary 1.4.

References

[1] Bar Alon, Amos Beimel, and Or Lasri. Simplified PIR and CDS protocols and improved linear secret-sharing schemes. In Benny Applebaum and Huijia (Rachel) Lin, editors, *Theory of Cryptography - 23rd International Conference, TCC 2025, Proceedings, Part II*, volume 16269 of *Lecture Notes in Computer Science*, pages 365–398. Springer, 2025.

[2] 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, 2018.

[3] 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, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 441–471. Springer, 2019.

[4] 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, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 280–293. ACM, 2020.

[5] Benny Applebaum, Amos Beimel, Oded Nir, Naty Peter, and Toniann Pitassi. Secret sharing, slice formulas, and monotone real circuits. *Electron. Colloquium Comput. Complex.*, TR22, 2022.

[6] Benny Applebaum, Amos Beimel, Oded Nir, Naty Peter, and Toniann Pitassi. Secret sharing, slice formulas, and monotone real circuits. *ACM Transactions on Computation Theory*, 18(2):1–32, 2026.

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

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

[9] Amos Beimel. Secret-sharing schemes for general access structures: An introduction. Cryptology ePrint Archive, Paper 2025/518, 2025.

[10] 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.

[11] 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, Proceedings, Part IV*, volume 15367 of *Lecture Notes in Computer Science*, pages 581–613. Springer, 2024.

[12] 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, Part II*, volume 10821 of *Lecture Notes in Computer Science*, pages 287–318. Springer, 2018.

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

[14] 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.

[15] Graham R Brightwell, Henry A Kierstead, Alexandr V Kostochka, and William T Trotter. The dimension of suborders of the boolean lattice. *Order*, 11(2):127–134, 1994.

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

[17] Yuval Dagan, Yuval Filmus, Ariel Gabizon, and Shay Moran. Twenty (simple) questions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, June 19-23*, pages 9–21. ACM, 2017.

[18] Mateus de Oliveira Oliveira and Pavel Pudlák. Representations of monotone boolean functions by linear programs. *ACM Trans. Comput. Theory*, 11(4):22:1–22:31, 2019.

[19] Ben Dushnik. Concerning a certain set of arrangements. *Proceedings of the American Mathematical Society*, 1(6):788–796, 1950.

[20] Ben Dushnik and Edwin W Miller. Partially ordered sets. *American journal of mathematics*, 63(3):600–610, 1941.

[21] Michael L. Fredman. How good is the information theory bound in sorting? *Theor. Comput. Sci.*, 1(4):355–361, 1976.

[22] Zoltan Füredi, Péter Hajnal, Vojtech Rödl, and William T. Trotter. Interval orders and shift graphs. In *Sets, Graphs and Numbers: Colloquia Mathematica Societatis János Bolyai*, volume 60, pages 297–313. North-Holland Publishing Co.; János Bolyai Mathematical Society, 1992.

[23] Zoltán Füredi and Jeff Kahn. On the dimensions of ordered sets of bounded degree. *Order*, 3(1):15–20, 1986.

[24] 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.

[25] Edgar N Gilbert and Edward F Moore. Variable-length binary encodings. *Bell System Technical Journal*, 38(4):933–967, 1959.

[26] Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. *SIAM J. Comput.*, 47(5):1778–1806, 2018.

[27] Milan Haiman. The dimension of divisibility orders and multiset posets. *Order*, 41(3):693–707, 2024.

[28] Pavel Hrubeš and Pavel Pudlák. A note on monotone real circuits. *Inf. Process. Lett.*, 131:15–19, 2018.

[29] 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.

[30] Stasys Jukna. Combinatorics of monotone computations. *Comb.*, 19(1):65–85, 1999.

[31] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 539–550. ACM, 1988.

[32] 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.

[33] Ehud D. Karnin, J. W. Greene, and Martin E. Hellman. On secret sharing systems. *IEEE Trans. Inf. Theory*, 29(1):35–41, 1983.

[34] Shahram Khazaei. Rank measures and exponential lower bounds for multilinear secret sharing. *Cryptology ePrint Archive*, 2026.

[35] Hal A Kierstead. The dimension of two levels of the boolean lattice. *Discrete mathematics*, 201(1-3):141–155, 1999.

