Improving Code Rate and Query Complexity for the Ligero-IOP System

Ignacio Cascudo[a,1], Carmit Hazay[2,3] and Muthu Venkitasubramaniam[2]

[1] IMDEA Software Institute, Madrid, Spain
[2] Ligero Inc.
[3] Bar-Ilan University, Ramat Gan, Israel

Abstract. This work presents two main contributions to the analysis and design of Ligero-style zero-knowledge proof systems. First, we analyze the original Ligero IOP in the list-decoding regime and show that mutual correlated agreement suffices to obtain improved soundness guarantees and lower query complexity, without modifying the protocol. Second, we propose a new variant of Ligero that incorporates out-of-domain sampling, leading to further soundness improvements and enabling Reed–Solomon code rates above 1/2. The new protocol preserves the constant-round structure of the Ligero IOP and its compatibility with memory-efficient proving, while achieving perfect honest-verifier zero knowledge.

1 Introduction

Zero-knowledge proof systems [GMR85] have emerged as a foundational primitive for privacy-preserving applications, enabling users to prove the correctness of computations or compliance with policies without revealing the underlying private data. Applications such as private payments, anonymous credentials, identity systems, confidential smart contracts, and privacy-preserving machine learning increasingly rely on practical proof systems that can operate efficiently at scale. As these applications move from theory to deployment, we list some important considerations for practical proof systems.

One important distinction is between *interactive* and *non-interactive* proof systems. Interactive protocols require multiple rounds of communication between the prover and verifier, which can limit usability in decentralized or asynchronous environments. In contrast, non-interactive proof systems allow a prover to generate a single proof that can later be verified independently, typically through the Fiat-Shamir transformation in the random oracle model. Non-interactivity is particularly important in privacy applications because it naturally enables *public verifiability*: proofs can be verified by arbitrary third parties without requiring interaction with the prover. This property is essential in blockchains and decentralized systems where proofs may need to be verified globally by validators, smart contracts, auditors, or counterparties. Consequently, non-interactive proof systems are generally the preferred choice for practical privacy-preserving deployments.

A second major distinction is between proof systems that provide only *soundness* and those that additionally provide the *zero-knowledge* property. Succinct arguments of knowledge (SNARKs) guarantee that a malicious prover cannot convince a verifier of a false statement except with negligible probability. However, in privacy applications,

E-mail: ignacio.cascudo@software.imdea.org (Ignacio Cascudo), carmit@ligero-inc.com (Carmit Hazay), muthu@ligero-inc.com (Muthu Venkitasubramaniam)

[a]Received support from Grant Excelencia María de Maeztu CEX2024-001471-M funded by MICIU/AEI/10.13039/501100011033. Work partially carried out while also being a researcher at Ligero Inc.

soundness alone is insufficient because the proof itself may leak information about the witness or computation. Zero-knowledge SNARKs (zkSNARKs) strengthen this guarantee by ensuring that the verifier learns nothing beyond the validity of the statement being proven. Since preserving confidentiality is the primary objective in privacy-preserving systems, practical deployments overwhelmingly require true zero-knowledge guarantees rather than soundness alone.

Practical proof systems also differ in the cryptographic assumptions on which they rely. Some systems are based primarily on *symmetric-key primitives*, especially cryptographic hash functions, while others rely on *public-key assumptions* such as pairings or discrete logarithms. Hash-based systems are often attractive for privacy applications because they tend to achieve high prover performance, benefit from hardware acceleration, and offer post-quantum security. Public-key based constructions, on the other hand, frequently achieve significantly shorter proofs and faster verification times, but rely on stronger algebraic assumptions and are generally not post-quantum secure. While succinct proofs are valuable, the increasing importance of long-term privacy and deployment on commodity hardware makes symmetric-key based systems particularly compelling for many real-world privacy applications.

Another important consideration for privacy applications is the memory requirement of the prover. Many practical systems require memory proportional to the size of the computation being proven, which can become a bottleneck for large computations or client-side proving environments. *Sublinear-memory proving* allows proofs to be generated using memory significantly smaller than the execution trace. Memory efficiency is especially critical for privacy-preserving applications because proof generation is ideally performed directly on user-controlled devices such as mobile phones, browsers, tablets, and lightweight edge hardware.

Finally, the language to specify the NP-statement for such proof systems can be low level (i.e. circuit specification) such as Circom, R1CS or a DSL like Cairo, Noir. A more general approach referred to as zkVMs[1] are proof systems that are designed to prove code expressed in the bytecode of standard VMs like WASM, RISCV.

In this work, we focus on these considerations taken together, where proof systems provide true zero-knowledge guarantees, support non-interactive publicly verifiable proofs, rely primarily on symmetric-key cryptography for efficiency and post-quantum security, and achieve sublinear-memory proving to enable execution on lightweight client devices.

Among zero-knowledge proofs, the Ligero IOP is one proof system that simultaneously achieves non-interactivity, sublinear proof lengths, and is plausibly post-quantum secure based on symmetric-key assumptions. Furthermore, for structured computation (such as batched computations) the Ligero system also achieves the SNARK property, namely sublinear[2] verification. Ligetron is a widely deployed concrete realization of the Ligero IOP in the form of a zkVM, demonstrating sublinear-memory proof generation and the ability to handle billions of constraints, even when run from lightweight client platforms like mobile browsers [WHV24] (cf. `https://platform.ligetron.com`). Recently, researchers at Google designed and implemented a privacy-preserving identity verification scheme based on a combination of the Ligero IOP and the GKR proof system [FS24]. Arguably, Ligetron / Ligero stands as the only system that is widely deployed and meets considerations for practical privacy applications (see related works in Section 1.2 for comparisons). Given the practical relevance for privacy and adoption, we focus on the Ligero IOP system in this work.

The Ligero IOP is built on the MPC-in-the-head paradigm introduced in the seminal work of Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS07], where the underlying MPC

[1] Some works like Jolt and others use the zkVM notion to describe proof systems even when they do not satisfy the zero-knowledge property. In this work, by zkVM we imply zero-knowledge.

[2] We distinguish succinct verification from sublinear verification where the former requires the verification to scale polylogarithmic in circuit size, the latter only requires it to be sub-linear.

protocol is instantiated using variants of the protocol of Damgård and Ishai [DI06]. These constructions rely on linear codes possessing an additional multiplicative structure and are typically instantiated using Reed-Solomon codes. In this work, we improve the efficiency of the Ligero IOP in two dimensions: proof length and prover efficiency.

1. The Ligero IOP relies on proximity testing of Reed-Solomon codes and was originally analyzed only in the unique-decoding regime with provable security guarantees. Subsequent systems, such as Aurora [BCR+19], STARKs [BBHR19], and related works, conjectured soundness of these tests beyond the unique-decoding regime. Over the past several years, substantial progress has been made in understanding this regime, including both positive and negative results [AHIV17, BCI+20, GKL24, Zei24, BCH+25, BCGM25, CHK25, CS25, DG25, FS25, GG25, Hab25]. In particular, the work of [Hab25, BCH+25, BCGM25] established that the relevant proximity testing protocol satisfies a strong property known as mutual correlated agreement up to the Johnson bound. However, the original Ligero analysis does not extend beyond the unique-decoding regime and therefore cannot directly leverage these recent advances. Extending Ligero to this setting has the potential to significantly reduce query complexity and consequently improve proof length.

2. Since the Ligero IOP is based on the MPC-in-the-head paradigm, the multiplication test checks products of codewords: if the Reed Solomon code has dimension k, the prover sends a polynomial of degree $2k-2$, twice the degree of the encodings, and the verifier checks it against the opened columns of the oracle. For Reed–Solomon codes of rate ρ, a polynomial of this degree can agree with another one on up to a fraction 2ρ of the evaluation domain, so, under the existing analysis, this check only gives soundness when ρ is below $1/2$, and practical parameter choices often require even lower rates. This harms the prover running time to some extent, as one needs to compute on an error-correcting code of length at least twice the size of the actual witness, and naturally raises the question of whether the Ligero IOP can be instantiated with codes of rate exceeding $1/2$. Improving the code rate directly translates to lower prover overhead and faster proving times, which is particularly important in client-side proving environments running on lightweight devices.

In this work, we address both points. We show how to extend the analysis of the Ligero IOP beyond the unique-decoding regime to leverage recent advances in proximity testing, and we present a variant of the protocol that moves the check of the degree-$(2k-2)$ polynomial out of the evaluation domain, removing the $1/2$ rate barrier while preserving the multiplicative structure required by the MPC-in-the-head framework. In addition, we provide proof-size estimates for a range of parameters, including rates above $1/2$.

1.1 Results and Techniques

In the Ligero IOP, the prover divides its witness into blocks and encodes each block with the Reed-Solomon code, resulting in a matrix all of whose rows are supposed to be codewords. Then, the prover provides oracle access to the matrix, where the verifier can query whole columns. The Ligero IOP consists of three tests: the *interleaved* (or degree) *test*, which, in a nutshell, tests whether the prover has encoded the witness using the Reed-Solomon code; and *the multiplication* (or quadratic) *and linear tests*, which test whether the witness satisfies some specified quadratic and linear relations. A common part of these tests consists of the verifier querying a number t of columns of the matrix.

Ligero's soundness error is approximately[3]

$$\max\left(\left(\frac{1+\rho}{2}\right)^t, (2\rho)^t\right),$$

where ρ denotes the rate of the Reed–Solomon code. The first term arises from the analysis of the interleaved test, while the second term comes from the multiplication and linearity tests.

At a high level, these errors arise from the following mechanism. In the interleaved test, the verifier asks the prover to reveal a random linear combination of the rows of the matrix, which, for an honest prover, should itself be a codeword. The verifier then checks two properties: first, that the received vector is indeed a codeword, and second, after querying several columns of the matrix, that the corresponding positions in the revealed codeword are consistent with those columns. This analysis relies on proximity-gap results, which guarantee that a random linear combination of the rows of a matrix that is sufficiently far from an interleaved Reed–Solomon code will itself remain far from the code. Consequently, if the prover commits to a matrix that is far from the code, then the revealed linear combination must either fail the codeword test (causing the proof to be rejected) or differ significantly from the true linear combination of the matrix rows, in which case the inconsistency will be detected during the query phase.

The precise notion of "far" is crucial here. In [BCI+20], it was shown that the above property can be guaranteed up to the Johnson bound, namely $1-\sqrt{\rho}$. However, the existing soundness analysis of Ligero only holds up to the unique-decoding bound since it requires not only the proximity-gap guarantee described above, but also the implicit guarantee that, whenever the test passes, the opened columns are in fact correct. The results of [BCI+20] provide this stronger guarantee only within the unique-decoding radius, where one can ensure the existence of at most a single nearby codeword. For Reed–Solomon codes, the unique-decoding radius is $\frac{1-\rho}{2}$. Accordingly, the soundness analysis shows that, conditioned on verifier acceptance, the desired consistency property holds with probability $(1-\frac{1-\rho}{2})^t$, yielding the term $\left(\frac{1+\rho}{2}\right)^t$.

On the other hand, the second term, $(2\rho)^t$, arises from the multiplication and linearity tests. In each of the tests, the prover must provide a polynomial constructed from the polynomials p_i encoding the rows of the oracle matrix together with verifier randomness. These polynomials have respective degrees at most $2k-2$, in the case of the multiplication test, and $k+\ell-2$ in the case of the linear test, where $k=\rho n$ is the code dimension (n being its length), and ℓ is the amount of elements encoded by each row, which is at most k. For now we can think of both degrees being bounded by $2k-2$. The verifier checks that these polynomials are consistent with the p_i's at the queried columns, and that they satisfy certain algebraic properties guaranteeing correctness of the witness. A malicious prover may instead provide a polynomial that satisfies the required algebraic properties but is consistent with the p_i's only on some coordinates, hoping that the verifier queries only those positions. Since the polynomial has degree at most $2k-2$, the Schwartz–Zippel lemma implies that the probability of escaping detection when the verifier queries t positions out of n is at most $\left(\frac{2k-2}{n}\right)^t \approx (2\rho)^t$, yielding the second soundness term.

In particular, this shows that, at least under the current analysis, soundness cannot be achieved when the rate ρ exceeds $1/2$.

In this work, we obtain two results that reduce these errors:

- First, we show that the interleaved test of the Ligero IOP can be extended to the *list-decoding* regime. The key property is the notion of mutual correlated agreement, introduced in [Zei24, ACFY25], and later shown to hold for Reed–Solomon codes

[3]Ignoring terms that become negligible over sufficiently large fields.

up to the Johnson bound in [Hab25, BCH+25, BCGM25] and even slightly beyond [DKT26], in some conditions. This improves the first soundness term from $(\frac{1+\rho}{2})^t$ to $\sqrt{\rho}^t$ (or even to some slightly smaller function $a_1(\rho)^t$ with the results of [DKT26]), without requiring any modification to the protocol.

- Second, we introduce a new modified protocol that allows us to *eliminate* the other major error term $(2\rho)^t$ and hence enables us to use high-rate Reed-Solomon codes. The main idea of this protocol is to use *out-of-domain sampling* (OOD), an idea that has been used, in different ways, in other Reed-Solomon based IOPs [BGKS20, ACFY24, ACFY25].

More precisely, our second technique is as follows. Recall that the soundness error in Ligero comes from testing prover-provided polynomials (called p_{mult}, p_{lin}) of degrees respectively at most $2k-2$ and $k+\ell-2 \leq 2k-2$ at t randomly chosen encoding positions among the n positions of the Reed–Solomon code. A malicious prover sending an incorrect polynomial can pass each test with probability at most $(2k-2)/n$. Our goal is to instead test the evaluations $p_{\mathsf{mult}}(\alpha)$, $p_{\mathsf{lin}}(\alpha)$ at a random point α outside the evaluation domain. Since α is sampled from a much larger set $\mathbb{F}_{\mathrm{OOD}}$, the probability that an incorrect polynomial agrees at this point drops to $(2k-2)/|\mathbb{F}_{\mathrm{OOD}}|$. The challenge is then how to verify consistency between $p_{\mathsf{mult}}(\alpha)$ (resp. $p_{\mathsf{lin}}(\alpha)$) and the witness-encoding polynomials p_i, given that the verifier only queries columns of the encoded matrix, which contain evaluations of the p_i over the code domain.

To address this, the prover additionally sends the values $p_i(\alpha)$ in the clear, and we must verify that these are consistent with the encoded polynomials. This is enforced via the interleaved test: recall that the prover already provides a linear combination v of the p_i's, so the verifier can also test the value $v(\alpha)$ and check its consistency with the claimed $p_i(\alpha)$'s. Crucially, the polynomials p_i have degree at most $k-1$, rather than $2k-2$, and therefore we can indeed apply the interleaved test with no penalty.

An additional challenge arises when incorporating zero knowledge. There, the prover blinds the polynomials p_{mult}, p_{lin} in the tests by adding random polynomials p_{mblind}, p_{lblind}. However, this would eliminate the soundness gains of our protocol, since proper masking requires these blinding polynomials to have the same degree bound (respectively $2k-2$ and $k+\ell-2$) and the verifier must later open it at queried codeword positions. Consequently, a malicious prover could again cheat with probability roughly $((2k-2)/n)^t \approx (2\rho)^t$. To overcome this, we build the blinding polynomials from random "pieces" of degree $\leq k-1$, whose degree can then be verified by the interleaved test; splitting a high-degree polynomial into lower-degree ones in this way is used in STARK implementations (e.g. [BS21, Section 3.6.2]), and has been used for zero knowledge in [HK24]. For example in the case of the linear test, it suffices to take $p_{\mathsf{lblind}} = p^{(0)}_{\mathsf{lblind}} + p^{(1)}_{\mathsf{lblind}} X^{\ell-1}$ where $p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}}$ are of degree $\leq k-1$. The prover then includes $p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}}$ in the interleaved test, which simultaneously allows to check their correct degree and provides the evaluations in the query points that allow to check they are consistent with p_{lin}. Unfortunately, a similar decomposition $p_{\mathsf{mblind}} = p^{(0)}_{\mathsf{mblind}} + p^{(1)}_{\mathsf{mblind}} X^{k-1}$ would not work in the case of p_{mblind}, as the queried evaluations of $p^{(0)}_{\mathsf{mblind}}$, $p^{(1)}_{\mathsf{mblind}}$ during the interleaved test leak too much information about p_{mblind} and hence, potentially, about the inputs. Instead, we use the alternative decomposition $p_{\mathsf{mblind}} = p^{(0)}_{\mathsf{mblind}} + p^{(1)}_{\mathsf{mblind}} X^{e} + p^{(2)}_{\mathsf{mblind}} X^{k-1}$ for a fixed $t \leq e \leq k-t-1$. See Section 5.4 for the details.

Comparison with [HK24] Our construction is structurally close to the zero-knowledge version of DEEP-ALI analysed in [HK24]: in both, the inputs are encoded by randomized polynomials, the prover reveals their evaluations at out-of-domain points to check the algebraic constraints, and consistency with the committed encodings is ensured by a degree

test on a masked random linear combination, which requires splitting high-degree masks into lower-degree pieces. The main difference is that [HK24] uses FRI for the degree test, giving a logarithmic number of rounds, whereas we use Ligero's interleaved test, which keeps the protocol constant-round at the cost of larger communication and verification. Moreover, a technical difference is that, since the interleaved test certifies only the single degree bound $k-1$, whereas FRI-based tests are more flexible with respect to the degree bounds of the tested polynomials, the splitting needs to be done slightly differently in our setting (in particular, the overlapping decomposition of p_{mblind}). [HK24] handles local trace constraints through vanishing quotients and FRI, while Ligero also supports arbitrary linear relations across packed witness positions which needs to be separately analyzed. Finally, we remark that the starting points are also complementary: [HK24] adds zero knowledge to DEEP-ALI, which already uses out-of-domain sampling, while we add out-of-domain sampling to a zero-knowledge protocol.

Comparison with [CFW26] Chiesa, Fenzi, and Weissenberg construct honest-verifier zero-knowledge IOPPs for constrained interleaved codes by composing sumcheck and code-switching reductions. Their key techniques are small encoded masks carried through these reductions and private zero-evaders that make out-of-domain sampling compatible with zero knowledge, yielding a $1+o(1)$ zero-knowledge overhead over corresponding non-private protocols. Both works use mutual correlated agreement and out-of-domain sampling. Our focus is on improving the existing Ligero protocol: we strengthen its soundness analysis and remove the degree-doubling bottleneck to permit Reed–Solomon rates above 1/2, while preserving its constant-round structure and streaming approach. Their recursive IOPP instantiations use logarithmic rounds and query only polylogarithmically many field elements, whereas our constant-round construction has argument size scaling as the square root of the circuit size.

Comparison with Vortex [BSU24] Vortex is a Ligero-based list polynomial commitment with out-of-domain openings in the Johnson regime, compiled with a polynomial IOP to obtain arguments of knowledge. Its opening mechanism is closely related to ours. Our contribution concerns the arithmetic layer, namely the list-decoding analysis of Ligero's arbitrary linear-constraint and quadratic tests together with their out-of-domain adaptation with perfect honest-verifier zero knowledge.

Our new protocol can be directly adapted for the Ligetron system and inherit both the memory efficiency as well as the zkVM properties. On a high-level, for each round or stage in the proof generation, the prover runs through the entire computation and streams the witness and constraints in a memory-efficient manner. The new protocol introduces additional rounds in the Ligero system and the prover can rely on the same witness and constraint generation procedure for the additional rounds.

