Arithmetic Sketching

Dan Boneh[1], Elette Boyle[2], Henry Corrigan-Gibbs[3],
Niv Gilboa[4], and Yuval Ishai[5]

[1] Stanford University dabo@cs.stanford.edu
[2] Reichman University and NTT Research eboyle@alum.mit.edu
[3] MIT henrycg@csail.mit.edu
[4] Ben-Gurion University gilboan@bgu.ac.il
[5] Technion yuvali@cs.technion.ac.il

Abstract. This paper introduces *arithmetic sketching*, an abstraction of a primitive that several previous works use to achieve lightweight, low-communication zero-knowledge verification of secret-shared vectors. An arithmetic sketching scheme for a language $\mathcal{L} \subseteq \mathbb{F}^n$ consists of (1) a randomized linear function compressing a long input x to a short "sketch," and (2) a small arithmetic circuit that accepts the sketch if and only if $x \in \mathcal{L}$, up to some small error. If the language $\mathcal{L}$ has an arithmetic sketching scheme with short sketches, then it is possible to test membership in $\mathcal{L}$ using an arithmetic circuit with few multiplication gates. Since multiplications are the dominant cost in protocols for computation on secret-shared, encrypted, and committed data, arithmetic sketching schemes give rise to lightweight protocols in each of these settings.

Beyond the formalization of arithmetic sketching, our contributions are:

- A general framework for constructing arithmetic sketching schemes from algebraic varieties. This framework unifies schemes from prior work and gives rise to schemes for useful new languages and with improved soundness error.
- The first arithmetic sketching schemes for languages of *sparse* vectors: vectors with bounded Hamming weight, bounded L_1 norm, and vectors whose few non-zero values satisfy a given predicate.
- A method for "compiling" any arithmetic sketching scheme for a language $\mathcal{L}$ into a low-communication malicious-secure multi-server protocol for securely testing that a client-provided secret-shared vector is in $\mathcal{L}$.

We also prove the first nontrivial lower bounds showing limits on the sketch size for certain languages (e.g., vectors of Hamming-weight one) and proving the non-existence of arithmetic sketching schemes for others (e.g., the language of all vectors that contain a specific value).

1 Introduction

In many cryptographic protocols, a server holds an encoding of a large client-provided vector. To protect against client misbehavior, the server must test whether the vector satisfies a simple predicate. For example:

- In private ad measurement [29, 48], an aggregation server holds a linearly homomorphic encryption of a client-provided vector; the server must test that the encrypted vector is zero everywhere except with a "1" at a single location.
- In PIR writing [42] and private messaging [1, 17, 24], a set of servers holds additive secret shares of a client-provided vector; the servers must test that the secret-shared vector is zero everywhere except that it contains an arbitrary value at a single location.
- In private-aggregation [10, 16, 20, 36, 45], a set of servers holds a secret-sharing of a large client-provided vector; the server must test whether the vector has bounded Hamming weight.
- In e-voting schemes [31], a tally server holds a linearly homomorphic encryption of a vector representing a ballot; the server must test that the vector has non-negative entries and bounded L_1 norm.
- In verifiable distributed multi-point functions [21], a set of servers holds (compressed) additive secret shares of a client-provided vector; the servers must test whether the vector has Hamming weight exactly w.
- In protocols for malicious-secure OT and MPC [19], a verifier holds a linearly homomorphic commitment to a prover-provided vector; the verifier must test whether the vector has bounded Hamming weight.

Each of these prior works gives a clever special-purpose protocol for its particular property-testing problem. And each scheme has the desirable feature that in the most important complexity measure—typically server-to-server communication or proof size—the server's cost is sublinear in the input size.

At the same time, each of the state-of-the-art schemes has at least one of three shortcomings: dependence on proofs, limited extensibility, and unclear optimality of their complexity measures. In particular, many of these protocols [10, 16, 17, 19, 24, 31] require the client to provide some auxiliary information ("proof") about the vector to the servers/verifier to perform the validity check, or even require interaction with the client [9, 19]. This dependence on the client precludes important use cases such as distributing the client, e.g. in an MPC protocol, or accepting the aggregate contribution of multiple clients. Also, in some applications, the servers must check a property on a subset of the vector that only the servers know [10]. So even if there is a single client, the client may not know what exactly it needs to prove. In addition, these protocols are purpose-built to test specific properties required for their applications, without necessarily considering extensions or generalizations. For example, testing that a vector has a single non-zero location is useful for different applications, but the set of authorized non-zero payloads might change between applications. Making these tools most broadly useful requires generalizing and unifying these constructions. Finally, in each of these protocols, it is not clear whether the known schemes are the best possible, in terms of the key complexity metrics.

In this work, we introduce arithmetic sketching schemes, giving a general framework for constructing protocols that enable testing simple properties on large vectors. Our schemes resolve the main shortcomings of those in prior work:

they require no auxiliary "proof" information from the client in the secret-shared setting, they are extensible, and in certain cases they match new efficiency lower bounds that we prove. The general framework provides simple, concretely efficient protocols for each of the aforementioned applications.

Formalization of arithmetic sketching. The first contribution of this paper is the definition of arithmetic sketching schemes. The purpose of an arithmetic sketching scheme for a language of vectors $\mathcal{L} \subseteq \mathbb{F}^n$, over a finite field $\mathbb{F}$ and input size $n \in \mathbb{N}$, is to test whether a given input $x \in \mathbb{F}^n$ is in the language $\mathcal{L}$. Typically, the languages $\mathcal{L}$ we consider are "simple" ones, such as the language of vectors of Hamming-weight one, of bounded-L_1 norm, etc.

More precisely, an arithmetic sketching scheme for a language $\mathcal{L}$ is a pair of algorithms: a sketching algorithm and a decision algorithm.

- The *sketching algorithm* is a randomized procedure that outputs a matrix $Q \in \mathbb{F}^{\ell \times n}$. The number of rows in the matrix ℓ (the "sketch size") is typically small—constant in the input length n.
- The *decision algorithm*, takes as input ℓ values, corresponding to the matrix-vector product of the sketching matrix $Q \in \mathbb{F}^{\ell \times n}$ with the input vector $x \in \mathbb{F}^n$. The decision algorithm must accept all $x \in \mathcal{L}$ and, with high probability, reject all $x \notin \mathcal{L}$. Furthermore, the decision algorithm must be *arithmetic*, in the sense that it is computed by an arithmetic circuit over the field $\mathbb{F}$, with size independent of the field size $|\mathbb{F}|$ and the input length n. Here, we interpret an output of "0" as accepting and a nonzero output as rejecting.

We can think of the vector $(Q \cdot x) \in \mathbb{F}^\ell$ as a succinct "sketch" of the large input $x \in \mathbb{F}^n$, since these $\ell \ll n$ values contain enough information to decide whether or not $x \in \mathcal{L}$. While all of the algorithms we construct are computationally efficient, when defined relative to infinite families of languages $\{\mathcal{L}_n\}_{n=1}^{\infty}$, computational efficiency plays no role in our definitions—an arithmetic sketching scheme is a purely information-theoretic object.

While linear sketching is a staple of data-structure and algorithm design [2, 15, 41], the key distinction here is the requirement for the decision predicate to be a small arithmetic circuit. Removing this arithmetic requirement trivializes the problem: as we discuss in Section 2.2, every sparse-enough language has a simple non-arithmetic sketching scheme with a small sketch size (albeit with a computationally inefficient decision procedure). The arithmetic requirement we place on the decision predicate limits the power of the sketches we consider: they cannot, say, test predicates of very large algebraic degree. At the same time, as we now discuss, arithmetic sketching schemes are a natural fit for cryptographic applications.

Applications of arithmetic sketching. Whenever verifiers can inexpensively apply *linear functions* to a large input x, arithmetic sketches yield asymptotically and concretely efficient protocols for testing properties on the input x. For example, consider a set of verifiers who hold linear secret shares of a large vector x, as is the case in multiparty computations [6] and many practical

protocols [1, 9, 10, 16, 17, 20, 21, 23, 24, 42], and who want to test whether $x \in \mathcal{L}$. An arithmetic sketching scheme for $\mathcal{L}$ gives a very simple multiparty protocol for this problem: The verifiers can use shared randomness to generate the sketching matrix $Q \in \mathbb{F}^{\ell \times n}$ and can locally compute secret shares of the "sketch" $Q \cdot x \in \mathbb{F}^{\ell}$. Then, the verifiers can run the decision procedure on these ℓ values in a small multiparty computation to determine whether $x \in \mathcal{L}$. The fact that the decision procedure is arithmetic and of small size (independent of $|\mathbb{F}|$ and n) ensures that this last step is inexpensive, even in the multiparty setting.

Analogously, consider a server who holds the encryption of a client-provided vector $x \in \mathbb{F}^n$ under a linearly homomorphic encryption (or commitment) scheme $E(\cdot)$, as is the case in a number of protocols [19, 29, 31, 45, 48]. In this case, the server can unilaterally compute the sketch under encryption: $E(Q \cdot x)$ using random coins it shares with the client. The client can then convince the server that $x \in \mathcal{L}$ using a small zero-knowledge proof—again, whose size is independent of $\mathbb{F}$ and n. While general-purpose succinct zero-knowledge techniques [8] can of course achieve the same goal, arithmetic sketching schemes can yield simpler protocols under simpler assumptions.

Efficiency metrics. When using arithmetic sketching schemes in cryptographic protocols, there are three main complexity metrics of interest:

- The *sketch size* dictates the computation required to evaluate the sketch. Since computing the sketch is the only part of the computation that depends on the (large) input length n, minimizing the sketch size—down to the constant—is crucial for concrete efficiency.
- The *algebraic degree* of the decision procedure dictates the round complexity of a multiparty computation for evaluating the sketch. Or, when s servers evaluate the sketch in a non-interactive multiparty computation, a sketch of degree d dictates the collusion threshold $t < s/d$—the number of colluding servers that the protocol can tolerate.
- The *number of multiplication gates* in the arithmetic circuit representing the decision procedure dictates the communication complexity of evaluating the sketch in a multiparty computation. Or, when the sketch is used to construct zero-knowledge proof, the gate count determines the size and time cost of generating and verifying the proof.

With an eye towards building the most concretely efficient multiparty protocols, our constructions aim to minimize these costs, which may each form a bottleneck in natural application scenarios.

1.1 Our contributions

We now summarize our technical contributions.

New framework for sketching weight-one vectors. Our first contribution is a new framework for constructing arithmetic sketching schemes for languages of vectors $\mathcal{L} \subseteq \mathbb{F}^n$ with at most one non-zero entry, where the non-zero entry must lie in a certain set $B \subseteq \mathbb{F}$. Such languages feature prominently in schemes

for PIR writing [1, 17, 24, 42], private ad-measurement systems [29, 48], private telemetry applications [10, 20], and two-party ORAM schemes [23]. Our new view of arithmetic sketching schemes for weight-one vectors yields the first asymptotically optimal arithmetic sketching schemes for weight-one vectors with $B = \mathbb{F}$ and $B = \{-1, 0, 1\}$ (for "like/abstain/dislike" voting), and also gives a clean and unified derivation of existing schemes.

Our approach works in three steps:

1. In Section 3.1, we define *algebraic manipulation detection distributions* ("AMD distributions"), a new object inspired by the earlier notion of AMD codes [18] that may be of independent interest. For $B \subseteq \mathbb{F}^\ell$, a B-multiplicative AMD distribution is a distribution $\mathcal{D}$ over codewords in $\mathbb{F}^\ell$, along with a verifier. For a word sampled from $\mathcal{D}$, verification succeeds if a "permitted" affine transformation $\mathbb{F}^\ell \to \mathbb{F}^\ell$ has been applied to the word, and verification fails (with high probability) otherwise. More precisely, the verifier must accept all words in the support of $\mathcal{D}$ scaled by any value in B. The verifier must reject, with high probability, all affine functions on words in the support of $\mathcal{D}$ that fall outside of this "permitted" set.
2. In Section 3.2, we show that any B-multiplicative AMD distribution gives a simple arithmetic sketching scheme for the language of vectors of weight at most one, whose non-zero element lies in B. The size ℓ of the sketch is equal to the length of codewords in the AMD distribution.
3. Finally, in Section 3.3, we show that pairs of polynomials over $\mathbb{F}$ whose zeros satisfy certain conditions yield AMD distributions.

Putting these three components together gives both a recipe for constructing new arithmetic sketching schemes for weight-one vectors, and a characterization of existing schemes for testing properties of weight-one vectors [14].

The first arithmetic sketching schemes for low-weight vectors. In Section 4, we use new techniques to construct arithmetic sketching schemes for vectors of various form with bounded weight. Here we consider a more refined version of the above notion of arithmetic sketching in which the decision algorithm is split into two parts: a *private* part, which is a randomized arithmetic circuit computing a "sanitized" version of the sketch, and a *public* part, which is a (possibly non-arithmetic) predicate deciding whether to accept or reject the sanitized sketch. The sanitized sketch computed by the private part should reveal nothing about an input $x \in \mathcal{L}$ except for the fact that it is in $\mathcal{L}$. Thus, in applications, we only need to securely evaluate the private part of the decision algorithm.

First, in Section 4.2, we show how to repurpose existing algorithms for black-box testing of sparse polynomials [30] to construct an arithmetic sketching scheme for vectors in $\mathbb{F}^n$ of Hamming weight w, where the non-zero entries can be arbitrary in $\mathbb{F}$.

Our idea is to view the input vector $x \in \mathbb{F}^n$ as the coefficients of a polynomial $f_x \in \mathbb{F}[Z]$ of degree at most $n-1$. Now, the input x has Hamming weight w if and only if the polynomial f_x has w non-zero coefficients. Prior work [30] shows that $O(w)$ polynomial-evaluation queries are enough to test whether a polynomial has

w non-zero coefficients. Furthermore the decision procedure for this test can be made arithmetic. With a single linear combination of the elements of x, of the form $(1, r, r^2, r^3, \ldots, r^{n-1}) \in \mathbb{F}^n$, we can compute an evaluation of f_x on r. So, we obtain an $O(w)$-size sketch for the language of weight-w vectors.

Next, in Section 4.3, we construct an arithmetic sketching scheme for vectors in $\mathbb{F}^n$ whose non-zero components are in the set $\{0, \ldots, w\}$ and sum to exactly w (i.e., the "L_1-norm" is w). This language comes up in voting applications, where a client may cast w votes for n candidates, and may vote multiple times for the same candidate. Our technical idea is to construct a low-degree multivariate polynomial whose coefficients are determined by the input vector $x \in \mathbb{F}^n$, such that the polynomial is (a) identically zero whenever the vector x has L_1-norm w and is (b) non-zero otherwise. We construct this polynomial via Newton's identities. Then, we show that it is possible to evaluate this polynomial at a random point using $w + 1$ queries to the input vector.

As a final arithmetic sketching scheme for the bounded-weight case, we show in Section 4.4 how to sketch for vectors $\mathbb{F}^n$ of weight w whose non-zero elements satisfy an arbitrary arithmetic circuit $C \colon \mathbb{F} \to \mathbb{F}$, where the size of C is independent of $\mathbb{F}$. To do so, we first use the aforementioned results to test that the input vector x has Hamming weight w. We next use linear queries to partition the input vector at random into w^2 chunks of size n/w^2, and to sum the values in each chunk. We then accept if either (a) the vector has Hamming weight $< w$, in which case two non-zero elements were hashed to the same chunk or (b) the sums all satisfy the circuit C, which happens whenever the vector is valid and there are no collisions. (By the birthday bound, the latter happens with constant probability.) This basic scheme has large correctness error, so we drive the correctness error down using a careful repetition of these steps.

We summarize our new sketch constructions in Table 1.

From sketching to malicious-secure client-server multiparty protocols. As we have discussed, arithmetic sketching schemes naturally give rise to protocols that a set of servers can use to check that a client-provided secret-shared vector x lies in a language $\mathcal{L}$. In essentially all applications, $x \in \mathcal{L}$ if the client is honest, and $x \notin \mathcal{L}$ otherwise. To recall: the servers, each holding additive shares of x, use shared randomness to generate the sketching matrix, they compute shares of the sketch locally, and then run a multiparty computation to run the decision procedure on these answers.

This simple protocol is *not* maliciously secure: any one of the servers can shift their share of the input $x \in \mathbb{F}^n$ by an additive offset $\Delta \in \mathbb{F}^n$. If the client is honest ($x \in \mathcal{L}$), the sketch should always accept and the servers learn nothing about x apart from the fact that $x \in \mathcal{L}$. However, once the malicious server has shifted its input by Δ, the output of the decision procedure reveals whether $(x + \Delta) \in \mathcal{L}$, which could reveal one bit of information about x. (In an e-voting setting, for example, this bit could reveal which candidate the client voted for.)

In Section 5, we give a general method for augmenting sketch-based client-server protocols of this form with malicious security at minimal cost. So, we show that any good arithmetic sketching scheme for a language $\mathcal{L}$ yields a malicious-

Table 1. Our new sketch constructions and their relation to prior work. Here, we assume that the sketch is over a finite field $\mathbb{F}$ of characteristic greater than two and the soundness error is $O(|\mathbb{F}|^{-1})$, unless otherwise noted. The note "Requires proof" indicates that the scheme is a fully linear probabilistically checkable proof [9] (see Section 5), rather than an arithmetic sketching scheme. The note "Only for DPF" indicates that the sketch only applies in the context of a specific secret-sharing scheme and $\omega < 2.38$ is the matrix-multiplication constant.

	Our sketch				
Language	Result	Size	Deg.	Muls.	Notes
Weight at most one, non-zero value in B...					
... with $B = \{0, 1\}$	n/a	2	2	1	*From BGI [14]*
... with $B = \{-1, 1\}$	n/a	2	2	1	
... with $B = \{1\}$	n/a	2	2	1	
... with $B = \mathbb{F}$	Thm. 13	3	2	2	*Requires proof [14]*
... with $B = \{-1, 0, 1\}$	"	2	3	2	" "
... with $B = \{0, 1\}$	Thm. 16	3	2	2	*Soundness err.* $O(\lvert\mathbb{F}\rvert^{-2})$
Weight at most w, non-zero values in B...					
... with $B = \mathbb{F}$	Cor. 23	$2w+1$	3	$w^{O(1)}$	*Requires proof [19]*
... with $B = \{0, \ldots, w\}$, entries sum to $\leq w$	Thm. 26	$w+1$	$w+2$	$O(w^2)$	" "
" "	"	"	3	$w^{O(1)}$	" "
Weight exactly w, non-zero values in B...					
... with $B = \mathbb{F}$	Thm. 22	$2w+1$	3	$O(w^\omega)$	*Only for DPF [21]*
... with $B = \{0, 1\}$	Cor. 28	$3w+2$	3	$O(w^\omega)$	*Requires proof [19]*

secure client-server protocol of this type for $\mathcal{L}$. The technique is to have the client provide a randomized encoding of its input to the servers, using a flavor of algebraic manipulation detection distributions (as in Section 3). Our construction generalizes beyond arithmetic sketching schemes to also provide malicious-secure client-server protocols for languages with "fully linear probabilistically checkable proofs" [9].

Limits of arithmetic sketching schemes. Our final contributions are to give lower bounds on the efficiency and power of arithmetic sketching schemes.

In Section 6, we use an algebraic argument to show that any arithmetic sketching scheme for vectors in $\mathbb{F}^n$ of weight at most one ($B = \mathbb{F}$) with sketch size ℓ and a decision predicate of degree d has soundness error at least $\left((d+1)\,|\mathbb{F}|^{\ell-2}\right)^{-1}$. This establishes that our arithmetic sketching scheme of Section 3 with sketch size $\ell = 3$ is essentially optimal in terms of its sketch size, providing an unexpected separation from the case of unit vectors in $\mathbb{F}^n$ ($B = \{0, 1\}$), in which sketch size $\ell = 2$ is enough for $O(1/|\mathbb{F}|)$ soundness error.

In Section 7, we use lower bounds from communication complexity to show that certain languages cannot have arithmetic sketching schemes. For example, define the L_p-norm of $x = (x_1, \ldots, x_n) \in \mathbb{F}^n$, for p relatively prime to $|\mathbb{F}| - 1$, as the scalar $(x_1^p + \cdots + x_n^p)^{1/p} \in \mathbb{F}$. In Section 4, we give a arithmetic sketching scheme for the language of vectors of L_1 norm w, with entries in $\{0, \ldots, w\}$. It is something of a surprise then that, as we show, there are no analogous arithmetic sketching schemes the language of vectors with L_p-norm, for any $p > 1$. For any $B \subseteq \mathbb{F}$, we also rule out arithmetic sketching schemes for the language of vectors that are all zeros with a contiguous run of values in B of arbitrary length, and the language of vectors that contains one value in B and arbitrary values elsewhere.

1.2 Related work

Boyle, Gilboa, and Ishai [14] implicitly constructed arithmetic sketching schemes for languages of vectors in $\mathbb{F}^n$ of Hamming weight one, where the non-zero element is in $B \subseteq \mathbb{F}$, for $B = \{0, 1\}, \{-1, 1\}, \{1\}$. They also presented a solution for the case of an unrestricted payload (i.e., $B = \mathbb{F}$), but their approaches required auxiliary proof information from the client. A variant of this construction that avoids the extra proof was given in the Blinder system [1].

Distributed point functions [14, 28] (DPFs) give a compressed representation of additive secret shares of a weight-one vector in $\mathbb{F}^n$. A work of de Castro and Polychroniadou [21] gives a technique that two parties, each holding a DPF key, can use to verify that their keys together indeed represent a vector of Hamming-weight one ($B = \mathbb{F}$ in our notation). When their protocol applies, it is extremely efficient. However, their protocol only applies to a specific tree-based DPF construction, thus not applying to input vectors that are secret-shared directly or encrypted. Moreover, their protocol cannot check further properties of the non-zero element (i.e., $B = \{0, 1\}$), which is critical for many applications.