[36] Aleksej Dmitrievich Korshunov. On the number of monotone boolean functions. *Problemy kibernetiki*, 38:5–109, 1981.

[37] Alexandr V Kostochka. The dimension of neighboring levels of the boolean lattice. *Order*, 14(3):267–268, 1997.

[38] Jan Krajícek. Interpolation by a game. *Math. Log. Q.*, 44:450–458, 1998.

[39] Jan Krajícek. Randomized feasible interpolation and monotone circuits with a local oracle. *J. Math. Log.*, 18(2):1850012:1–1850012:27, 2018.

[40] Jan Krajícek and Igor C. Oliveira. On monotone circuits with local oracles and clique lower bounds. *Chic. J. Theor. Comput. Sci.*, 2018, 2018.

[41] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[42] 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.

[43] 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.

[44] Or Meir. An efficient randomized protocol for every karchmer-wigderson relation with two rounds. *Electron. Colloquium Comput. Complex.*, TR17, 2017.

[45] Shay Moran and Amir Yehudayoff. A note on average-case sorting. *Order*, 33(1):23–28, 2016.

[46] Oded Nir. Resolving the worst-case complexity of linear secret sharing. Cryptology ePrint Archive, Paper 2026/1667, 2026.

[47] Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1(301-315):6, 1993.

[48] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997.

[49] Arnold Rosenbloom. Monotone real circuits are more powerful than monotone Boolean circuits. *Inf. Process. Lett.*, 61(3):161–164, 1997.

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

[51] J Spencer. Minimal scrambling sets of simple orders. *Acta Mathematica Academiae Scientiarum Hungaricae*, 22(3):349–353, 1971.

[52] Ulfberg Stafan. *On Lower Bounds for Circuits and Selection*. Ph.D., Royal Institute of Technology, Stockholm, Sweden, 1999.

[53] William T. Trotter. *Combinatorics and Partially Ordered Sets: Dimension Theory*. Johns Hopkins University Press, 1992.

[54] Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35(6):703–747, 2015.

[55] Christopher Williamson. Secret sharing at the shannon ceiling. *arXiv preprint arXiv:2608.17047*, 2026.

A Proof of the Gilbert-Moore alphabetic-tree theorem

In this section, we prove that for every $p_1, \ldots, p_m > 0$ satisfying $\sum_{i \in [m]} p_i = 1$, there is a rooted binary tree in which every internal node has exactly two children, whose leaves are $1, \ldots, m$ in left-to-right order, and in which leaf i has at most $\lceil \log_2(1/p_i) \rceil + 1$ ancestors.

Proof of Theorem 3.5. We construct the required tree by assigning leaf i a binary codeword σ_i. I.e., a path from a root to a leaf in an ordered binary tree. To obtain these codewords, we use the probabilities $p_1, \ldots, p_m$ to partition $[0, 1)$ into consecutive intervals $I_1, \ldots, I_m$, where I_i has length p_i; these intervals fill $[0, 1)$ because $\sum_{i \in [m]} p_i = 1$. Inside each I_i we will choose a short interval described by a binary prefix, and this prefix will be the codeword σ_i.

We first explain how binary prefixes describe subintervals of $[0, 1)$. A binary string σ of length ℓ represents an integer $q \in \{0, \ldots, 2^\ell - 1\}$ and determines the interval

$$D(\sigma) = \left[\frac{q}{2^\ell}, \frac{q+1}{2^\ell} \right).$$

This interval consists precisely of the numbers in $[0, 1)$ whose binary expansion starts with σ, where for numbers with two binary expansions we choose the one that does not end in an infinite sequence of 1's. It has length $2^{-\ell}$ and is called a *dyadic interval*. Appending a bit to σ selects either the left or the right half of $D(\sigma)$.