We estimate proof sizes and present them in Tables 1 and 2 in Appendix A. They quantify two complementary effects. In the first table, we consider $N = 2^{20}$ constraints and a 256-bit field. At rate 1/32, the improved analysis reduces the estimated proof size by approximately 39%. OOD provides its main benefit at higher rates where it reduces the queries at rate 1/3 and permits rates 1/2 and 5/8.

The second table fixes rate 1/32 and reports OOD proof sizes for different circuit sizes over 32-, 64-, and 256-bit base fields. At 2^{20} gates, the 32-bit witness variant has estimated sizes of 0.677 and 0.748 MB at the 80- and 100-bit targets, compared with 1.705 and 1.907 MB over a 256-bit base field. See Section 7 for the analysis.

1.2 Related Work

Time and memory efficient ZK-SNARKs were first designed by Bitansky and Chiesa [BC12] and by Holmgren and Rothblum [HR18] in the designated verifier (i.e., not publicly verifiable) setting. Publicly-verifiable ZK-SNARKs with similar overheads can be accomplished via recursive composition [BGH19, COS20, BCMS20]. However, these constructions have significant overheads as they typically rely on non-black-box usage of the underlying primitives. More recently, two works by Block et al. [BHR+20, BHR+21], designed the first black-box construction of a ZK-SNARKs with polylogarithmic overhead in space and time based on the hardness of discrete logarithm in prime-order groups and hardness assumptions on hidden order groups, respectively. These are based on public-key assumptions that are not post-quantum secure and not practically efficient.

For hash-based proof systems, Bootle et al. [BCL22] and Ron-Zewi and Weiss [RW24] design zero-knowledge IOPs based on tensor-codes that can be compiled to ZK-SNARKs, however, these works have not been implemented in practice. More recently, Dalal et al. [DHRR26] construct a hash-Based multilinear proof system that is zero-knowledge where they benchmark its practical efficiency but are not demonstrated to be memory efficient.

There are numerous zk-SNARKs for VMs that are production ready and deployed in practice today by companies like Succinct Labs, RiscZero, Starkware, Zisk. These works design SNARKs (or STARKs) that are hash-based and achieve zero-knowledge property by recursively composing with ZK-SNARKs such as [Gro16, GWC19, BGH19]. These systems break the computation into computational shards that each individually proved via the hash-based SNARK and then combined via recursive composition first using the hash-based SNARK and in the final step with a ZK-SNARK. These systems have not been to be shown viable for privacy applications as they are not scalable on memory constrained devices.

There is a long line of work based on the sum-check protocol [LFKN90] designing concretely efficient SNARKs [Set20, CBBZ23, STW24, AST24]. Some recent works have explored making them memory efficient [BCF+25, NTZ25]. These works have not shown to be practical in memory-constrained environments, let alone have the zero-knowledge property. While some of these works [GLS+23a] can be used to design hash-based SNARKs, the ones that are implemented in practice have been based on public-key assumptions that are not post-quantum secure [AST24].

Another class of proof systems based on folding schemes [KST22, KS24, Azt26] have emerged that allows for efficient design of SNARKs and zkSNARKs but the ones that have been implemented are based on public-key assumptions that are not post-quantum secure. These systems have the promise of being memory efficient. Lattice-based folding schemes [NS25, BC25a, BC25b] are plausibly post-quantum, but the current constructions are only SNARKs and are yet to be demonstrated to be practical with zero-knowledge.

The work of Ephraim et al. [EFKP20] show that assuming the existence of standard (circuit) SNARKs one can construct a non-interactive succinct argument of knowledge (i.e., SNARK) for parallel RAM computations where the complexities are preserved on the prover's side, and the verifier requires polylogarithmic in T time and space based on collision-resistant hash functions (CRHF), where the underlying CRHF and SNARK is used in a non-black-box manner. This work demonstrates asymptotic feasibility but is not practically viable.

1.3 Overview

Appendix B gives a qualitative comparison with other hash-based constructions, distinguishing proof size, verification complexity, interaction, prover memory, and zero knowledge.

In Section 3, we recall recent results on proximity testing. In Section 4, we analyze the soundness of the original Ligero IOP [AHIV17] beyond the unique decoding regime.

In Section 5, we present a new variant of the Ligero IOP that leverages out-of-domain sampling, and discuss how to add zero knowledge to this variant. In Section 6 we describe a zero-knowledge proof for circuit satisfiability based on these tests, prove its security properties including round-by-round soundness which we use to discuss the security of the non-interactive version via the BCS compiler. Finally, in Section 7 we provide some concrete proof lengths and query complexity for various parameter regimes.

2 Preliminaries

Basic notations. We denote the security parameter by κ. We say that a function $\mu : \mathbb{N} \to \mathbb{N}$ is *negligible* if for every positive polynomial $p(\cdot)$ and all sufficiently large κ's it holds that $\mu(\kappa) < \frac{1}{p(\kappa)}$. We use the abbreviation PPT to denote probabilistic polynomial-time and denote by $[n]$ the set of elements $\{1, \ldots, n\}$ for some $n \in \mathbb{N}$. We will generally represent vectors over some sets with bold font, e.g. $\mathbf{v}$, and matrices with upper case fonts A. We denote by $\mathbf{v}[i]$ the ith coordinate of $\mathbf{v}$. Moreover we denote by $A[i]$ the i-th column of matrix A. We extend these notations to set of coordinates so if $\mathbf{v}$ (resp. A) have n coordinates (resp. columns) and $S \subset [n]$, then $\mathbf{v}[S] = (\mathbf{v}[i])_{i \in S}$ (resp. $A[S] = (A[i])_{i \in S}$). Less often, we will need to refer to the i-th row of A, and we will do so as A_i. Given vectors $\mathbf{v}, \mathbf{w} \in \mathbb{F}^n$ its componentwise product (also called Hadamard product or Schur product) is $\mathbf{v} \circ \mathbf{w} = (\mathbf{v}[1] \cdot \mathbf{w}[1], \mathbf{v}[2] \cdot \mathbf{w}[2], \ldots, \mathbf{v}[n] \cdot \mathbf{w}[n])$

Coding notation. A code of length n over an alphabet Σ is simply a subset $C \subseteq \Sigma^n$. For vectors $\mathbf{v}$, $\mathbf{w} \in \Sigma^n$ the Hamming distance $d(\mathbf{v}, \mathbf{w})$ is the number of coordinates in which the differ, i.e. $|\{i \in [n] : \mathbf{v}[i] \neq \mathbf{w}[i]\}|$. Moreover, denote by $d(\mathbf{v}, C)$ the minimum distance of $\mathbf{v}$ from C, i.e. $d(\mathbf{v}, C) = \min_{\mathbf{c} \in C} d(\mathbf{v}, \mathbf{c})$. Finally denote by $d(V, C)$ the minimal distance between a vector set V and a code C, namely $d(V, C) = \min_{\mathbf{v} \in V}\{d(\mathbf{v}, C)\}$. For a code C with $|C| \geq 2$, its minimum distance is $d(C) = \min_{\substack{\mathbf{c}, \mathbf{c}' \in C \\ \mathbf{c} \neq \mathbf{c}'}} d(\mathbf{c}, \mathbf{c}')$.

A linear code L of length n over a finite field $\mathbb{F}$ is a code $L \subseteq \mathbb{F}^n$ which is also a vector space over $\mathbb{F}$. Consequently it has some dimension k as a vector space. We then say that L is a $[n, k, d]$-linear code, where $d = d(L)$ is its minimum distance. The rate of a linear code is $\rho = k/n$ and the relative minimum distance is $\delta = d/n$. Our ZKIPCP protocol uses Reed-Solomon (RS) codes, defined next.

Definition 1 (Reed-Solomon Code)**.** *For positive integers n, k, finite field $\mathbb{F}$ with $|\mathbb{F}| \geq n \geq k$, and a vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n) \in \mathbb{F}^n$ of pairwise distinct field elements, the code $\mathsf{RS}_{\mathbb{F},n,k,\boldsymbol{\eta}}$ is the $[n, k, n-k+1]$-linear code over $\mathbb{F}$ that consists of all n-tuples $(p(\eta_1), \ldots, p(\eta_n))$ where p is a polynomial of degree $< k$ over $\mathbb{F}$.*

Note that for a Reed-Solomon code, the rate ρ and relative minimum distance δ are related as $\rho = 1 - \delta + 1/n$. In its classical use, a $[n, k, n-k+1]$-Reed Solomon code L would be used to deterministically encode messages from $\mathbb{F}^k$ into codewords in $\mathbb{F}^n$ (note $\mathbb{F}^k$ is in bijection with L). In this paper we use a randomized encoding where our inputs are in $\mathbb{F}^\ell$, $\ell \leq k$, and we encode them into L by sampling randomness in $\mathbb{F}^{k-\ell}$ and deterministically encoding the joint vector of input and randomness. Therefore each input in $\mathbb{F}^\ell$ has $|\mathbb{F}|^{k-\ell}$ possible encoding codewords, while each codeword in L decodes to a unique input. We will set up the encoding procedure so that decoding is as follows:

Definition 2 (Message decoding for RS codes)**.** *Let $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ be an RS code and $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_\ell)$ be a sequence of distinct elements of $\mathbb{F}$ for $\ell \leq k$. For $\mathbf{u} \in L$ we define the message $\mathsf{Decode}_{\boldsymbol{\zeta}}(\mathbf{u})$ to be $(p_{\mathbf{u}}(\zeta_1), \ldots, p_{\mathbf{u}}(\zeta_\ell))$, where $p_{\mathbf{u}}$ is the polynomial (of degree $< k$) corresponding to $\mathbf{u}$.*

The above message decoding procedure applies to codewords that have not undergone errors. In general, error correcting codes allow to decode even in the presence of a small number of errors. The unique decoding radius marks the limit for which it is guaranteed that, for any received vector, there exists at most one codeword within that distance of the vector.

Definition 3 (Unique decoding radius)**.** *For a code $C \subseteq \Sigma^n$ of minimum distance $d(C) = d$, its unique decoding bound is $\frac{d-1}{2}$. For every $\mathbf{v} \in \Sigma^n$, there is at most one word $\mathbf{c} \in C$ such that $d(\mathbf{v}, \mathbf{c}) \leq \frac{d-1}{2}$.*

For arbitrary codes, it is not generally known if there is an efficient (polynomial in the length n) algorithm to find the codeword that is at distance $\leq \frac{d-1}{2}$ from a given vector, if this codeword exists. But in the case of Reed Solomon codes, there are efficient algorithms that accomplish this goal, such as the Berlekamp-Welch algorithm.

Beyond the unique decoding bound, one has the notion of list decoding, where the goal is to output a list of codewords that are at at most a given distance τn of the input vector. As τ grows, the list may eventually become exponentially large in n. The Johnson bound is a value of τ below which the list is guaranteed to be polynomial in n, regardless of the input vector.

Lemma 4 (Johnson bound)**.** *Let $C \subset \Sigma^n$ be a code of length n and minimum distance d. Let $\delta = d/n$ be its relative minimum distance. For $\gamma \in (0, 1)$ and $\mathbf{v} \in \Sigma^n$, let*

$$\Lambda = \Lambda(C, \mathbf{v}, \gamma) := \{\mathbf{c} \in C : d(\mathbf{c}, \mathbf{v}) \leq \gamma n\}.$$

If $\gamma < 1 - \sqrt{1-\delta}$, then $|\Lambda| = O(\mathsf{poly}(n))$. In fact $|\Lambda| \leq \frac{\delta - \gamma}{(1-\gamma)^2 - (1-\delta)}$.

The quantity $J(\delta) = 1 - \sqrt{1-\delta}$ is called the Johnson bound of C. Note $\frac{\delta}{2} < J(\delta) < \delta$ for $\delta \in (0, 1)$.

For Reed-Solomon codes $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$, write $\rho^- = 1 - \delta = \rho - \frac{1}{n}$, and then $J(\delta) = 1 - \sqrt{\rho^-} > 1 - \sqrt{\rho}$. Moreover, the Guruswami-Sudan algorithm [GS98] computes $\Lambda(L, \mathbf{v}, \gamma)$ in polynomial time in n for any $\mathbf{v} \in \mathbb{F}^n$ and $\gamma < J(\delta)$, in particular for any $\gamma < 1 - \sqrt{\rho}$.

Remark 1. For Reed Solomon codes, if $\gamma < 1 - \sqrt{\rho} < J(\delta)$ and we call $\sigma := 1 - \sqrt{\rho} - \gamma$, and recall again that $\delta = 1 - \rho + 1/n$ one can bound $|\Lambda|$ as

$$|\Lambda| \leq \frac{\sigma - \rho + \sqrt{\rho} + 1/n}{(\sigma + \sqrt{\rho})^2 - \rho + 1/n} \leq \frac{\sigma + \sqrt{\rho} - \rho}{2\sigma\sqrt{\rho}} = \frac{1}{2\sqrt{\rho}} + \frac{1 - \sqrt{\rho}}{2\sigma}.$$

Interleaved codes Given a linear code $L \subseteq \mathbb{F}^n$ and an integer $m \geq 1$, we denote L^m the set whose elements are all $m \times n$ matrices U over $\mathbb{F}$ such that every row U_i of U is a codeword of L. By seeing matrices in $\mathbb{F}^{m \times n}$ as vectors in $(\mathbb{F}^m)^n$ (i.e., regarding each column as a sole coordinate in the space $\mathbb{F}^m$), L^m can be regarded as a code over the alphabet $\mathbb{F}^m$. Moreover, under such interpretation, the Hamming distance between matrices $A, B \in \mathbb{F}^{m \times n} = (\mathbb{F}^m)^n$ (not necessarily in L^m) becomes the number of columns j such that $A[j] \neq B[j]$. Then the distance between A and L^m is $d(A, L^m) = \min\{d(A, U) : U \in L^m\}$, and the minimum distance of L^m is $d(L^m) = \min_{\substack{U,V \in L^m \\ U \neq V}} d(U, V)$. It is easy to verify that $d(L^m) = d(L)$. When $L = \mathsf{RS}_{\mathbb{F},n,k,\eta}$ is an RS code and $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_\ell)$ as in Definition 2, then we let $\mathsf{Decode}_{\boldsymbol{\zeta}}(U)$ be the length-$m\ell$ vector $\mathbf{x} = (x_{11}, \ldots, x_{1\ell}, \ldots, x_{m1}, \ldots, x_{m\ell})$ such that $(x_{i1}, \ldots, x_{i\ell}) = \mathsf{Decode}_{\boldsymbol{\zeta}}(U_i)$ for $i \in [m]$.

3 Proximity Testing

In this section, we recall some recent results on so-called proximity gaps for (Reed-Solomon) codes, which play an important role in the Ligero construction. These results ensure that

if a certain matrix U^* is γ-far from an interleaved code L^m (meaning $d(U^*, L^m) > \gamma n$ for some fixed $\gamma \in (0,1)$), then for certain distributions of randomness $\mathbf{r} \in \mathbb{F}^m$, a random linear combination $\mathbf{r}^\top U^*$ of its rows is, with high probability, γ-far from the base code L (i.e. $d(\mathbf{r}^\top U^*, L) > \gamma n$). Diverse results have been reported for various values of γ and randomness distributions in [AHIV17, BCI+20], and in a series of recent results [GKL24, Zei24, BCH+25, BCGM25, CHK25, CS25, DG25, FS25, GG25, Hab25]. We will describe the results that are relevant for us below.

Moreover, the soundness analysis for Ligero is simplified and can be extended to the list-decoding regime when the strong property of mutual correlated agreement [ACFY25] holds. Roughly, this property claims that if $\mathbf{r}^\top U^*$ is close to L then, not only U^* is close to L^m with high probability, but also the set of coordinates in which U^* and $\mathbf{r}^\top U^*$ differ from the closest codewords in the corresponding codes L^m and L coincide. We describe these notions formally in the next section.

3.1 Proximity Generators and Mutual Correlated Agreement

To capture the different randomness sources we can use for the coefficients of this linear combination, and their induced soundness errors, we use the following definition.

Definition 5 (Proximity generator, [ACFY25])**.** *Let $L \subseteq \mathbb{F}^n$ be a linear code. A proximity generator $\mathsf{Gen}(m)$ with proximity bound $B(L,m)$ and error $\varepsilon(L,m,\gamma)$ is a probability distribution on $\mathbb{F}^m$ such that for all $\gamma \in (0, 1 - B(L,m))$ and for all $U^* \in \mathbb{F}^{m\times n}$, if $\Pr_{\mathbf{r}\leftarrow\mathsf{Gen}(m)}(d(\mathbf{r}^\top U^*, L) \le \gamma n) > \varepsilon(L,m,\gamma)$, then $d(U^*, L^m) \le \gamma n$. I.e. there exists a coordinate subset $S \subseteq [n]$ of size $|S| \ge (1-\gamma)n$ and an $U \in L^m$ such that $U[S] = U^*[S]$.*

In the conditions from the definition above we do not make any assumption on whether $\mathbf{r}^\top U^*$ coincides with some codeword in L on the same set of coordinates S where U^* coincides with a codeword in L^m. The notion of mutual correlated agreement introduced in [ACFY25] imposes this additional condition. Note that because we are not assuming we are within the unique decoding radius, it may happen that $\mathbf{r}^\top U^*$ (resp. U^*) is close to several codewords in L (resp. L^m) and coincides in different sets of coordinates with all of them; the following definition also takes care of this.

Definition 6 (Mutual correlated agreement, [ACFY25])**.** *Let $L \subseteq \mathbb{F}^n$ be a linear code. A proximity generator $\mathsf{Gen}(m)$ **with mutual correlated agreement**, proximity bound $B'(L,m)$ and error $\varepsilon'(L,m,\gamma)$ is a probability distribution on $\mathbb{F}^m$ such that for all $\gamma \in (0, 1 - B'(L,m))$ and for all $U^* \in \mathbb{F}^{m\times n}$, the following holds:*

$$\Pr_{\mathbf{r}\leftarrow\mathsf{Gen}(m)}\left(\exists S \subseteq [n] : \begin{array}{lll} & |S| \ge (1-\gamma)n \\ \wedge & \exists \mathbf{c} \in L,\ (\mathbf{r}^\top U^*)[S] = \mathbf{c}[S] \\ \wedge & \forall U \in L^m,\ U^*[S] \neq U[S] \end{array}\right) \le \varepsilon'(L,m,\gamma)$$

It is not hard to see (see e.g. [ACFY25, Lemma 4.9]) that, within the unique decoding radius $\gamma < \delta/2$, a proximity generator automatically has mutual correlated agreement. In other words, if Gen is a proximity generator with proximity bound $B(L,m) > 1 - \delta/2$ then it also has mutual correlated agreement for the same bound $B'(L,m) = B(L,m)$.