Arithmetic sketching schemes are closely related to fully linear probabilistically checkable proofs [9] (FL-PCPs). Essentially an FL-PCP for a language $\mathcal{L} \subseteq \mathbb{F}^n$ is a Merlin-Arthur analogue of arithmetic sketching scheme. In an FL-PCP, the verifier's queries compute an inner product with the input x concatenated with a proof string π, whose length may grow with the input length n. (The number of queries is typically constant in n.) If $x \in \mathcal{L}$, there is a proof π that causes the verifier to accept. If $x \notin \mathcal{L}$ the verifier almost always rejects. Arithmetic sketching schemes are thus FL-PCPs with an empty proof string. FL-PCPs are more powerful than arithmetic sketching schemes—given a large enough proof, they can check whether the input satisfies an arbitrary arithmetic circuit. Yet FL-PCPs require additional proof, which can be costly in communication or infeasible when there is no one party that knows the full input.

Traditional sketching data structures [2, 15, 41], which summarize a large data stream in small space, inspire our approach. As we discuss in Section 2, the arithmetic nature of our sketches means that most traditional sketching techniques do not naturally apply in our setting.

Our sketches for low-weight vectors (Section 4) view the secret-shared vector as coefficients of a polynomial and then use polynomial-sparsity testing [30] to test

that the coefficient vector has low weight. Ghosh and Simkin [27] use polynomial-sparsity testing as a low-weight test in the setting of private set intersection. There, each of two parties holds a set—represented as its characteristic vector—and they view the difference of their vectors as the coefficients of a polynomial. This coefficient vector will be sparse if and only if the input sets differ in few elements.

Finally, the combination of secure computation and sketching was previously considered in the context of sublinear-communication secure computation protocols for approximations [25, 32, 35]. Since the focus of these works was on crude asymptotic efficiency rather than concrete efficiency, they could rely on traditional (non-arithmetic) linear sketching. Moreover, in contrast to the approximate sketching problems considered in these works, here we consider arithmetic sketching schemes that yield the exact output except with negligible failure probability.

Notation. For a finite set S, we use $x \xleftarrow{R} S$ to denote a uniformly random sample from S. We use $x \stackrel{\text{def}}{=} 3$ to note definition and $\langle \cdot, \cdot \rangle$ to denote inner product. For a finite field $\mathbb{F}$, when $|\mathbb{F}| > n$ we use $1, 2, 3, \ldots, n$ to denote distinct non-zero field elements.

2 A Formalization of Arithmetic Sketching Schemes

We characterize our goal by the following notion of an *arithmetic sketching* scheme. Our definition (Section 2.2) is roughly analogous to the definition of fully linear PCPs [9], with the role of the witness and prover removed.

2.1 Overview

We define arithmetic sketching schemes with respect to a finite field $\mathbb{F}$, a dimension $n \in \mathbb{N}$, and a language $\mathcal{L} \subseteq \mathbb{F}^n$. For a given input $x \in \mathbb{F}^n$, the task is to determine whether $x \in \mathcal{L}$ given only the output of a *linear function* applied to the input vector x. That is, in a arithmetic sketching scheme with sketch size ℓ, we consider a verifier who uses randomness to produce a sketching matrix $Q \in \mathbb{F}^{\ell \times n}$. The verifier receives the matrix-vector product of the sketching matrix Q with the input vector x: that is, $Q \cdot x \in \mathbb{F}^\ell$. The verifier decides whether to accept ($x \in \mathcal{L}$) or reject ($x \notin \mathcal{L}$) by running a decision algorithm D on the vector $Q \cdot x$.

All of the specific languages $\mathcal{L}$ we will consider in this work are simple, and can be decided in linear time in the input size. What makes constructing arithmetic sketching schemes non-trivial is that (1) the decision algorithm takes as input a small linear sketch of the (large) input vector and (2) the decision algorithm must be a small arithmetic circuit, as we discuss now.

Arithmetic verifier. We will typically consider infinite families of languages $\mathcal{L}$, parameterized by the field $\mathbb{F}$ and input length n. For example, we could consider the family of all unit vectors in $\mathbb{F}^n$. We will require our arithmetic sketching constructions to have an *arithmetic verifier*: that is, the decision algorithm D

applies an arithmetic circuit to the sketch, and accepts if and only if the circuit outputs the all-zeros vector in $\mathbb{F}$. Restricting the decision circuit to being arithmetic means that it is possible to compute with good concrete efficiency in a secure multi-party computation, as in our motivating applications. (Later, in Section 4.1, we will also consider relaxed arithmetic sketching schemes with partially arithmetic decision predicates.)

Universal family of sketching schemes. We require by default that the family be *universal* in the sense that both the sketch size ℓ and the decision algorithm D are independent of the field $\mathbb{F}$ and input size n.

Complexity measures. In our constructions, we will aim to minimize the *sketch size* ℓ (influencing the computation time), the algebraic *degree* of D (influencing the round complexity or security threshold), and the multiplicative *size* of D (number of multiplication gates, influencing communication complexity).

2.2 Formal definitions

We now formalize the above discussion.

Definition 1 (Arithmetic sketching scheme: Syntax)**.** *Let $\mathbb{F}$ be a finite field. A* arithmetic sketching *scheme for a language $\mathcal{L} \subseteq \mathbb{F}^n$ with sketch size $\ell \in \mathbb{N}$ consists of a pair of algorithms (S, D):*

- *$S() \to Q \in \mathbb{F}^{\ell \times n}$. The randomized sketching algorithm outputs a query matrix $Q \in \mathbb{F}^{\ell \times n}$. (We also refer to Q as the "sketching matrix.")*
- *$D(a) \to y \in \mathbb{F}^m$. The decision algorithm takes as input a sketch vector $a \in \mathbb{F}^\ell$ and outputs a vector $y \in \mathbb{F}^m$. (To enable stronger security, we will later allow the decision algorithm D to be randomized, taking random field elements $r_1, \ldots, r_k$ as additional inputs.)*

We assume by default that the decision algorithm D is implemented by an arithmetic circuit over $\mathbb{F}$, consisting of addition, subtraction and multiplication gates, as well as a unit gate outputting the constant $\mathbf{1} \in \mathbb{F}$. We measure the complexity of D by its algebraic degree *and* multiplicative size *(counting the number of multiplication gates).*

A arithmetic sketching scheme as above must satisfy the following properties.

Definition 2 (Completness and soundness)**.** *A arithmetic sketching scheme (S, D) for a language $\mathcal{L} \subseteq \mathbb{F}^n$ is*

- ***Complete** if, for all $x \in \mathcal{L}$, the verifier accepts:*

$$\Pr[D(Q \cdot x) = \mathbf{0}^m : Q \leftarrow S()] = 1.$$

  (If this probability is at least $1 - \epsilon$, we say that the arithmetic sketching scheme has completeness error ϵ.)
- ***Sound** with soundness error ϵ if, for all $x \notin \mathcal{L}$, the acceptance probability, as in the completeness definition, is at most ϵ.*

A few remarks on the definition of arithmetic sketching schemes:

Linear decision not interesting. The decision algorithm D must contain at least one multiplication gate. If not, the arithmetic sketching accepts the vectors in the kernel of a linear map and thus can only test linear predicates.

Non-arithmetic sketching. We restrict the decision algorithm D to have low arithmetic complexity. Without this restriction, any language $\mathcal{L} \subseteq \mathbb{F}^n$ can be decided using a sketch of size $\ell = O(\log |\mathcal{L}|)$. The sketch, given input $x \in \mathbb{F}^n$, just computes a random linear combination $r \leftarrow R(x) \in \mathbb{F}^\ell$ of the input vector. The decision predicate searches—via brute-force search—for a value $x' \in \mathcal{L}$ such that $R(x') = r$ and accepts if, and only if, one exists. Since a random linear code has good distance, the sketch will be sound. When the decision predicate must be a small arithmetic circuit, constructing arithmetic sketching schemes is non-trivial, even in the information-theoretic setting.

2.3 Zero knowledge

In the context of cryptographic applications, it is important that if $x \in \mathcal{L}$, then the sketching-based verification reveal nothing beyond the fact that $x \in \mathcal{L}$. For the above notion of arithmetic sketching, this *zero-knowledge* property is automatically guaranteed by the completeness requirement if we use a secure computation protocol to compute the output of the decision algorithm D. Indeed, for $x \in \mathcal{L}$, the output of D is always $\mathbf{0}^m$. However, some applications motivate a stronger notion of *two-sided zero knowledge*, requiring that the output of the decision algorithm D hides all information about the input x apart from whether $x \in \mathcal{L}$. To this end, we allow the decision algorithm D to be randomized, taking secret random field elements as additional inputs.

Definition 3 (Two-sided zero knowledge)**.** *We say that an arithmetic sketching scheme has* δ-two-sided zero knowledge *if there exists a simulator* Sim *such that for all* $x \in \mathbb{F}^n$*, the following distributions are* δ*-close in statistical distance:*

$$\mathcal{D}_{\text{ideal}} = \mathsf{Sim}(\mathbb{1}\{x \in \mathcal{L}\}) \quad // \ \mathsf{Sim} \text{ gets the bit indicating whether } x \in \mathcal{L}$$

$$\mathcal{D}_{\text{real}} = \left\{ \left(Q, v\right) : \begin{array}{l} Q \leftarrow S() \\ r \xleftarrow{\text{R}} \mathbb{F}^k \\ v \leftarrow D\big(Q \cdot x; r\big) \end{array} \right\}.$$

If $\delta = 0$*, we say that the scheme satisfies* p*erfect zero knowledge.*

An arithmetic sketching scheme may not necessarily satisfy two-sided zero knowledge. In particular, if $x \notin \mathcal{L}$ the output of D could leak additional information about the input x. But the following generic transformation, whose proof we give in Appendix A, uses a standard random-linear-combination technique to modify any arithmetic sketching scheme into one with two-sided zero knowledge at a small additional cost:

Fact 4. *It is possible to modify any arithmetic sketching scheme over finite field* $\mathbb{F}$ *with decision predicate* $D\colon \mathbb{F}^\ell \times \mathbb{F}^k \to \mathbb{F}^m$ *to satisfy perfect two-sided zero*

knowledge by allowing D to use m secret random field elements. This modification increases the multiplicative size of D by m, the algebraic degree of D by 1, and the soundness error by $1/|\mathbb{F}|$.

3 Sketching via Algebraic Manipulation Detection

We will be particularly interested in languages consisting of vectors whose Hamming weight is at most 1, possibly with additional restrictions on the nonzero entry. (For simplicity, in the following we will sometimes use "weight 1" to refer to weight *at most* 1.) In this section, we present a new framework for constructing arithmetic sketching schemes for such languages. The framework captures several existing ad-hoc constructions in a unified way and gives rise to efficient new constructions. The central tool is a new object that we call an *algebraic manipulation detection (AMD) distribution*. This notion is inspired by the notion of AMD codes of Cramer et al. [18], extending it to provide limited forms of *targeted malleability* [11].

Background: AMD codes. An AMD code gives a (randomized) way to encode a message into a codeword in a way that allows detection of "additive tampering" of the encoded message. In particular, an AMD code consists of a randomized encoder $\mathsf{Enc}\colon \mathbb{F}^n \to \mathbb{F}^\ell$ and decoder $\mathsf{Dec}\colon \mathbb{F}^\ell \to \mathbb{F}^n$ such that:

- it is possible to recover the message from the codeword: for all messages $m \in \mathbb{F}^n$, $\Pr[m = \mathsf{Dec}(\mathsf{Enc}(m))] = 1$, and
- the codeword shifted by any additive offset either decodes to the correct message m or a failure symbol $\perp$: for all messages $m \in \mathbb{F}^n$ and $\Delta \in F^\ell$,

$$\Pr\big[\mathsf{Dec}(\mathsf{Enc}(m) + \Delta) \notin \{m, \perp\}\big] \leq \epsilon,$$

  where the probability is taken over the randomness of Enc and ϵ is some small value $\epsilon \approx 1/|\mathbb{F}|$.

Our notion: AMD distributions. Our notion of AMD distributions differs from standard AMD codes in three ways:

1. First, there is no encoder and no message to be encoded. Instead we define a sampling algorithm S_{AMD} that outputs, effectively, a random codeword r.
2. Second, we demand that attempting to decode a codeword r that has been shifted by a non-zero additive offset results in a decoding failure with high probability. (In traditional AMD codes, a shifted codeword may decode to the failure symbol $\perp$ *or* to the original message.) This stronger AMD requirement appears in several works on secure computation [26].
3. Third, and most important, we demand that the manipulation-detection property of the coding scheme holds even for *affine* tampering: that is, taking r to $\beta r + \Delta$ for nonzero scalar β and vector Δ. (Traditional AMD codes detect only additive shifts.) In some cases, we allow decoding to succeed if the codeword r is multiplied by an "allowable" value $\beta \in B$, for some prespecified set $B \subseteq \mathbb{F}$. We call these "B-multiplicative" AMD distributions.

Looking ahead, we will show that any B-multiplicative AMD distribution yields a sketching scheme for the language of vectors of weight one whose payload is in the set B.

3.1 Definition

We now make our notion of AMD distributions formal.

Definition 5 (AMD Distribution)**.** *An* algebraic manipulation detection (AMD) distribution *over finite field $\mathbb{F}$ with codeword length $\ell \in \mathbb{N}$ and error ϵ is given by a pair of procedures:*

- $S_{\mathsf{AMD}}() \to r \in \mathbb{F}^\ell$*. A randomized algorithm that samples a codeword r.*
- $V_{\mathsf{AMD}}(r) \to y \in \mathbb{F}$*. A verification algorithm, represented as an arithmetic circuit, that accepts a vector $r \in \mathbb{F}^\ell$ and outputs an element $y \in \mathbb{F}$. We interpret the output as accepting if $y = 0$ and rejecting otherwise.*

These procedures must satisfy the following properties:

- **Nontriviality:** *The distribution of S_{AMD} is supported by at least two vectors.*
- **Completeness:** *We have that:*

$$\Pr[V_{\mathsf{AMD}}(r) = 0 : r \leftarrow S_{\mathsf{AMD}}()] = 1.$$

- **Affine manipulation detection:** *For every scalar $\beta \in \mathbb{F}$ and additive offset $\Delta \in \mathbb{F}^\ell$ with $\beta \neq 0$ and $\Delta \neq \mathbf{0}^\ell$, it holds that*

$$\Pr[V_{\mathsf{AMD}}(\beta r + \Delta) = 0 : r \leftarrow S_{\mathsf{AMD}}()] \leq \epsilon.$$

Remark. In Section 3.5, we will consider a more refined notion of AMD distributions, requiring that the decoder detect any nontrivial linear combination of two or more (distinct) random samples from S_{AMD} with high probability.

For our applications, we will often need AMD distributions that satisfy forms of *targeted malleability*. That is, for some set $B \subseteq \mathbb{F}$, we would like the AMD decoder to accept codewords that have been scaled by a constant $\beta \in B$ and to reject, with high probability, all codewords scaled by constants $\beta' \in \mathbb{F} \setminus B$.

Note that the completeness requirement implies that $1 \in B$. In the context of the sketching application this is without loss of generality, since for any $\beta \in B \setminus \{0\}$ scaling the queries by β effectively converts B to $\beta^{-1} \cdot B$.

The following definition captures the above notion of targeted malleability:

Definition 6 (B-Multiplicative AMD Distribution)**.** *Let $\mathbb{F}$ be a finite field and let $B \subseteq \mathbb{F}$ such that $1 \in B$. We say that an AMD distribution $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ over $\mathbb{F}$ and with error ϵ is* B-multiplicative *if it additionally satisfies the following properties:*

- *For every $\beta \in B$, it holds that $\Pr[V_{\mathsf{AMD}}(\beta r) = 0 : r \leftarrow S_{\mathsf{AMD}}] = 1$.*
- *For every $\beta' \in \mathbb{F} \setminus B$, it holds that $\Pr[V_{\mathsf{AMD}}(\beta' r) = 0 : r \leftarrow S_{\mathsf{AMD}}] \leq \epsilon$.*

We will need the following lemma:

Lemma 7. *Let $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ be an AMD distribution with codeword length ℓ over finite field $\mathbb{F}$, where $|\mathbb{F}| > 2$, with error ϵ. Then for all $r_0 \in \mathbb{F}^\ell$, we have $\Pr[S_{\mathsf{AMD}} = r_0] \leq \epsilon$.*

Proof. Towards a contradiction, assume $\Pr[S_{\mathsf{AMD}} = r_0] > \epsilon$. If $r_0 = \mathbf{0}$, then we can set $\beta = 1$ and Δ to be any non-zero vector accepted by V_{AMD}, which exists by the non-triviality requirement. Otherwise, if $r_0 \neq \mathbf{0}$, pick any $\beta \in \mathbb{F} \setminus \{0, 1\}$, and let $\Delta = -(\beta - 1)r_0 \in \mathbb{F}^\ell$. Note that $\beta \neq 0$ and $\Delta \neq \mathbf{0}$. Then we have the following, where the probabilities are all taken over the random choice of $r \leftarrow S_{\mathsf{AMD}}$:

$$\begin{aligned}
&\Pr[V_{\mathsf{AMD}}(\beta r + \Delta) = 0] \\
&\quad \geq \Pr[V_{\mathsf{AMD}}(\beta r + \Delta) = 0 \mid r = r_0] \cdot \Pr[r = r_0] \\
&\quad \geq \Pr[V_{\mathsf{AMD}}(\beta r_0 - (\beta - 1)r_0) = 0 \mid r = r_0] \cdot \Pr[r = r_0] \\
&\quad \geq \Pr[V_{\mathsf{AMD}}(r_0) = 0 \mid r = r_0] \cdot \Pr[r = r_0] \\
&\quad > 1 \cdot \epsilon.
\end{aligned}$$

We derive the last line by invoking (a) the completeness of the AMD distribution and (b) our initial assumption. This contradicts the AMD property of the distribution. □

3.2 From AMD distributions to sketching schemes

We now use AMD distributions to construct arithmetic sketching schemes for the language of vectors of Hamming weight (at most) one, whose non-zero entry lies within a specific set B. For our discussion, the following notation will be useful:

Definition 8 (Language $\mathcal{L}_B$)**.** *For a finite field $\mathbb{F}$, dimension $n \in \mathbb{N}$, and set $B \subseteq F$, we define the language $\mathcal{L}_B$ to be the set of B-multiples of unit vectors in $\mathbb{F}^n$. That is, $\mathcal{L}_B = \{\beta \cdot \boldsymbol{e}_i \mid i \in [n], \beta \in B\}$, where $\boldsymbol{e}_i \in \mathbb{F}^n$ is the vector of zeros with a single one at coordinate i.*

The following construction shows that B-multiplicative AMD distributions immediately give rise to arithmetic sketching schemes for the language $\mathcal{L}_B$.

Construction 9 (Sketch for Hamming-weight one from AMD distributions)**.** *The construction is parameterized by a finite field $\mathbb{F}$, an input size $n \in \mathbb{N}$, a set $B \subseteq \mathbb{F}$ such that $1 \in B$, and a B-multiplicative AMD distribution $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ with codeword length $\ell \in \mathbb{N}$. The arithmetic sketching scheme (S, D) is over the field $\mathbb{F}$, has sketch size ℓ, and is defined as follows:*

- $S() \to Q \in \mathbb{F}^{\ell \times n}$.
  - *For $i \in [n]$, compute $r_i \xleftarrow{R} S_{\mathsf{AMD}}() \in \mathbb{F}^\ell$.*
  - *Output the matrix $Q \in \mathbb{F}^{\ell \times n}$ whose columns are $(r_1, \ldots, r_n)$.*
- $D(a) \to y \in \mathbb{F}^m$.
  - *Output $V_{\mathsf{AMD}}(a)$.*

Theorem 10. *If $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ is a B-multiplicative AMD distribution over finite field $\mathbb{F}$ ($|\mathbb{F}| > 2$) with codeword length ℓ and error ϵ then, for all $n \in \mathbb{N}$, Construction 9 instantiated with $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ is an arithmetic sketching scheme for the language $\mathcal{L}_B \subseteq \mathbb{F}^n$ (as in Definition 8) with soundness error 2ϵ.*

Proof. To show completeness: We can write any valid input $x \in \mathcal{L}_B$ as $x = \beta \boldsymbol{e}_i$ for some $\beta \in B$ and $i \in [n]$. Then, for sketching query matrix $Q \leftarrow S_{\mathsf{AMD}}()$, we have that $a = Q \cdot x = Q \cdot (\beta \boldsymbol{e}_i) \in \mathbb{F}^\ell$; i.e., the i-th column of the query matrix multiplied by the scalar β. By construction, the i-th column of the matrix Q is computed as $(r_{i1}, \ldots, r_{i\ell}) \leftarrow S_{\mathsf{AMD}}()$. So the sketch decision procedure outputs $D(\beta r_{i1}, \ldots, \beta r_{i\ell}) = V_{\mathsf{AMD}}(\beta r_{i1}, \ldots, \beta r_{i\ell})$. This value is always 0 by the B-multiplicative property of the AMD distribution $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$, so the verifier always accepts.

To show soundness: Let $x' \notin \mathcal{L}_B$. There are two cases.