We now define the promised partition of $[0, 1)$. We use half-open intervals so that consecutive intervals do not overlap at their common endpoint. For each $i \in [m]$, let

$$s_i = \sum_{i' < i} p_{i'}, \qquad I_i = [s_i, s_i + p_i), \qquad \text{and} \qquad c_i = s_i + \frac{p_i}{2}.$$

Thus, I_i has length p_i, the intervals $I_1, \ldots, I_m$ appear from left to right in this order, and c_i is the midpoint of I_i. Set $\ell_i = \lceil \log_2(1/p_i) \rceil + 1$, and let σ_i be the first ℓ_i bits in the binary expansion of c_i. By definition, $c_i \in D(\sigma_i)$. Moreover, $|D(\sigma_i)| = 2^{-\ell_i} \le 2^{-\log_2(1/p_i)-1} \le \frac{p_i}{2}$. Every point of $D(\sigma_i)$ is at distance less than $|D(\sigma_i)|$ from c_i, while c_i is at distance exactly $p_i/2$ from each endpoint of I_i. Therefore, $D(\sigma_i) \subseteq I_i$.

This containment gives the two properties needed to construct an alphabetic code. First, the intervals $D(\sigma_i)$ are pairwise disjoint because the intervals I_i are pairwise disjoint. Hence no σ_i is a prefix of another $\sigma_{i'}$: if it were, then $D(\sigma_{i'}) \subseteq D(\sigma_i)$, contradicting disjointness. Thus the strings $\sigma_1, \ldots, \sigma_m$ are prefix-free. Second, if $i < i'$, then $D(\sigma_i)$ lies entirely to the left of $D(\sigma_{i'})$. Consequently, σ_i precedes $\sigma_{i'}$ in lexicographic order.

Finally, we form the binary tree of the strings $\sigma_1, \ldots, \sigma_m$: its root is the empty string, and following a left or right edge appends 0 or 1, respectively. Since the strings are prefix-free, they are exactly the leaves of this tree. Since they occur in lexicographic order as $\sigma_1, \ldots, \sigma_m$, the corresponding leaves occur from left to right as $1, \ldots, m$. Now repeatedly contract the edge from every node with exactly one child to its unique child. These contractions preserve the leaves and their left-to-right order, do not increase any leaf depth, and leave a tree in which every node is either a leaf or has two children. Thus, every ancestor of a leaf has two children, and the depth of leaf i is at most

$$\mathrm{depth}_T(i) \le \ell_i = \lceil \log_2(1/p_i) \rceil + 1$$

as required. □

B Proof of the Input Fixing Lemma

Proof of Lemma 5.1. It suffices to fix one input p at a time. Fixing p to 0 deletes p: for an ideal linear scheme, the share of p is removed, and the remaining shares form an ideal linear scheme for g'; for a CDS gate, the vertex p and the hyperedges containing it are deleted, which gives a CDS gate.

Next, let p be fixed to 1, so that a set B of the remaining inputs is authorized in g' if and only if $B \cup \{p\}$ is authorized in g. If g is an ideal $\mathbb{F}$-linear gate, write the share of p as $\alpha s + \beta \cdot r$, where s is the secret and r is the randomness. If $\beta = 0$, then this share is either identically 0, in which case we simply delete p, or it determines s, in which case $g' \equiv 1$. Suppose therefore that $\beta \neq 0$. We condition the scheme on the share of p being 0 and then omit this fixed share. Concretely, choose v with $\beta \cdot v = 1$, sample r' uniformly from $\ker \beta$, and use randomness $r' - \alpha s v$ in the original scheme. All remaining shares are still linear functions of s and r'. A set B reconstructs the secret in the new scheme exactly when $B \cup \{p\}$ reconstructs it in the original scheme; moreover, if $B \cup \{p\}$ is unauthorized, then its shares are independent of the secret, and this remains true after conditioning the share of p to be 0. Thus the remaining shares form an ideal linear scheme realizing g'.

If g is a CDS gate and $p = z_{i,a}$, then g' accepts B if and only if B contains an input of part V_i, two inputs of the same part V_j for some $j \neq i$, or inputs z_{j,a_j} for all $j \neq i$ such that the tuple $(a_1, \ldots, a_{q_g})$ with $a_i = a$ is in R_g. Hence $g' = h \vee \bigvee_{z \in V_i \setminus \{p\}} z$, where h is the CDS gate with parts V_j, $j \neq i$, and relation $\{(a_j)_{j \neq i} : (a_1, \ldots, a_{q_g}) \in R_g \text{ with } a_i = a\}$ (if a second input of V_i is fixed to 1, then $g' \equiv 1$). □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il