In fact what we will use is the following consequence of the definition, which says that proximity generators with mutual correlated agreement are *"neighbourhood-preserving"*: the set of codewords in L which are γ-close to $\mathbf{r}^\top U^*$ is exactly the set that we obtain when we apply $\mathbf{r}$ to the list $\Lambda(L^m, U^*, \gamma)$ of codewords in L^m which are γ-close to U^*:

Lemma 7 (Lemma 4.12. in [ACFY25])**.** *Let $L \subseteq \mathbb{F}^n$ be a linear code, let Gen be a proximity generator for L with mutual correlated agreement, proximity bound B', and error ε'. Then, for any $U^* \in \mathbb{F}^{m\times n}$ and any $\gamma \in (0, \min\{\delta_L, 1 - B'(L,m)\})$ we have $\Pr_{\mathbf{r}\leftarrow\mathsf{Gen}(m)}(\Lambda(L, \mathbf{r}^\top U^*, \gamma) \neq \{\mathbf{r}^\top U : U \in \Lambda(L^m, U^*, \gamma)\}) \le \varepsilon'(L,m,\gamma)$*

3.2 Known Results for Proximity Generators

As mentioned above, several works have obtained diverse results on proximity generators both for general linear codes and for Reed-Solomon codes, and this question remains an actively researched topic at the time of this writing. Here, we summarize the results most relevant to us: we focus on positive results (i.e., that a given generator is a proximity one and has mutual correlated agreement) for Reed-Solomon codes, and we consider the following two types of distributions. To simplify the exposition, we will not state the best known error ε' in every case as we would need to separate in more cases, and we do not consider some more general notions where different gaps γ and $\gamma+\eta$ (for small η) are used in L and L^m. In this work we will consider the following proximity generators:

Definition 8. *For an integer $m>1$, define the following distributions over $\mathbb{F}^m$*

- $\mathsf{Gen}_{\mathtt{unif}}(m)$ *the uniform distribution over* $\{(r_1, r_2, \ldots, r_{m-1}, 1) : r_i \in \mathbb{F} \ \ \forall i \in [m-1]\}$
- $\mathsf{Gen}_{\mathtt{pows}}(m)$ *the uniform distribution over* $\{(r^{m-1}, r^{m-2}, \ldots, r, 1) : r \in \mathbb{F}\}$

Before describing the results for proximity generators, we introduce some notation and an elementary lemma that we will use later.

Definition 9. *We define the following function. For* $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{pows}\}$*, let*

$$c_{\mathtt{rand}}(m) = \begin{cases} 1 & \text{if } \mathtt{rand} = \mathtt{unif} \\ m-1 & \text{if } \mathtt{rand} = \mathtt{pows}. \end{cases}$$

Lemma 10. *Let $\mathbf{v} \neq \mathbf{w}$ any two distinct vectors in $\mathbb{F}^m$. For* $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{pows}\}$*, we have* $\Pr_{\mathbf{r} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(m)}[\mathbf{r}^\top \mathbf{v} = \mathbf{r}^\top \mathbf{w}] \leq c_{\mathtt{rand}}(m)/|\mathbb{F}|$

Proof. Note that $\mathbf{r}^\top \mathbf{v} = \mathbf{r}^\top \mathbf{w}$ iff $\mathbf{r}^\top(\mathbf{v}-\mathbf{w}) = 0$. For $\mathtt{rand} = \mathtt{unif}$, if $\mathbf{v} \neq \mathbf{w}$ then there is some coordinate i for which $v_i - w_i \neq 0$. Then $\mathbf{r}^\top(\mathbf{v}-\mathbf{w}) = r_i(v_i - w_i) + \sum_{j=1, j\neq i}^{m} r_j(v_j - w_j)$. If $i \neq m$, then r_i is uniform in $\mathbb{F}$ and therefore $r_i(v_i - w_i)$ is clearly uniformly distributed in $\mathbb{F}$ so the probability that $\mathbf{r}^\top(\mathbf{v}-\mathbf{w}) = 0$ is $1/|\mathbb{F}|$. If $\mathbf{v}$ and $\mathbf{w}$ only differ in the last coordinate ($i = m$), then $\mathbf{r}^\top(\mathbf{v}-\mathbf{w}) = v_m - w_m \neq 0$, so in this case the probability is 0.

For $\mathtt{rand} = \mathtt{pows}$, the expression $\mathbf{r}^\top(\mathbf{v}-\mathbf{w}) = \sum_{j=1}^{m}(v_j - w_j) r^{m-j}$ can be seen as the evaluation of polynomial $\sum_{j=1}^{m}(v_j - w_j)X^{m-j}$ on the point r. The polynomial is nonzero since $\mathbf{v} \neq \mathbf{w}$. Since a nonzero polynomial of degree $\leq m-1$ can have at most $m-1$ zeros in $\mathbb{F}$ and r is chosen uniformly in $\mathbb{F}$, the probability of this expression being 0 is at most $\frac{m-1}{|\mathbb{F}|}$. ∎

Now we can describe the aforementioned results on proximity generators.

Lemma 11. *[Proximity generators for RS codes] Let $L = \mathsf{RS}_{\mathbb{F},n,k,\boldsymbol{\eta}}$ be a Reed-Solomon code and let $\rho = k/n$ be its rate*

- *[BCI+20, Theorems 1.6, 6.1] (Unique decoding radius).* $\mathsf{Gen}_{\mathtt{unif}}(m), \mathsf{Gen}_{\mathtt{pows}}(m)$ *are proximity generators for L with mutual correlated agreement* [4] *for proximity bound $B'(L,m) = \frac{1+\rho}{2}$,* [5] *and error $\varepsilon'_{\mathtt{rand}}(L,m,\gamma) = \frac{c_{\mathtt{rand}}(m)\cdot n}{|\mathbb{F}|}$ for each case* $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{pows}\}$.

[4] While mutual correlated agreement was not explicitly considered in [BCI+20] (as the notion was defined later), this follows immediately from the fact that within unique decoding every proximity generator has mutual correlated agreement.

[5] Technically we could use here $\rho^- = 1 - \delta = \rho - 1/n$but we ignore this for simplicity.

- *[BCH+25, Theorem 1.5] (List decoding - Johnson radius).* $\mathsf{Gen}_{\mathtt{unif}}(m), \mathsf{Gen}_{\mathtt{pows}}(m)$ *are proximity generators for* L *with mutual correlated agreement for proximity bound* $B'(L,m) = \sqrt{\rho}$ *and error (for each case* $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{pows}\}$*)*

$$\varepsilon'_{\mathtt{rand}}(L,m,\gamma) = c_{\mathtt{rand}}(m)\left(\frac{2u^5+3u\gamma\rho}{3\rho^{3/2}}n + \frac{u}{\rho}\right)\cdot\frac{1}{|\mathbb{F}|},$$

where $u = \max(\lceil \frac{\sqrt{\rho}}{1-\sqrt{\rho}-\gamma} \rceil, 3) + \frac{1}{2}$.

3.3 Beyond the Johnson bound

Several recent results have established list decoding and mutual correlated agreement for Reed–Solomon codes beyond the Johnson radius, on arbitrary evaluation sets, and even up to capacity [BCP+26, DKT26, Jer26, Jo26]. Approaching capacity, i.e. list decoding up to radius $1-\rho-\sigma$ (agreement on a fraction $\rho+\sigma$ of the coordinates) for small $\sigma > 0$, requires interpolation with derivatives of high order d, and the resulting list sizes and exceptional counts grow as n^d and n^{d+1} (with, e.g., $d = \lceil e^{1.5/\sigma} \rceil$ in [DKT26, Theorem 1.2]), which makes them unsuitable for concrete parameters.

The first-order case $d = 1$ of [DKT26], however, comes with explicit bounds of order $O(n)$ and $O(n^2)$ respectively, and lowers the agreement threshold from $\sqrt{\rho}$ to an explicit $a_1(\rho) < \sqrt{\rho}$. To our understanding, these are the most refined results applicable to concrete parameters at the time of writing; [DKT26] also improves the error of the Johnson-radius result above. We include these results below.

Lemma 12.

- *[DKT26, Theorem 5.12, Corollaries 7.2 and 7.7] (List decoding, Johnson radius, improved error). Over a field of any characteristic,* $\mathsf{Gen}_{\mathtt{unif}}(m)$ *is a proximity generator for* L *with mutual correlated agreement for proximity bound* $B'(L,m) = \sqrt{\rho^-}$*, where* $\rho^- = \frac{k-1}{n}$ *and, writing* $\eta_0 := 1-\gamma-\sqrt{\rho^-}$*,*

$$\varepsilon'_{\mathtt{unif}}(L,m,\gamma) = \frac{B_{\mathrm{cf}}}{|\mathbb{F}|-1}, \qquad B_{\mathrm{cf}} < \frac{8nu^3}{3\sqrt{\rho^-}}, \qquad u = \max\left(\left\lceil \frac{\sqrt{\rho^-}}{2\eta_0} \right\rceil, 3\right) + \frac{1}{2},$$

independently of m*, where* B_{cf} *is given exactly by [DKT26, Eq. (68)]. This improves the dependence on the gap from* $O_\rho(n/\eta_0^5)$ *to* $O_\rho(n/\eta_0^3)$*. For* $\mathsf{Gen}_{\mathtt{pows}}(m)$*, [DKT26, Eq. (88) and Corollary 7.7] give an analogous explicit error, linear in* $m-1$*.*

- *[DKT26, Theorems 1.1, 5.8, Corollaries 7.2, 7.3 and 7.7] (List decoding beyond the Johnson radius). Let*

$$a_1(\rho) = \begin{cases} \sqrt{\rho/2}\,(1+u_\rho), \text{ where } u_\rho > 0 \text{ is the root of } u^2(u+3) = \sqrt{\rho/2}, & \rho < \rho_c, \\ \dfrac{3\rho + 2\sqrt{\rho(5-\rho)(2-\rho)}}{8-\rho}, & \rho \geq \rho_c, \end{cases}$$

with $\rho_c = 11 - 3\sqrt{13} \approx 0.1833$*; note* $\rho < a_1(\rho) < \sqrt{\rho}$*. Let* $\gamma < 1 - a_1(\rho)$ *and* $\eta_1 := 1-\gamma-a_1(\rho) > 0$*, and suppose* $k \geq 2$ *and that* $\mathbb{F}$ *has characteristic* $p > \max\{k-1, B_\partial\}$*, where* $B_\partial = O_\rho((\eta_1 + 1/n)^{-1})$ *is explicit [DKT26, Eq. (34)]. Then* $\mathsf{Gen}_{\mathtt{unif}}(m)$ *is a proximity generator for* L *with mutual correlated agreement for proximity bound* $B'(L,m) = a_1(\rho)$ *(i.e., for every* $\gamma < 1 - a_1(\rho)$*), with error*

$$\varepsilon'_{\mathtt{unif}}(L,m,\gamma) = \frac{E_1}{|\mathbb{F}|-1}, \qquad E_1 = O_\rho\left(\frac{n^2}{(\eta_1+1/n)^4}\right),$$

independently of m*, where* E_1 *is given explicitly by [DKT26, Eq. (58)]. For* $\mathsf{Gen}_{\mathtt{pows}}(m)$*, [DKT26, Corollary 7.3] gives error* $(m-1)C_{\mathrm{pow}}/|\mathbb{F}|$ *for an explicit* C_{pow} *under a slightly stronger condition.*

For the second item, as long as $\eta_1 \leq \frac{1-a_1(\rho)}{2}$ (which anyway we should assume, as otherwise γ falls in the unique decoding bound), a direct evaluation of the explicit parameters in [DKT26] shows $B_\partial < 13\eta_1^{-1}$ and then if $k > 13\eta_1^{-1}$ the condition $p > k-1$ suffices, so for example taking prime fields (so that the characteristic is the size of the field) is enough to be able to use these results, since Reed-Solomon codes anyway require the field size to be at least n.

We will also need some estimation of the size of the list of codewords we obtain when list decoding in this regime.

Lemma 13. *[DKT26, Theorem 5.7, Corollary 7.5] Under the assumptions of Lemma 12 (second item), for every $U^* \in \mathbb{F}^{m\times n}$,*

$$|\Lambda(L^m, U^*, \gamma)| \leq 4(k-1)\lambda_{\text{agr}} B_{\text{jet}} B_\partial + 2B_{\text{jet}} B_\partial + B_{\text{jet}} = O_\rho\left(\frac{n}{(\eta_1 + 1/n)^2}\right),$$

where $\lambda_{\text{agr}} = \frac{n-k+1}{A-k+1}$ with $A = \lceil (1-\gamma)n \rceil$, and $B_{\text{jet}}, B_\partial$ are as in [DKT26, Eq. (34)].

In our soundness analysis, $|\Lambda|$ will only appear in the error multiplying terms of order $n/|\mathbb{F}|$, so over large fields we will be still able to claim that this remains negligible compared to other terms in the soundness error, as happens within the Johnson bound regime.

4 Revisiting the Soundness Analysis of the Ligero Tests

In this section, we describe the Ligero proof system from [AHIV17] and generalize its soundness analysis to incorporate the recent results on proximity testing; more precisely, we show that the soundness analysis in [AHIV17] can be extended to the list decoding regime as long as the proximity generator has the property of mutual correlated agreement.

In Ligero, the witness is encoded as a matrix U^* of an interleaved Reed-Solomon code L^m, to which the verifier has oracle access, and then three tests are performed: the Interleaved Test is intended to prove that the oracle is indeed a codeword of the interleaved code, or at least close to it. The other two tests, the Multiplication Test and the Linear Test, check that the inputs encoded by that codeword satisfy some multiplicative and linear relations.

4.1 Interleaved Code Test

We first consider the test Test-Interleaved as described in Figure 1.

Completeness follows directly from the linearity of L.

Lemma 14. *If $U \in L^m$ and $\mathcal{P}$ is honest, then $\mathcal{V}$ always accepts.*

Theorem 15. *Let $L = RS_{\mathbb{F},n,k,\eta}$ be a Reed-Solomon code. Let $\texttt{rand} \in \{\texttt{unif}, \texttt{pows}\}$. Suppose $\mathsf{Gen}(m) := \mathsf{Gen}_{\texttt{rand}}(m)$ is a proximity generator for L with mutual correlated agreement, proximity bound $B' = B'(L,m)$ and error $\varepsilon' = \varepsilon'_{\texttt{rand}}(L,m,\gamma)$, where $\gamma \in (0, 1-B')$. Let*

$$\Lambda := \Lambda(L^m, U^*, \gamma) = \{U \in L^m : d(U, U^*) \leq \gamma n\}.$$

Then, except with probability at most

$$\mathsf{Error}_{\mathsf{int}}(\mathsf{Gen}, m, \gamma) := \varepsilon' + (1-\gamma)^t + |\Lambda| \cdot \gamma n \cdot \frac{c_{\texttt{rand}}(m)}{|\mathbb{F}|},$$

if $\mathcal{V}$ accepts Test-Interleaved where the prover uses oracle U^*, then Λ contains at least one element U such that $U[j] = U^*[j]$ for all the queried $j \in Q$.

Test-Interleaved

The test is parametrized by finite field $\mathbb{F}$, $[n, k, d]$-Reed Solomon code L (over $\mathbb{F}$), and an integer m. Let $\mathsf{Gen}(m)$ be a distribution on $\mathbb{F}^m$.
Oracle: A matrix $U \in \mathbb{F}^{m \times n}$, claimed to be a L^m-codeword. $\mathcal{V}$ can query any $j \in [n]$, and obtains the full column $U[j]$.

Interactive testing:

1. $\mathcal{V}$ samples randomness $\mathbf{r}$ according to $\mathsf{Gen}(m)$ and sends it to the prover.
2. $\mathcal{P}$ responds with $w(X)$, the polynomial of degree $\leq k-1$ such that $w(\eta_i) = w_i$ for $i \in [n]$ where $\mathbf{w} = \mathbf{r}^\top U \in L$ (this can also be computed as $\sum_{j=1}^{m} r_j p_j(X)$ where p_j is the polynomial of degree $\leq k-1$ interpolating the j-th row of U)
3. $\mathcal{V}$ queries a uniformly random set $Q \subset [n]$ of size t, and receives $U[j]$, $j \in Q$.
4. $\mathcal{V}$ accepts iff $\deg w \leq k-1$ and for every $j \in Q$ it holds that

$$w(\eta_j) = \mathbf{r}^\top U[j].$$

Figure 1: Test-Interleaved $(\mathbb{F}, L[n, k, d], m, t; U)$

Proof. Let $w^*(X)$ be the polynomial sent by the prover, which we can assume of degree $\leq k-1$, otherwise the verifier rejects. Let $\mathbf{w}^* = (w^*(\eta_1), \ldots, w^*(\eta_n)) \in L$. Note that Λ depends only on U^*, and hence is fixed before $\mathcal{V}$ samples any randomness, while $\mathbf{r}$ and $\mathbf{w}^*$ are fixed before Q is sampled. We say that $U \in \Lambda$ is *good* if $U[j] = U^*[j]$ for all $j \in Q$; we need to bound the probability that the test passes and no $U \in \Lambda$ is good.

First, we consider the following events. E_1 is the event that the committed U^* is more than γ-far from L^m, i.e. $\Lambda = \emptyset$; this event depends only on U^*. The other two events depend only on the randomness $\mathbf{r}$:

- E_2: $\Lambda(L, \mathbf{r}^\top U^*, \gamma) \neq \{\mathbf{r}^\top U : U \in \Lambda\}$. Since Gen has mutual correlated agreement, by Lemma 7 we have $\Pr_{\mathbf{r}}[E_2] \leq \varepsilon'$. (Note that this covers the case $\Lambda = \emptyset$, where the lemma states that $\mathbf{r}^\top U^*$ is γ-far from L except with probability ε'.)
- E_3: there exist $U \in \Lambda$ and $j \in [n]$ with $U[j] \neq U^*[j]$ but $\mathbf{r}^\top U[j] = \mathbf{r}^\top U^*[j]$. For a fixed $U \in \Lambda$ there are $d(U, U^*) \leq \gamma n$ columns j where the first inequality happens, and for each of them the latter equality holds with probability at most $c_{\mathtt{rand}}(m)/|\mathbb{F}|$ by Lemma 10. Hence, by a union bound, $\Pr_{\mathbf{r}}[E_3] \leq |\Lambda| \cdot \gamma n \cdot c_{\mathtt{rand}}(m)/|\mathbb{F}|$.

Assume from now on that neither E_2 nor E_3 happens. We distinguish two cases, depending only on $\mathbf{w}^*$ (and hence fixed before Q is sampled).

Case 1: $\mathbf{w}^* \neq \mathbf{r}^\top U$ for all $U \in \Lambda$. This includes the case E_1, where $\Lambda = \emptyset$ (and in the "unique-decoding regime" $\gamma \in (0, \frac{\delta}{2})$, Λ contains at most one element, but if $\gamma \geq \frac{\delta}{2}$, it could contain more than one). Since E_2 does not happen, $\mathbf{w}^* \notin \Lambda(L, \mathbf{r}^\top U^*, \gamma)$, i.e. $d(\mathbf{r}^\top U^*, \mathbf{w}^*) > \gamma n$. Since t coordinates of $\mathbf{r}^\top U^*$ will be revealed and compared with those of $\mathbf{w}^*$, and Q is sampled after $\mathbf{w}^*$ is fixed, the test passes with probability at most $\binom{n-\gamma n-1}{t} / \binom{n}{t} \leq (1-\gamma)^t$.

Case 2: $\mathbf{w}^* = \mathbf{r}^\top U_0$ for some $U_0 \in \Lambda$. Fix one such U_0. If Test-Interleaved passes then $\mathbf{r}^\top U_0[j] = \mathbf{w}^*[j] = \mathbf{r}^\top U^*[j]$ for all $j \in Q$, and since E_3 does not happen, $U_0[j] = U^*[j]$ for all $j \in Q$. Hence U_0 is good, and the claim holds.