- Case 1: $x' = \beta' \boldsymbol{e}_i$ for some $i \in [n]$, $\beta' \notin B$.
  By the B-multiplicative property of the AMD distribution:

$$\begin{aligned} & \Pr\left[D(Q \cdot x') = \mathbf{0} \colon Q \leftarrow S()\right] \\ = \ & \Pr\left[V_{\mathsf{AMD}}(\beta' r_{i1}, \ldots, \beta' r_{i\ell}) = \mathbf{0} \colon (r_{i1}, \ldots, r_{i\ell}) \leftarrow S_{\mathsf{AMD}}()\right] \\ \leq \ & \epsilon. \end{aligned}$$

- Case 2: x' has Hamming weight $k \geq 2$. Letting Q be the random $\ell \times n$ query matrix, we can write the answer vector $a = Q \cdot x'$ as a linear combination of $k \geq 2$ independent samples r^i from the AMD distribution S_{AMD}. Namely, $a = \sum_{i=1}^{k} \beta_i r^i$ for nonzero $\beta_i \in \mathbb{F}$. Letting $\Delta = \sum_{i=2}^{k} \beta_i r^i$, we can now write:

$$\begin{aligned} \Pr\left[D(a) = 0\right] &= \Pr\left[V_{\mathsf{AMD}}(\beta_1 r^1 + \Delta) = 0\right] \\ &\leq \Pr\left[\Delta = 0\right] + \Pr\left[V_{\mathsf{AMD}}(\beta_1 r^1 + \Delta) = 0 \,|\, \Delta \neq 0\right] \\ &\leq \epsilon + \epsilon = 2\epsilon, \end{aligned}$$

  where the last inequality follows from Lemma 7 and the affine manipulation detection property of S_{AMD}. □

3.3 Constructing AMD distributions from algebraic varieties

We now show how to construct AMD distributions based on algebraic varieties. These AMD distributions, via Theorem 10, give rise to practical arithmetic sketching schemes for useful instances of the Hamming-weight one languages $\mathcal{L}_B$ that minimize the relevant complexity measures.

Construction 11 (AMD distribution from varieties)**.** *The construction is parameterized by a finite field $\mathbb{F}$, randomness complexity $k \in \mathbb{N}$, a codeword size $\ell \in \mathbb{N}$, an arithmetic AMD sampler, defined by a polynomial map $\mathbf{g} \colon \mathbb{F}^k \to \mathbb{F}^\ell$, and an AMD verifier, defined by a polynomial $f \in \mathbb{F}[R_1, \ldots, R_\ell]$. The construction is then:*

- $S_{\mathsf{AMD}}() \to r \in \mathbb{F}^\ell$*:*

  - *Sample a random vector* $(s_1, \ldots, s_k) \xleftarrow{R} \mathbb{F}^k$.
  - *Output* $r \leftarrow \mathbf{g}(s_1, \ldots, s_k) \in \mathbb{F}^\ell$.
- $V_{\mathsf{AMD}}(r_1, \ldots, r_\ell) \to y \in \mathbb{F}$*: Output* $y \leftarrow f(r_1, \ldots, r_\ell) \in \mathbb{F}$.

The following theorem follows almost immediately by construction:

Theorem 12. *When instantiated with a polynomial map* $\mathbf{g}\colon \mathbb{F}^k \to \mathbb{F}^\ell$ *with total degree* $\deg \mathbf{g}$ *and polynomial* $f \in \mathbb{F}[R_1, \ldots, R_\ell]$ *with total degree* $\deg f$*, Construction 11 is a* B*-multiplicative AMD distribution, in the sense of Definition 6, with soundness error* $(\deg f \cdot \deg \mathbf{g})/|\mathbb{F}|$ *when:*

(1) [for completeness] the polynomial $f \circ \mathbf{g} \equiv 0 \in \mathbb{F}[S_1, \ldots, S_k]$*,*
(2) [for AMD] for all non-zero $\beta \in \mathbb{F}$ *and non-zero* $\Delta \in \mathbb{F}^\ell$*, the formal polynomial* $f \circ (\beta\mathbf{g} + \Delta) \not\equiv 0 \in \mathbb{F}[S_1, \ldots, S_k]$*, and*
(3) [for B*-multiplicativity] it holds that* $f \circ \beta\mathbf{g} \equiv 0$ *if and only if* $\beta \in B$*.*

Proof. For completeness: By property (1), $V_{\mathsf{AMD}}(S_{\mathsf{AMD}}) = f(\mathbf{g}(s_1, \ldots, s_k)) = 0$.

For AMD: When at least one of $\beta \in \mathbb{F}$ and $\Delta \in \mathbb{F}^\ell$ is non-zero, we have:

$$\Pr[V_{\mathsf{AMD}}(\beta r + \Delta) = 0 \colon r \leftarrow S_{\mathsf{AMD}}()] = \Pr[f(\beta\mathbf{g}(s) + \Delta) \colon s \xleftarrow{R} \mathbb{F}^k].$$

By property (2), $f \circ (\beta\mathbf{g} + \Delta) \in \mathbb{F}[S_1, \ldots, S_k]$ is a non-zero polynomial, and its total degree is at most $\deg f \cdot \deg \mathbf{g}$. Then by the Schwartz-Zippel Lemma, the probability that a random point in $\mathbb{F}^k$ is a zero is at most $(\deg f \cdot \deg \mathbf{g})/|\mathbb{F}|$.

For B-multiplicativity: we invoke property (3). When $\beta \in B$, the argument is the same as for completeness. When $\beta \notin B$, the argument is the same as for AMD. □

If the pair $(f, \mathbf{g})$ satisfy properties (1) and (2) of Theorem 12, we can use the theorem to characterize the B-multiplicativity properties of the AMD distribution resulting from Construction 11. In particular, when f has monomials of only two degrees, we have the following:

- **One degree:** Suppose f is homogeneous; i.e., every monomial of f has the same total degree $d \geq 0$. Then the pair $(f, \mathbf{g})$ give a B-multiplicative AMD distribution for $B = \mathbb{F}$. That is because, since for all $\beta \in \mathbb{F}$,

  $$f \circ \beta\mathbf{g} = \beta^d \cdot (f \circ \mathbf{g}) \equiv 0,$$

  where the final step holds by the completeness guarantee $f \circ \mathbf{g} \equiv 0$.
- **Two degrees:** Suppose $f(R_1, \ldots, R_\ell) = f_1(R_1, \ldots, R_\ell) + f_2(R_1, \ldots, R_\ell)$, where $f_1, f_2 \not\equiv 0$ and every monomial of f_1 (respectively, f_2) has total degree $d_1 + d_2$ (resp, d_2), where $d_1 > 0, d_2 \geq 0$. Then

  $$\begin{aligned} f \circ \beta\mathbf{g} &= f_1 \circ \beta\mathbf{g} + f_2 \circ \beta\mathbf{g} \\ &= \beta^{d_1+d_2} \cdot (f_1 \circ \mathbf{g}) + \beta^{d_2} \cdot (f_2 \circ \mathbf{g}) \\ &= \beta^{d_2}(1 - \beta^{d_1}) \cdot (f_2 \circ \mathbf{g}), \end{aligned}$$

  where the final step holds since $f_1 \circ \mathbf{g} \equiv -f_2 \circ \mathbf{g}$ by the completeness guarantee. Since $f_2 \not\equiv 0$, this polynomial is the zero polynomial precisely for values of β which annihilate the prefixed multiplicative term. Depending on the choices of d_1, d_2, this directly corresponds to the sets $B = \{1\}, \{0, 1\}, \{-1, 1\}, \{-1, 0, 1\}$.

3.4 New sketching schemes for vectors of weight one (or at most one)

We now consider several useful instantiations of Construction 11, including both new sketching schemes and abstractions of existing ones.

New instantiations. Our characterization also yields useful new arithmetic sketching schemes for the case $\mathrm{char}(\mathbb{F}) > 2$:

$$\begin{aligned} B = \{-1,0,1\}: &\qquad f(r_1,r_2) = r_1^3 - r_2 &&\qquad \mathbf{g}(s) = (s, s^3) \\ B = \mathbb{F}: &\qquad f(r_1,r_2,r_3) = r_1 r_2 - r_3^2 &&\qquad \mathbf{g}(s_1,s_2) = (s_1^2, s_2^2, s_1 s_2). \end{aligned}$$

Constructions from [14]. The following sketching schemes from [14] fit directly into the characterization above, yielding a simple unified derivation based on Theorem 12:

$$\begin{aligned} B = \{0,1\}: &\qquad f(r_1,r_2,r_3) = r_3 - r_1 r_2 &&\qquad \mathbf{g}(s_1,s_2) = (s_1, s_2, s_1 s_2) \\ B = \{0,1\}: &\qquad f(r_1,r_2) = r_1^2 - r_2 &&\qquad \mathbf{g}(s) = (s, s^2) \\ B = \{1\}: &\qquad f(r_1,r_2) = (r_1+1)^2 - r_2 &&\qquad \mathbf{g}(s) = (s, (s+1)^2), \end{aligned}$$

where the last two schemes require $\mathrm{char}(\mathbb{F}) > 2$.

We illustrate checking the conditions of Theorem 12 for the case $B = \mathbb{F}$:

- Completeness: $f \circ \mathbf{g} = s_1^2 s_2^2 - (s_1 s_2)^2 \equiv 0$.
- AMD: for $\beta \in \mathbb{F}$ and $\Delta = (\Delta_1, \Delta_2, \Delta_3) \in \mathbb{F}^3$, we have $f \circ (\beta \mathbf{g} + \Delta)(s_1, s_2) = \beta\Delta_2 s_1^2 + \beta\Delta_1 s_2^2 - 2\beta\Delta_3 s_1 s_2 + (\Delta_1\Delta_2 - \Delta_3^2)$. For any non-zero β and non-zero Δ, this polynomial in (s_1, s_2) is not identically zero.
- B-multiplicative: Since f is homogeneous of degree two, it holds by the completeness property that $f \circ \beta\mathbf{g} \equiv 0$ for all $\beta \in \mathbb{F}$.

We summarize the above by the following theorem.

Theorem 13 (Arithmetic sketching with a single decision polynomial). *Let n be a positive integer and $\mathbb{F}$ be a finite field with char$(\mathbb{F}) > 2$. There are arithmetic sketching schemes for the language $\mathcal{L}_B \subseteq \mathbb{F}^n$ (as in Definition 8), with soundness error $O(1/|\mathbb{F}|)$ and a single decision polynomial f, for the following choices of B.*

- *$B = \{0,1\}$: sketch size 2, f of degree 2 and multiplicative size 1;*
- *$B = \{1\}$: sketch size 2, f of degree 2 and multiplicative size 1;*
- *$B = \{-1,0,1\}$: sketch size 2, f of degree 3 and multiplicative size 2;*
- *$B = \mathbb{F}$: sketch size 3, f of degree 2 and multiplicative size 2.*

Jumping ahead, in Theorem 30 we will show that the extra query required for the case $B = \mathbb{F}$ is not an artifact of our construction, and it is in fact necessary even if a decision algorithm D consisting of multiple decision polynomials f_i is allowed.

Remark (General $\mathcal{L}_B$). An arithmetic sketching scheme for $\mathcal{L}_B$ with an arbitrary $B \subset \mathbb{F}$ of size k can be reduced to the case of $\mathcal{L}_{\mathbb{F}}$ by adding the single additional query $q_0 = (1, 1, \ldots, 1)$ and another decision polynomial of degree k, checking that $a_0 = \langle x, q \rangle \in B$. However, this general solution is significantly less efficient than the sketching schemes from Theorem 13.

3.5 A sketch with $1/|\mathbb{F}|^2$ soundness for binary weight at most 1

In Section 3.2 we used AMD distributions to construct sketching schemes for the language $\mathcal{L}_B \subseteq \mathbb{F}^n$ (Definition 8) for certain sets $B \subseteq \mathbb{F}$. When $\mathrm{char}(\mathbb{F}) > 2$, the sketch from Section 3.2 for $\mathcal{L}_{\{0,1\}}$ contains two elements and has soundness error $O(1/|\mathbb{F}|)$.

In this section we construct a sketch for $\mathcal{L}_{\{0,1\}}$ that contains *three* elements and has soundness error $O(1/|\mathbb{F}|^2)$. One can repeat the sketch for $\mathcal{L}_{\{0,1\}}$ from Section 3.2 twice to obtain the same soundness level, however the resulting sketch would have length four, whereas our sketch has length three.

The previous notion of AMD distributions is insufficient for this purpose. First, the verification algorithm $V_{\mathsf{AMD}}(r)$ outputs only a single field element, which is insufficient to achieve soundness error $\epsilon < 1/|\mathbb{F}|$. More inherently, the failure probability ϵ must be bigger than the inverse of the support size of S_{AMD}, since a successful additive attack can be based on a guess of the codeword r. In the following, we will modify original notion of AMD distributions (Definition 5) to address both limitations. Concretely, the new notion allows $V_{\mathsf{AMD}}(r)$ to output multiple field elements, and requires detection of linear combinations of two or more *distinct* samples from S_{AMD}. Somewhat surprisingly, the latter requirement will allow us to break the inverse-support-size barrier.

Definition 14 (Low-Error AMD Distribution)**.** *A* low-error AMD distribution *with error ϵ is defined similarly to a standard AMD distribution from Definition 5, with the following modified syntax and detection property.*

- **Syntax:** $V_{\mathsf{AMD}}(r) \to y \in \mathbb{F}^m$*, namely the verification circuit outputs m field elements. We interpret the all-0 output as accepting.*
- **Completeness:** $\Pr\left[V_{\mathsf{AMD}}(r) = 0^m \ : \ r \leftarrow S_{\mathsf{AMD}}()\right] = 1$.
- **Detection:** *Let S be the support size of S_{AMD} and let $n \in \mathbb{N}$ satisfy $2 \leq n < S$. Let $S^{(n)}_{\mathsf{AMD}}$ denote the probability distribution of n independent samples from S_{AMD},* conditioned on all samples being distinct*. Then for all $(\beta_1, \ldots, \beta_n)$ in $\mathbb{F}^n \setminus \mathcal{L}_{\{0,1\}}$ it holds that*

$$\Pr\left[V_{\mathsf{AMD}}(\beta_1 r^{(1)} + \ldots + \beta_k r^{(n)}) = 0^m : (r^{(1)}, \ldots, r^{(n)}) \leftarrow S^{(n)}_{\mathsf{AMD}}\right] \leq \epsilon. \quad (1)$$

A low-error AMD distribution as above naturally gives rise to an arithmetic sketching scheme detecting inputs in $\mathcal{L}_{\{0,1\}}$, as in Construction 9. The only difference is that instead of sampling the columns of the sketching matrix uniformly at random from S_{AMD}, their choice is now conditioned on the distinctness requirement. We next show how to instantiate this approach with S_{AMD} that samples powers of a random field element.

Construction 15 (A low-error AMD distribution)**.** *The construction is parameterized by a codeword length $\ell \in \mathbb{N}$, a finite field $\mathbb{F}$, and a polynomial map $\mathbf{f} : \mathbb{F}^\ell \to \mathbb{F}^m$. The construction is then:*

- $S_{\mathsf{AMD}}() \to r \in \mathbb{F}^\ell$*:*
  - *Sample a random* $s \xleftarrow{R} \mathbb{F}^*$.
  - *Output* $r \leftarrow (s, s^2, \ldots, s^\ell) \in \mathbb{F}^\ell$
- $V_{\mathsf{AMD}}(r_1, \ldots, r_\ell) \to y \in \mathbb{F}^m$*: Output* $\mathbf{f}(r_1, \ldots, r_\ell) \in \mathbb{F}^m$.

The next theorem shows that instantiating this construction with $\ell = 3$, which implies a sketch of size three, gives a low-error AMD distribution with error $\epsilon = O(1/|\mathbb{F}|^2)$.

Theorem 16. *Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be a field where $|\mathbb{F}| > n^2$. Then the S_{AMD} in Construction 15, instantiated with $\ell = 3$, $m = 2$, and the polynomial map*

$$\mathbf{f}(r_1, r_2, r_3) = \left(r_1^2 - r_2, \quad r_3 - r_1 r_2\right) \in \mathbb{F}^2$$

is a low-error AMD distribution (as in Def. 14) with error $\epsilon \leq 6e/(|\mathbb{F}|^2 - 3|\mathbb{F}|)$.

Proof idea. We give the full proof in Appendix B. Completeness follows directly from the construction. To prove the detection property defined in (1), we have two cases: First, suppose the input vector has weight 1 but that its non-zero element u, say in position i, is not in $\{0, 1\}$. Then $u^2 \neq u$ and therefore $(s_i u)^2 \neq s_i^2 u$ because $s_i \in \mathbb{F}^*$. This means that $r_1^2 - r_2 \neq 0$ as required. Second, suppose that the input vector β has weight w greater than one. Define the set $(\mathbb{F}_{\neq})^n$ as the set of all n-tuples in $(\mathbb{F}^*)^n$ whose elements are pairwise distinct. Then for the sketch to fail, the random vector $\mathbf{s} \in (\mathbb{F}_{\neq})^n$, whose elements are pairwise distinct, must lie on the intersection of the two n-variate polynomials

$$\langle \mathbf{X}, \beta \rangle^2 = \langle \mathbf{X}^2, \beta \rangle \qquad \text{and} \qquad \langle \mathbf{X}^3, \beta \rangle = \langle \mathbf{X}, \beta \rangle \cdot \langle \mathbf{X}^2, \beta \rangle. \tag{2}$$

For a fixed β of weight at least two, we will appeal to Bézout's theorem to argue that the intersection contains at most $6|\mathbb{F}|^{n-2}$ points $\mathbf{s} \in (\mathbb{F}_{\neq})^n$. Given that, we can bound the detection error as required. □

In this subsection we used sketching matrices that are restricted to contain distinct non-zero columns. This restriction can also be applied to the sketches in Theorem 13. For example, for the $B = \{0, 1\}$ sketch of Theorem 13, whose exact soundness error can be shown to be about $2/|\mathbb{F}|$, the soundness error can be reduced by about a factor of two by ensuring that the sampled sketching matrix contains distinct non-zero columns, as we do in this section.

4 Sketching for Low-Weight Vectors

In this section, we construct sketching schemes for languages of vectors of low Hamming weight.

Throughout this section, when working with input vectors of dimension n, we assume that the finite field $\mathbb{F}$ over which we work contains an element of order at least n. If the field $\mathbb{F}$ is too small for such an element to exist, we can lift the input vector to an extension $\mathbb{K}/\mathbb{F}$ such that $|\mathbb{K}| \geq n$. When working with a secret-shared input vector, this lifting requires no communication between the verifiers. Using such an extension increases the communication cost, in terms of bits, by at most a multiplicative $\lceil \log_2 n \rceil$ factor.

4.1 Refined definitions: Arithmetic sketching with private decision

In this section, we give refined definitions of arithmetic sketching schemes, which give more efficient constructions when implementing the decision predicate in a multiparty computation (as in many applications). In particular, we now split the decision predicate D into two parts: a *private predicate* D^{priv} and a *public predicate* D^{pub}:

- The private predicate, D^{priv}, operates on the verifier's sketch of the input vector x and evaluates a (typically small) randomized arithmetic circuit on the sketch. The output of D^{priv} is essentially a "sanitized" version of the sketch—the sanitized sketch leaks nothing about the input x, in a sense that we will define shortly. In applications, we will typically evaluate D^{priv} via secure multiparty computation.
- The public predicate, D^{pub}, takes the "sanitized" sketch as input and determines whether the verifier will accept or reject. The public predicate may compute a complicated high-degree function of its inputs. Since the inputs to D^{pub} leak nothing about the instance x, in applications, it is safe to publish these values and compute D^{pub} in the clear.

The refined syntax is then:

- $S() \to Q$. The sketching algorithm outputs a query matrix $Q \in \mathbb{F}^{\ell \times n}$.
- $D^{\text{priv}}(a, (r_1, \ldots, r_k)) \to y \in \mathbb{F}^m$. The private decision algorithm takes as input the sketch $a = Q \cdot x \in \mathbb{F}^\ell$ and random field elements $r_1, \ldots, r_k$, and outputs a vector $y \in \mathbb{F}^m$.
- $D^{\text{pub}}(y) \to \{0, 1\}$. The public decision algorithm takes as input a vector $y \in \mathbb{F}^m$ and outputs an accept/reject bit.

Completeness and soundness are as in the standard definition of arithmetic sketching schemes (Definition 2) with the combined decision algorithm $D^{\text{pub}} \circ D^{\text{priv}}$ replacing the single algorithm D. For zero knowledge, we require that the output of the *private* decision algorithm leak nothing about the input x apart from the fact that $x \in \mathcal{L}$.

Definition 17 (Zero knowledge with split decision predicate)**.** *We say that an arithmetic sketching scheme $(S, D^{\text{priv}}, D^{\text{pub}})$ for a language $\mathcal{L} \subseteq \mathbb{F}$ satisfies δ-zero knowledge if there exists a simulator* Sim *such that for all $x \in \mathcal{L}$, the following*

distributions are δ-close in statistical distance:

$$\mathcal{D}_{\text{ideal}} = \mathsf{Sim}()$$
$$\mathcal{D}_{\text{real}} = \left\{ \Big(Q,\ D^{\text{priv}}\big(Q \cdot x, r\big)\Big) \colon \begin{array}{c} Q \leftarrow S() \\ r \xleftarrow{\text{R}} \mathbb{F}^k \end{array} \right\}.$$

If an arithmetic sketching scheme has δ-zero knowledge for $\delta = 0$, we say that the sketch has perfect zero knowledge*.*

Our notion of zero knowledge is analogous to the notion of *strong* zero-knowledge in the setting of zero-knowledge proof systems on secret-shared data [9].

Fact 4 shows that when the decision algorithm is a single arithmetic circuit, it is easy to modify any arithmetic sketching scheme to satisfy the stronger notion of *two-sided* zero knowledge (Definition 3) by augmenting the decision algorithm with a bit of additional randomness. Since the decision routine here may have a non-algebraic (high-degree) public predicate, it is no longer always easy to add two-sided zero knowledge to such arithmetic sketching schemes.

4.2 Weight-w vectors with arbitrary payload

We first give a linear sketch-verification scheme that recognizes vectors of Hamming weight w with arbitrary payload.