Altogether, the probability that $\mathcal{V}$ accepts and no $U \in \Lambda$ is good is at most

$$\Pr[E_2] + \Pr[E_3] + (1-\gamma)^t \leq \mathsf{Error}_{\mathsf{int}}(\mathsf{Gen}, m, \gamma).$$

■

Remark 2 (Comparison between $\texttt{rand} = \texttt{pows}$ and $\texttt{rand} = \texttt{unif}$). While in general using $\mathsf{Gen}_{\texttt{pows}}(m)$ as $\mathsf{Gen}(m)$ (as opposed to using $\mathsf{Gen}_{\texttt{unif}}(m)$) gives a worse error according to current results, one benefit of its use in Test-Interleaved is that $\mathcal{V}$ only needs to send a single element $r \in \mathbb{F}$ to specify the random vector $\mathbf{r} = (r^{m-1}, \ldots, r, 1)$, which decreases communication.

4.2 Multiplication and Linear Tests

We now consider the multiplication and linear tests in Ligero. We begin with the multiplication test, described in Figure 2 where the prover claims that the oracle encodes $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ each in $\mathbb{F}^{m\ell}$ such that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$, for $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{m\ell}$ which are part of the statement, and hence known by both parties (recall $\circ$ stands for componentwise product). The oracle sent by the prover is actually to a matrix $U = [U^x|U^y|U^z|U^w]^\top$ where each U^x, U^y, U^z is in $\mathbb{F}^{m\times n}$ and U^w is in $\mathbb{F}^{m'\times n}$,[6] and they respectively encode $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{w}$. Here $\mathbf{w} \in \mathbb{F}^{m'\ell}$ is an auxiliary input not used in this protocol, but we include it for consistency with how the multiplication test is applied in the circuit satisfiability proof.

Test-Multiplication

The test is parametrized by finite field $\mathbb{F}$, $[n, k, d]$-Reed Solomon code L (over $\mathbb{F}$), integers m, m', ℓ, t (with $t \leq n$ and $\ell \leq k < n$) and a selector $\texttt{rand} \in \{\texttt{unif}, \texttt{pows}\}$.

Public input: $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{m\ell}$. We assume public fixed encodings $U^a, U^b \in L^m$ of $\mathbf{a}, \mathbf{b}$ (more precisely $\mathbf{a}$ is parsed as $\mathbf{a}_1||\mathbf{a}_2||\ldots||\mathbf{a}_m$, where each block is in $\mathbb{F}^\ell$, and the i-th row of U^a encodes $\mathbf{a}_i$ in L, for $i \in [m]$ - and analogously for $\mathbf{b}$).

Oracle: A purported $L^{3m+m'}$-codeword $U = [U^x|U^y|U^z|U^w]^\top$ where it is claimed that $U^x, U^y, U^z \in \mathbb{F}^{m\times n}$ respectively encode $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ with $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$.
We assume we are running the following test in parallel with the Test-Interleaved, and that step 3 ($\mathcal{V}$ choosing Q and receiving $U[j]$, $j \in Q$) is common to both tests.

Interactive testing:

1. $\mathcal{V}$ samples randomness $\mathbf{r} = (r_1, \ldots, r_m)$ according to $\mathsf{Gen}_{\texttt{rand}}(m)$ and sends it to the prover.
2. $\mathcal{P}$ computes the polynomial $p_{\mathsf{mult}}(X) = \sum_{i=1}^m r_i p_i(X)$ where $p_i(X) = p_i^x(X)p_i^y(X) + p_i^a(X)p_i^z(X) - p_i^b(X)$ where p_i^x, p_i^y, p_i^z, p_i^a, p_i^b are the polynomials associated to the codewords in row i of U^x, U^y, U^z, U^a, U^b respectively.
   $\mathcal{P}$ sends p_{mult} to $\mathcal{V}$.
3. $\mathcal{V}$ queries a set $Q \subset [n]$ from the oracle and receives $U[j]$, $j \in Q$.
4. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{mult}} \leq 2k - 2$.
   (b) $p_{\mathsf{mult}}(\zeta_c) = 0$ for every $c \in [\ell]$.
   (c) $\sum_{i=1}^m r_i \cdot (U_i^x[j] \cdot U_i^y[j] + U_i^a[j] \cdot U_i^z[j] - U_i^b[j]) = p_{\mathsf{mult}}(\eta_j)$ for every j in Q.

Figure 2: Test-Multiplication $(\mathbb{F}, L[n, k, d], m, m', t, \mathbf{a}, \mathbf{b}; U)$

We obtain the following:

[6] Here and in the rest of the paper, we will be slightly abusing notation: by $U = [U^x|U^y|U^z|U^w]^\top$ we will always mean $U = \begin{bmatrix} U^x \\ U^y \\ U^z \\ U^w \end{bmatrix}$. That is, we transpose the "position of the blocks", but not the blocks themselves.

Theorem 16. *Let $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{m\ell}$ be public vectors encoded by $U^a, U^b \in L^m$ respectively. Let $U^* = [(U^x)^* | (U^y)^* | (U^z)^* | (U^w)^*]^\top \in \mathbb{F}^{(3m+m')\times n}$ be a possibly malformed oracle for inputs $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}, \mathbf{w} \in \mathbb{F}^{m'\ell}$ where the prover claims that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$. Let $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{pows}\}$. Suppose that Test-Multiplication is run in parallel with Test-Interleaved, and the latter uses a proximity generator $\mathsf{Gen} = \mathsf{Gen}_{\mathtt{rand}}(3m + m')$ with mutual correlated agreement, under the conditions of Theorem 15, with proximity bound $B' = B'(L, 3m + m')$ and error $\varepsilon' = \varepsilon'_{\mathtt{rand}}(L, 3m + m', \gamma)$, where $\gamma \in (0, 1 - B')$. Let*

$$\Lambda := \{U \in L^{3m+m'} : \ d(U, U^*) \le \gamma n\} \quad \text{and} \quad \Lambda_Q := \{U \in \Lambda : \ U[j] = U^*[j] \ \forall j \in Q\}.$$

Then, for any malicious prover $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts and no $U \in \Lambda_Q$ encodes $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ (and $\mathbf{w} \in \mathbb{F}^{m'\ell}$) such that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$ is at most

$$\mathsf{Error}_{\mathsf{mult}} := \varepsilon' + \max\left\{(1-\gamma)^t, \left(2\rho - \frac{2}{n}\right)^t\right\} + |\Lambda| \cdot \frac{c_{\mathtt{rand}}(m) + \gamma n \, c_{\mathtt{rand}}(3m + m')}{|\mathbb{F}|}.$$

In other words, if $\mathcal{V}$ accepts then, except with probability at most $\mathsf{Error}_{\mathsf{mult}}$, some $U \in \Lambda_Q$ encodes $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ (and $\mathbf{w} \in \mathbb{F}^{m'\ell}$) such that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$.

Proof. First introduce the following notation: each U in $L^{3m+m'}$ defines $\mathbf{x}^U = \mathsf{Decode}_{\boldsymbol{\zeta}}(U^x)$, $\mathbf{y}^U = \mathsf{Decode}_{\boldsymbol{\zeta}}(U^y)$, $\mathbf{z}^U = \mathsf{Decode}_{\boldsymbol{\zeta}}(U^z) \in \mathbb{F}^{m\ell}$, and, for each $i \in [m]$, polynomials $p_i^{U,x}$, $p_i^{U,y}$, $p_i^{U,z}$ of degree $\le k-1$ (the polynomials that the prover would use in an honest proof with encoding matrix U). Let $p_i^U(X) = p_i^{U,x}(X)p_i^{U,y}(X) + p_i^a(X)p_i^{U,z}(X) - p_i^b(X)$ and $p^U_{\mathsf{mult}}(X) = \sum_{i=1}^m r_i p_i^U(X)$. We say that U is *valid* if $\mathbf{x}^U \circ \mathbf{y}^U + \mathbf{a} \circ \mathbf{z}^U = \mathbf{b}$, and *invalid* otherwise. We denote by $\mathbf{s} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(3m + m')$ the randomness sent by $\mathcal{V}$ in Test-Interleaved, and by w^* and p^*_{mult} the polynomials sent by $\mathcal{P}^*$ in Test-Interleaved and Test-Multiplication respectively. We can assume $\deg w^* \le k-1$ and $\deg p^*_{\mathsf{mult}} \le 2k-2$, as otherwise the verifier rejects. This defines the word $\mathbf{w}^* := (w^*(\eta_1), \ldots, w^*(\eta_n))$ in L.

Now note that Λ depends only on U^*, and hence is fixed before $\mathcal{V}$ samples any randomness, while $\mathbf{r}$, $\mathbf{s}$, w^* and p^*_{mult} are all fixed before Q is sampled. We consider the following bad events, which only depend on the randomness $\mathbf{r}$ (for B_1), and $\mathbf{s}$ (for B_2, B_3):

- B_1: some invalid $U \in \Lambda$ satisfies, for the choice of $\mathbf{r}$, $p^U_{\mathsf{mult}}(\zeta_c) = 0$ for all $c \in [\ell]$. For a fixed invalid U, there are $i \in [m]$ and $c \in [\ell]$ such that $p_i^U(\zeta_c) \neq 0$, and then by Lemma 10, $p^U_{\mathsf{mult}}(\zeta_c) = \sum_{i=1}^m r_i p_i^U(\zeta_c) = 0$ with probability at most $c_{\mathtt{rand}}(m)/|\mathbb{F}|$. By a union bound, $\Pr_{\mathbf{r}}[B_1] \le |\Lambda| \, c_{\mathtt{rand}}(m)/|\mathbb{F}|$.
- B_2: there exist $U \in \Lambda$ and $j \in [n]$ with $U[j] \neq U^*[j]$ but $\mathbf{s}^\top U[j] = \mathbf{s}^\top U^*[j]$. For a fixed $U \in \Lambda$ there are $d(U, U^*) \le \gamma n$ columns j where the first inequality happens, and for each of them the latter equality holds with probability at most $c_{\mathtt{rand}}(3m + m')/|\mathbb{F}|$ by Lemma 10. Hence $\Pr_{\mathbf{s}}[B_2] \le |\Lambda| \, \gamma n \, c_{\mathtt{rand}}(3m + m')/|\mathbb{F}|$.
- B_3: $\Lambda(L, \mathbf{s}^\top U^*, \gamma) \neq \{\mathbf{s}^\top U : U \in \Lambda\}$. By Lemma 7, $\Pr_{\mathbf{s}}[B_3] \le \varepsilon'$.

Assume from now on that none of B_1, B_2, B_3 happens, and that no $U \in \Lambda_Q$ is valid. We distinguish two cases, depending only on $\mathbf{w}^*$ (and hence fixed before Q is sampled).

Case 1: $\mathbf{w}^* \neq \mathbf{s}^\top U$ for all $U \in \Lambda$ (including if $\Lambda = \emptyset$). Since B_3 does not happen, $\mathbf{w}^* \notin \Lambda(L, \mathbf{s}^\top U^*, \gamma)$, i.e. $d(\mathbf{w}^*, \mathbf{s}^\top U^*) > \gamma n$. The check of Test-Interleaved passes only if $w^*(\eta_j) = \mathbf{s}^\top U^*[j]$ for all $j \in Q$, which, since Q is sampled after w^* is fixed, happens with probability at most $\binom{n-\gamma n-1}{t}/\binom{n}{t} \le (1-\gamma)^t$.

Case 2: $\mathbf{w}^* = \mathbf{s}^\top U_0$ for some $U_0 \in \Lambda$. Fix one such U_0. If the check of Test-Interleaved passes, then for all $j \in Q$ we have $\mathbf{s}^\top U_0[j] = w^*(\eta_j) = \mathbf{s}^\top U^*[j]$, and since B_2 does not happen, $U_0[j] = U^*[j]$. Hence $U_0 \in \Lambda_Q$, so by assumption U_0 is invalid, and since B_1 does not happen, $p^{U_0}_{\mathsf{mult}}(\zeta_c) \neq 0$ for some $c \in [\ell]$. In order to pass check 4b, it must be then that

$p^*_{\mathsf{mult}} \neq p^{U_0}_{\mathsf{mult}}$. Both polynomials have degree $\leq 2k-2$, so they can agree on at most $2k-2$ points. Moreover, since $U_0[j] = U^*[j]$ for $j \in Q$, the left-hand side of check 4c equals $p^{U_0}_{\mathsf{mult}}(\eta_j)$, so 4c passes only if $p^{U_0}_{\mathsf{mult}}(\eta_j) = p^*_{\mathsf{mult}}(\eta_j)$ for all $j \in Q$. Since U_0 and p^*_{mult} are fixed before Q is sampled, this happens with probability at most $\binom{2k-2}{t}/\binom{n}{t} \leq \left(2\rho - \frac{2}{n}\right)^t$.

Therefore, the probability that $\mathcal{V}$ accepts and no $U \in \Lambda_Q$ is valid is at most

$$\Pr[B_1] + \Pr[B_2] + \Pr[B_3] + \max\left\{(1-\gamma)^t, \left(2\rho - \frac{2}{n}\right)^t\right\} \leq \mathsf{Error}_{\mathsf{mult}}.$$

■

Now we consider the linear test, where the prover shows that $U \in \mathbb{F}^{m\times n}$ encodes a $\mathbf{x} \in \mathbb{F}^{m\ell}$ with $A\mathbf{x} = \mathbf{b}$ where $A \in \mathbb{F}^{m'\times m\ell}$, $\mathbf{b} \in \mathbb{F}^{m'}$ are part of the statement. This is presented in Figure 3.

Test-Linear

The test is parametrized by finite field $\mathbb{F}$, $[n,k,d]$-Reed Solomon code L (over $\mathbb{F}$), integers m', m, ℓ, t (with $t \leq n$ and $\ell \leq k$) and a selector $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{powS}\}$.

Public input: Matrix $A \in \mathbb{F}^{m'\times m\ell}$, vector $\mathbf{b} \in \mathbb{F}^{m'}$.

Oracle: A purported L^m-codeword U, which is claimed to encode $\mathbf{x} \in \mathbb{F}^{m\ell}$ such that $A\mathbf{x} = \mathbf{b}$. We assume we are running the following test in parallel with the Test-Interleaved, and that step 3 ($\mathcal{V}$ choosing Q and receiving $U[j]$, $j \in Q$) is common to both tests.

Interactive testing:

1. $\mathcal{V}$ samples randomness $\mathbf{r}$ according to $\mathsf{Gen}_{\mathtt{rand}}(m')$ and sends it to the prover.
2. $\mathcal{P}$ and $\mathcal{V}$ compute $\mathbf{r}^\top A \in \mathbb{F}^{m\ell}$ which is parsed as $(\widetilde{r}_{11}, \ldots, \widetilde{r}_{1\ell}, \ldots, \widetilde{r}_{m1}, \ldots, \widetilde{r}_{m\ell})$ and for $i \in [m]$, the unique polynomials $\widetilde{r}_i(X)$ of degree $\leq \ell - 1$ with $\widetilde{r}_i(\zeta_c) = \widetilde{r}_{ic}$ for $c \in [\ell]$. $\mathcal{P}$ sends $p_{\mathsf{lin}}(X) = \sum_{i=1}^m \widetilde{r}_i(X) p_i(X)$ where $p_i(X)$ is the polynomial encoding U_i, the i-th row of U.
3. $\mathcal{V}$ queries a set $Q \subset [n]$ from the oracle, receives $U[j]$, $j \in Q$.
4. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{lin}} \leq k + \ell - 2$
   (b) $\sum_{c=1}^{\ell} p_{\mathsf{lin}}(\zeta_c) = \mathbf{r}^\top \mathbf{b}$
   (c) For every $j \in Q$, $\sum_{i=1}^m \widetilde{r}_i(\eta_j) \cdot U_i[j] = p_{\mathsf{lin}}(\eta_j)$

Figure 3: Test-Linear $(\mathbb{F}, L[n,k,d], m', m, \ell, t, A, \mathbf{b}; U)$

Theorem 17. *Let $A \in \mathbb{F}^{m'\times m\ell}$ be a public matrix and $\mathbf{b} \in \mathbb{F}^{m'}$ a public vector, where $\ell \leq k$. Let $U^* \in \mathbb{F}^{m\times n}$ be a possibly malformed oracle for an input $\mathbf{x} \in \mathbb{F}^{m\ell}$, where the prover claims that $A\mathbf{x} = \mathbf{b}$. Let $\mathtt{rand} \in \{\mathtt{unif}, \mathtt{powS}\}$. Suppose that Test-Linear is run in parallel with Test-Interleaved, and the latter uses a proximity generator $\mathsf{Gen} = \mathsf{Gen}_{\mathtt{rand}}(m)$ with mutual correlated agreement, under the conditions of Theorem 15, with proximity bound $B' = B'(L,m)$ and error $\varepsilon' = \varepsilon'_{\mathtt{rand}}(L,m,\gamma)$, where $\gamma \in (0, 1-B')$. Let*

$$\Lambda := \{U \in L^m : \ d(U,U^*) \leq \gamma n\} \quad \text{and} \quad \Lambda_Q := \{U \in \Lambda : \ U[j] = U^*[j] \ \forall j \in Q\}.$$

Then, for any malicious prover $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts and no $U \in \Lambda_Q$

encodes $\mathbf{x} \in \mathbb{F}^{m\ell}$ *such that* $A\mathbf{x} = \mathbf{b}$ *is at most*

$$\mathsf{Error}_{\mathsf{lin}} := \varepsilon' + \max\left\{(1-\gamma)^t, \left(\rho + \frac{\ell-2}{n}\right)^t\right\} + |\Lambda| \cdot \frac{c_{\mathtt{rand}}(m') + \gamma n\, c_{\mathtt{rand}}(m)}{|\mathbb{F}|}.$$

Proof. We follow the same structure as in the proof of soundness of the multiplication test. Each U in L^m defines $\mathbf{x}^U = \mathsf{Decode}_{\boldsymbol{\zeta}}(U) \in \mathbb{F}^{m\ell}$ and, for each $i \in [m]$, a polynomial p_i^U of degree $\leq k-1$ such that the i-th row of U is $(p_i^U(\eta_1), \ldots, p_i^U(\eta_n))$. Let $p_{\mathsf{lin}}^U(X) = \sum_{i=1}^m \widetilde{r}_i(X) p_i^U(X)$, which has degree $\leq k+\ell-2$, where the polynomials $\widetilde{r}_i$ are computed from $\mathbf{r}$ and A as in Figure 3. Note that

$$\sum_{c=1}^{\ell} p_{\mathsf{lin}}^U(\zeta_c) = \sum_{i=1}^{m}\sum_{c=1}^{\ell} \widetilde{r}_{ic}\, p_i^U(\zeta_c) = \mathbf{r}^\top A \mathbf{x}^U.$$

Again as in the multiplication test, we say U is valid if $A\mathbf{x}^U = \mathbf{b}$, and otherwise invalid. We denote by w^* and p_{lin}^* the polynomials sent by $\mathcal{P}^*$, which we assume of degrees $\leq k-1$ and $\leq k+\ell-2$, as otherwise the verifier rejects. Let $\mathbf{w}^*$ the codeword associated to w^*. We consider the following bad events:

- B_1: some invalid $U \in \Lambda$ satisfies $\mathbf{r}^\top A\mathbf{x}^U = \mathbf{r}^\top \mathbf{b}$. For invalid U, we have $A\mathbf{x}^U \neq \mathbf{b}$ so by Lemma 10, $\mathbf{r}^\top A\mathbf{x}^U = \mathbf{r}^\top \mathbf{b}$ with probability at most $c_{\mathtt{rand}}(m')/|\mathbb{F}|$. By a union bound, $\Pr_{\mathbf{r}}[B_1] \leq |\Lambda| \cdot c_{\mathtt{rand}}(m')/|\mathbb{F}|$.
- B_2: there exists $U \in \Lambda$ and $j \in [n]$ with $U[j] \neq U^*[j]$ but $\mathbf{s}^\top U[j] = \mathbf{s}^\top U^*[j]$. As in the proof of Theorem 16, $\Pr_{\mathbf{s}}[B_2] \leq |\Lambda| \cdot \gamma \cdot n \cdot c_{\mathtt{rand}}(m)/|\mathbb{F}|$.
- B_3: $\Lambda(L, \mathbf{s}^\top U^*, \gamma) \neq \{\mathbf{s}^\top U : U \in \Lambda\}$. By Lemma 7, $\Pr[B_3] \leq \varepsilon'$.

Assume that none of B_1, B_2, B_3 happens, and no $U \in \Lambda_Q$ is valid. We have two cases.

Case 1: $\mathbf{w}^* \neq \mathbf{s}^\top U$ for all $U \in \Lambda$. By the arguments of Case 1 in Theorem 16 verbatim, the test passes with probability at most $\binom{n-\gamma n-1}{t}/\binom{n}{t} \leq (1-\gamma)^t$.

Case 2: $\mathbf{w}^* = \mathbf{s}^\top U_0$ for some $U_0 \in \Lambda$. The same arguments as in Theorem 16 yield $U_0 \in \Lambda_Q$, so by assumption U_0 is invalid, and since B_1 does not happen,

$$\sum_{c=1}^{\ell} p_{\mathsf{lin}}^{U_0}(\zeta_c) = \mathbf{r}^\top A\mathbf{x}^{U_0} \neq \mathbf{r}^\top \mathbf{b}.$$

In order to pass check 4b, $\mathcal{P}$ needs to send $p_{\mathsf{lin}}^* \neq p_{\mathsf{lin}}^{U_0}$. Both polynomials have degree $\leq k+\ell-2$, so they can agree on at most $k+\ell-2$ points. Now we can argue as in Theorem 16 to conclude that check 4c passes with probability at most $\binom{k+\ell-2}{t}/\binom{n}{t} \leq \left(\rho + \frac{\ell-2}{n}\right)^t$.

As in Theorem 16, we obtain the probability bound in the statement. ∎

On knowledge soundness. In our setting, combining soundness with the fact that there is efficient error-correction up to the Johnson bound (or even the bound $a_1(\rho)$ in the cases from Lemma 12) will give us knowledge soundness as well. The extractor obtains the matrix to which the prover has sent the oracle, error-corrects it to the (polynomially-sized) list of codewords U in L^m that are γ-close to U^*. For each such U, the extractor decodes and obtains the corresponding inputs. By the results above, with high probability, some of these inputs should satisfy the statements (but since the list is polynomially-sized, the extractor can also just check which of the inputs actually satisfy the proposition). We will argue more in detail round-by-round knowledge soundness for our final protocol for proving circuit satisfiability in Section 6.

Comparison with the unique-decoding regime. We briefly compare the soundness errors that we have derived for Test-Linear and Test-Multiplication with the ones originally stated in [AHIV17]. Later in Section 6, we give more precise numbers for the application of these tests to proofs for circuit satisfiability.

Remark 3. According to Lemma 11, for every fixed $\gamma = 1 - \sqrt{\rho} - \sigma$ with $\sigma > 0$, and $\texttt{rand} \in \{\texttt{unif}, \texttt{powS}\}$ we have $\varepsilon'_{\texttt{rand}}(L, 3m + m', \gamma) = O_{m,\sigma,\rho,n}(|\mathbb{F}|^{-1})$. On the other hand, by Remark 1, the size of Λ in Theorem 16 satisfies $|\Lambda| \leq \frac{1-\sqrt{\rho}}{2\sigma} + \frac{1}{2\sqrt{\rho}} = O_{\sigma,\rho}(1)$.

Hence $\mathsf{Error}_{\mathsf{mult}} \leq \max\{(1-\gamma)^t, (2\rho - \frac{2}{n})^t\} + O_{m,m',\sigma,\rho,n}(|\mathbb{F}|^{-1})$ for arbitrarily small constant $\sigma > 0$. An analogous statement holds for $\mathsf{Error}_{\mathsf{lin}}$.

For a large enough finite field $\mathbb{F}$, we can neglect the latter factor. In the Johnson bound regime, as we increase the query size t, the error will decrease as $\lesssim \max\{\sqrt{\rho},\ 2\rho\}^t$. In comparison with this, a soundness analysis that only takes into account the unique decoding regime, as in [AHIV17, AHIV23] gives us $\lesssim \max\{(\frac{1+\rho}{2}), 2\rho\}^t$. For a more visual comparison, we plot these two curves $\max\{\sqrt{\rho}, 2\rho\}$ and $\max\{\frac{1+\rho}{2},\ 2\rho\}$ in the graph in Section C (together with curves that will show the error in the out-of-domain strategy in next section). We can see that the improvement of the list decoding based analysis takes place in the regime where $\rho \leq 1/3$ and it is more significant for smaller values of the rate ρ.

If we assume the regime of more recent results (Lemma 12) we obtain even smaller error, namely $\lesssim \max\{a_1(\rho) + \eta_1,\ 2\rho\}^t$ for the $a_1(\rho)$ and η_1 defined there. See again the graph for the plot of $a_1(\rho)$.

5 Protocols with Out-of-Domain Sampling

In the previous section, we have seen that extending the soundness analysis of Ligero to the list decoding regime, paired with the latest proximity testing results improves one of the terms of the soundness error; namely, ignoring terms that are very small when the field size is large, the soundness error is reduced from $\max\{(\frac{1+\rho}{2})^t, (2\rho)^t\}$ to $\max\{(\sqrt{\rho})^t, (2\rho)^t\}$ (up to the Johnson bound) or even $\max\{(a_1(\rho) + \eta_1)^t, (2\rho)^t\}$ using the results from [DKT26] in Lemma 12. This improves the overall soundness error significantly when the rate ρ is smaller than $1/3$. However, when the Reed-Solomon rate ρ is relatively large, the overall soundness error is dominated by the term $(2\rho)^t$ that comes from Test-Linear and Test-Multiplication, where in particular, we cannot expect any soundness for $\rho \geq 1/2$.

In this section, we adopt an out-of-domain sampling strategy that has been used, in different ways, in other Reed-Solomon based IOPs [BGKS20, ACFY24, ACFY25] and which will remove this $(2\rho)^t$ error term, largely improving the overall soundness error when $\rho \geq 1/3$ and enabling us to go beyond the bound $\rho = 1/2$. As a consequence, the overall soundness error of the resulting protocol will be dominated by the term $\sqrt{\rho}^t$ (or $a_1(\rho)^t$) from the interleaved test when the field size is large.

5.1 Main Idea

In order to understand how out-of-domain sampling achieves this, let us consider the origin of the $(2\rho)^t$-error term in the tests Test-Multiplication and Test-Linear from the previous section. We illustrate this for Test-Multiplication and the reasoning for Test-Linear is akin.

In Test-Multiplication, the prover must send a polynomial p_{mult} of degree $\leq 2k - 2$ constructed from the witness-encoding polynomials p_i^x, p_i^y, p_i^z; if the prover is honest and the statement holds, then p_{mult} should evaluate to 0 at all ζ_c, $c \in [\ell]$. However, a malicious prover can commit to some matrix U which encodes an invalid witness and then, instead of revealing the p^U_{mult} constructed honestly from U, the prover may reveal some other polynomial p^*_{mult} of degree $\leq 2k - 2$ with $p^*_{\mathsf{mult}}(\zeta_c) = 0\ \forall c \in [\ell]$.

The consistency between p^U_{mult} and p^*_{mult} is then tested on a set of t positions η_j within the code domain; therefore, the best strategy of the malicious prover is to choose a p^*_{mult} that coincides with p^U_{mult} as much as possible in the code domain. Since both polynomials are of degree $\leq 2k-2$, the malicious prover can make p^*_{mult} coincide with p^U_{mult} at up to a fraction $\frac{2k-2}{n} \leq 2\rho$ of the domain points, which leads to the $(2\rho)^t$ error.

For the linear test Test-Linear, the situation is analogous, with a polynomial p_{lin} of degree $k+\ell-2$ instead of p_{mult}; in this case the prover can make p^*_{lin} coincide with p^U_{lin} in a fraction $\frac{k+\ell-2}{n}$ of the code domain, which we also bound by 2ρ.

Our new testing procedures instead will check the consistency between p^U_{mult} and p^*_{mult} (and similarly between p^U_{lin} and p^*_{lin} for the linear test) on a *random* point α of the finite field, *outside the domain* $\{\eta_1, \ldots, \eta_n\}$ of the code and the set of evaluation points $\{\zeta_1, \ldots, \zeta_\ell\}$ for the witness. We will use the notation $\mathbb{F}_{\mathrm{OOD}} = \mathbb{F} \setminus \{\zeta_1, \ldots, \zeta_\ell, \eta_1, \ldots, \eta_n\}$. Now the malicious prover can only succeed in having $p^U_{\mathsf{mult}}(\alpha) = p^*_{\mathsf{mult}}(\alpha)$ with probability at most $\frac{2k-2}{|\mathbb{F}_{\mathrm{OOD}}|}$, which is negligible if the field is large. However, this is not for free, as we now need the verifier to compute $p^U_{\mathsf{mult}}(\alpha)$ and, while in Test-Multiplication the values $p^U_{\mathsf{mult}}(\eta_j)$ could be computed simply by querying the committed columns $U[j]$, there is no such commitment available to compute $p^U_{\mathsf{mult}}(\alpha)$.

Instead, the prover will send all necessary values $p^x_i(\alpha)$, $p^y_i(\alpha)$, $p^z_i(\alpha)$ (from which the verifier can compute $p^U_{\mathsf{mult}}(\alpha)$) and then prove that these values are consistent with the committed U. This proof will reuse Test-Interleaved (Figure 1): recall that in that test the prover is revealing some polynomial $v(X)$ of degree $\leq k-1$ which corresponds to a random linear combination of the polynomials p^x_i, p^y_i, p^z_i. Now the verifier checks the consistency of the polynomial v with p^x_i, p^y_i, p^z_i when evaluated in both α and the queried positions η_j, $j \in Q$. In order to pass the test, the prover needs to send a polynomial $v^*(X)$ of degree $\leq k-1$ such that $v^*(\alpha)$ is consistent with $p^x_i(\alpha)$, $p^y_i(\alpha)$, $p^z_i(\alpha)$, and the randomness. But with overwhelming probability this cannot be the "correct" $v^U(X)$ computed from U if $p^U_{\mathsf{mult}} \neq p^*_{\mathsf{mult}}$ as we are assuming. And then, this will be detected "in-domain" except with the error probability of Test-Interleaved. Note that the key point here is that this part of the test is carried out with polynomials of degree $k-1$ and not $2k-2$, which is what allows to use the guarantees from Test-Interleaved.

The case of Test-Linear is analogous.

In summary we have been able to turn a test of evaluation equality of polynomials of degree $2k-2$ on t points in the domain (leading to error $(2\rho)^t$), into one test of evaluation equality of these polynomials on one point α *outside* the domain, plus one test for the evaluation of polynomials *of degree* $k-1$ on both α and t points in the domain, the latter of which is actually Test-Interleaved, leading to an $\sqrt{\rho}^t + O(|\mathbb{F}|^{-1})$ error. We now discuss these tests in detail.

5.2 Multiplication Test

We first present the multiplication test using out-of-domain sampling, which we call Test-MultOOD, in Figure 4.

Remark 4. Note that Test-MultOOD contains Test-Interleaved from Figure 1 on $3m+m'$ rows. Indeed this is formed by steps 5-7, together with the degree check on v in 8a and check 8e. On the other hand, steps 1-2, the degree check on p_{mult} in 8a and check 8b are part of Test-Multiplication from Figure 2; however, the other check from Test-Multiplication, which compares the evaluation of p_{mult} in all points indexed by Q with the opened columns in U, is now replaced by check 8c which compares instead the evaluation on α with the vector $\mathbf{h}$ sent in step 4. This $\mathbf{h}$ is purportedly the "extension" of U to the out-of-domain challenge point α, whose consistency with U is ensured by checks 8d and 8e.

Test-MultOOD (Multiplication with Out of Domain sampling)

The test is parametrized by finite field $\mathbb{F}$, $[n,k,d]$-Reed Solomon code L (over $\mathbb{F}$), integers m, m', ℓ, t (with $t \leq n$ and $\ell \leq k < n$) and a selector $\texttt{rand} \in \{\texttt{unif}, \texttt{powS}\}$.

Public input: $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{m\ell}$. We assume public fixed encodings $U^a, U^b \in L^m$ of $\mathbf{a}, \mathbf{b}$ (more precisely $\mathbf{a}$ is parsed as $\mathbf{a}_1||\mathbf{a}_2||\ldots||\mathbf{a}_m$, where each block is in $\mathbb{F}^\ell$, and the i-th row of U^a encodes $\mathbf{a}_i$ in L, for $i \in [m]$ - and analogously for $\mathbf{b}$).

Oracle: A purported $L^{3m+m'}$-codeword $U = [U^x|U^y|U^z|U^w]^\top$ where it is claimed that $U^x, U^y, U^z \in \mathbb{F}^{m\times n}$ respectively encode (as explained above for $\mathbf{a}$, $\mathbf{b}$) $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ with $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$ and U^w encodes some $\mathbf{w} \in \mathbb{F}^{m'\ell}$.

Interactive testing:

1. $\mathcal{V}$ samples randomness $\mathbf{r} = (r_1, \ldots, r_m) \leftarrow \mathsf{Gen}_{\texttt{rand}}(m)$ and sends it to the prover.
2. $\mathcal{P}$ computes the polynomial $p_{\mathsf{mult}}(X) = \sum_{i=1}^m r_i p_i(X)$ where $p_i(X) = p_i^x(X)p_i^y(X) + p_i^a(X)p_i^z(X) - p_i^b(X)$ where p_i^x, p_i^y, p_i^z, p_i^a, p_i^b are the polynomials associated to the codewords in row i of U^x, U^y, U^z, U^a, U^b respectively. $\mathcal{P}$ sends p_{mult} to $\mathcal{V}$.
3. $\mathcal{V}$ chooses $\alpha \in \mathbb{F}_{\mathrm{OOD}}$ and sends it to prover.
4. $\mathcal{P}$ sends $\mathbf{h} = ((p_i^x(\alpha))_{i=1}^m, (p_i^y(\alpha))_{i=1}^m, (p_i^z(\alpha))_{i=1}^m, (p_i^w(\alpha))_{i=1}^{m'}) \in \mathbb{F}^{3m+m'}$.
   $\mathcal{V}$ parses it as $((h_i^x)_{i=1}^m, (h_i^y)_{i=1}^m, (h_i^z)_{i=1}^m, (h_i^w)_{i=1}^{m'})$.
5. $\mathcal{V}$ samples $\mathbf{s} \leftarrow \mathsf{Gen}_{\texttt{rand}}(3m+m')$, sends it to $\mathcal{P}$.
   Parse $\mathbf{s}$ as $((s_i^x)_{i=1}^m, (s_i^y)_{i=1}^m, (s_i^z)_{i=1}^m, (s_i^w)_{i=1}^{m'})$.
6. $\mathcal{P}$ sends $v(X) = \sum_{i=1}^m (s_i^x p_i^x + s_i^y p_i^y + s_i^z p_i^z) + \sum_{i=1}^{m'} s_i^w p_i^w$ of degree $\leq k-1$.
7. $\mathcal{V}$ selects $Q \subseteq [n]$ of size t and queries U in Q obtaining columns $U[j]$, $j \in Q$.
8. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{mult}} \leq 2k-2$, $\deg v \leq k-1$
   (b) $p_{\mathsf{mult}}(\zeta_c) = 0$ for every $c \in [\ell]$
   (c) $p_{\mathsf{mult}}(\alpha) = \sum_{i=1}^m r_i(h_i^x h_i^y + p_i^a(\alpha)h_i^z - p_i^b(\alpha))$
   (d) $v(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}$
   (e) $v(\eta_j) = \mathbf{s}^\top \cdot U[j]$ for $j \in Q$

Figure 4: Test-MultOOD $(\mathbb{F}, L, m, m', \ell, t, \texttt{rand}, \mathbf{a}, \mathbf{b}; U)$

Theorem 18. *With the same notation in Theorem 16, for any malicious prover $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts in the protocol of Figure 4 and no $U \in \Lambda$ encodes $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ (and $\mathbf{w} \in \mathbb{F}^{m'\ell}$) such that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$ is at most*

$$\mathsf{Error}_{\mathsf{mult}} := \varepsilon' + (1-\gamma)^t + |\Lambda| \left(\frac{c_{\texttt{rand}}(m) + c_{\texttt{rand}}(3m+m')}{|\mathbb{F}|} + \frac{2k-2}{|\mathbb{F}| - (n+\ell)} \right).$$

Proof. We use the same notation as in the proof of Theorem 16 (a $*$ superindex means a value sent by the malicious prover, or derived from what they send, and a superindex U is a value sent in an honest proof with encoding matrix $U \in L^{3m+m'}$). In particular, for the new element $\mathbf{h}$ in this proof, we have notations $\mathbf{h}^*$, $\mathbf{h}^U$ with respective coordinates $h_i^{*,x}$, $h_i^{U,x}$ and the corresponding notations for y and z. Again U being *valid* means it encodes a witness, and *invalid* that it encodes a non-witness. In this case we consider the following bad events:

- B_1: some invalid $U \in \Lambda$ satisfies $p_{\mathsf{mult}}^U(\zeta_c) = 0$ for all $c \in [\ell]$. This is the same as in Theorem 16, where we deduced $\Pr[B_1] \leq |\Lambda|\, c_{\texttt{rand}}(m)/|\mathbb{F}|$.

- B_2: some $U \in \Lambda$ satisfies $p^U_{\mathsf{mult}} \neq p^*_{\mathsf{mult}}$ but $p^U_{\mathsf{mult}}(\alpha) = p^*_{\mathsf{mult}}(\alpha)$. We may assume $\deg p^*_{\mathsf{mult}} \leq 2k-2$, since otherwise check 8a is rejected. For a fixed U, p^U_{mult} depends only on U and $\mathbf{r}$, so both polynomials are fixed before α is sampled, and they can coincide in at most $2k-2$ points. Since α is uniform in $\mathbb{F}_{\mathrm{OOD}}$,

$$\Pr[B_2] \leq |\Lambda| \frac{2k-2}{|\mathbb{F}_{\mathrm{OOD}}|} = |\Lambda| \frac{2k-2}{|\mathbb{F}| - (n+\ell)}.$$

- B_3: some $U \in \Lambda$ satisfies $\mathbf{h}^U \neq \mathbf{h}^*$ but $\mathbf{s}^\top \cdot \mathbf{h}^U = \mathbf{s}^\top \cdot \mathbf{h}^*$. By Lemma 10,

$$\Pr[B_3] \leq |\Lambda|\, c_{\mathtt{rand}}(3m+m')/|\mathbb{F}|.$$

- B_4: $\Lambda(L, \mathbf{s}^\top U^*, \gamma) \neq \{\mathbf{s}^\top U : U \in \Lambda\}$. By Lemma 7, $\Pr[B_4] \leq \varepsilon'$.