Given an input vector $x = (x_0, \ldots, x_{n-1}) \in \mathbb{F}^n$, we view the vector x as holding the coefficients of the polynomial $p(Z) = \sum_{i=0}^{n-1} x_i Z^i \in \mathbb{F}[Z]$. Notice that the number of non-zero coefficients of the polynomial p is exactly equal to the number of non-zero elements of the input vector x. Furthermore, for all $r \in \mathbb{F}$, it is possible to compute the evaluation of the polynomial p at the point r by taking a single linear combination of the elements of x, of the form $(1, r, r^2, \ldots, r^{n-1}) \in \mathbb{F}^n$. So, if we can test whether a polynomial p has w non-zero coefficients using ℓ polynomial-evaluation queries to p, we can test whether the input vector x has w non-zero coefficients using a linear sketch of size ℓ. By applying existing algorithms for testing whether a polynomial's coefficient vector is sparse [30], we thus construct linear sketch-verification schemes for low-weight vectors.

The sketch-verification schemes in this section rely on the following lemma, which appears in a similar form in the work of Grigorescu, Jung, and Rubinfeld [30]:

Lemma 18 (Ben-Or and Tiwari [7])**.** *Let $\mathbb{F}$ be a finite field, let $p \in \mathbb{F}[U]$ be a polynomial, and let ℓ be a non-negative integer. Then for any $u \in \mathbb{F}$, define the Hankel matrix $H_p(u)$ as:*

$$H_p(u) \stackrel{\text{def}}{=} \begin{pmatrix} p(1) & p(u) & p(u^2) & \ldots & p(u^\ell) \\ p(u) & p(u^2) & p(u^3) & \ldots & p(u^{\ell+1}) \\ p(u^2) & p(u^3) & p(u^4) & \ldots & p(u^{\ell+2}) \\ \vdots & \vdots & & \ddots & \vdots \\ p(u^\ell) & p(u^{\ell+1}) & p(u^{\ell+2}) & \ldots & p(u^{2\ell}) \end{pmatrix} \in \mathbb{F}^{(\ell+1)\times(\ell+1)}.$$

Then for a polynomial p with w non-zero cofficients,

- *if $w > \ell$, the determinant $\det(H_p(U))$ is a non-zero polynomial in U of degree at most $2\binom{\ell+1}{2}\deg(p)$, and*
- *if $w \leq \ell$, the determinant $\det(H_p(U)) \equiv 0$, as a polynomial in U.*

In particular, if the polynomial p of Lemma 18 has more than ℓ non-zero coefficients, then for a random $u \xleftarrow{R} \mathbb{F}$, the matrix $H_p(u)$ will be full rank and its determinant $\det(H_p(u))$ will be non-zero with high probability. Moreover, if the polynomial p has $w \leq \ell$ non-zero coefficients, the matrix $H_p(u)$ will be of rank w, with high probability over the random choice of $u \xleftarrow{R} \mathbb{F}$. We prove the following in Appendix D:

Corollary 19. *Let $\mathbb{F}$ be a finite field, let $p \in \mathbb{F}[U]$ be a polynomial, and let ℓ be a non-negative integer. If p has ℓ non-zero coefficients then, if we sample $u \xleftarrow{R} \mathbb{F}$, for the matrix $H_p(u) \in \mathbb{F}^{(\ell+1)\times(\ell+1)}$ defined in Lemma 18, it holds that:*

$$\Pr\left[\,\mathrm{rank}(H_p(u)) = \ell \colon u \xleftarrow{R} \mathbb{F}\right] \geq 1 - 2\binom{\ell}{2}\frac{\deg(p)}{|\mathbb{F}|}.$$

We will also need the following standard fact, proven in Appendix D:

Fact 20. *Let $\mathbb{F}$ be a finite field and let n be a positive integer. Then the probability that a random matrix $R \xleftarrow{R} \mathbb{F}^{n\times n}$ is full rank is at least $\frac{1}{|\mathbb{F}|-1}$.*

Construction 21 (Sketching for Hamming weight w)**.** *The construction is parameterized by a finite field $\mathbb{F}$, and integers w and n with $w < n < |\mathbb{F}|$. The sketch accepts the language of vectors in $\mathbb{F}^n$ of Hamming weight exactly w. The scheme has sketch size $\ell = 2w + 1$, completeness error $2/(|\mathbb{F}| - 1)$ soundness error $O(w^2 n/|\mathbb{F}|)$, and a decision algorithm implementable by an arithmetic circuit with $O(w^\omega)$ multiplication gates, where $\omega < 2.38$ is the (algebraic) matrix-multiplication constant.*

- $S() \to Q$.
  - *Sample a random value $u \xleftarrow{R} \mathbb{F}$.*
  - *Return*

$$Q = \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & u & u^2 & u^3 & \dots & u^{(n-1)} \\ 1 & u^2 & u^4 & u^6 & \dots & u^{2(n-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u^{\ell-1} & u^{2(\ell-1)} & u^{3(\ell-1)} & \dots & u^{(\ell-1)(n-1)} \end{pmatrix} \in \mathbb{F}^{\ell\times n}.$$

- $D^{\mathrm{priv}}(a, (r_1, \dots, r_k)) \to y \in \mathbb{F}^m$.

- *Use the sketch $a \in \mathbb{F}^\ell$ of size $\ell = 2w+1$ to form a Hankel matrix H:*

$$H = \begin{pmatrix} a_1 & a_2 & a_3 & \dots & a_{w+1} \\ a_2 & a_3 & a_4 & \dots & a_{w+2} \\ a_3 & a_4 & a_5 & \dots & a_{w+3} \\ \vdots & \vdots & & \ddots & \vdots \\ a_{w+1} & a_{w+2} & a_{w+3} & \dots & a_{2w+1} \end{pmatrix} \in \mathbb{F}^{(w+1)\times(w+1)}.$$

  - *Use the $k = 2(w+1)^2$ random values $r_1, \dots, r_k \in \mathbb{F}$ given as input to sample two square matrices $R_1, R_2 \in \mathbb{F}^{(w+1)\times(w+1)}$.*
  - *Compute the matrix-matrix product $R_1 \cdot H \cdot R_2 \in \mathbb{F}^{(w+1)\times(w+1)}$.*
  - *Output this product as a vector of dimension $m = (w+1)^2$.*
- $D^{\text{pub}}(y) \to \{0,1\}$. *Interpret the input $y \in \mathbb{F}^m$ as a matrix Y of dimension $(w+1)\times(w+1)$. Accept if and only if* $\operatorname{rank}(Y) = w$.

Theorem 22. *Construction 21 has completeness error $\frac{2}{|\mathbb{F}|-1}$, soundness error $O(w^2 n/|\mathbb{F}|)$, and δ-zero knowledge, in the sense of Definition 17, for $\delta = O(w^2 n/|\mathbb{F}|)$.*

Proof of Theorem 22. For completeness: By Corollary 19, when the input vector x has Hamming weight w, the matrix H will have rank w with probability $1 - O(w^2 n/|\mathbb{F}|)$. Then provided that the random matrices R_1 and R_2 that D^{priv} samples are of full rank, the matrix $R_1 \cdot H \cdot R_2$ will have rank w and the verifier will accept. By Fact 20 and the union bound, the probability that either matrix fails to be invertible is $2/(|\mathbb{F}|-1)$.

For soundness: When the input vector x has Hamming weight $w' < w$, the matrix H has rank $w' < w$ (by Lemma 18), so the matrix $H \cdot R$ has rank at most w' and the verifier will always reject. When the input vector x has Hamming weight $w' > w$, again by Lemma 18, the determinant is a polynomial in u of degree $O(w^2 n)$. So in this case the determinant is non-zero, and the verifier will accept with probability $O(w^2 n/|\mathbb{F}|)$.

For zero knowledge: The simulator takes a fixed matrix $M \in \mathbb{F}^{(w+1)\times(w+1)}$ of rank w and outputs $R_1 \cdot M \cdot R_2$ for random matrices $R_1, R_2 \in \mathbb{F}^{(w+1)\times(w+1)}$. To explain why the simulation is correct, if x has Hamming weight w, the matrix H has rank w with probability $1 - O(w^2 n/|\mathbb{F}|)$, by Corollary 19. (If $\operatorname{rank}(H) \neq w$, the simulation fails.) Multiplying a rank-w matrix M on the left and right by random matrices is always distributed in the same way, regardless of the choice of the rank-w matrix M [33]. □

Bounded weight. Construction 21 accepts vectors of Hamming weight *exactly* w. In some applications, we want to accept the language $\mathcal{L}_B^{\times \le w}$ of vectors of Hammming weight *at most* w. We prove the following in Appendix D:

Corollary 23. *There is an explicit arithmetic sketching scheme for the language of vectors in $\mathbb{F}^n$ of Hamming weight* at most *w. The scheme has sketch size $2w+1$, completeness error $2/(|\mathbb{F}|-1)$ soundness error $O(w^2 n/|\mathbb{F}|)$, δ-zero knowledge,*

in the sense of Definition 17, for $\delta = O(w^2 n/|\mathbb{F}|)$, and has a private decision predicate that can be implemented using an arithmetic circuit of $\mathsf{poly}(w)$ *gates and degree three.*

4.3 Sketching for vectors with L_1 norm w

Let w and n be a positive integers, and let $\mathbb{F}$ be a finite field of characteristic greater than wn. In this section we give a linear sketch-verification scheme that recognizes vectors of a bounded L_1 norm, specifically the set

$$\mathcal{L}_1^{(=w)} \stackrel{\text{def}}{=} \left\{ (x_1, \ldots, x_n) \in \mathbb{F}^n \ : \ \sum_{i=1}^{n} x_i = w, \quad x_i \in \{0, \ldots, w\} \right\}. \tag{3}$$

This sketch is useful for voting or private-telemetry applications in which each client casts w votes for n candidates, while allowing the client to vote for the same candidate multiple times.

To outline the main technical idea: given an input vector $x = (x_1, \ldots, x_n) \in \mathbb{F}^n$, the sketch chooses a random vector $(r_1, \ldots, r_n) \in \mathbb{F}^n$ and evaluates the power sums

$$p_j \stackrel{\text{def}}{=} \sum_{i=1}^{n} x_i \cdot r_i^j \in \mathbb{F}, \quad \text{for } j = 1, \ldots, w+1 \tag{4}$$

using a total of $w+1$ linear queries to x (i.e., using a sketch of size $w+1$). To decide whether or not to accept the vector x, we apply the Newton identities (Theorem 24) to the sketch $(p_1, \ldots, p_{w+1})$ to obtain the quantities e_w and e_{w+1} in $\mathbb{F}$. Then the decision algorithm accepts the instance x if $e_{w+1} = 0$ and $e_w \neq 0$.

The sketch can be adapted to test membership in $\mathcal{L}_1^{(\leq w)}$ where the equality in (3) is changed to $\sum_{i=1}^{n} x_i \leq w$. The only modification to the sketch is that x is accepted whenever $e_{w+1} = 0$ (we drop the check that $e_w \neq 0$).

To explain why this approach is sound, let us first review the Newton identities. For $w \geq 0$

- let $P_w(X_1, \ldots, X_n)$ be the w-th power sum polynomial defined as $P_w(X_1, \ldots, X_n) = \sum_{j=1}^{n} X_j^w$.
- Let $E_w(X_1, \ldots, X_n)$ be the w-th symmetric polynomial, defined as $E_w(X_1, \ldots, X_n) = \sum_{1 \leq j_1 < j_2 < \cdots < j_w \leq n} X_{j_1} X_{j_2} \ldots X_{j_w}$.

The Newton identities relate these two families via a recurrence relation.

Theorem 24 (Newton identities [39]). *For all finite fields $\mathbb{F}$ and integers $k, n \in \mathbb{N}$ with $k \leq n$:*

$$k \cdot E_k(X_1, \ldots, X_n) = \sum_{i=1}^{k} (-1)^{i-1} \cdot E_{k-i}(X_1, \ldots, X_n) \cdot P_i(X_1, \ldots, X_n).$$

The recurrence lets us express E_k purely in terms of $P_1, \ldots, P_k$. For example

$$E_1 = P_1, \qquad 2E_2 = P_1^2 - P_2, \qquad 6E_3 = P_1^3 - 3P_1P_2 + 2P_3, \tag{5}$$

and so on. In other words, for every $k > 0$ there is a polynomial N_k such that $E_k = N_k(P_1, \ldots, P_k)$. This N_k has total degree k.

Now, observe that if $(r_1, \ldots, r_n) \in \mathbb{F}^n$ is a vector that has w or fewer non-zero entries, then $E_{w+1}(r_1, \ldots, r_n) = 0$. However, if a random vector $(r_1, \ldots, r_n) \in \mathbb{F}^n$ has more than w non-zero entries, then $E_{w+1}(r_1, \ldots, r_n)$ is very likely to be non-zero. With this in mind, we can present the sketch.

Construction 25 (Sketching for L_1 weight w)**.** *The construction is parameterized by integers w and n and a finite field $\mathbb{F}$ of characteristic greater than nw. The sketch accepts the language of vectors $\mathcal{L}_1^{(=w)}$ in $\mathbb{F}^n$. The scheme has sketch size $w+1$, soundness error $(w+1)/|\mathbb{F}|$, and a decision predicate with $O(w^2)$ multiplication gates.*

- $S() \to (q_1, \ldots, q_{w+1})$.
  - *Sample random values $r_1, \ldots, r_n \xleftarrow{R} \mathbb{F}$.*
  - *Return*
$$Q = \begin{pmatrix} r_1 & r_2 & r_3 & r_4 & \cdots & r_n \\ r_1^2 & r_2^2 & r_3^2 & r_4^2 & \cdots & r_n^2 \\ \vdots & \vdots & & \ddots & & \vdots \\ r_1^\ell & r_2^\ell & r_3^\ell & r_4^\ell & \cdots & r_n^\ell \end{pmatrix} \in \mathbb{F}^{\ell \times n}.$$
- $D^{\text{priv}}\big((p_1, \ldots, p_{w+1}), (d_1, d_2)\big) \to y \in \mathbb{F}^2$.
  - *Use the $\ell = w+1$ sketch values $(p_1, \ldots, p_{w+1})$ to compute the quantities e_w and e_{w+1} using (5), namely, $e_w = N_w(p_1, \ldots, p_w)$ and $e_{w+1} = N_{w+1}(p_1, \ldots, p_{w+1})$.*
  - *Use the two random values $d_1, d_2 \in \mathbb{F}$ to blind e_w and e_{w+1} by computing $\hat{e}_w = e_w \cdot d_1$ and $\hat{e}_{w+1} = e_{w+1} \cdot d_2$.*
  - *Output $y = (\hat{e}_w, \hat{e}_{w+1}) \in \mathbb{F}^2$.*
- $D^{\text{pub}}(y) \to \{0, 1\}$. *Parse $y = (\hat{e}_w, \hat{e}_{w+1}) \in \mathbb{F}^2$. Accept if and only if $\hat{e}_w \neq 0$ and $\hat{e}_{w+1} = 0$. (Completeness fails when $d_1 = 0$.)*

Theorem 26. *Construction 25 is complete (with completeness error $1/|\mathbb{F}|$), sound, and δ-zero knowledge, in the sense of Definition 17, for $\delta = (w+2)/|\mathbb{F}|$. Moreover, the scheme has a private decision predicate that can be implemented using either:*

- *an arithmetic circuit of size $O(w^2)$ and of degree $w+2$, or*
- *an arithmetic circuit of size $\mathsf{poly}(w)$ and of degree 3.*

Zero knowledge when $x \in \mathcal{L}_1^{(=w)}$ is immediate. To obtain the degree-three circuit computing the decision predicate, we apply standard results on randomized encodings [34]. All that remains it to prove completeness and soundness of the construction, which we do with the following theorem, proved in Appendix D.1:

Theorem 27. *Let $\mathbb{F}$ be a finite field of characteristic greater than nw. Then the following holds for all $x = (x_1, \ldots, x_n) \in \mathbb{F}^n$. Sample a random $(r_1, \ldots, r_n) \in \mathbb{F}^n$, compute the power sums $p_1, \ldots, p_{w+1} \in \mathbb{F}$ as in (4), and use (5) to compute $e_{w+1} \in \mathbb{F}$ as $e_{w+1} = N_{w+1}(p_1, \ldots, p_{w+1})$. Then*

- *if $x \in \mathcal{L}_1^{(\leq w)}$ then $\Pr[e_{w+1} = 0] = 1$,*
- *if $x \notin \mathcal{L}_1^{(\leq w)}$ then $\Pr[e_{w+1} = 0] \leq (w+1)/|\mathbb{F}|$.*

Here the probability is over the random choice of $(r_1, \ldots, r_n)$ in $\mathbb{F}^n$.

Construction 25 uses Theorem 27 twice: once to prove that $x \in \mathcal{L}_1^{(\leq w)}$ and once to prove that $x \notin \mathcal{L}_1^{(\leq w-1)}$. Together this proves that x is in $\mathcal{L}_1^{(=w)}$.

Binary vectors of Hamming weight w. By combining Construction 21 and Construction 25 we obtain a sketch that accepts vectors $x \in \mathbb{F}^n$ that (i) have Hamming weight w and (ii) are in $\mathcal{L}_1^{(=w)}$. The only such vectors are vectors in $\{0,1\}^n$ that have Hamming weight w. We have:

Corollary 28. *There is an arithmetic sketch for binary vectors of Hamming weight w; the cost is that of running both Construction 21 and Construction 25.*

Vectors of Hamming weight w with equal non-zero entries. To test if a vector is of Hamming weight w with all non-zero entries equal to one another, the verifier can apply a sketch vector of the form $(1, 1, 1, \ldots, 1)$ to the input to compute the sum $\sigma \in \mathbb{F}$ of the entries of the input vector. If the input vector $x \in \mathbb{F}^n$ is well formed, then the value $\sigma w^{-1} \in \mathbb{F}$ is the value of the non-zero entries. Then the vector $(w \cdot \sigma^{-1})x \in \mathbb{F}^n$ is a binary vector of Hamming weight w if and only if $x \in \mathbb{F}^n$ is a vector of Hamming weight w with equal non-zero entries. The verifier can then test that the vector $(w \cdot \sigma^{-1})x \in \mathbb{F}^n$ is a binary vector of Hamming weight w. If so, the verifier can conclude that x is of Hamming weight w with equal non-zero entries.

More generally, the verifier can apply an arbitrary arithmetic circuit $C \colon \mathbb{F} \to \mathbb{F}$ to $\sigma w^{-1} \in \mathbb{F}$, such as a bounds check, to ensure that all of the non-zero entries are equal and satisfy circuit C.

4.4 Bounded-weight vectors with arbitrarily restricted payloads

In this section, we construct arithmetic sketching schemes for vectors in $\mathbb{F}^n$ with exactly w non-zero entries *and* where each non-zero entry satisfies a arithmetic circuit $C \colon \mathbb{F} \to \mathbb{F}$. (We assume that $C(0) \neq 0 \in \mathbb{F}$. Otherwise we can use the remark below to sketch for the langauge of vectors of Hamming weight at most w whose non-zero entries satisfy C.) We have:

Theorem 29. *For all $\lambda, w, n \in \mathbb{N}$ with $w \leq n^{1/2} < |\mathbb{F}|$, and arithmetic circuits $C \colon \mathbb{F} \to \mathbb{F}$ (with size independent of $\mathbb{F}$), there is a arithmetic sketching scheme for the language of vectors in $x \in \mathbb{F}^n$ of Hamming weight w such that each non-zero element satisfies C. The scheme (Construction 46) has completeness error $\epsilon = 4/(|\mathbb{F}| - 1) + 2^{-\lambda}$, soundness error $O(w^2 n/|\mathbb{F}|)$, sketch size $O(w\lambda)$ and δ-zero knowledge, in the sense of Definition 17, for $\delta = O(w^2 n/|\mathbb{F}|) + 2^{-\lambda}$.*

Proof idea. The full proof appears in Appendix D.2. This sketch has two parts:

- First, we check that the input vector has Hamming weight exactly w, using Construction 21.
- Next, we repeat the following test λ times and accept if all accept:
  - Hash the n components of the vector down into w^2 "bins." That is, we partition the n components of the input vector into w^2 chunks at random construct a test vector $t = (t_1, \ldots, t_{w^2}) \in \mathbb{F}^{w^2}$ where t_i is the sum of the entries of x in chunk i.
  - Let $C'(z) \stackrel{\text{def}}{=} z \cdot C(z) \in \mathbb{F}$ be circuit that accepts all inputs that C does in addition to the value $0 \in \mathbb{F}$.
  - Accept if $t \in \mathbb{F}^{w^2}$ has Hamming weight less than w (i.e., two non-zero elements of x landed in the same bucket) OR if, for the circuit C', $C'(t_1) = C'(t_2) = \cdots = C'(t_{w^2}) = 0 \in \mathbb{F}$.

The first check ensures an upper bound on the Hamming weight of the vector. The second check ensures that all of the non-zero entries of the vector satisfy the circuit C. □

Remark. The same arithmetic sketching construction as in Theorem 29 serves to check that a vector in $\mathbb{F}^n$ has *at most* w non-zero entries and where each non-zero entry satisfies an arithmetic circuit $C \colon \mathbb{F} \to \mathbb{F}$. To do so, we change only the first step: just check that the input vector has at most w non-zero entries, using Corollary 23, instead of checking that the input vector has Hamming weight exactly w.

5 From Arithmetic Sketching to Client-Server Protocols

As discussed, a natural motivating application scenario of our arithmetic sketching schemes is in a client-server setting, where a client secret shares a sensitive vector across servers, and the servers must verify that the shared vector has a proper form. The linear nature of the sketching scheme enables the servers to individually compute shares of the sketch of the input. The low arithmetic complexity of the decision algorithm D means that servers can securely secure evaluate D on their shares of the sketch using a via multi-party computation (MPC). If the arithmetic sketching scheme satisfies the property of zero knowledge from Definition 17, then the corresponding client-server verification scheme further provides zero-knowledge guarantees against *semi-honest* servers. We refer the reader to prior work [9] for a detailed treatment of this transformation in the semi-honest model. Indeed, our arithmetic sketching schemes can be seen as a special case of *fully linear probabilistically checkable proofs* (FL-PCP) from prior work [9], without a Prover procedure.