Assume from now on that none of $B_1, \ldots, B_4$ happens and that no $U \in \Lambda$ is valid. We can also assume $\deg v^* \leq k-1$, as otherwise check 8a is rejected. We distinguish two cases, depending only on v^* (and hence determined before Q is sampled).

Case 1: $\mathbf{v}^* \neq \mathbf{s}^\top U$ for all $U \in \Lambda$, including the case $\Lambda = \emptyset$. Exactly as in previous proofs, the test passes with probability at most $\binom{n-\gamma n-1}{t}/\binom{n}{t} \leq (1-\gamma)^t$.

Case 2: $\mathbf{v}^* = \mathbf{s}^\top U_0$ for some $U_0 \in \Lambda$, i.e. $v^* = v^{U_0}$. This case is a bit different than in Theorem 16, as in this case $\mathcal{V}$ always rejects, regardless of Q.[7] First, check 8d tests whether $v^*(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^*$. Since $v^*(\alpha) = v^{U_0}(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^{U_0}$ and B_3 does not happen, 8d can only pass if $\mathbf{h}^* = \mathbf{h}^{U_0}$. Assume this is the case. Then the right-hand side of check 8c is

$$\sum_{i=1}^{m} r_i \big(h_i^{U_0,x} h_i^{U_0,y} + p_i^a(\alpha) h_i^{U_0,z} - p_i^b(\alpha) \big) = p^{U_0}_{\mathsf{mult}}(\alpha),$$

so check 8c can only pass if $p^*_{\mathsf{mult}}(\alpha) = p^{U_0}_{\mathsf{mult}}(\alpha)$, which, since B_2 does not happen, implies $p^*_{\mathsf{mult}} = p^{U_0}_{\mathsf{mult}}$. Then 8b can only pass if $p^{U_0}_{\mathsf{mult}}(\zeta_c) = 0$ for all $c \in [\ell]$. But we are assuming U_0 is invalid and B_1 does not happen, so $p^{U_0}_{\mathsf{mult}}(\zeta_c) \neq 0$ for some $c \in [\ell]$, and 8b is rejected.

Altogether, we obtain the probability bound in the statement. ∎

5.3 Linear Test

We now present the linear test using out-of-domain sampling, Test-LinOOD, in Figure 5.

Remark 5. As happened with Figure 4, Test-LinOOD contains Test-Interleaved as steps 5-7, together with the degree check on v in 8a and check 8e. On the other hand, steps 1-2, the degree check on p_{lin} in 8a and check 8b are part of Test-Linear (Figure 3) and the other check from Test-Linear, which compares the evaluation of p_{lin} in all points indexed by Q with the opened columns in U, is now replaced by check 8c which compares instead the evaluation on α with the vector $\mathbf{h}$ sent in step 4, whose consistency with U is ensured by checks 8d and 8e.

Theorem 19. *With the same notation as in Theorem 17, for any malicious prover $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts in the protocol of Figure 5 and no $U \in \Lambda$ encodes $\mathbf{x} \in \mathbb{F}^{m\ell}$ such that $A\mathbf{x} = \mathbf{b}$ is at most*

$$\mathsf{Error}_{\mathsf{lin}} := \varepsilon' + (1-\gamma)^t + |\Lambda| \left(\frac{c_{\mathtt{rand}}(m') + c_{\mathtt{rand}}(m)}{|\mathbb{F}|} + \frac{k+\ell-2}{|\mathbb{F}| - (n+\ell)} \right).$$

[7] Seen in another way, we are proving that if the proof passes in Case 2 when B_1 and B_3 (equivalent to B_1 and B_2 in Theorem 16) do not happen, then B_2 has to happen, but the difference is that this is an event which is independent from Q, and that is why we have dealt with it already.

Test-LinOOD (Linear with Out of Domain sampling)

The test is parametrized by finite field $\mathbb{F}$, $[n,k,d]$-Reed Solomon code L (over $\mathbb{F}$), integers m', m, ℓ, t (with $t \leq n$ and $\ell \leq k$) and a selector $\texttt{rand} \in \{\texttt{unif}, \texttt{pows}\}$.

Public input: Matrix $A \in \mathbb{F}^{m' \times m\ell}$, vector $\mathbf{b} \in \mathbb{F}^{m'}$

Oracle: A purported L^m-codeword U, which is supposed to encode $\mathbf{x} \in \mathbb{F}^{m\ell}$ such that $A\mathbf{x} = \mathbf{b}$ (more precisely, $\mathbf{x}$ is parsed as $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 || \ldots || \mathbf{x}_m$ where each $\mathbf{x}_i \in \mathbb{F}^\ell$, and the i-th row of U encodes $\mathbf{x}_i$).

Interactive testing:

1. $\mathcal{V}$ samples randomness $\mathbf{r}$ according to $\mathsf{Gen}_{\texttt{rand}}(m')$ and sends it to the prover.
2. $\mathcal{P}$ and $\mathcal{V}$ compute $\mathbf{r}^\top A \in \mathbb{F}^{m\ell}$ which is parsed as $(\widetilde{r}_{11}, \ldots, \widetilde{r}_{1\ell}, \ldots, \widetilde{r}_{m1}, \ldots, \widetilde{r}_{m\ell})$ and, for $i \in [m]$, they compute the unique polynomials $\widetilde{r}_i(X)$ of degree $\leq \ell - 1$ with $\widetilde{r}_i(\zeta_c) = \widetilde{r}_{ic}$ for $c \in [\ell]$.
   $\mathcal{P}$ sends $p_{\mathsf{lin}}(X) = \sum_{i=1}^m \widetilde{r}_i(X) p_i(X)$ where $p_i(X)$ is the polynomial encoding U_i.
3. $\mathcal{V}$ chooses $\alpha \in \mathbb{F}_{\mathrm{OOD}}$ and sends it to prover.
4. $\mathcal{P}$ sends to $\mathcal{V}$ the vector $\mathbf{h} = (p_i(\alpha))_{i=1}^m \in \mathbb{F}^m$.
5. $\mathcal{V}$ samples $\mathbf{s} \leftarrow \mathsf{Gen}_{\texttt{rand}}(m)$, sends it to $\mathcal{P}$
6. $\mathcal{P}$ constructs $v(X) = \sum_{i=1}^m s_i p_i(X)$ and sends it to $\mathcal{V}$
7. $\mathcal{V}$ selects $Q \subseteq [n]$ of size t and queries U in Q obtaining columns $U[j], j \in Q$
8. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{lin}} \leq k + \ell - 2$, $\deg v \leq k - 1$
   (b) $\sum_{c=1}^\ell p_{\mathsf{lin}}(\zeta_c) = \mathbf{r}^\top \cdot \mathbf{b}$
   (c) $p_{\mathsf{lin}}(\alpha) = \bar{\mathbf{r}}^\top \cdot \mathbf{h}$, where $\bar{\mathbf{r}} = (\widetilde{r}_i(\alpha))_{i=1}^m$
   (d) $v(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}$,
   (e) $v(\eta_j) = \mathbf{s}^\top \cdot U[j]$ for $j \in Q$

Figure 5: Test-LinOOD $(\mathbb{F}, L[n,k,d], m', m, \ell, t, A, \mathbf{b}; U)$ (Linear with Out Of Domain sampling)

Proof. The proof is similar in structure to previous ones, and we use the same notations. We consider the following bad events:

- B_1: some invalid $U \in \Lambda$ satisfies $\mathbf{r}^\top A \mathbf{x}^U = \mathbf{r}^\top \mathbf{b}$. Exactly as in the proof of Theorem 17, $\Pr[B_1] \leq |\Lambda|\, c_{\texttt{rand}}(m')/|\mathbb{F}|$.
- B_2: some $U \in \Lambda$ satisfies $p^U_{\mathsf{lin}} \neq p^*_{\mathsf{lin}}$ but $p^U_{\mathsf{lin}}(\alpha) = p^*_{\mathsf{lin}}(\alpha)$. We may assume $\deg p^*_{\mathsf{lin}} \leq k + \ell - 2$, since otherwise check (a) is rejected. For a fixed U, p^U_{lin} depends only on U and $\mathbf{r}$, so both polynomials are fixed before α is sampled, and they can coincide in at most $k + \ell - 2$ points. Since α is uniform in $\mathbb{F}_{\mathrm{OOD}}$, by a union bound $\Pr[B_2] \leq |\Lambda| \frac{k+\ell-2}{|\mathbb{F}_{\mathrm{OOD}}|} = |\Lambda| \frac{k+\ell-2}{|\mathbb{F}|-(n+\ell)}$.
- B_3: some $U \in \Lambda$ satisfies $\mathbf{h}^U \neq \mathbf{h}^*$ but $\mathbf{s}^\top \cdot \mathbf{h}^U = \mathbf{s}^\top \cdot \mathbf{h}^*$. As in Theorem 18, $\Pr[B_3] \leq |\Lambda|\, c_{\texttt{rand}}(m)/|\mathbb{F}|$.
- B_4: $\Lambda(L, \mathbf{s}^\top U^*, \gamma) \neq \{\mathbf{s}^\top U : U \in \Lambda\}$. By Lemma 7, $\Pr[B_4] \leq \varepsilon'$.

As usual, assume none of $B_1, \ldots, B_4$ happens and no $U \in \Lambda$ is valid.

Case 1: $\mathbf{v}^* \neq \mathbf{s}^\top U$ *for all* $U \in \Lambda$ (this includes the case $\Lambda = \emptyset$).As in previous proofs, the proof passes with probability at most $\binom{n-\gamma n-1}{t} / \binom{n}{t} \leq (1-\gamma)^t$.

Case 2: $\mathbf{v}^* = \mathbf{s}^\top U_0$ *for some* $U_0 \in \Lambda$, i.e. $v^* = v^{U_0}$. We show that in this case $\mathcal{V}$ rejects. First, check 8d tests whether $v^*(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^*$. Since $v^*(\alpha) = v^{U_0}(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^{U_0}$ and B_3 does not happen, check 8d can only pass if $\mathbf{h}^* = \mathbf{h}^{U_0}$. Assume this is the case. Then the right-hand side of check 8c is $\bar{\mathbf{r}}^\top \cdot \mathbf{h}^{U_0} = \sum_{i=1}^m \widetilde{r}_i(\alpha) p_i^{U_0}(\alpha) = p_{\mathsf{lin}}^{U_0}(\alpha)$, so check 8c can only pass if $p_{\mathsf{lin}}^*(\alpha) = p_{\mathsf{lin}}^{U_0}(\alpha)$, which, since B_2 does not happen, implies $p_{\mathsf{lin}}^* = p_{\mathsf{lin}}^{U_0}$. Assume this is the case. Then check 8b can only pass if $\sum_{c=1}^{\ell} p_{\mathsf{lin}}^{U_0}(\zeta_c) = \mathbf{r}^\top \mathbf{b}$, i.e. $\mathbf{r}^\top A \mathbf{x}^{U_0} = \mathbf{r}^\top \mathbf{b}$. But U_0 is invalid by assumption, so since B_1 does not happen, check 8b is rejected.

Altogether, we obtain the probability bound in the statement. ∎

5.4 Adding Zero-Knowledge

In this section we discuss how to add the property of honest-verifier zero-knowledge to the protocols presented in this section.

We first explain why Test-LinOOD and Test-MultOOD are not directly honest-verifier zero-knowledge. First, the verifier learns the evaluations of the input-encoding polynomials at the $t+1$ points η_j, $j \in Q$, and α, which must not reveal anything about their values at $\zeta_1, \ldots, \zeta_\ell$, i.e. the inputs. This is however easily fixed by taking $k \geq \ell + t + 1$, as already done in [AHIV17] (where $k \geq \ell + t$ suffices, since there is no OOD evaluation).

Second, the prover reveals the polynomials v (from Test-Interleaved, part of both tests), p_{mult} (in Test-MultOOD) and p_{lin} (in Test-LinOOD), which may leak further information. For v we use the solution of [AHIV17]: the prover commits to an additional uniformly random row $\mathbf{u}_{\mathsf{iblind}} \in L$, encoding p_{iblind} of degree $\leq k-1$, which enters the linear combination v with coefficient 1, thereby perfectly masking the polynomial that would be sent in the non-ZK version. This makes v uniformly random, and the degree of p_{iblind} is tested by Test-Interleaved together with the other rows.

For p_{mult} and p_{lin} we would also like to add uniformly random masks p_{mblind} and p_{lblind} of degree $\leq 2k-2$ and $\leq k+\ell-2$ respectively, subject to $p_{\mathsf{mblind}}(\zeta_c) = 0$ for all $c \in [\ell]$ and $\sum_{c=1}^{\ell} p_{\mathsf{lblind}}(\zeta_c) = 0$ (so that the masked polynomials still pass the corresponding checks). The problem is soundness: the degrees of the masks must be enforced, but Test-Interleaved only tests degree $\leq k-1$.

As in STARKs [BS21, HK24], we have the prover instead build each mask from random polynomials of degree $\leq k-1$, which it commits to as additional rows of the oracle, so that their degrees are tested by Test-Interleaved. For the linear test it suffices to take $p_{\mathsf{lblind}} = p_{\mathsf{lblind}}^{(0)} + X^{\ell-1} p_{\mathsf{lblind}}^{(1)}$, with $p_{\mathsf{lblind}}^{(0)}, p_{\mathsf{lblind}}^{(1)}$ uniformly random of degree $\leq k-1$ subject to $\sum_c p_{\mathsf{lblind}}(\zeta_c) = 0$ (Lemma 24).

However, the analogous split $p_{\mathsf{mblind}} = p_{\mathsf{mblind}}^{(0)} + X^{k-1} p_{\mathsf{mblind}}^{(1)}$ does not give zero-knowledge. The reason is that Test-Interleaved requires revealing $p_{\mathsf{mblind}}^{(0)}(\eta_j)$, $p_{\mathsf{mblind}}^{(1)}(\eta_j)$, for $j \in Q$, and this reveals *too much information* about p_{mblind}. Note that the problem is not that the verifier learns $p_{\mathsf{mblind}}(\eta_j)$ (this is unavoidable as it is anyway revealed from the rest of the proof, as happens in the linear test too) but that these values actually leak more information than that about p_{mblind}.

Intuitively, the two pieces overlap in a single coefficient of p_{mblind}, whereas explaining $t+1$ arbitrary evaluations requires an overlap of at least $t+1$ coefficients (in the linear case the overlap is $k - \ell + 1 \geq t+1$). We therefore use three pieces, $p_{\mathsf{mblind}} = p_{\mathsf{mblind}}^{(0)} + X^e p_{\mathsf{mblind}}^{(1)} + X^{k-1} p_{\mathsf{mblind}}^{(2)}$, for some $t \leq e \leq k-t-1$, so that consecutive pieces overlap in $k - e \geq t+1$ and $e+1 \geq t+1$ coefficients (this requires $k \geq 2t+1$). Lemma 21 shows that this preserves zero-knowledge.

We describe the ZK-version Test-ZKMultOOD of Test-MultOOD in Figure 6 and the ZK-version Test-ZKLinOOD of Test-LinOOD in Figure 7.

Test-ZKMultOOD (Multiplication with Out of Domain sampling)

The test is parametrized by finite field $\mathbb{F}$, $[n,k,d]$-Reed Solomon code L (over $\mathbb{F}$), integers m, m', ℓ, t, e with $\ell + t + 1 \leq k \leq n$ and $t \leq e \leq k - t - 1$ and $\texttt{rand} \in \{\texttt{unif}, \texttt{powS}\}$.

Public input: $\mathbf{a}, \mathbf{b} \in \mathbb{F}^{m\ell}$ encoded by $U^a, U^b \in L^m$ as in Test-MultOOD.

Oracle: A purported $L^{3m+m'}$-codeword $U = [U^x|U^y|U^z|U^w]^\top$ as in Test-MultOOD

Interactive testing:

1. Let $\Phi : \mathbb{F}[X]^3 \to \mathbb{F}[X]$ be defined by $(f_0, f_1, f_2) \mapsto f_0 + X^e f_1 + X^{k-1} f_2$.
   $\mathcal{P}$ samples $(p^{(0)}_{\mathsf{mblind}}, p^{(1)}_{\mathsf{mblind}}, p^{(2)}_{\mathsf{mblind}})$ uniformly at random in
   $$W = \{\mathbf{f} := (f_0, f_1, f_2) \in (\mathbb{F}[X])^3 : \deg f_i \leq k-1, (\Phi(\mathbf{f}))(\zeta_c) = 0 \ \forall\ c \in [\ell]\}$$
   $\mathcal{P}$ also chooses p_{iblind} uniformly at random of degree $\leq k-1$.
   $\mathcal{P}$ commits to the L-codewords $(f(\eta_j))_{j\in[n]}$, $f \in \{p^{(0)}_{\mathsf{mblind}}, p^{(1)}_{\mathsf{mblind}}, p^{(2)}_{\mathsf{mblind}}, p_{\mathsf{iblind}}\}$. Let $U' = [U|U_{\mathsf{mblind}}|U_{\mathsf{iblind}}]^\top$ where U_{mblind} has as 3 rows the 3 first codewords above and U_{iblind} has as only row the last codeword. Note that $\mathcal{V}$ can query $U'[j]$ by querying $U[j]$ and the j-th components of these new rows.
   Moreover set $p_{\mathsf{mblind}} = \Phi((p^{(0)}_{\mathsf{mblind}}, p^{(1)}_{\mathsf{mblind}}, p^{(2)}_{\mathsf{mblind}})) = p^{(0)}_{\mathsf{mblind}} + X^e p^{(1)}_{\mathsf{mblind}} + X^{k-1} p^{(2)}_{\mathsf{mblind}}$.
2. $\mathcal{V}$ samples randomness $\mathbf{r} = (r_1, \ldots, r_m, 1)$ according to $\mathsf{Gen}_{\texttt{rand}}(m+1)$ and sends it to the prover.
3. $\mathcal{P}$ computes $p_{\mathsf{mult}}(X) = \sum_{i=1}^m r_i p_i(X) + p_{\mathsf{mblind}}$ where $p_i(X) = p^x_i(X)p^y_i(X) + p^a_i(X)p^z_i(X) - p^b_i(X)$ where p^x_i, p^y_i, p^z_i, p^a_i, p^b_i are the polynomials associated to the codewords in row i of U^x, U^y, U^z, U^a, U^b respectively. $\mathcal{P}$ sends p_{mult} to $\mathcal{V}$.
4. $\mathcal{V}$ chooses $\alpha \in \mathbb{F}_{\mathrm{OOD}}$ and sends it to prover.
5. Call the polynomials associated to the codewords of U' as $u'_1(X), \ldots, u'_{3m+m'+4}(X)$ (i.e. the polynomials $p^x_i, p^y_i, p^z_i, p^w_i, p^{(j)}_{\mathsf{mblind}}, p_{\mathsf{iblind}}$).
   $\mathcal{P}$ sends $\mathbf{h} = (u'_\iota(\alpha))_{\iota=1}^{3m+m'+4} \in \mathbb{F}^{3m+m'+4}$. Denote h^x_i, h^y_i, h^z_i ($i \in [m]$), and $h^{(j)}_{\mathsf{mblind}}$ ($j \in [0,2]$) the values corresponding to $p^x_i(\alpha)$, $p^y_i(\alpha)$, $p^z_i(\alpha)$, $p^{(j)}_{\mathsf{mblind}}(\alpha)$, respectively.
6. $\mathcal{V}$ samples $\mathbf{s} \leftarrow \mathsf{Gen}_{\texttt{rand}}(3m + m' + 4)$, sends it to $\mathcal{P}$.
7. $\mathcal{P}$ sends the polynomial $v(X) = \sum_{\iota=1}^{3m+m'+4} s_\iota u'_\iota(X)$ of degree $\leq k-1$.
8. $\mathcal{V}$ selects $Q \subseteq [n]$ of size t and queries U' in Q obtaining columns $U'[j]$, $j \in Q$.
9. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{mult}} \leq 2k-2$, $\deg v \leq k-1$
   (b) $p_{\mathsf{mult}}(\zeta_c) = 0$ for every $c \in [\ell]$
   (c) $p_{\mathsf{mult}}(\alpha) = \sum_{i=1}^m r_i(h^x_i h^y_i + p^a_i(\alpha)h^z_i - p^b_i(\alpha)) + h^{(0)}_{\mathsf{mblind}} + \alpha^e h^{(1)}_{\mathsf{mblind}} + \alpha^{k-1} h^{(2)}_{\mathsf{mblind}}$
   (d) $v(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}$
   (e) $v(\eta_j) = \mathbf{s}^\top \cdot U'[j]$ for $j \in Q$

Figure 6: Test-ZKMultOOD $(\mathbb{F}, L, m, m', \ell, t, e, \texttt{rand}, \mathbf{a}, \mathbf{b}; U)$

Theorem 20. *The Protocol Test-ZKMultOOD in Figure 6 is perfectly honest-verifier zero-knowledge*

The proof relies on the following algebraic lemma.

Lemma 21. *Let $\ell + t + 1 \leq k$ and $t \leq e \leq k - t - 1$.*

Test-ZKLinOOD (Linear with Out of Domain sampling)

The test is parametrized by finite field $\mathbb{F}$, $[n,k,d]$-Reed Solomon code L (over $\mathbb{F}$), integers m', m, ℓ, t (with $t \leq n$ and $\ell + t + 1 \leq k$) and a selector $\texttt{rand} \in \{\texttt{unif}, \texttt{pows}\}$.

Public input: Matrix $A \in \mathbb{F}^{m' \times m\ell}$, vector $\mathbf{b} \in \mathbb{F}^{m'}$

Oracle: A purported L^m-codeword U, which is supposed to encode $\mathbf{x} \in \mathbb{F}^{m\ell}$ such that $A\mathbf{x} = \mathbf{b}$ (more precisely, $\mathbf{x}$ is parsed as $\mathbf{x} = \mathbf{x}_1 || \mathbf{x}_2 || \ldots || \mathbf{x}_m$ where each $\mathbf{x}_i \in \mathbb{F}^\ell$, and the i-th row of U encodes $\mathbf{x}_i$).

Interactive testing:

1. Let $\Phi' : \mathbb{F}[X]^2 \to \mathbb{F}[X]$ be defined by $(f_0, f_1) \mapsto f_0 + X^{\ell-1} f_1$.
   $\mathcal{P}$ samples $(p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}})$ uniformly at random in
   $$W = \{\mathbf{f} := (f_0, f_1) \in (\mathbb{F}[X])^2 : \deg f_i \leq k-1, \sum_{c=1}^{\ell} (\Phi'(\mathbf{f}))(\zeta_c) = 0 \}$$
   $\mathcal{P}$ also chooses p_{iblind} uniformly at random of degree $\leq k-1$.
   $\mathcal{P}$ commits to the L-codewords $(f(\eta_j))_{j\in[n]}$, $f \in \{p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}}, p_{\mathsf{iblind}}\}$. Let $U' = [U|U_{\mathsf{lblind}}|U_{\mathsf{iblind}}]^\top$ where U_{lblind} has as 2 rows the 2 first codewords above and U_{iblind} has as only row the last codeword. Note that $\mathcal{V}$ can query $U'[j]$ by querying $U[j]$ and the j-th components of these new rows.
   Call p_i, $i = 1, \ldots, m+3$ the polynomials encoded by the rows of U' (hence $p_{m+1} = p^{(0)}_{\mathsf{lblind}}$, $p_{m+2} = p^{(1)}_{\mathsf{lblind}}$, $p_{m+3} = p_{\mathsf{iblind}}$). Moreover set
   $$p_{\mathsf{lblind}} = \Phi'((p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}})) = p^{(0)}_{\mathsf{lblind}} + X^{\ell-1} p^{(1)}_{\mathsf{lblind}}$$
2. $\mathcal{V}$ samples and sends randomness $\mathbf{r}$ according to $\mathsf{Gen}_{\texttt{rand}}(m'+1)$.
3. $\mathcal{P}$ and $\mathcal{V}$ compute $\mathbf{r}_*^\top A := (\widetilde{r}_{11}, \ldots, \widetilde{r}_{1\ell}, \ldots, \widetilde{r}_{m1}, \ldots, \widetilde{r}_{m\ell}) \in \mathbb{F}^{m\ell}$ where $\mathbf{r}_* \in \mathbb{F}^{m'}$ is $\mathbf{r}$ without its last component (which we set to 1 in our paper). For $i \in [m]$, they compute the unique polynomials $\widetilde{r}_i(X)$ of degree $\leq \ell - 1$ with $\widetilde{r}_i(\zeta_c) = \widetilde{r}_{ic}$ for $c \in [\ell]$. Also define $\widetilde{r}_{m+1}(X) := 1$, $\widetilde{r}_{m+2}(X) := X^{\ell-1}$, $\widetilde{r}_{m+3} := 0$.
   $\mathcal{P}$ sends $p_{\mathsf{lin}}(X) := \sum_{i=1}^{m+3} \widetilde{r}_i(X) p_i(X)$. Note $p_{\mathsf{lin}}(X) = \sum_{i=1}^{m} \widetilde{r}_i(X) p_i(X) + p_{\mathsf{lblind}}$.
4. $\mathcal{V}$ chooses $\alpha \in \mathbb{F}_{\mathrm{OOD}}$ and sends it to prover.
5. $\mathcal{P}$ sends to $\mathcal{V}$ the vector $\mathbf{h} = (p_i(\alpha))_{i=1}^{m+3} \in \mathbb{F}^{m+3}$.
6. $\mathcal{V}$ samples $\mathbf{s} \leftarrow \mathsf{Gen}_{\texttt{rand}}(m+3)$, sends it to $\mathcal{P}$.
7. $\mathcal{P}$ constructs $v(X) = \sum_{i=1}^{m+3} s_i p_i(X)$ and sends it to $\mathcal{V}$.
8. $\mathcal{V}$ selects $Q \subseteq [n]$ of size t and queries U' in Q obtaining columns $U'[j], j \in Q$.
9. $\mathcal{V}$ accepts iff
   (a) $\deg p_{\mathsf{lin}} \leq k + \ell - 2$, $\deg v \leq k - 1$
   (b) $\sum_{c=1}^{\ell} p_{\mathsf{lin}}(\zeta_c) = \mathbf{r}_*^\top \cdot \mathbf{b}$
   (c) $p_{\mathsf{lin}}(\alpha) = \bar{\mathbf{r}}^\top \cdot \mathbf{h}$, where $\bar{\mathbf{r}} = (\widetilde{r}_i(\alpha))_{i=1}^{m+3}$
   (d) $v(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}$
   (e) $v(\eta_j) = \mathbf{s}^\top \cdot U'[j]$ for $j \in Q$

Figure 7: Test-ZKLinOOD $(\mathbb{F}, L, m', m, \ell, t, \texttt{rand}, A, \mathbf{b}; U)$

Let $V_0 = \{g \in \mathbb{F}[X] : \deg g \leq 2k-2, \quad g(\zeta_c) = 0, \ \forall\ c \in [\ell]\ \}$ *and*

$$W = \{\mathbf{f} := (f_0, f_1, f_2) \in (\mathbb{F}[X])^3 : \deg f_i \leq k-1, \qquad i = 0,1,2$$
$$f_0(\zeta_c) + \zeta_c^e f_1(\zeta_c) + \zeta_c^{k-1} f_2(\zeta_c) = 0, \quad \forall\ c \in [\ell]\}.$$

Define $\Phi : (\mathbb{F}[X])^3 \to \mathbb{F}[X]$ *given by* $\Phi(\mathbf{f}) = f_0(X) + X^e f_1(X) + X^{k-1} f_2(X)$
For any $T \subseteq \mathbb{F} \setminus \{\zeta_1, \ldots, \zeta_\ell\}$ *of size* $|T| = t+1$*, define the map* $\pi_T : (\mathbb{F}[X])^3 \to \mathbb{F}^{2(t+1)}$ *given by* $\pi_T(\mathbf{f}) = (f_1(\theta), f_2(\theta))_{\theta \in T}$*. Then*

1. $\Phi(W) = V_0$*. In particular, if* $\mathbf{f}$ *is uniform in* W *then* $\Phi(\mathbf{f})$ *is uniform in* V_0*.*
2. *For every* $g \in V_0$*,* $\pi_T(W \cap \Phi^{-1}(\{g\})) = \mathbb{F}^{2(t+1)}$*. In particular if* $\mathbf{f}$ *is uniform in* $W \cap \Phi^{-1}(\{g\})$*, then* $\pi_T(\mathbf{f})$ *is uniform in* $\mathbb{F}^{2(t+1)}$*.*

Proof. 1. It is clear that $\Phi(W) \subseteq V_0$. On the other hand every $g \in V_0$ can be written as $f_0 + X^{k-1} f_2$, such that $\deg f_0, \deg f_2 \leq k-1$ and this automatically satisfies $f_0(\zeta_c) + \zeta_c^{k-1} f_2(\zeta_c) = 0$. Then, clearly, $(f_0, 0, f_2) \in W \cap \Phi^{-1}(g)$ and the map is surjective when restricted to W, i.e. $\Phi(W) = V_0$. Since W and V_0 are vector spaces and Φ is a linear function, the part about uniformity follows.

2. By part 1., the $W \cap \Phi^{-1}(\{g\})$ is non-empty. Moreover, it is a coset of the vector space $W \cap \Phi^{-1}(\{0\})$ (namely $W \cap \Phi^{-1}(\{g\}) = \mathbf{h} + W \cap \Phi^{-1}(\{0\})$ for any fixed $\mathbf{h} \in W \cap \Phi^{-1}(\{g\})$); therefore it is enough to prove the result for $g = 0$, as the corresponding distributions of $(f_1(\theta), f_2(\theta))$ will be just shifted by a fixed $(h_1(\theta), h_2(\theta))$.

Let $(y_{1,\theta}, y_{2,\theta})_{\theta \in T}$ an arbitrary element in $\mathbb{F}^{2(t+1)}$ and we want to find $\mathbf{f}$ in $W \cap \Phi^{-1}(\{0\})$ with $f_i(\theta) = y_{i,\theta}$ for $i = 1, 2$ and $\theta \in T$.

Before explicitly constructing this $\mathbf{f}$, note that for $\mathbf{f}$ to be in $\Phi^{-1}(\{0\})$, we need $f_0 + X^e f_1 + X^{k-1} f_2 = 0$, so in particular to cancel out the coefficients of $X^i, i \geq k$, we need to set $f_1(X) = a(X) + b(X) X^{k-e}$, $f_2 = d - b(X) X$, where $\deg a \leq k - e - 1$, $d \in \mathbb{F}$ and $\deg b(X) \leq e - 1$. In particular note that f_2 actually has to be of degree $\leq e$.

Given that, we fix it in the following way:

- First choose f_2 of degree $\leq e$, satisfying $f_2(\theta) = y_{2,\theta}$ for $\theta \in T$. We can do this because $|T| = t+1$ and $e \geq t$. Split $f_2(X) = d - b(X) X$ where $d \in \mathbb{F}$ and $\deg b(X) \leq e - 1$.
- Now for $\theta \in T$, the values $y'_{1,\theta} = y_{1,\theta} - b(\theta)\theta^{k-e}$ are fixed. Choose $a(X)$ of degree $\deg a \leq k - e - 1$ with $a(\theta) = y'_{1,\theta}$ for $\theta \in T$. Again we can do this because $|T| = t+1$ and $k - e - 1 \geq t$. Writing $f_1(X) = a(X) + b(X) X^{k-e}$ we have $f_1(\theta) = y_{1,\theta}$ for θ in T.
- Take $f_0 = -X^e f_1 - X^{k-1} f_2$. Then clearly $\mathbf{f} \in \Phi^{-1}(\{0\})$ and $f_0(\zeta_c) + \zeta_c^e f_1(\zeta_c) + \zeta_c^{k-1} f_2(\zeta_c) = 0$ for all ζ_c. Moreover the degree of f_0 is $\leq k - 1$ because of our choice of f_1 and f_2, so $\mathbf{f} \in W$ as well.

Finally, as in 1., the fact that π_T is a linear surjective map between vector spaces implies the uniformity condition. ∎

Now we can prove Theorem 20

Proof of Theorem 20. The simulator samples the verifier's randomness $\mathbf{r} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(m+1)$, $\alpha \leftarrow \mathbb{F}_{\mathrm{OOD}}$, $\mathbf{s} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(3m + m' + 4)$ and $Q \subseteq [n]$ with $|Q| = t$.

The simulator queries $U[j]$, $j \in Q$, which provides $U^x[j], U^y[j], U^z[j], U^w[j]$. It now fixes p_i^x, p_i^y, p_i^z (and p_i^w) of degree $\leq k-1$ such that $p_i^x(\eta_j) = U_i^x[j]$ for $j \in Q$ (same for y, z) and that the polynomial $S = \sum_{i=1}^m r_i \left(p_i^x p_i^y + p_i^a p_i^z - p_i^b\right)$ satisfies $S(\zeta_c) = 0$ for $c \in [\ell]$.

To do that, pick $i_0 \in [m]$ with $r_{i_0} \neq 0$. If none exists set $i_0 = \perp$. For $i \in [m] \setminus \{i_0\}$ the simulator picks p_i^x uniformly among polynomials of degree $\leq k-1$ with $p_i^x(\eta_j) = U_i^x[j]$ for $j \in Q$, and analogously for y, z (and for w with $i \in [m']$). Let

$$S_* = \sum_{i=1, i \neq i_0}^{m} r_i \left(p_i^x p_i^y + p_i^a p_i^z - p_i^b\right).$$

If $i_0 = \perp$, then $S = S_*$ and notice that this case happens if $r_i = 0$ for all $i \in [m]$, so $S = 0$ and consequently $S(\zeta_c) = 0$. Otherwise, we use $i = i_0$ to "correct" S_* so that $S(\zeta_c) = 0$ as follows: the simulator picks, as before, $p^z_{i_0}$ uniformly among polynomials of degree $\leq k-1$ with $p^z_{i_0}(\eta_j) = U^z_{i_0}[j]$ for $j \in Q$, but for $p^x_{i_0}, p^y_{i_0}$ it sets the additional restrictions: $p^y_{i_0}(\zeta_c) = 1$ for all $c \in [\ell]$, and $p^x_{i_0}(\zeta_c) = -r^{-1}_{i_0} S_*(\zeta_c) - p^a_{i_0} p^z_{i_0}(\zeta_c) + p^b_{i_0}(\zeta_c)$. It is possible to choose them in this way because in total we have $t + \ell$ restrictions, and $k \geq \ell + t$. In this way $S := \sum_{i=1}^m r_i \left(p^x_i p^y_i + p^a_i p^z_i - p^b_i\right)$ satisfies $S(\zeta_c) = 0$ for all $c \in [\ell]$.

Let V_0, W be as in Lemma 21. Note $S \in V_0$ by the previous observations. The simulator samples p_0 uniformly in V_0 and sets $g = p_0 - S$. By construction, $g \in V_0$. It then samples $\mathbf{f}$ uniformly in $W \cap \Phi^{-1}(g)$. The simulator sets $p^{(j)}_{\mathsf{mblind}} = f_j$, $j \in [0, 2]$.

The simulator samples v uniformly of degree $\leq k-1$ and defines p_{iblind} as the unique polynomial for which $v(X)$ is the "$\mathbf{s}$-combination" of all rows of U' (the coefficient of p_{iblind} is the last coordinate of $\mathbf{s}$, which equals 1, so we can always do this).

The simulator outputs p_0, v, the vector h of evaluations at α of all row polynomials of U', and the simulated entries $U'[j]$, $j \in Q$, of the rows $\mathbf{u}^{(i)}_{\mathsf{mblind}}, i \in [0, 2]$ and $\mathbf{u}_{\mathsf{iblind}}$.

We compare with the real view, writing $T = \{\eta_j : j \in Q\} \cup \{\alpha\}$.

(a) The entries of $U[Q]$ are the same in both views, since the simulator obtains them by querying the oracle. Conditioned on them, in the real view the evaluations on α of the witness-row polynomials $p^x_i, p^y_i, p^z_i, p^w_i$ are uniform in $\mathbb{F}^{3m+m'}$: they have degree $\leq k-1$ and hence their evaluations in α are uniformly random given their evaluations at the $\ell + t$ values ζ_c, $c \in [\ell]$ and η_j, $j \in Q$, since $\ell + t + 1 \leq k$. In the simulated view these evaluations at α are also uniform given the constraints: namely the evaluations on the points η_j, $j \in Q$ for all polynomials p^*_i and additionally, for $p^x_{i_0}$, $p^y_{i_0}$, on the points $\zeta_c, c \in [\ell]$. This is again because we take the polynomials uniform of degree $\leq k-1$ constrained to these (at most) $t + \ell$ evaluations, and $t + \ell + 1 \leq k$.

(b) In the real view $p_0 = S + \Phi(\mathbf{f})$ with $S \in V_0$, and by Lemma 21, part 1, p_0 is uniform in V_0 and independent of the elements fixed in (a). The same happens in the simulation.

(c) Conditioned on (a) and (b), in both views $\mathbf{f}$ is uniform in $W \cap \Phi^{-1}(g)$ for some $g \in V_0$. For $\theta \in T$, $g(\theta)$ is given (in the same way in the real run and the simulation) by p_0, $\mathbf{r}$, α, Q, the public polynomials p^a_i, p^b_i, and the entries of $U'[Q]$ and $\mathbf{h}$ corresponding to the rows of U^x, U^y, U^z (i.e. excluding the blinding rows).[8] By Lemma 21, part 2, the values $(f_1(\theta), f_2(\theta))_{\theta \in T}$ are uniform, and $f_0(\theta)$ is determined by them and $g(\theta)$. Hence the blinding entries of h and of $U'[Q]$ are identically distributed.

(d) In the real view, p_{iblind} is uniform of degree $\leq k-1$ and independent of everything in (a)-(c). Hence v is uniform among polynomials of degree $\leq k-1$ conditioned on everything fixed in (a)-(c). In the simulation v is sampled uniformly by construction. Finally, in both views $p_{\mathsf{iblind}}(\theta)$ for $\theta \in T$ are given by $p_{\mathsf{iblind}}(\theta) = v(\theta) - \sum_R s_R\, p_R(\theta)$, where the sum ranges over all rows R of U' other than $\mathbf{u}_{\mathsf{iblind}}$, so they are also identically distributed.