However, note that the properties of the arithmetic sketching scheme do not directly give any guarantees against a potentially adversarial verifier. Indeed, in the solutions thus far, a *malicious* server may be able to reveal information about the input. In what follows, we analyze this more adversarial setting, and provide general techniques for achieving privacy against malicious verifiers.

We give the technical details on our client-server schemes in Appendix C, and we give an overview of the results here.

Auxiliary client information required. We first observe that, in general, achieving security against a malicious verifier/server *requires* the client to send additional information to the servers, beyond simply its secret shared vector. In particular, without some form of auxiliary client information π, the scheme will inevitably be subject to a selective-failure attack. Consider, for example, the language $\mathcal{L}_{\{0,1\}} \subseteq \mathbb{F}^n$ of binary vectors of weight at most 1. Given only secret shares of x, a malicious server can learn whether x is the all-zero vector, by adding $+1$ offset to one position of his secret share, and seeing if the resulting vector is accepted (or whether the vector is any other particular unit vector, by analogous adjustments).

Given this state of affairs, we thus turn back to the more general notion of fully linear PCPs from [9], incorporating also a procedure for the client to generate auxiliary information to provide to the servers. We provide a formal definition in Section C.1. Note that in the definition of FL-PCP, queries are restricted to make linear access to both the input x *and* the auxiliary proof material π, i.e., in our notation the sketch is computed as $Q \cdot (x \| \pi)$ for sketching matrix Q over $\mathbb{F}$. Requiring the sketch to be a linear function of the proof π is not inherent, but will be convenient for us. In particular, the client can send the auxiliary information π to the servers in *additive secret-shared form*, meaning that we can include sensitive information within π to aid in authentication without revealing it to malicious servers.

Toward malicious security. We next observe that a malicious-secure protocol must enforce the correct execution of the sketching algorithm S and private decision algorithm D, even in the presence of malicious servers. We can achieve this using malicious-secure multiparty computation protocols. Given the low arithmetic complexity of the decision algorithm D, this resulting overhead will be minimal. In most cases, to run the sketching algorithm S in a multiparty computation, the verifiers may simply run a coin tossing protocol to sample the randomness for query generation.

Privacy against additive *attacks.* At this point, only one attack surface remains: incorrectly performing query evaluation. Equivalently, the adversary's remaining power is to submit improper inputs to the secure computation protocol for the decision algorithm D. Recall the inputs to the execution of D are (allegedly) each server's additive share of the query answers. Since the verifiers only hold additive secret shares of both the input x and proof material π by design, the adversarially chosen input must be independent of these secret values; this corresponds to an arbitrary fixed *additive* offset attack on the combined value. That is, a malicious verifier can learn the output of D on the shifted vector $a + \Delta \in \mathbb{F}^\ell$ where $a \in \mathbb{F}^\ell$ is the honest sketch value and $\Delta \in \mathbb{F}^\ell$ is an adversarially chosen additive offset.

This kind of additive attack can reveal sensitive information, as the selective-failure attack discussed above demonstrates. Even with auxiliary proof material π, one must be careful. Suppose, for example, the value of π for x depends only on

the product of two symbols $x_i x_j$ and not on the symbols themselves; a malicious verifier can then learn whether $x_i = 0$ for a secret input x by adding a garbage offset to his share of x_j and seeing if the value is still accepted.

We formulate a notion of *additive-attack privacy* for the FL-PCP, strengthening zero knowledge by requiring no information is revealed even in the face of additive attacks. Utilizing an FL-PCP with this extra property, together with MPC secure against malicious parties for jointly executing the sketching algorithm S and decision algorithm D suffices to yield a secure client-server protocol in the manner described above.

Privacy against additive attacks via AMD distributions. As the final step, we develop a general approach for obtaining this notion of additive-attack privacy. This will be done by once again making use of *algebraic manipulation detection (AMD) distributions*, as put forth in Section 3. We show that if the distribution of (honestly generated) FL-PCP query and answer values satisfies a form of AMD guarantee, then the FL-PCP indeed provides privacy against additive attacks. Then, we provide general transformations for "hardening" general FL-PCPs to ones providing this additional AMD property, with mild overhead.

In the general case, the additional authentication values introduced in our approach result in a constant multiplicative overhead above the secret sharing of the vector x. However, this approach is particularly useful for the setting of sparse vectors x (as is predominantly the focus of this paper), in which the cost of secret sharing the additional proof information adds little overhead beyond secret sharing the input itself. See also the discussion in Section C.2.

6 Lower Bound on Sketch Size

In this section we establish a lower bound on the sketch size of an arithmetic sketching scheme for weight-one vectors using an algebraic argument. We show that the soundness error of an arithmetic sketching scheme for this language depends on its sketch size. More precisely, the soundness error is at least $\frac{1}{(d+1)|\mathbb{F}|^{\ell-2}}$ if the sketch size is ℓ and the algebraic degree of the decision algorithm is d. Consequently, any arithmetic sketching scheme with sketch size only two has constant soundness error. This bound proves that the positive result of Section 3 constructing a scheme for weight-one vectors with sketch size $\ell = 3$ is optimal for any scheme with soundness error of the form $\frac{f(n)}{|\mathbb{F}|}$ for some function f.

In the rest of the section we slightly abuse the terminology and say that any scheme with soundness error larger than $\frac{f(n)}{|\mathbb{F}|}$ is not an arithmetic sketching scheme, instead of stating the exact error. Additional terminology and notation follow. Let $\mathcal{L}_{\mathbb{F}}$ be the language of vectors of weight at most 1 and length $n \geq 1$ over a field $\mathbb{F}$. That is, $\mathcal{L}_{\mathbb{F}} = \{\beta \boldsymbol{e}_i : i \in [n], \beta \in \mathbb{F}\} \subset \mathbb{F}^n$ for unit vectors $e_i \in \mathbb{F}^n$. Recall that the decision predicate $D : \mathbb{F}^\ell \to \mathbb{F}^m$ is arithmetic and therefore $D(a_1, \ldots, a_\ell) = (z_1, \ldots, z_m)$ can be represented as m polynomials $D_1, \ldots, D_m$ over $\mathbb{F}$ such that $D_j(a_1, \ldots, a_\ell) = z_j$.

Theorem 30 states the main result of this section, which bounds the soundness error of an arithmetic sketching scheme for $\mathcal{L}_{\mathbb{F}}$ as a function of its sketch size. We prove Theorem 30 in Appendix E.

Theorem 30. *Let (S, D) be a arithmetic sketching scheme for the language $\mathcal{L}_{\mathbb{F}}$. If the sketch size of the scheme is ℓ, and the algebraic degree of D is d then the soundness error of the scheme is at least $\frac{1}{(d+1)|\mathbb{F}|^{\ell-2}}$.*

Proof idea. The full proof appears in Appendix E. The proof proceeds in three steps. First, we prove in Lemma 47 that any polynomial D_j is a sum of homogeneous polynomials that all vanish on any possible column of the sketching matrix. Then, we prove that the set of roots of a homogeneous polynomial over a field is a union of a bounded number of linear subspaces of $\mathbb{F}^\ell$ that are of dimension 1 or 0. Finally, we use the fact that each column in the sketching matrix must necessarily be a root of D_j to prove that the decision procedure will accept a vector with two non-zero locations if the associated columns of the sketching matrix happen to be part of the same linear subspace. The probability of that event occurring is too high if the sketch size is low. □

Corollary 31. *Any arithmetic sketching scheme (S, D) for $\mathcal{L}_{\mathbb{F}}$ with sketch size two has soundness error at least $\frac{1}{d+1}$. Therefore, any arithmetic sketching scheme for $\mathcal{L}_{\mathbb{F}}$ with $O(|\mathbb{F}|^{-1})$ soundness error must have sketch size at least three.*

7 Languages Without Arithmetic Sketching Schemes

Consider two parties, one holding an input $x \in \mathbb{F}^n$, and the other holding an input $y \in \mathbb{F}^n$ who wish to determine if $(x, y) \in \mathcal{L}'$ for some language $\mathcal{L}' \subseteq (\mathbb{F}^n)^2$, using public coins. Let $R(\mathcal{L}')$ denote the minimal communication complexity to decide whether $(x, y) \in \mathcal{L}'$ for all x, y with probability bounded away from $1/2$.

An arithmetic sketching scheme for the language $\mathcal{L} = \{x + y \mid (x, y) \in \mathcal{L}'\}$ induces an instance of a protocol that decides $\mathcal{L}'$ in two steps. First, the two parties derive the sketching matrix from the public random coins, and locally compute shares of the sketch on $x + y$ using linearity. Then, the parties use an interactive protocol to compute the decision on the sketch.

The communication complexity of this type of scheme depends only on the decision algorithm D. However, since the families of sketching schemes we consider are universal, D is independent of $\mathbb{F}$ and n, i.e. the communication complexity of the scheme is $O(1)$.

In this section, we use known lower bounds on the communication complexity of any protocol that computes a language $\mathcal{L}'$ to derive lower bounds on the size of a decision circuit D in an arithmetic sketching scheme for the associated language $\mathcal{L}$. Indeed, we show that several natural languages $\mathcal{L}$ do not have a universal family of arithmetic sketching scheme by proving that their decision circuits must depend on n or on $\mathbb{F}$.

The communication complexity lower bounds we use are on the problems of Set Disjointness (DISJ) and Greater Than (GT). In Set Disjointness the inputs

x, y are subsets of a universe $\{1, \ldots, n\}$, and the goal is to decide the language $\mathcal{L}^{DISJ} = \{(x, y) \subseteq \{1, \ldots, n\} \mid x \cap y = \emptyset\}$. A series of works [5,37,46] established that $R(DISJ) = \Theta(n)$. The inputs x, y in the problem of Greater Than are the binary representation of two non-negative integers, and the goal is to decide the language $\mathcal{L}^{GT} = \{(x, y) \mid x > y\}$. Viola [49] proved that $R(\mathcal{L}^{GT}) = \Omega(\log n)$.

7.1 L_p norm

An arithmetic sketch for the language of vectors with L_1 norm equal to some w is presented in Section 4.3. It is natural to ask whether this construction can be generalized to an arithmetic sketch for L_p such that $p > 1$ is a constant integer. Even though L_p is not necessarily a norm over the field, computing it as a function is interesting and useful.

Previous work [3,38] on computing moments in the streaming model [2] with low space complexity is sufficient to establish that arithmetic sketching schemes for the L_p norm are impossible. In this model there are n real-valued changes to a given vector of real numbers, and the goal is estimate the L_p norm of the vector with small error and minimal space. Translating the bounds on space complexity from the streaming model to sketch size in an arithmetic sketching shows that $\ell = \Omega(\log n)$ for L_2 and $\ell = \Theta(n^{1-2/p} \log n)$ for $p > 2$. However, the streaming model accepts a decision algorithm that is not necessarily arithmetic.

In this section we show a tighter bound for arithmetic sketching schemes via a communication complexity argument. Specifically, we show that the decision algorithm for L_p, $p > 1$, is of multiplicative size $\Omega(n)$ via a reduction of a general protocol for DISJ to an arithmetic sketching for L_p.

More formally, for an integer $p \geq 2$ and a field $\mathbb{F}$ such that p and $|\mathbb{F}| - 1$ are co-prime let $\mathcal{L}_p^{(=w)} \stackrel{\text{def}}{=} \left\{(x_1, \ldots, x_n) \in \mathbb{F}^n \ : \ \left(\sum_{i=1}^n x_i^p\right)^{1/p} = w\right\}$. This definition of L_p restricts the field to ensure that a p-th root exists for every field element. For L_2, which is the most useful case of L_p, we also consider the norm squared, which can be defined for all fields. Define the language $\mathcal{L}_2^{2(=w)} \stackrel{\text{def}}{=} \left\{(x_1, \ldots, x_n) \in \mathbb{F}^n \ : \ \sum_{i=1}^n x_i^2 = w\right\}$.

The proof of the following theorem appears in Appendix F:

Theorem 32. *Let $n, p \geq 2$ be integers, $p \geq 2$, let $\mathbb{F}$ be a finite field such that $|\mathbb{F}| - 1$ and p are co-prime, and let $\mathbb{E}$ be a finite field, $|\mathbb{E}| > 2$. If $w_f \in \mathbb{F}$ and $w_e \in \mathbb{E}$ then the decision algorithm for any arithmetic sketching scheme for the languages $\mathcal{L}_p^{(=w_f)}$ over $\mathbb{F}$ or $\mathcal{L}_2^{2(=w_e)}$ over $\mathbb{E}$ is of multiplicative size $\Omega(n)$.*

7.2 Specified value in arbitrary vector

In Section 3 we construct arithmetic sketching schemes for the language $\mathcal{L}_B$ of B-multiples of unit vectors, i.e. vectors in which all entries are 0, except for one entry which is in the set $B \subseteq \mathbb{F}$. It seems natural to ask whether there is an arithmetic sketching scheme for the language of all vectors in which one entry is in B, while all other entries are arbitrary elements in $\mathbb{F}$. The next theorem proves that this language, namely $\mathcal{L}_{B,\mathbb{F}} \stackrel{\text{def}}{=} \{(x_1, \ldots, x_n) \in \mathbb{F}^n \mid \exists i, x_i \in B \ \}$, has

no arithmetic sketching scheme for B that is not too large, via a reduction from DISJ. We prove the following theorem in Appendix F:

Theorem 33. *Let $n \geq 2$ be an integer, let $\mathbb{F}$ be a field, and let $B \subseteq \mathbb{F}$ such that $\frac{|B|}{|\mathbb{F}|} \leq \frac{1}{3n}$. Then, the language $\mathcal{L}_{B,\mathbb{F}}$ has no arithmetic sketching scheme.*

7.3 Intervals

Shared vectors that are all zero, except for a secret interval $[a, b], 1 \leq a \leq b \leq n$ on which their value is some constant $\beta \in B$ have several different applications, e.g. [12, 14]. An arithmetic sketching scheme for $B = \{-1, 0, 1\}$ was presented in [14]. The more natural setting of $B = \{0, 1\}$, i.e. accepting either the all zero vector or a vector with an interval of 1, was considered in that work though without giving a arithmetic sketching scheme for the language.

At first glance the difference between $B = \{-1, 0, 1\}$ and $B = \{0, 1\}$ seems to be minor, and constructing an arithmetic sketching scheme for intervals with $B = \{0, 1\}$ should be achievable with the correct technical approach. However, we show in this section that such an arithmetic sketching scheme is impossible via a reduction from the Greater Than (GT) problem using the lower bound on its communication complexity.Let

$$\mathcal{L}_{int,B} \stackrel{\text{def}}{=} \left\{ \begin{array}{c} (x_1, \ldots, x_n) \in \mathbb{F}^n : \ \exists\ 1 \leq a \leq b \leq n, \exists \beta \in B\ \forall a \leq i \leq b\ x_i = \beta, \\ \forall\ (i < a \vee i > b)\ x_i = 0 \end{array} \right\}.$$

We prove the following theorem in Appendix F:

Theorem 34. *If $B = \{0, 1\}$ then there does not exist an arithmetic sketching scheme for the language $\mathcal{L}_{int,B}$.*

8 Open Questions

Our results leave several natural open questions. A broad question is to obtain a tight understanding of the achievable tradeoffs between sketch size, decision degree, and soundness error for languages of interest. Even for the simple "weight-1" languages covered by Theorem 13, where our sketching schemes are optimal with respect to both sketch size and decision degree, the soundness error we obtain is only optimal up to a constant factor.

The gaps are bigger for more complex languages. For example, in the case of vectors of Hamming weight *at most* w (Corollary 23), the number of multiplication gates required by the decision algorithm is polynomially bigger than in the case of Hamming weight *exactly* w (Theorem 22), and even the latter may not be asymptotically optimal.

Finally, it may be useful to extend the scope of arithmetic sketching to capture *approximate* computations, which are commonly considered in the algorithmic literature on (insecure) sketching and streaming.

Acknowledgements. We thank Mark Simkin for pointers to related work. D. Boneh is supported by NSF, the DARPA SIEVE program, the Simons Foundation, UBRI, and NTT Research. E. Boyle is supported by AFOSR Award FA9550-21-1-0046, ERC Project HSS (852952), and a Google Research Award. H. Corrigan-Gibbs is supported by Capital One, Facebook, Google, Mozilla, Seagate, MIT's FinTech@CSAIL Initiative, and NSF Award CNS-2054869. N. Gilboa is supported by ISF grant 2951/20, ERC grant 876110, and a grant by the BGU Cyber Center. Y. Ishai is supported by ERC Project NTSC (742754), BSF grant 2018393, and ISF grant 2774/20. Opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

References

1. Ittai Abraham, Benny Pinkas, and Avishay Yanai. Blinder: MPC based scalable and robust anonymous committed broadcast., 2020.
2. Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *STOC*, pages 20–29, 1996.
3. Alexandr Andoni, Huy L Nguyen, Yury Polyanskiy, and Yihong Wu. Tight lower bound for linear sketches of moments. In *Automata, Languages, and Programming: 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I 40*, pages 25–32. Springer, 2013.
4. Saikrishna Badrinarayanan, Peihan Miao, Srinivasan Raghuraman, and Peter Rindal. Multi-party threshold private set intersection with sublinear communication. In *PKC*, 2021.
5. Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.
6. Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC*. 1988.
7. Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 301–309, 1988.
8. Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *Journal of Cryptology*, 30(4):989–1066, 2017.
9. Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Zero-knowledge proofs on secret-shared data via fully linear PCPs. In *CRYPTO*, pages 67–97. Springer, 2019.
10. Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Lightweight techniques for private heavy hitters. In *IEEE Symposium on Security and Privacy*. IEEE, 2021.
11. Dan Boneh, Gil Segev, and Brent Waters. Targeted malleability: homomorphic encryption for restricted computations. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 350–366. ACM, 2012.
12. Elette Boyle, Nishanth Chandran, Niv Gilboa, Divya Gupta, Yuval Ishai, Nishant Kumar, and Mayank Rathee. Function secret sharing for mixed-mode and fixed-point secure computation. In *EUROCRYPT*, pages 871–900, 2021.

13. Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *EUROCRYPT*, 2015.
14. Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. In *CCS*, 2016.
15. Graham Cormode and Shan Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.
16. Henry Corrigan-Gibbs and Dan Boneh. Prio: Private, robust, and scalable computation of aggregate statistics. In *NSDI*, pages 259–282, 2017.
17. Henry Corrigan-Gibbs, Dan Boneh, and David Mazières. Riposte: An anonymous messaging system handling millions of users. In *IEEE Symposium on Security and Privacy*, 2015.
18. Ronald Cramer, Yevgeniy Dodis, Serge Fehr, Carles Padró, and Daniel Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT*, pages 471–488, 2008.
19. Ivan Damgård, Ji Luo, Sabine Oechsner, Peter Scholl, and Mark Simkin. Compact zero-knowledge proofs of small hamming weight. In *PKC*. Springer, 2018.
20. Hannah Davis, Christopher Patton, MIke Rosulek, and Phillipp Schoppmann. Verifiable distributed aggregation functions. *Cryptology ePrint Archive*, 2023.
21. Leo de Castro and Anitgoni Polychroniadou. Lightweight, maliciously secure verifiable function secret sharing. In *EUROCRYPT*, pages 150–179. Springer, 2022.
22. Richard A DeMillo and Richard J Lipton. A probabilistic remark on algebraic program testing. Technical report, Georgia Inst. OF Tech Atlanta School of Information and COmputer Science, 1977.
23. Jack Doerner and Abhi Shelat. Scaling ORAM for secure computation. In *CCS*, 2017.
24. Saba Eskandarian, Henry Corrigan-Gibbs, Matei Zaharia, and Dan Boneh. Express: Lowering the cost of metadata-hiding communication with cryptographic privacy. *arXiv preprint arXiv:1911.09215*, 2019.
25. Joan Feigenbaum, Yuval Ishai, Tal Malkin, Kobbi Nissim, Martin J. Strauss, and Rebecca N. Wright. Secure multiparty computation of approximations. *ACM Trans. Algorithms*, 2(3):435–472, 2006.
26. Daniel Genkin, Yuval Ishai, Manoj Prabhakaran, Amit Sahai, and Eran Tromer. Circuits resilient to additive attacks with applications to secure computation. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC*, pages 495–504. ACM, 2014.
27. Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In *CRYPTO*, 2019.
28. Niv Gilboa and Yuval Ishai. Distributed point functions and their applications. In *EUROCRYPT*, pages 640–658, 2014.
29. Matthew Green, Watson Ladd, and Ian Miers. A protocol for privately reporting ad impressions at scale. In *CCS*, 2016.
30. Elena Grigorescu, Kyomin Jung, and Ronitt Rubinfeld. A local decision test for sparse polynomials. *Information Processing Letters*, 110(20):898–901, 2010.
31. Jens Groth. Non-interactive zero-knowledge arguments for voting. In *ACNS*, pages 467–482. Springer, 2005.
32. Piotr Indyk and David P. Woodruff. Polylogarithmic private approximations and efficient matching. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 245–264. Springer, 2006.

33. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *FOCS*, 2000.
34. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP*, pages 244–256, 2002.
35. Yuval Ishai, Tal Malkin, Martin J. Strauss, and Rebecca N. Wright. Private multiparty sampling and approximation of vector combinations. *Theor. Comput. Sci.*, 410(18):1730–1745, 2009.
36. Rob Jansen and Aaron Johnson. Safely measuring Tor. In *CCS*, pages 1553–1567, 2016.
37. Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.
38. Daniel M Kane, Jelani Nelson, and David P Woodruff. On the exact space complexity of sketching and streaming small norms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1161–1178. SIAM, 2010.
39. DG Mead. Newton's identities. *The American mathematical monthly*, 99(8):749–751, 1992.
40. Payman Mohassel and Enav Weinreb. Efficient secure linear algebra in the presence of covert or computationally unbounded adversaries. In *CRYPTO*, pages 481–496. Springer, 2008.
41. Jelani Jelani Osei Nelson. *Sketching and streaming high-dimensional vectors*. PhD thesis, Massachusetts Institute of Technology, 2011.
42. Rafail Ostrovsky and Victor Shoup. Private information storage. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 294–303, 1997.
43. Fedor Petrov. Lower bound of q pochhammer symbol. `https://mathoverflow.net/questions/327384/lower-bound-of-q-pochhammer-symbol`.
44. Nicholas Pippenger. A formula for the determinant. *arXiv preprint arXiv:2206.00134*, 2022.
45. Raluca Ada Popa, Hari Balakrishnan, and Andrew J. Blumberg. VPriv: Protecting privacy in location-based vehicular services. In *USENIX Security*, pages 335–350, 2009.
46. Alexander A Razborov. On the distributional complexity of disjointness. In *Automata, Languages and Programming: 17th International Colloquium Warwick University, England, July 16–20, 1990 Proceedings 17*, pages 249–253. Springer, 1990.
47. Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.
48. Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *NDSS*, 2010.
49. Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35:703–747, 2015.
50. Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979.

A Deferred proofs from Section 2

Proof of Fact 4. The idea is to augment the decision predicate D with additional randomness that masks the output in the case that its output is non-zero (i.e., when $x \notin \mathcal{L}$). More precisely, if the decision predicate has the form $D\colon \mathbb{F}^\ell \to \mathbb{F}^m$, we define an augmented decision predicate D' that takes m additional random values $(r_1, \ldots, r_m) \in \mathbb{F}^m$ as input: $D'(a_1, \ldots, a_\ell, r_1, \ldots, r_m)$. The new predicate D' first computes the output of the original decision predicate $(v_1, \ldots, v_m) \leftarrow D(a_1, \ldots, a_\ell)$ and then outputs a random linear combination of the blinded values: $\sum_{i=1}^m v_i r_i \in \mathbb{F}$. Whenever $x \in \mathcal{L}$, the decision predicate D' outputs zero, as before. But now whenever $x \notin \mathcal{L}$ the output of the decision predicate D' is distributed uniformly at random over $\mathbb{F}$—the output leaks nothing about the input x. In either case, the output of the decision predicate is now easy to simulate. This transformation increases the probability that the verifier incorrectly accepts by $1/|\mathbb{F}|$. □

We note that by using a simple generalization of the above transformation, we can increase the soundness error by only $1/|\mathbb{F}|^d$ by increasing the multiplicative size of D by dm and the algebraic degree of D by 1.

B Deferred proofs from Section 3

Proof of Theorem 16. Completeness follows directly from the construction. To prove the detection property defined in (1), fix some $\beta \notin \mathcal{L}_{\{0,1\}}$. It is convenient to define the set $(\mathbb{F}_{\neq})^n$ be the set of all n-tuples in $(\mathbb{F}^*)^n$ whose elements are pairwise distinct. We need to show that for a random vector $\mathbf{s}$ sampled from $(\mathbb{F}_{\neq})^n$, if we define z_i as the inner product $z_i = \langle \beta, \mathbf{s}^i \rangle$ for $i = 1, 2, 3$ then

$$\Pr\big[\mathbf{f}(z_1, z_2, z_3) = (0, 0)\big] \leq \epsilon$$

for the ϵ in the theorem statement.

There are two cases. First, suppose $\beta = u\mathbf{e}_i$ for $u \notin \{0, 1\}$. Then $u^2 \neq u$ and therefore $(s_i u)^2 \neq s_i^2 u$ because $s_i \in \mathbb{F}^*$. This means that $z_1^2 - z_2 \neq 0$ as required.

Second, suppose β has weight w greater than one. Then for the sketch to fail, the random vector $\mathbf{s} \in (\mathbb{F}_{\neq})^n$ must lie on the intersection of the two n-variate polynomials

$$\langle \mathbf{X}, \beta \rangle^2 = \langle \mathbf{X}^2, \beta \rangle \qquad \text{and} \qquad \langle \mathbf{X}^3, \beta \rangle = \langle \mathbf{X}, \beta \rangle \cdot \langle \mathbf{X}^2, \beta \rangle. \tag{6}$$

For a fixed β of weight at least two, we will argue that the intersection contains at most $6\,|\mathbb{F}|^{n-2}$ points $\mathbf{s} \in (\mathbb{F}_{\neq})^n$. Given that, we can bound the detection error as

$$\begin{aligned} \epsilon \leq \frac{6\,|\mathbb{F}|^{n-2}}{|(\mathbb{F}_{\neq})^n|} \leq \frac{6\,|\mathbb{F}|^{n-2}}{\prod_{i=1}^n (|\mathbb{F}| - i)} &\leq \frac{6}{(|\mathbb{F}| - 1)(|\mathbb{F}| - 2)} \cdot \left(\frac{|\mathbb{F}|}{|\mathbb{F}| - n}\right)^{n-2} \\ &\leq \frac{6}{|\mathbb{F}|^2 - 3\,|\mathbb{F}|} \cdot \big(1 - \tfrac{1}{n}\big)^{-(n-2)} \leq \frac{6e}{|\mathbb{F}|^2 - 3\,|\mathbb{F}|} \end{aligned}$$

as required. The third inequality follows from the assumption that $|\mathbb{F}| \geq n^2$, and therefore $n/|\mathbb{F}| \leq 1/n$.

It remains to show that when the weight of β is two or more, the intersection in (6) contains at most $6\,|\mathbb{F}|^{n-2}$ points in $(\mathbb{F}_{\neq})^n$. First, consider the case when $\beta \in \mathbb{F}^n$ has weight exactly two. Say, its first two positions, β_1, β_2 are the non-zero positions. Then a direct calculation shows that the resultant of the two equations in (6) with respect to X_1 is zero when X_2 is zero, or $\beta_1 = 1$, or $\beta_2 = 1$, or $\beta_1 + \beta_2 \in \{0, 1\}$. In all four cases the points $(s_1, s_2) \in \mathbb{F}^2$ that satisfy (6) also satisfy $s_1 = 0$ or $s_2 = 0$ or $s_1 = s_2$. Hence, there are no points $(s_1, s_2) \in (\mathbb{F}_{\neq})^2$ that satisfy (6) when β has weight exactly two.

Second, let us consider the case when $\beta \in \mathbb{F}^n$ has weight three or more. Observe that there are exactly $|\mathbb{F}|$ triples $(u_1, u_2, u_3) \in \mathbb{F}^3$ such that $\mathbf{f}(u_1, u_2, u_3) = (0, 0)$, namely $u_1^2 = u_2$ and $u_3 = u_1 u_2$. This is because once u_1 is chosen, u_2 and u_3 are determined. Hence, it suffices to show that for all $u_1, u_2, u_3 \in \mathbb{F}^3$ and all $\beta \in \mathbb{F}^n$ of weight three or more, the following system

$$\langle \mathbf{X}, \beta \rangle = u_1 \qquad \text{and} \qquad \langle \mathbf{X}^2, \beta \rangle = u_2 \qquad \text{and} \qquad \langle \mathbf{X}^3, \beta \rangle = u_3 \tag{7}$$

has at most $6\,|\mathbb{F}|^{n-3}$ solutions in $(\mathbb{F}_{\neq})^n$. By symmetry, we can assume that the first three positions in β, namely $\beta_1, \beta_2, \beta_3$, are non-zero. Next, let's fix the variables $X_4, X_5, \ldots, X_n \in \mathbb{F}$ arbitrarily. Then (7) becomes a system in the three variables X_1, X_2, X_3 of the same shape as (7). It remains to show that for all $\beta_1, \beta_2, \beta_3 \in \mathbb{F}^*$ and $u_1, u_2, u_3 \in \mathbb{F}$, this system of three equations in three variables has at most six solutions in $(\mathbb{F}_{\neq})^3$. This will prove that (6) has at most $6\,|\mathbb{F}|^{n-2}$ solutions in $(\mathbb{F}_{\neq})^n$, as required.

There are two cases. First, if $\beta_i + \beta_j \neq 0$ for all $1 \leq i < j \leq 3$ and $\beta_1 + \beta_2 + \beta_3 \neq 0$ then the three equations in (7) are algebraically independent, and therefore the intersection of these three surfaces in $\mathbb{F}^3$ contains only six points by Bézout's theorem. Second, if $\beta_1 + \beta_2 = 0$, then by linearity we can assume that $\beta_1 = 1$ and $\beta_2 = -1$. Then (7) becomes

$$X_1 - X_2 + \beta X_3 = u_1, \quad X_1^2 - X_2^2 + \beta X_3^2 = u_2, \quad X_1^3 - X_2^3 + \beta X_3^3 = u_3,$$

where $\beta \neq 0$. In this case the intersection of the three surfaces can have $|\mathbb{F}|$ or more points in $\mathbb{F}^3$. However, once we exclude the points in the intersection where $X_1 = X_2$, then the intersection contains at most six points in $(\mathbb{F}_{\neq})^3$. Finally, when $\beta_1 + \beta_2 + \beta_3 = 0$ the intersection of the three surfaces can also have $|\mathbb{F}|$ or more points in $\mathbb{F}^3$. However once we exclude the points in the intersection where $X_1 = X_2 = X_3$, then again the intersection contains at most six points in $(\mathbb{F}_{\neq})^3$. Hence, for all $\beta_1, \beta_2, \beta_3 \in \mathbb{F}^*$, the intersection contains at most six points in $(\mathbb{F}_{\neq})^n$, as required. □

C Deferred material from Section 5

In this section we make formal the definitions and constructions outlined in Section 5.

C.1 Client-Server Verification via Additive-Attack Security

We begin with the definition from [9] of a *fully linear PCP* (FL-PCP); the arithmetic sketching schemes of the present work are a special simplified case of FL-PCPs with the absence of a prover algorithm and proof π. In particular, the "queries" in the FL-PCP correspond to the sketching matrix in a arithmetic sketching scheme and the "query answers" in the FL-PCP correspond to the sketch in a arithmetic sketching scheme.

Definition 35 (FL-PCP [9])**.** *Let $\mathbb{F}$ be a finite field. A* fully linear probabilistically checkable proof *(FL-PCP) scheme for a language $\mathcal{L} \subseteq \mathbb{F}^n$ with query complexity $\ell \in \mathbb{N}$ consists of algorithms $(P_{PCP}, Q_{PCP}, D_{PCP})$:*

- *$P_{PCP}(x) \to \pi$. The randomized prover algorithm takes as input $x \in \mathbb{F}^n$ and outputs auxiliary proof information $\pi \in \mathbb{F}^h$.*
- *$Q_{PCP}() \to (q_1, \ldots, q_\ell)$. The randomized query algorithm outputs queries $q_1, \ldots, q_\ell \in \mathbb{F}^n \times \mathbb{F}^h$.*
- *$D_{PCP}(a_1, \ldots, a_\ell) \to y \in \mathbb{F}^m$. The randomized decision algorithm takes as input the query answers $a_i = \langle (x||\pi), q_i \rangle \in \mathbb{F}$, for $i \in [\ell]$, and outputs a vector $y \in \mathbb{F}^m$.*

We consider the following completeness and soundness properties of the algorithms $(P_{PCP}, Q_{PCP}, D_{PCP})$:

- ***Completeness.*** *For all $x \in \mathcal{L}$, the verifier accepts a valid proof:*

$$\Pr\left[D_{PCP}\big(\langle (x||\pi), q_1 \rangle, \cdots, \langle (x||\pi), q_\ell \rangle\big) = \mathbf{0} \;:\; \begin{array}{c} \pi \leftarrow P_{PCP}(x); \\ (q_1, \ldots, q_\ell) \leftarrow Q_{PCP}(1^n) \end{array} \right] = 1.$$

- ***Soundness*** *with soundness error ϵ holds if for every $x^* \notin \mathcal{L}$ and for all false proofs $\pi^* \in \mathbb{F}^h$, the probability that the verifier accepts is at most ϵ:*

$$\Pr\left[D_{PCP}\big(\langle (x^*||\pi^*), q_1 \rangle, \cdots, \langle (x^*||\pi^*), q_\ell \rangle\big) = \mathbf{0} \;:\; (q_1, \ldots, q_\ell) \leftarrow Q_{PCP}(1^n) \right] \leq \epsilon.$$

As discussed, we consider a strengthened notion of "*additive-attack*" privacy, an extension of zero knowledge that requires no information be revealed even to an adversary who can introduce additive offsets to the inputs to D_{PCP} and learns the corresponding output y.

Definition 36 (Additive-Attack Privacy)**.** *We say that an FL-PCP $(P_{PCP}, Q_{PCP}, D_{PCP})$ for language $\mathcal{L} \subseteq \mathbb{F}^n$ further satisfies* δ-additive-attack privacy *if every adversary $\mathcal{A}$ there exists a simulator S_{PCP} such that for all $x \in \mathcal{L}$ the following distributions are δ-close:*

$$S_{PCP}() \;\approx_\delta\; \left\{ \big((q_1, \ldots, q_\ell), y\big) \;:\; \begin{array}{r} \pi \leftarrow P_{PCP}(x) \\ (q_1, \ldots, q_\ell) \leftarrow Q_{PCP}() \\ (v_1, \ldots, v_\ell) \leftarrow \mathcal{A}(q_1, \ldots, q_\ell) \\ \forall j \in [\ell],\; a_j = \langle (x||\pi), q_j \rangle \\ y \leftarrow D_{PCP}\big(a_1 + v_1, \ldots, a_\ell + v_\ell\big) \end{array} \right\}.$$

As the final item in this subsection, we provide a more formal description of the client-server protocol given such an FL-PCP with additive-attack privacy.

Construction 37 (Client-server verification protocol)**.** *The construction is parameterized by finite field $\mathbb{F}$, an input size $n \in \mathbb{N}$, a language $\mathcal{L} \subseteq \mathbb{F}^n$, and an FL-PCP $(P_{PCP}, Q_{PCP}, D_{PCP})$. The parties involved are a single client and $k \in \mathbb{N}$ servers. The client-server verification protocol Π_{cs} for verifying the servers hold secret shares of $x \in \mathcal{L}$ is defined as follows:*

- **Input:** *The client holds vector $x \in \mathbb{F}^n$.*
- **Client shares:** *The client samples auxiliary information $\pi \leftarrow P_{PCP}(x)$, and generates and distributes additive secret shares $(x^{(i)}, \pi^{(i)}) \in \mathbb{F}^n \times \mathbb{F}^h$ of x, π to each server $i \in [k]$.*
- **Generate query:** *The servers jointly execute the randomized algorithm $Q()$ to generate and reveal a set of public queries $q_1, \ldots, q_\ell \in \mathbb{F}^{n+h}$, via MPC.*
- **Answer query:** *Each server $i \in [k]$ locally computes its contribution $a_j^{(i)} = \langle (x^{(i)} || \pi^{(i)}), q_j \rangle$ to each (linear) query response, $j \in [\ell]$.*
- **Decision:** *The servers execute an MPC which takes as input $(a_j^{(i)})_{j \in [\ell]}$ from each server i, combines the contributions as $a_j = \sum_{i \in [k]} a_j^{(i)}$, evaluates $y \leftarrow D_{PCP}(a_1, \ldots, a_\ell)$, and outputs the resulting vector $y \in \mathbb{F}^m$. Each server accepts iff $y = \mathbf{0}$ is the all-0 vector.*

Claim 38. *If the underlying k-party MPC protocol is secure against t malicious corruptions (for $t < k$), and the FL-PCP satisfies ϵ-soundness error and δ-additive-attack security, then the protocol Π_{cs} above provides the following guarantees:*

- *Completeness: That is, for every $x \in \mathcal{L}$, then an honest execution of Π_{cs} results in accepting x.*
- *Soundness against malicious client. That is, for every $x^* \notin \mathcal{L}$, and any false proof $\pi^* \in \mathbb{F}^h$, then the (honest) servers accept with probability bounded above by ϵ.*
- *Zero knowledge against t malicious servers. That is, for any malicious adversary $\mathcal{A}$ controlling up to t of the k servers, there exists a simulator $\mathcal{S}$ such that for every input $x \in \mathcal{L}$, the view of $\mathcal{A}$ within execution of Π_{cs} on input x is within δ distance of the simulated view $\mathcal{S}(1^n)$. (This distance is statistical if the MPC provides statistical security, and computational otherwise.)*

Proof. Completeness and soundness against a malicious client follow directly by the corresponding properties of the FL-PCP, together with correctness of the MPC protocol. (Note that these follow as in the semi-honest server case given in [9]). Consider zero knowledge against t malicious servers $T \subset [k]$. By the t-malicious-party security of the underlying MPC protocol, for any t-server-corrupting adversary $\mathcal{A}$, the view of $\mathcal{A}$ within the two MPC protocols can be simulated given their respective corrupt-party inputs and outputs. These input and output values correspond to precisely: (1) the queries $q_1, \ldots, q_\ell$, and (2) the final output $y \leftarrow D_{\text{PCP}}(a'_1, \ldots, a'_\ell)$ evaluated on the potentially malicious values

a'_j, defined by the honest parties' shares of each a_j combined with the corrupt servers' contributions, as a function of values seen thus far: in particular, of the queries $(q_1, \ldots, q_\ell)$. This corresponds *exactly* to the distribution simulatable by the definition of δ-additive-attack security of the FL-PCP. Zero knowledge thus follows. □

Note that the framework extends in a straightforward way to more general secret sharing schemes, beyond additive sharing, with a linear reconstruction procedure.

C.2 Generic Approach: Additive-Attack Security via AMD Distributions

As a central tool, we once again make use of *algebraic manipulation detection (AMD) distributions*, as put forth in Section 3. We consider a stronger version of AMD distributions than in Definition 5. Here, the distribution output contains a public and private portion, and additive-attack detection must succeed even when the additive offset may depend on the public part of the given sample. In our setting, the public portion will correspond to the *queries* that servers will see in the clear, and the private portion will correspond to the query *answers*.

Definition 39 (Public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD Distribution)**.** *A* public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD distribution *is given by a pair of procedures:*

- $S_{\mathsf{AMD}}() \to (r_{\mathsf{pub}}, r_{\mathsf{priv}}) \in \mathbb{F}^{\ell_{\mathsf{pub}}} \times \mathbb{F}^{\ell_{\mathsf{priv}}}$*: A randomized sampling algorithm that outputs a vector* $(r_{\mathsf{pub}}, r_{\mathsf{priv}}) \in \mathbb{F}^{\ell_{\mathsf{pub}}} \times \mathbb{F}^{\ell_{\mathsf{priv}}}$*.*
- $V_{\mathsf{AMD}}(v) \to y \in \mathbb{F}$*: A verification algorithm, represented as an arithmetic circuit, which accepts a vector* $v \in \mathbb{F}^{\ell_{\mathsf{pub}}} \times \mathbb{F}^{\ell_{\mathsf{priv}}}$ *and outputs an element* $y \in \mathbb{F}$ *(interpreted as accept if* $y = 0$ *and reject if* $y \neq 0$*). As before, we sometimes consider* $y \in \mathbb{F}^m$*, where acceptance corresponds to* $y = \mathbf{0}$*.*

satisfying the following properties:

- **Completeness***:* $\Pr[V_{\mathsf{AMD}}(r_{\mathsf{pub}}, r_{\mathsf{priv}}) = 0 : (r_{\mathsf{pub}}, r_{\mathsf{priv}}) \leftarrow S_{\mathsf{AMD}}()] = 1$.
- **AMD***: For every adversarial strategy* $\mathcal{A}$*, the following holds:*

$$\Pr\left[\begin{array}{c} (\Delta_1, \Delta_2) \neq \mathbf{0} \\ V_{\mathsf{AMD}}(r_{\mathsf{pub}} + \Delta_1, r_{\mathsf{priv}} + \Delta_2) = 0 \end{array} : \begin{array}{c} (r_{\mathsf{pub}}, r_{\mathsf{priv}}) \leftarrow S_{\mathsf{AMD}}() \\ (\Delta_1, \Delta_2) \leftarrow \mathcal{A}(r_{\mathsf{pub}}) \end{array}\right] \leq \epsilon.$$

In some sense, a public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD distribution directly captures the necessary property of additive-attack privacy: namely, any fully linear PCP whose (honestly generated) queries and answers form a public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD distribution together with verification algorithm D_{PCP} already satisfies this stronger privacy guarantee. We now formalize precisely this statement.

Claim 40 (Additive-Attack FL-PCP)**.** *Suppose* $(P_{PCP}, Q_{PCP}, D_{PCP})$ *is a fully linear PCP for the language* $\mathcal{L}$*, with* $D_{PCP} : \mathbb{F}^\ell \to \mathbb{F}^m$ *and soundness error*

ϵ, such that for any valid $x \in \mathcal{L}$ it holds that the distribution S_{AMD} defined by honestly generated queries/answers:

$$S_{\mathsf{AMD}} := \left\{ (r_{\mathsf{pub}} = (q_1, \ldots, q_\ell), r_{\mathsf{priv}} = (a_1, \ldots, a_\ell)) : \begin{array}{r} \pi \leftarrow P_{PCP}(x, w), \\ (q_1, \ldots, q_\ell) \leftarrow Q_{PCP}(1^n) \\ \forall j \in [\ell], a_j = \langle (x||\pi), q_j \rangle \end{array} \right\},$$

together with V_{AMD} given by D_{PCP}, form a public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD distribution. Then $(P_{PCP}, Q_{PCP}, D'_{PCP})$ satisfies ϵ-additive-attack privacy, where D'_{PCP} is a modified algorithm $D'_{PCP} = \mathsf{ReRand} \circ D_{PCP}$, where ReRand on input $y \in \mathbb{F}^m$ samples random $r \leftarrow \mathbb{F}^m$ and outputs the linear combination $\sum y_i r_i$.

Proof. First note that completeness of the FL-PCP is preserved. Soundness is preserved up to an additive $|\mathbb{F}|^{-1}$ error, since a nonzero vector $y \leftarrow D_{\text{PCP}}$ in the underlying scheme will revert to a false positive only with probability $|\mathbb{F}|^{-1}$ via the random linear combination of Rerand. For privacy against additive attacks, consider the following simulator. The simulator S_{PCP} first honestly samples queries $(q_1, \ldots, q_\ell) \leftarrow Q_{\text{PCP}}$ and runs the adversary on these values to identify an additive attack offset vector $v := (v_1, \ldots, v_\ell) \leftarrow \mathcal{A}(q_1, \ldots, q_\ell)$. It then outputs $((q_1, \ldots, q_\ell), 0)$ if $v = \mathbf{0}$, or $((q_1, \ldots, q_\ell), r)$ for a *random* element $r \leftarrow \mathbb{F}$ if $v \neq \mathbf{0}$. Note that $v = \mathbf{0}$ corresponds to a lack of attack, in which case S_{PCP} properly simulates acceptance. On the other hand, by the AMD property of the corresponding $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ pair, any offset $v \neq \mathbf{0}$ will result in $V_{\mathsf{AMD}} = D_{\text{PCP}}$ *rejecting* (i.e., outputting $y \neq \mathbf{0}$) except with probability ϵ. This means that S_{PCP} properly simulates a random field element output as the result of Rerand. □

It thus suffices to provide a general transformation from any fully linear PCP to one whose queries and answers satisfy the above public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD property. We next develop a sequence of claims toward this goal, beginning with constructions of AMD distributions following from the literature.

Claim 41 (AMD Distribution Construction [18])**.** *Let $\mathbb{F}$ such that $char(\mathbb{F}) > 2$, and $k \in \mathbb{N}$. Then for any $v = (v_1, \ldots, v_k) \in \mathbb{F}^k$, the following corresponding pair $(S_{\mathsf{AMD}}, V_{\mathsf{AMD}})$ is a public $((\mathbb{F}^k, \mathbb{F}^{k+1}), \epsilon = |\mathbb{F}|^{-1})$-AMD Distribution:*

- *$S_{\mathsf{AMD}}()$: Sample $r \leftarrow \mathbb{F}$, and output $((v_1, \ldots, v_k), (v_1 r + r^2, \ldots, v_k r + r^2, r))$.*
- *$V_{\mathsf{AMD}}((v_1, \ldots, v_k), (v_{k+1}, \ldots, v_{2k+1}))$: Output 0 iff for every $j \in [k]$, $(v_{k+j} - v_{2k+1}^2 - v_{2k+1} v_j) = 0$.*

Remark (Alternative AMD Constructions).

- For $char(\mathbb{F}) = 2$, this construction can be replaced by replacing each of the coordinates $k+1$ through $2k$ of the form $(a_i r + r^2)$ instead with the value $(a_i r + r^3)$ for similar parameters at slightly greater computational cost [18].
- For non-zero inputs $(a_1, \ldots, a_k) \in (\mathbb{F} \backslash \{0\})^k$, it suffices to just give $(a_1, \ldots, a_k, a_1 r, , a_k r, r)$ for random $r \in \mathbb{F}$, with corresponding verifications.

– Cramer *et al.* [18] gave a construction with better output length given by $\mathsf{AMD}'(a_1, ..., a_k) = (a_1, ..., a_k, \sum_{i=1}^{k} r^i a_i + r^{i+1}, r) \in \mathbb{F}^{k+2}$ for randomly sampled $r \leftarrow \mathbb{F}$. However, although this gives a shorter output length, to support its evaluation in a *fully linear manner* will require greater overhead in the proof size, as $r^i(x||\pi)$ must be appended for every value $i \in [k]$.

We present a compiler making use of the AMD distribution construction from Claim 41. The compiler modifies an underlying FL-PCP by incorporating extra auxiliary proof elements π' which enable the linear queries to compute *AMD-authenticated* versions of the underlying x, π. Concretely, in addition to giving out secret shares of x, π, for a random "MAC" $r \in \mathbb{F}$ it will also give shares of the r-scaled vector $r(x||\pi)$, as well as shares of r and r^2. These can be used to augment the underlying query and decision process with additional authentication queries, computing $ra_i + r^2$ for each answer value a_i.

Plugging in alternative AMD distribution constructions (e.g., from Remark C.2) will yield analogous versions of the following theorem with different parameter tradeoffs.

Theorem 42 (Generic Compiler: Protecting Against Additive Attacks). *Suppose $(P_{PCP}, Q_{PCP}, D_{PCP})$ is an ℓ-query fully linear PCP for the language $\mathcal{L} \subseteq \mathbb{F}^n$ with auxiliary proof size s. Then there exists a fully linear PCP* with $|\mathbb{F}|^{-1}$-additive-attack privacy *with auxiliary proof size $(n + 2s + 2)$, with $(2\ell + 1)$ queries, and comparable arithmetic decision complexity.*

Proof. Consider the following modified fully linear PCP:

– $P'_{\mathrm{PCP}}(x)$:
  1. Sample $\pi \leftarrow P_{\mathrm{PCP}}(x)$.
  2. Sample $r \leftarrow \mathbb{F}$.
  3. Output $\pi' = (\pi, r(x||\pi), (r, r^2)) \in \mathbb{F}^s \times \mathbb{F}^{n+s} \times \mathbb{F}^2$.

– $Q'_{\mathrm{PCP}}()$: Recall the output vectors $q'_j \in \mathbb{F}^{2n+2s+2}$ define linear queries to be made to $(x||\pi')$.
  1. Sample $(q_1 \ldots, q_\ell) \leftarrow Q_{\mathrm{PCP}}()$, where each $q_j \in \mathbb{F}^{n+s}$.
  2. For each $j \in [\ell]$, let $q'_j := q_j||(0, \ldots, 0)||(0, 0) \in \mathbb{F}^{2n+2s+2}$.
  3. For each $j \in [\ell]$, let $q'_{\ell+j} := (0, \ldots, 0)||q_j||(0, 1) \in \mathbb{F}^{2n+2s+2}$.
  4. Let $q'_{2\ell+1} = (0, \ldots, 0)||(0, \ldots, 0)||(1, 0) \in \mathbb{F}^{2n+2s+2}$.

– $(D_{\mathrm{PCP}})'(a_1, \ldots, a_{2\ell+1})$:
  1. For $j \in [\ell]$, let $y_j = a_{\ell+j} - a_{2\ell+1}^2 - a_{2\ell+1}a_j$; i.e., verifying the AMD authentication relation.
  2. Compute $y_{\ell+1} = D_{\mathrm{PCP}}(a_1, \ldots, a_\ell)$.
  3. Output $(y_1, \ldots, y_{\ell+1})$.

Note that completeness and soundness are preserved from the underlying FL-PCP (in particular, the new D'_{PCP} rejects if the original D_{PCP} rejects).

It remains to show that the distribution of honestly generated queries and answers

$$S_{\mathsf{AMD}} := ((q'_1, \ldots, q'_{2\ell+1}), (a'_1, \ldots, a'_{2\ell+1}))$$

together with decision function D'_{PCP} form a public $((\mathbb{F}^{\ell_{\mathsf{pub}}}, \mathbb{F}^{\ell_{\mathsf{priv}}}), \epsilon)$-AMD distribution, where $(q'_1, \ldots, q'_{2\ell+1}) \in \mathbb{F}^{\ell_{\mathsf{pub}}} = (\mathbb{F}^{2n+2s+2})^{2\ell+1}$ and $(a'_1, \ldots, a'_{2\ell+1}) \in \mathbb{F}^{\ell_{\mathsf{priv}}} = \mathbb{F}^{2\ell+1}$.

Note that $q'_{\ell+1}, \ldots, q'_{2\ell+1}$ are determined completely given $q'_1, \ldots, q'_\ell$. Further, as D'_{PCP} applies only to the answer portion of the S_{AMD} vector, applying additive offsets to the query portion of irrelevant for acceptance. Now, by construction, the honest answer distribution portion of S_{AMD} has structure

$$a' := (a_1, \ldots, a_\ell, \quad ra_1 + r^2, ra_2 + r^2, \ldots, ra_\ell + r^2, \quad r),$$

corresponding to the AMD distribution structure from Claim 40, meaning that even given the values of $(a_1, \ldots, a_\ell)$, (in particular, given just the queries $q_1, \ldots, q_\ell$), then an adversary cannot choose additive offsets for the vector a' such that it will satisfy the AMD verification tests with probability greater than $|\mathbb{F}|^{-1}$ over the random choice of r. The claim follows.

□

For arbitrary statements, the overhead of this AMD encoding approach corresponds to essentially doubling the amount of secret shared material. However, for useful special cases, much of this overhead is redundant. An interesting example is when the vector x already has some form of AMD robustness, e.g. when proving well-formedness of authenticated Beaver triples. (The corresponding distribution on statement x which the parties hold additive secret shares of consists of a global random Δ together with many instances of the form $(a, b, ab, a\Delta, b\Delta, ab\Delta)$ for random $a, b \in \mathbb{F}$.)

Optimizing for sparse vectors x. Another particularly useful example is applying the above ideas to the sketching schemes for *sparse* vectors x, as is a focus of this work. We observe that the additional proof elements corresponding to rx (where $r \in \mathbb{F}$ is random and $x \in \mathbb{F}^n$ is an input of low Hamming Weight) will itself be a vector of low Hamming weight with the same support. In applications such as proving on secret-shared data, this means this supplementary proof information can be secret-shared together with x in an efficient fashion. For example, existing constructions for distributed point functions [13, 14, 28] enable secret-sharing a vector $\boldsymbol{v} \in (\mathbb{F}^d)^n$ of Hamming Weight 1 at cost scaling with $(n + d)$ instead of nd. Interpreting the pair (x, rx) as a single secret vector in $(\mathbb{F}^2)^n$ thus has minimal additive cost over secret sharing x itself, and not twice the cost.

D Deferred proofs from Section 4

Proof of Corollary 19. We first show that $\text{rank}(H_p(u)) \geq \ell$ with high probability over the random choice of u. Let $H'_p(U) \in \mathbb{F}^{\ell \times \ell}$ be the ℓ-by-ℓ submatrix formed from the first ℓ rows and ℓ columns of $H_p(U)$. Notice that $H'_p(U)$ has exactly the same structure as $H_p(U)$, except with smaller dimension. Therefore, we can apply Lemma 18 to $H'_p(U)$. In particular, if the polynomial p has ℓ non-zero coefficients, then $\det H'_p(U)$ is a non-zero polynomial of degree at most $d = 2\binom{\ell}{2} \deg(p)$. For a

random $u \leftarrow \mathbb{F}$, the probability that $H'_p(u)$ is full rank is then at least $1 - d/|\mathbb{F}|$ as required. Since $H'_p(u)$ is a submatrix of $H_p(u)$, this implies that $\text{rank}(H_p(u)) \geq \ell$ with high probability as well.

To complete the proof, we show that $\text{rank}(H_p(u)) < \ell + 1$. By Lemma 18, $\det(H_p(u)) = 0$. Therefore $H_p(u)$ cannot be of full rank. □

Proof of Fact 20. Let $q = |\mathbb{F}|$. Consider the process of sampling the rows of R one at a time from q^n. The probability that ith row is in the span of the first $(i-1)$ rows is at most $q^{i-1}/q^n = q^{i-1-n}$. Then the probability that all rows are linearly independent is at least $p_{\text{good}} = \prod_{i=1}^{n}(1 - q^{i-1-n}) = \prod_{i=1}^{n}(1 - q^{-i})$. Then $p_{\text{good}} \geq \prod_{i=1}^{\infty}(1 - q^{-i}) \geq 1 - q^{-1} - q^{-2}(1 - q^{-3})(1 - q^{-4}) \cdots \geq 1 - \sum_{i=1}^{\infty} q^{-i}$ [43]. We conclude that $p_{\text{good}} \geq 1 - \frac{1}{q-1}$. □

Proof sketch for Corollary 23. The construction is very similar to that of Construction 21. The only difference is that the decision procedure directly computes the determinant of the $w \times w$ matrix H_n formed from the first w rows and w columns of H. The sketch accepts if the output of the decision predicate is zero. By Lemma 18, it holds that $\det(H_n) \equiv 0$ if and only if the input x has Hamming weight at most w. We can express the determinant computation as an arithmetic branching program and then use standard results on randomized encodings [34] to represent the private portion of the decision computation as a degree-three arithmetic circuit of size $\mathsf{poly}(w)$. □

There are more efficient algorithms for computing the determinant of a Hankel matrix than the ones we use in the proof of Corollary 23 [4]. However, those faster algorithms are not algebraic (i.e., they degree that depends on the field size) so they do not yield improved arithmetic-sketching schemes.

An alternative approach for computing the determinant in the proof of Corollary 23 is to directly represent the determinant computation as an arithmetic circuit of of size $O(w^4 \log w)$ and degree $O(w)$ [44]. Yet another option is to use a special-purpose multiparty computation protocol, such as the one of Mohaseel and Weinreb [40], that requires only $O(w^{2+\epsilon})$ communication, for small $\epsilon > 0$. This latter protocol does not strictly fit into our framework, since the decision predicate is a multi-round protocol rather than a single arithmetic circuit. At the same time, when implementing the decision predicate in a multiparty computation, this interactive variant may be preferable.

D.1 Proof of Theorem 27

Proof of Theorem 27. First, consider a vector $(x_1, \ldots, x_n) \in \mathbb{F}^n$ where $x_i \in \{0, 1, \ldots, w\}$ for $i = 1, \ldots, n$. Let us treat $x_1, \ldots, x_n$ as integers in $[0, w]$. Then, for $(r_1, \ldots, r_n) \in \mathbb{F}^n$ the Newton identities for $k = w + 1$ imply that e_{w+1} in the statement of the theorem satisfies

$$e_{w+1} = \sum_{\substack{(y_1,\ldots,y_n)\in\mathbb{Z}^n \\ \sum_{i=1}^{n} y_i = w+1 \\ 0 \leq y_i \leq x_i}} r_1^{y_1} \cdots r_n^{y_n}. \tag{8}$$

Now, to prove the first part of the theorem, let $x = (x_1, \ldots, x_n) \in \mathcal{L}_1^{(\leq w)}$ and let $(r_1, \ldots, r_n) \in \mathbb{F}^n$. Because $x \in \mathcal{L}_1^{(\leq w)}$, the sum in (8) is over an empty set, and therefore e_{w+1} is zero, as required.

To prove the second part of the theorem recall that $e_{w+1} = \tilde{E}_x(r_1, \ldots, r_n)$ where $\tilde{E}_x(R_1, \ldots, R_n)$ is the polynomial

$$\tilde{E}_x(R_1, \ldots, R_n) = N_{w+1}\left(\sum_{i=1}^{n} x_i R_i,\ \sum_{i=1}^{n} x_i R_i^2,\ \ldots,\ \sum_{i=1}^{n} x_i R_i^{w+1}\right). \tag{9}$$

We show that if $x \notin \mathcal{L}_1^{(\leq w)}$ then $\tilde{E}_x(R_1, \ldots, R_n)$ is a non-zero polynomial of total degree $w+1$. Now the theorem follows by the Schwartz-Zippel lemma.

By construction, we know that $\tilde{E}_x$ has at most total degree $w+1$. It remains to prove that $\tilde{E}_x$ is not the zero polynomial for all $x = (x_1, \ldots, x_n) \notin \mathcal{L}_1^{(\leq w)}$. We argue this in two steps.

- First, suppose that there is some $i \in [n]$ for which $x_i \notin \{0, 1, \ldots, w\}$. Then Lemma 43 shows that the coefficient of R_i^{w+1} in $\tilde{E}_x(R_1, \ldots, R_n)$ is not zero, which proves that $\tilde{E}_x$ is not the zero polynomial.
- Second, suppose that $x_i \in \{0, 1, \ldots, w\}$ for all $i \in [n]$. In this case (8) shows that we can write $\tilde{E}_x$ explicitly as

$$\tilde{E}_x(R_1, \ldots, R_n) = \sum_{\substack{(y_1,\ldots,y_n)\in\mathbb{Z}^n \\ \sum_{i=1}^n y_i = w+1 \\ 0 \leq y_i \leq x_i}} R_1^{y_1} \cdots R_n^{y_n}.$$

  Since $x \notin \mathcal{L}_1^{(\leq w)}$ this sum is over a non-empty set, again proving that $\tilde{E}_x$ is not the zero polynomial.

Either way, $\tilde{E}_x$ is not the zero polynomial, which completes the proof of the theorem. □

Lemma 43. *Let $(x_1, \ldots, x_n) \in \mathbb{F}^n$ and let $\tilde{E}_x(R_1, \ldots, R_n)$ be the polynomial defined in (9). Then for all $j \in [n]$, the coefficient of the monomial R_j^{w+1} in $\tilde{E}_x$ is $\frac{x_j(x_j-1)\cdots(x_j-w)}{(w+1)!} \in \mathbb{F}$. In particular, if $x_j \notin \{0, 1, \ldots, w\}$ then the coefficient of R_j^{w+1} in $\tilde{E}_x$ is non-zero.*

Proof of Lemma 43. By symmetry, it suffices to prove the lemma for $j = 1$, namely prove that the coefficient of R_1^{w+1} in $\tilde{E}_x$ satisfies the lemma. We prove the lemma by induction on w. When $w = 0$ we have that $\tilde{E}_x(R_1, \ldots, R_n) = \sum_{i=1}^n x_i R_i$ and the lemma holds.

Next, assume the lemma holds for all $0 \leq w' < w$ and we prove it for w. The Newton identity (Theorem 24) implies the following recurrence on N_k:

$$k \cdot N_k(Z_1, \ldots, Z_k) = \sum_{i=1}^{k} (-1)^{i-1} \cdot N_{k-i}(Z_1, \ldots, Z_{k-i}) \cdot Z_i$$

where $N_0 = 1$. We use the recurrence for $k = w+1$. Plugging in $Z_j = \sum_{i=1}^{n} x_i R_i^j$ for $j = 1, \ldots, w+1$, and using the inductive hypothesis, we obtain that the coefficient of R_1^{w+1} in the polynomial $\tilde{E}_x(R_1, \ldots, R_n) = N_{w+1}(Z_1, \ldots, Z_{w+1})$ is

$$\frac{1}{w+1}\left[(-1)^w x_1 + \sum_{i=1}^{w}(-1)^{i-1} \cdot \left[\frac{x_1(x_1-1)\cdots(x_1-w+i)}{(w+1-i)!}\right] \cdot x_1\right]$$

This simplifies to $\frac{x_1(x_1-1)\cdots(x_1-w)}{(w+1)!}$, as required. □

D.2 Proof of Theorem 29

Our construction uses the following sketch as a subroutine:

Construction 44 (Sketch for all non-zero elements satisfying predicate)**.** *The construction is parameterized by a finite field $\mathbb{F}$, integers n and w with $w \leq n^{1/2}$, and an arithmetic circuit $C\colon \mathbb{F} \to \mathbb{F}$ (of size independent of $|\mathbb{F}|$) with $C(0) \neq 0 \in \mathbb{F}$. The sketch accepts the language of vectors in $x \in \mathbb{F}^n$ such that either:*

- *each non-zero element x_i of x is such that $C(x_i) = 0 \in \mathbb{F}$, OR*
- *the vector has Hamming weight strictly less than w.*

The construction makes use of the zero-knowledge sketch $(S', D^{\text{priv}\prime}, D^{\text{pub}\prime})$ of Corollary 23 for vectors in $\mathbb{F}^n$ and Hamming weight at most $w-1$.

- $S() \to Q \in \mathbb{F}^{(\ell'+n)\times n}$.
  - *Run $Q' \leftarrow S'() \in \mathbb{F}^{\ell' \times n}$.*
  - *Output the matrix $Q \in \mathbb{F}^{(n+\ell')\times n}$ formed by the $n \times n$ identity matrix in $\mathbb{F}$ stacked on top of the matrix Q'.*
- $D(a, r_1, \ldots, r_n) \to y$.
  - *Let $(a_1, \ldots, a_n) \in \mathbb{F}^n$ be the first n elements of a and let a' be the rest.*
  - *Compute $y' \leftarrow D(a')$.*
  - *For $i \in [n]$, compute $y_i \leftarrow a_i \cdot C(a_i) \in \mathbb{F}$.*
  - *Return $y \leftarrow y' \left(\prod_{i=1}^{n} r_i y_i \in \mathbb{F}\right)$.*

Lemma 45. *Construction 44 has the same completeness and soundness error as Corollary 23. The construction has perfect zero knowledge. The sketch size is $2w + n + 1$.*

Proof. For correctness: If the input vector x has weight strictly less than w, then $y' = 0$ with probability at least $1 - 2/(|\mathbb{F}| - 1)$, by Corollary 23 and then $y = 0$. If each non-zero element of the input vector x is either 0 or satisfies C, then $x_i \cdot C(x_i) = 0$ for all $i \in [n]$. So then $\prod_{i=1}^{n} r_i y_i = 0$ for all $(r_1, \ldots, r_n)$ and the output $y = 0$.

For soundness: If the input vector has weight at least w, then $y' \neq 0$ with probability $1 - O(w^2 n/|\mathbb{F}|)$. If the input vector has some non-zero element x_i such that $C(x_i) \neq 0$, then $y_i = x_i C(x_i) \neq 0$ and the vector $(y_1, \ldots, y_n) \neq \mathbf{0}^n \in \mathbb{F}^n$. The decision predicate outputs a random linear combination of the elements in this non-zero vector, which is non-zero with probability $1 - 1/|\mathbb{F}|$. Then the

output is the product of two non-zero elements of $\mathbb{F}$, except with probability $O(w^2 n/|\mathbb{F}|)$.

For zero knowledge: If the input is in the language, the decision predicate outputs zero always. If the input is not in the language, the output is a uniform random value over $\mathbb{F}$. □

Construction 46 (Sketch for bounded Hamming weight satisfying predicate)**.** *The construction is parameterized by a finite field $\mathbb{F}$, integers w and n with $w < n < |\mathbb{F}|$, and an arithmetic circuit $C\colon \mathbb{F} \to \mathbb{F}$ (of size independent of $|\mathbb{F}|$), and a statistical security parameter $\lambda \in \mathbb{N}$. The sketch accepts the language of vectors in $x \in \mathbb{F}^n$ of Hamming weight w such that each non-zero element x_i of x is such that $C(x_i) = 0 \in \mathbb{F}$.*

The construction makes use of:

- *the sketch $(Q_w, {D^{\text{priv}}}_w, {D^{\text{pub}}}_w)$ of Construction 21 for vectors in $\mathbb{F}^n$ of Hamming weight w and*
- *the sketch of $(Q_C, {D^{\text{priv}}}_C, {D^{\text{pub}}}_C)$ Construction 44 for vectors in $\mathbb{F}^{w^2}$ that either have Hamming weight $< w$ or whose whose non-zero elements satisfy a circuit $C\colon \mathbb{F} \to \mathbb{F}$, of size independent of $|\mathbb{F}|$.*

We describe the construction informally:

- *Use Construction 21 to check that the input vector has Hamming weight exactly w.*
- *Repeat for $i \in [\lambda]$:*
  - *Partition the input vector $x \in \mathbb{F}^n$ into w^2 chunks, then sum up the values in each chunk to form a test vector $t_i \in \mathbb{F}^{w^2}$. We can form the vector t using w^2 linear combinations of the input.*
  - *Apply the sketch of Construction 44 to test whether the test vector t_i is either (a) of Hamming weight $< w$ or (b) has all non-zero elements satisfying C.*
  - *Call the output of this sketch y_i.*
- *The private decision predicate outputs the output of ${D^{\text{priv}}}_w$ and $y = \prod_{i=1}^{\lambda} y_i \in \mathbb{F}$.*

Now we return to prove the main theorem:

Proof of Theorem 29. Construction 46 proves the theorem. We must show that for all $\lambda \in \mathbb{N}$, Construction 46 has completeness error $\epsilon = 4/(|\mathbb{F}| - 1) + 2^{-\lambda}$, soundness error $O(w^2 n/|\mathbb{F}|)$ and ϵ-zero knowledge.

Let $\mathcal{L} \subseteq \mathbb{F}^n$ be the language of vectors of Hamming weight w whose non-zero components satisfy $C\colon \mathbb{F} \to \mathbb{F}$.

To show completeness: When the input $x \in \mathcal{L}$, the sketch of Construction 21 accepts with all but probability $2/(|\mathbb{F}| - 1)$. Then the sketch accepts whenever $\prod_{i=1}^{\lambda} y_i = 0$, which happens when at least one value $y_i \in \{y_1, \ldots, y_\lambda\}$ is zero.

This is happens with the probability that when we throw w balls independently at random in to $w^2 < n$ bins, each ball lands in a distinct bin. The probability

of a collision is at most $\binom{w}{2}(1/w^2) \leq 1/2$. The probability that each of the λ runs has a collision is then at most $2^{-\lambda}$. For any run without a collision, the probability that Construction 44 rejects is at most $2/(|\mathbb{F}| - 1)$. So the overall completeness error is at most $4/(|\mathbb{F}| - 1) + 2^{-\lambda}$.

For soundness:

- If the input vector has Hamming weight not equal to w, Construction 21 will accept with probability at most $O(w^2 n/|\mathbb{F}|)$.
- If the input vector has Hamming weight equal to w but one of its non-zero elements does not satisfy C, then Construction 44 will accept with probability at most $O(w^2 n/|\mathbb{F}|)$.

For zero knowledge: We invoke the zero-knowledge property of *Construction* 21 and the fact that $y = 0$, except with probability $4/(|\mathbb{F}| - 1) + 2^{-\lambda}$ when x is well formed. □

E Proof of Theorem 30

We begin with two technical lemmata and then prove Theorem 30.

Lemma 47. *Let (S, D) be an arithmetic sketching scheme for the language $\mathcal{L}_{\mathbb{F}}$ and let $D = (D_1, \ldots, D_m)$ for ℓ-variate polynomials $D_j, 1 \leq j \leq m$, of degree at most d. Then, for every $1 \leq j \leq m$ there exist at most $d + 1$ homogeneous, ℓ-variate polynomials, $p_{j,0}, p_{j,1}, \ldots, p_{j,d}$ such that $D_j = \sum_{k=0}^{d} p_{j,k}$, and every column of the matrix generated by S is a root of $p_{j,k}$ for all $0 \leq k \leq d$.*

Proof. Let $p_{j,k}$ be the sum of all monomials of D_j that have degree k. Then, obviously $p_{j,k}$ is homogeneous for all k and furthermore $D_j = \sum_{k=0}^{d} p_{j,k}$.

Let $(r_1, \ldots, r_\ell)$ be the i-th column of the sketching matrix generated by S for some $1 \leq i \leq n$. On input vector $\boldsymbol{e}_i$ we have that $D_j(a_1, \ldots, a_\ell) = D_j(r_1, \ldots, r_\ell) = 0$, since $\boldsymbol{e}_i \in \mathcal{L}_{\mathbb{F}}$ and due to completeness. For every $\beta \in \mathbb{F}$ we have that $\beta \boldsymbol{e}_i \in \mathcal{L}_{\mathbb{F}}$ and again due to completeness $D_j(\beta \cdot r_1, \ldots, \beta \cdot r_\ell) = \sum_{k=0}^{d} p_{j,k}(\beta \cdot r_1, \ldots, \beta \cdot r_\ell) = 0$. It follows that $\sum_{k=0}^{d} \beta^k \cdot p_{j,k}(r_1, \ldots, r_\ell) = 0$ since $p_{j,k}$ is homogeneous of degree k.

Regard $D_j(\beta \cdot r_1, \ldots, \beta \cdot r_\ell)$ as a univariate polynomial in β, while $(r_1, \ldots, r_\ell)$ is constant. This is a degree d polynomial, which evaluates to 0 on every field element β. Since a non-zero, degree-d univariate polynomial over a field has at most d roots, and since (S, D) is universal which means that $|\mathbb{F}|$ could be larger than d, it follows that D_j must be the zero polynomial and therefore $p_{j,k}(r_1, \ldots, r_\ell) = 0$ for all j, k and every column $(r_1, \ldots, r_\ell)$ of the sketching matrix generated by S. □

We next prove an algebraic lemma on the distribution of roots of multivariate, homogeneous polynomials.

Lemma 48. *Let f be an ℓ-variate and homogeneous polynomial of degree $d > 0$ over a field $\mathbb{F}$ that is not identically zero. Then, the set of roots of f is a union of at most $(d + 1)|\mathbb{F}|^{\ell-2}$ linear subspaces of $\mathbb{F}^\ell$ of dimension at most 1.*

Proof. The proof proceeds by induction on ℓ. In the base case, $\ell = 1$, a homogeneous polynomial of degree d is of the form $f(x) = ax^d$, for $a \in \mathbb{F}$, and it has a single root at $x = 0$, which forms a linear subspace of dimension 0.

Assuming that the statement is true for $\ell - 1$, partition the roots of an ℓ-variate polynomial $f(x_1, \ldots, x_\ell)$ into two sets: all the roots in which $x_\ell \neq 0$, and the roots in which $x_\ell = 0$.

Since f is homogeneous of degree d, every monomial in f is of the form $x_1^{i_1} \cdots x_\ell^{i_\ell}$, where $i_1, \ldots, i_\ell$ are non-negative integers, and $\sum_{j=1}^{\ell} i_j = d$. Let $\mathcal{I} = \{I = (i_1, \ldots, i_\ell) \mid 0 \leq i_1, \ldots, i_\ell, \sum_{j=1}^{\ell} i_j = d\}$. Then, f can be represented as $f(x_1, \ldots, x_\ell) = \sum_{I \in \mathcal{I}} a_I \cdot x_1^{i_1} \cdots x_\ell^{i_\ell}$, such that $a_I \in \mathbb{F}$ for all I.

Consider the $\ell-1$ variables $y_j = x_j / x_\ell, j = 1, \ldots, \ell-1$, and let $g(y_1, \ldots, y_{\ell-1}) = \sum_{(i_1,\ldots,i_\ell) \in I} a_I \cdot y_1^{i_1} \cdots y_{\ell-1}^{i_{\ell-1}}$. Then, it holds that $f(x_1, \ldots, x_\ell) = x_\ell^d \cdot g(y_1, \ldots, y_{\ell-1})$ on any $(x_1, \ldots, x_\ell) \in \mathbb{F}^{\ell-1} \times (\mathbb{F} \setminus \{0\})$.

Any point $(x_1, \ldots, x_\ell) \in \mathbb{F}^{\ell-1} \times \mathbb{F} \setminus \{0\}$ is a root of f if and only if $\left(\frac{x_1}{x_\ell}, \ldots, \frac{x_{\ell-1}}{x_\ell}\right) \in \mathbb{F}^{\ell-1}$ is a root of g. Furthermore, if $(y_1, \ldots, y_\ell)$ is a root of g, then all points $(y_1 \cdot x_\ell, \ldots, y_{\ell-1} \cdot x_\ell, x_\ell)$ are roots of f, forming exactly a dimension-1 linear subspace of $\mathbb{F}^\ell$, i.e. a line. The number of roots of g is at most $d|\mathbb{F}|^{\ell-2}$ by the Schwartz-Zippel lemma [22, 47, 50], since g is an $(\ell-1)$-variate, degree-d polynomial. Therefore, the roots of f such that $x_\ell \neq 0$ are arranged in at most $d|\mathbb{F}|^{\ell-2}$ linear subspaces of dimension 1.

Restricting f to the set of points such that $x_\ell = 0$ consider two cases. If f on this set is the zero polynomial then this set contributes $|\mathbb{F}|^{\ell-1}$ points to the set of roots of f, and they can be regarded as forming $|\mathbb{F}|^{\ell-2}$ dimension-1 linear subspaces of $|\mathbb{F}|^\ell$. If f on this set is not identically zero then its restriction to this set is an $(\ell-1)$-variate and homogeneous polynomial and by the induction all of its roots are in a union of linear subspaces of dimension at most 1. The number of these subspaces is less than $|\mathbb{F}|^{\ell-2}$, which is the total number of dimension-1 linear subspaces in the set. In both cases, all the roots of f lie in at most $(d+1)|\mathbb{F}|^{\ell-2}$ linear subspaces of $\mathbb{F}^\ell$ which have dimension at most 1.

□

We now prove the main result of Section 6:

Proof of Theorem 30. Consider a vector $z \in \mathbb{F}^n \setminus \mathcal{L}_\mathbb{F}$ which is 1 in two random locations i, i' and is 0 anywhere else. Assume that the answers to the queries of S are $a = (a_1, \ldots, a_\ell)$. Recall that D is a sequence of m polynomials $(D_1, \ldots, D_m), m \geq 1$, and D accepts z if and only if $D_j(a) = 0$ for all $1 \leq j \leq m$.

For every $j = 1, \ldots, m$, the polynomial D_j is an ℓ-variate polynomial over $\mathbb{F}$ of degree at most d. By Lemma 47, $D_j = \sum_{k=0}^{d} p_{j,k}$, all $p_{j,k}$ are homogeneous, and any column of the sketching matrix is a root of all $p_{j,k}$.

By Lemma 48, the roots of $p_{j,k}$ form a union of at most $(d+1)|\mathbb{F}|^{\ell-2}$ linear subspaces of $\mathbb{F}^\ell$. Since any column of the sketching matrix is a root of all $p_{j,k}$ it follows that all the columns of the sketching matrix are in the union of at most $(d+1)|\mathbb{F}|^{\ell-2}$ linear subspaces. If the i-th and i'-th columns of the matrix are sampled from the same linear subspace then their sum is also in the same

subspace and is thus a root of $p_{j,k}$ for all j, k. In this case, the sum, and therefore z, will be accepted by D. The probability of both columns being sampled from the same linear subspace leading to erroneously accepting z is at least the inverse of the number of these subspaces, i.e. $\frac{1}{(d+1)|\mathbb{F}|^{\ell-2}}$. □

F Deferred proofs from Section 7

Remark (Consequences of Theorem 32). Consider the language $\mathcal{L}_p^{(\in B)}$ that includes all vectors in which the L_p norm is in a set B. For example, this language could include all vectors with L_p norm smaller than some bound. Theorem 32 generalizes to preclude an arithmetic sketching scheme for this language if B is not too large, more specifically if $|B| \leq \frac{c|\mathbb{F}|}{p-1}$ for a constant $c < 1/2$. That is, any arithmetic sketching scheme for $\mathcal{L}_p^{(\in B)}$ must have a decision algorithm with super-constant multiplicative size. By repeating the proof of Theorem 32, sets x, y are used to construct vectors v_x, v_y over a field such that if the sets are disjoint then $v_x + v_y$ always has a fixed norm, which can be programmed to be in B, while if the sets are not disjoint then the probability of the norm being in B is $(p-1)|B|/|\mathbb{F} \leq c$. This induces a protocol for DISJ, which contradicts the existence of an arithmetic sketching scheme for $\mathcal{L}_p^{(\in B)}$. A similar generalization of Theorem 32 holds for the language $\mathcal{L}_2^{2(\in B)}$ that includes all vectors whose L_2 norm squared is in B for $|B| \leq c|\mathbb{F}|$.

Proof of Theorem 32. Assume that (S_p, D_p) for is an arithmetic sketching scheme for $\mathcal{L}_p^{(=w_f)}$. Let x, y be subsets of $\{1, \ldots, n\}$ with one party holding each set. To decide whether they are disjoint the parties locally construct vectors $v_x, v_y \in \mathbb{F}^{n+2}$. The first n entries of each vector are an indicator vector for the respective sets with set members represented by random field elements. That is, $v_x[i] = \alpha_i$ for all $i \in x$, where $\alpha_i \in \mathbb{F}$ is a random and independently chosen field element, and $v_x[i] = 0$ for all $i \in \{1, \ldots, n\} \setminus \{x\}$. Similarly $v_y[i] = \beta_i$ for all $i \in y$, such that $\beta_i \in \mathbb{F}$ is sampled randomly and independently, and $v_y[0]$ for all $i \in \{1, \ldots, n\} \setminus \{y\}$. The last two entries of the vectors are defined by: $v_x[n+1] = (w^p - \sum_{i=1}^n \alpha_i^p)^{1/p}, v_x[n+2] = 0$ for $w = w_f$, and $v_y[n+1] = 0, v_y[n+2] = (-\sum_{i=1}^n \beta_i^p)^{1/p}$.

If x and y are disjoint then $(v_x + v_y)[i] = v_x[i]$ or $(v_x + v_y)[i] = v_y[i]$ for every $1 \leq i \leq n+2$. Therefore, the L_p norm of $v_x + v_y$ is

$$\left(\sum_{i=1}^{n+2}(v_x + v_y)[i]^p\right)^{1/p} = \left(\sum_{i=1}^{n} v_x[i]^p + \sum_{i=1}^{n} v_y[i]^p + v_x[n+1]^p + v_y[n+2]^p\right)^{1/p} = w$$

If x and y are not disjoint then there exists some $j \in x \cap y$. Let $s = x \cap y \setminus \{j\}$, and let $c = \sum_{i \in s}(v_x + v_y)[i]^p - v_x[i]^p - v_y[i]^p$. In this case the L_p norm of $v_x + v_y$

is

$$\left(\sum_{i=1}^{n+2}(v_x+v_y)[i]^p\right)^{1/p} = \left(w^p + \sum_{i\in x\cap y}(v_x+v_y)[i]^p - v_x[i]^p - v_y[i]^p\right)^{1/p}$$
$$= \left(w^p + (\alpha_j+\beta_j)^p - \alpha_j^p - \beta_j^p + c\right)^{1/p}$$

The expression $(\alpha_j+\beta_j)^p - \alpha_j^p - \beta_j^p + c$ is a degree $p-1$ polynomial in α_j, β_j. By the Schwartz-Zippel lemma the probability that this polynomial has a root for a random choice of α_j, β_j is at most $\frac{p-1}{|\mathbb{F}|}$. Therefore, the probability that the L_p norm of $v_x + v_y$ is w is at most $\frac{p-1}{|\mathbb{F}|}$.

DISJ can therefore be decided by locally computing v_x and v_y and then running the arithmetic sketching scheme on $v_x + v_y$. If the sets are disjoint then the sketching scheme always accepts, and if they are not disjoint the sketching scheme only accepts with probability $\frac{p-1}{|\mathbb{F}|}$. Since the communication induced by the arithmetic sketching scheme is the multiplicative size of D_p that size is also $\Omega(n)$.

Proving the second part of the theorem uses a similar strategy to prove that the decision algorithm of any arithmetic sketching scheme for $\mathcal{L}_2^{2(=w_e)}$ over some finite field $E, |E| > 2$ has multiplicative size $\Omega(n)$. The difference from the previous case is that field elements are not necessarily quadratic residues, and thus may not have quadratic roots. To get around this problem recall the well-known fact that every element in a finite field is the sum of two quadratic residues.

Given sets $x, y \subseteq \{1, \ldots, n\}$ held respectively by two parties. Each party locally constructs $v_x, v_y \in \mathbb{F}^{n+4}$ such that the first n entries of v_x and v_y are the same as in the previous construction. Let q_1, q_2 be two quadratic residues in $\mathbb{F}$ such that $q_1 + q_2 = w - \sum_{i=1}^n \alpha_i^2$, and let q_3, q_4 be two quadratic residues in $\mathbb{F}$ such that $q_3 + q_4 = -\sum_{i=1}^n \beta_i^2$. Complete v_x by setting $v_x[n+1] = q_1^{1/2}, v_x[n+1] = q_2^{1/2}$ and $v_x[n+3] = v_x[n+4] = 0$. Complete v_y by setting $v_y[n+1] = v_y[n+2] = 0$ and $v_x[n+3] = q_3^{1/2}, v_x[n+4] = q_4^{1/2}$.

Repeating the previous analysis shows that if x and y are disjoint then the L_2 norm squared of $v_x + v_y$ is always w, while if x and y are not disjoint then the L_2 norm squared of $v_x + v_y$ is w with probability at most $1/|\mathbb{E}|$. The $\Omega(n)$ lower bound on the communication complexity of disjointness again proves the same $\Omega(n)$ lower bound on the multiplicative size of the decision algorithm. □

Proof of Theorem 33. Assume towards a contradiction that (S, D) is an arithmetic sketching scheme for $\mathcal{L}_{B,\mathbb{F}}$. Let β be a fixed element in B.

Let $x, y \subseteq \{1, \ldots, n\}$ be two subsets that are each held by a different party. The party holding x constructs $v_x \in \mathbb{F}^n$ by setting $v_x[i] = \beta$ if $x_i = 1$, and samples $v_x[i] \in \mathbb{F}$ randomly if $x_i = 0$. The party holding y constructs v_y by defining $v_y[i] = 0$ if $y_i = 1$, and samples $v_y[i] \in \mathbb{F}$ randomly if $x_i = 0$.

If x and y are disjoint then every entry in $v_x + v_y$ is random in $\mathbb{F}$. Therefore, the probability that a specific entry is in B is $|B|/|\mathbb{F}|$. By the union bound the

probability that some entry of $v_x + v_y$ is in B is at most $n|B|/|\mathbb{F}| = 1/3$. If x and y are not disjoint then $i \in x \cap y$ for some $1 \leq i \leq n$ and by construction $(v_x + v_y)[i] = \beta \in B$.

The two parties decide DISJ by locally constructing v_x, v_y, and accepting if and only if (S, D) rejects $v_x + v_y$. This protocol has an error of at most a third, and therefore its communication complexity, and thus the multiplicative size of D is $\Omega(n)$ contradicting the assumption that (S, D) is an arithmetic sketching scheme. □

Proof of Theorem 34. Assume towards a contradiction that there exists an arithmetic sketching scheme (S, D) for $\mathcal{L}_{int,B}$. Let n be a natural number, and Let $0 \leq x, y < 2^n$ be two integers, each held by a different party. The parties wish to decide whether $x > y$. To do so they first choose a field $\mathbb{F}, |\mathbb{F}| > 2$. Then, the first party constructs $v_x \in \mathbb{F}^{2^n}$ such that $v_x[i] = -1$ for all $i \leq x$, and $v_x[i] = 0$ for all $i > x$. The second party constructs $v_y \in \mathbb{F}^{2^n}$ such that $v_y[i] = 1$ for all $i \leq y$, and $v_y[i] = 0$ for all $i > y$. The two parties then invoke (S, D) on $v_x + v_y$ and accept (x, y) if and only if (S, D) rejects $v_x + v_y$.

If $x > y$ then $v_x + v_y$ is a vector of zeroes, except for the interval $[y + 1, x]$ which is all -1. Therefore, (S, D) rejects $v_x + v_y$ and the two parties accept (x, y). If $x \leq y$ then either $x = y$ and $v_x + v_y$ is a vector of zeroes or $x < y$ in which case the vector includes an interval of 1 in $[x + 1, y]$. In either case $v_x + v_y \in \mathcal{L}_{int,B}$, and (S, D) accepts the vector leading to the parties rejecting (x, y).

Since GT can be reduced to an arithmetic sketching scheme for $\mathcal{L}_{int,B}$ with the same communication complexity as the multiplicative size of D, and due to the lower bound $R(GT) = \Omega(\log n)$, there does not exist an arithmetic sketching scheme for $\mathcal{L}_{int,B}$. □