■

Theorem 22. *The conditions are as in Theorems 16 and 18 with the following changes: $(U')^* \in \mathbb{F}^{(3m+m'+4)\times n}$ denotes a possibly malformed oracle, consisting of the oracle $U^* = [(U^x)^* | (U^y)^* | (U^z)^* | (U^w)^*]^\top$ as in previous theorems, together with the four additional rows committed by $\mathcal{P}^*$ in step 1 of Figure 6. Now we have $\mathsf{Gen}_{\mathtt{rand}}(3m + m' + 4)$, a proximity generator for L with mutual correlated agreement, with proximity bound B' and error ε'. For $\gamma \in (0, 1 - B')$, let $\Lambda := \Lambda(L^{3m+m'+4}, (U')^*, \gamma)$.*

Then, for any malicious prover $\mathcal{P}^*$, the probability that $\mathcal{V}$ accepts in the protocol of Figure 6 and no $U' \in \Lambda$ has its first $3m + m'$ rows encoding $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ (and $\mathbf{w} \in \mathbb{F}^{m'\ell}$)

[8] Namely, $g(\theta) = p_0(\theta) - \sum_{i=1}^m r_i (p^x_i(\theta) p^y_i(\theta) + p^a_i(\theta) p^z_i(\theta) - p^b_i(\theta))$, and the evaluations $p^x_i(\theta), p^y_i(\theta), p^z_i(\theta)$ are given by $U[Q]$ (if $\theta = \eta_j$, $j \in Q$) or by the corresponding entries in $\mathbf{h}$ (if $\theta = \alpha$).

such that $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$ *is at most*

$$\mathsf{Error}^{\mathsf{zk}}_{\mathsf{mult}} := \varepsilon' + (1-\gamma)^t + |\Lambda| \left(\frac{c_{\mathtt{rand}}(m+1) + c_{\mathtt{rand}}(3m+m'+4)}{|\mathbb{F}|} + \frac{2k-2}{|\mathbb{F}| - (n+\ell)} \right).$$

Proof. We again use the notations with U' superindices to refer to elements of the proof honestly computed from a matrix $U' \in L^{3m+m'+4}$. Note now U' is valid if *its first* $3m+m'$ *rows* encode $\mathbf{x}, \mathbf{y}, \mathbf{z}$ with $\mathbf{x} \circ \mathbf{y} + \mathbf{a} \circ \mathbf{z} = \mathbf{b}$, and invalid otherwise. Note that Λ depends only on U'^*, and hence is fixed before $\mathcal{V}$ samples any randomness.

We consider the following bad events:

- B_1: some invalid $U' \in \Lambda$ satisfies $p^{U'}_{\mathsf{mult}}(\zeta_c) = 0$ for all $c \in [\ell]$. We proceed as in previous proofs: the only change is that now we are considering the inner product between $\left(p^{U'}_1(\zeta_c), \ldots, p^{U'}_m(\zeta_c), p^{U'}_{\mathsf{mblind}}(\zeta_c)\right) \in \mathbb{F}^{m+1}$ and $\mathbf{r} = (r_1, \ldots, r_m, 1) \leftarrow \mathsf{Gen}_{\mathtt{rand}}(m+1)$. We still have that since the former is fixed before $\mathbf{r}$ is chosen, by Lemma 10, $p^{U'}_{\mathsf{mult}}(\zeta_c) = \sum_{i=1}^m r_i p^{U'}_i(\zeta_c) + p^{U'}_{\mathsf{mblind}}(\zeta_c) = 0$ with probability at most $c_{\mathtt{rand}}(m+1)/|\mathbb{F}|$. Therefore (again using the union bound) $\Pr[B_1] \leq |\Lambda|\, c_{\mathtt{rand}}(m+1)/|\mathbb{F}|$.
- B_2: some $U' \in \Lambda$ satisfies $p^{U'}_{\mathsf{mult}} \neq p^*_{\mathsf{mult}}$ but $p^{U'}_{\mathsf{mult}}(\alpha) = p^*_{\mathsf{mult}}(\alpha)$. This is the same exact argument in Theorem 18, and $\Pr[B_2] \leq |\Lambda| \frac{2k-2}{|\mathbb{F}|-(n+\ell)}$.
- B_3: some $U' \in \Lambda$ satisfies $\mathbf{h}^{U'} \neq \mathbf{h}^*$ but $\mathbf{s}^\top \cdot \mathbf{h}^{U'} = \mathbf{s}^\top \cdot \mathbf{h}^*$. By Lemma 10 and the union bound, $\Pr[B_3] \leq |\Lambda|\, c_{\mathtt{rand}}(3m+m'+4)/|\mathbb{F}|$.
- B_4: $\Lambda(L, \mathbf{s}^\top U'^*, \gamma) \neq \{\mathbf{s}^\top U' : U' \in \Lambda\}$. By Lemma 7, $\Pr[B_4] \leq \varepsilon'$.

Assume as usual none of $B_1, \ldots, B_4$ happens, $\deg v^* \leq k-1$, and no $U' \in \Lambda$ is valid.

Case 1: $v^* \neq \mathbf{s}^\top U'$ for all $U' \in \Lambda$. There is nothing new here: by the arguments from previous proofs, the proof still passes with probability at most $\binom{n-\gamma n-1}{t}/\binom{n}{t} \leq (1-\gamma)^t$.

Case 2: $v^* = \mathbf{s}^\top U'_0$ for some $U'_0 \in \Lambda$, i.e. $v^* = v^{U_0}$. Again we show that in this case $\mathcal{V}$ rejects. First, check 9d tests whether $v^*(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^*$. Since $v^*(\alpha) = v^{U'_0}(\alpha) = \mathbf{s}^\top \cdot \mathbf{h}^{U'_0}$ and B_3 does not happen, check 9d can only pass if $\mathbf{h}^* = \mathbf{h}^{U'_0}$. Assume this is the case. Then the right-hand side of check 9c is

$$\sum_{i=1}^m r_i \left(h_i^{U'_0,x} h_i^{U'_0,y} + p_i^a(\alpha) h_i^{U'_0,z} - p_i^b(\alpha) \right) + h^{U'_0,(0)}_{\mathsf{mblind}} + \alpha^e h^{U_0,(1)}_{\mathsf{mblind}} + \alpha^{k-1} h^{U'_0,(2)}_{\mathsf{mblind}} = p^{U'_0}_{\mathsf{mult}}(\alpha),$$

so check 9c can only pass if $p^*_{\mathsf{mult}}(\alpha) = p^{U'_0}_{\mathsf{mult}}(\alpha)$, which, since B_2 does not happen, implies $p^*_{\mathsf{mult}} = p^{U_0}_{\mathsf{mult}}$. But then check 9b can only pass if $p^{U'_0}_{\mathsf{mult}}(\zeta_c) = 0$ for all $c \in [\ell]$. This cannot happen because U'_0 is invalid and B_1 does not happen, so check 9b is rejected.

Altogether, we obtain the probability bound in the statement. ∎

Theorem 23. *The Protocol Test-ZKLinOOD in Figure 7 is perfectly honest-verifier zero-knowledge.*

The proof follows the same structure as that of Theorem 20, with Lemma 24 below playing the role of Lemma 21.

Lemma 24. *Let* $\ell + t + 1 \leq k$. *Let* $V'_0 = \{g \in \mathbb{F}[X] : \deg g \leq k+\ell-2,\ \sum_{c=1}^\ell g(\zeta_c) = 0\}$ *and* $W' = \{\mathbf{f} := (f_0, f_1) \in (\mathbb{F}[X])^2 : \deg f_i \leq k-1,\ \sum_{c=1}^\ell \left(f_0(\zeta_c) + \zeta_c^{\ell-1} f_1(\zeta_c)\right) = 0\}$.

Let $\Phi' : (\mathbb{F}[X])^2 \to \mathbb{F}[X]$ *be given by* $\Phi'(\mathbf{f}) = f_0(X) + X^{\ell-1} f_1(X)$.

For $T \subseteq \mathbb{F} \setminus \{\zeta_1, \ldots, \zeta_\ell\}$ *of size* $|T| = t+1$, *define* $\pi'_T : (\mathbb{F}[X])^2 \to \mathbb{F}^{t+1}$ *given by* $\pi'_T(\mathbf{f}) = (f_1(\theta))_{\theta \in T}$. *Then*

1. $\Phi'(W') = V'_0$. *In particular, if* $\mathbf{f}$ *is uniform in* W' *then* $\Phi'(\mathbf{f})$ *is uniform in* V'_0.

2. *For every* $g \in V'_0$, $\pi'_T(W' \cap \Phi'^{-1}(\{g\})) = \mathbb{F}^{t+1}$. *In particular if* $\mathbf{f}$ *is uniform in* $W' \cap \Phi'^{-1}(\{g\})$, *then* $\pi'_T(\mathbf{f})$ *is uniform in* $\mathbb{F}^{t+1}$.

Proof. 1. Since $\sum_{c=1}^{\ell} \Phi'(\mathbf{f})(\zeta_c) = \sum_{c=1}^{\ell} \big(f_0(\zeta_c) + \zeta_c^{\ell-1} f_1(\zeta_c)\big)$ and $\deg \Phi'(\mathbf{f}) \le k + \ell - 2$ whenever $\deg f_0, \deg f_1 \le k-1$, it is clear that $\Phi'(W') \subseteq V'_0$. On the other hand, every g in V'_0, can be written as $g = f_0 + X^{\ell-1} f_1$ with $f_0 = g \bmod X^{\ell-1}$, and $f_1 = (g - f_0)/X^{\ell-1}$. Then $\deg f_0 \le \ell - 2 \le k-1$, $\deg f_1 \le k-1$ and $\Phi'((f_0, f_1)) = g$. Moreover $\sum_c \big(f_0(\zeta_c) + \zeta_c^{\ell-1} f_1(\zeta_c)\big) = \sum_c g(\zeta_c) = 0$, so $(f_0, f_1) \in W' \cap \Phi'^{-1}(\{g\})$. Hence $\Phi'(W') = V'_0$. Since W' and V'_0 are vector spaces and Φ' is linear, the part about uniformity follows.

2. By part 1., $W' \cap \Phi'^{-1}(\{g\})$ is non-empty. Moreover, it is a coset of the vector space $W' \cap \Phi'^{-1}(\{0\})$ (namely $W' \cap \Phi'^{-1}(\{g\}) = \mathbf{h} + W' \cap \Phi'^{-1}(\{0\})$ for any fixed $\mathbf{h} \in W' \cap \Phi'^{-1}(\{g\})$); therefore it is enough to prove the result for $g = 0$, as the corresponding distribution of $(f_1(\theta))_{\theta \in T}$ will be just shifted by $(h_1(\theta))_{\theta \in T}$.

Let $(y_\theta)_{\theta \in T}$ be an arbitrary element in $\mathbb{F}^{t+1}$; we want to find $\mathbf{f}$ in $W' \cap \Phi'^{-1}(\{0\})$ with $f_1(\theta) = y_\theta$ for $\theta \in T$.

Note that any f_1 of degree $\le k - \ell$ gives $\mathbf{f} = (-X^{\ell-1} f_1, f_1) \in \Phi'^{-1}(\{0\})$, and $\mathbf{f} \in W'$ as well, since $\sum_c \Phi'(\mathbf{f})(\zeta_c) = 0$. Hence it suffices to choose f_1 of degree $\le k - \ell$ with $f_1(\theta) = y_\theta$ for $\theta \in T$. We can do this because $|T| = t + 1$ and $k - \ell \ge t + 1 > t$.

Finally, as in 1., the fact that π'_T is a linear surjective map between vector spaces implies the uniformity condition. ∎

Proof of Theorem 23. Recall that the rows of U' in Test-ZKLinOOD (Figure 7) are encoded by $p_1, \ldots, p_{m+3}$, where $p_{m+1} = p^{(0)}_{\mathsf{lblind}}$, $p_{m+2} = p^{(1)}_{\mathsf{lblind}}$ and $p_{m+3} = p_{\mathsf{iblind}}$. The simulator proceeds as in the proof of Theorem 20, with the following changes.

After sampling $\mathbf{r} \leftarrow \mathsf{Gen}_{\mathsf{rand}}(m' + 1)$, $\alpha \leftarrow \mathbb{F}_{\mathrm{OOD}}$, $\mathbf{s} \leftarrow \mathsf{Gen}_{\mathsf{rand}}(m + 3)$, Q, and querying $U[Q]$, the simulator samples $p_1, \ldots, p_m$ of degree $\le k - 1$ with $p_i(\eta_j) = U_i[j]$ for $j \in Q$, now under the condition $\sum_{c=1}^{\ell} S(\zeta_c) = \mathbf{r}_*^\top \mathbf{b}$ for $S := \sum_{i=1}^{m} \widetilde{r}_i\, p_i$, i.e. $\sum_{i,c} \widetilde{r}_{ic}\, p_i(\zeta_c) = \mathbf{r}_*^\top \mathbf{b}$. We can do this using similar arguments to the ones in the proof of Theorem 20.

The simulator samples p_{lin} uniformly in $\{q : \deg q \le k + \ell - 2,\ \sum_c q(\zeta_c) = \mathbf{r}_*^\top \mathbf{b}\} = S + V'_0$, sets $g = p_{\mathsf{lin}} - S \in V'_0$, samples $\mathbf{f}$ uniformly in $W' \cap \Phi'^{-1}(\{g\})$, and sets $(p_{m+1}, p_{m+2}) = (f_0, f_1)$. Finally, v and p_{m+3} are sampled as in the proof of Theorem 20, and the simulator outputs p_{lin}, v, $\mathbf{h} = (p_i(\alpha))_{i=1}^{m+3}$ and the entries $U'_i[j]$, $j \in Q$, $i \in \{m+1, m+2, m+3\}$.

The comparison with the real view is as in items (a)–(d) of the proof of Theorem 20, with $T = \{\eta_j : j \in Q\} \cup \{\alpha\}$. Items (a) and (d) are verbatim, since the witness-row polynomials are again uniform of degree $\le k - 1$ subject to at most $\ell + t < k$ evaluations. For (b), in the real view $p_{\mathsf{lin}} = S + \Phi'(\mathbf{f})$ with $\sum_c S(\zeta_c) = \mathbf{r}_*^\top A\mathbf{x} = \mathbf{r}_*^\top \mathbf{b}$, so by Lemma 24, part 1, p_{lin} is uniform in $S + V'_0$, which does not depend on S, as in the simulation. For (c), $g(\theta) = p_{\mathsf{lin}}(\theta) - \sum_{i=1}^{m} \widetilde{r}_i(\theta) p_i(\theta)$ is determined in both views by the elements in (a) and (b), so by Lemma 24, part 2, $(f_1(\theta))_{\theta \in T}$ is uniform and $f_0(\theta) = g(\theta) - \theta^{\ell-1} f_1(\theta)$; hence the blinding entries of $\mathbf{h}$ and $U'[Q]$ are identically distributed.

∎

Theorem 25. *The conditions are as in Theorems 17 and 19 with the following changes:* $U'^* \in \mathbb{F}^{(m+3)\times n}$ *denotes a possibly malformed oracle consisting of* $U^* \in \mathbb{F}^{m \times n}$ *together with the three rows committed by* $\mathcal{P}^*$ *in step 1 of Figure 7;* $\mathsf{Gen}_{\mathsf{rand}}(m + 3)$ *is a proximity generator for* L *with mutual correlated agreement, with proximity bound* B' *and error* ε'*; and, for* $\gamma \in (0, 1 - B')$, $\Lambda := \Lambda(L^{m+3}, U'^*, \gamma)$. *Then, for any malicious prover* P^*, *the probability that* V *accepts in the protocol of Figure 7 and no* $U \in \Lambda$ *has its first* m *rows*

encoding $x \in \mathbb{F}^{m\ell}$ with $Ax = b$ is at most

$$\mathsf{Error}^{\mathsf{zk}}_{\mathsf{lin}} := \varepsilon' + (1-\gamma)^t + |\Lambda| \left(\frac{c_{\mathsf{rand}}(m'+1) + c_{\mathsf{rand}}(m+3)}{|\mathbb{F}|} + \frac{k+\ell-2}{|\mathbb{F}| - (n+\ell)} \right).$$

Proof. We follow the proof of Theorem 19, with U'^*, $\mathsf{Gen}_{\mathsf{rand}}(m+3)$ in place of U^*, $\mathsf{Gen}_{\mathsf{rand}}(m)$. For $U' \in L^{m+3}$ with row polynomials $p_1^{U'}, \ldots, p_{m+3}^{U'}$, let $\mathbf{x}^{U'}$ be the vector encoded by its first m rows, $p^{U'}_{\mathsf{lblind}} := p^{U'}_{m+1} + X^{\ell-1} p^{U'}_{m+2}$, and $p^{U'}_{\mathsf{lin}} := \sum_{i=1}^{m+3} \tilde{r}_i\, p_i^{U'} = \sum_{i=1}^{m} \tilde{r}_i\, p_i^{U'} + p^{U'}_{\mathsf{lblind}}$. Since $\tilde{r}_i(\zeta_c) = \tilde{r}_{ic}$, we have $\sum_{c=1}^{\ell} p^{U'}_{\mathsf{lin}}(\zeta_c) = \mathbf{r}_*^\top A \mathbf{x}^{U'} + \beta^{U'}$, with $\beta^{U'} := \sum_{c=1}^{\ell} p^{U'}_{\mathsf{lblind}}(\zeta_c)$.

As in Theorem 22, the only event whose analysis changes with respect to the non-ZK version (Theorem 19) is B_1, since a malicious prover need not sample the blinding rows in W', so we cannot assume $\beta^{U'} = 0$:

- B_1: some invalid $U' \in \Lambda$ satisfies $\sum_{c=1}^{\ell} p^{U'}_{\mathsf{lin}}(\zeta_c) = \mathbf{r}_*^\top \mathbf{b}$. For a fixed invalid U', the vector $\mathbf{d}^{U'} := (A\mathbf{x}^{U'} - \mathbf{b}, \beta^{U'}) \in \mathbb{F}^{m'+1}$ is non-zero in one of its first m' coordinates and is fixed before $\mathbf{r} = (\mathbf{r}_*, 1)$ is sampled. Since $\mathbf{r}^\top \mathbf{d}^{U'} = \sum_c p^{U}_{\mathsf{lin}}(\zeta_c) - \mathbf{r}_*^\top b$, Lemma 10 and a union bound give $\Pr[B_1] \le |\Lambda|\, c_{\mathsf{rand}}(m'+1)/|\mathbb{F}|$.

B_2, B_3, B_4 are defined and bounded as in Theorem 19, with $c_{\mathsf{rand}}(m+3)$ in place of $c_{\mathsf{rand}}(m)$ in B_3. Cases 1 and 2 are verbatim, noting that in Case 2 the right-hand side of check 9c is $\bar{\mathbf{r}}^\top \mathbf{h}^{U'_0} = \sum_{i=1}^{m+3} \tilde{r}_i(\alpha) p_i^{U'_0}(\alpha) = p^{U'_0}_{\mathsf{lin}}(\alpha)$. ∎

We combine the three tests to obtain a proof system for circuit satisfiability, following the original Ligero protocol [AHIV17]. We present the construction in Section 6.

6 Complete Circuit Satisfiability Protocol

In this section, we describe a honest-verifier zero-knowledge proof for circuit satisfiability that uses the tests that we have presented in the previous section. We recall how [AHIV17] encodes an arithmetic circuit as a pair of linear and multiplicative statements, and then we combine tests Test-ZKMultOOD and Test-ZKLinOOD in a manner where the interleaved code test, as well as the queried evaluations of the polynomials (both in and out of the domain), are common to both tests. Moreover, the prover creates directly oracles to a matrix U that already contains the evaluations of the blinding polynomials created in Test-ZKMultOOD and Test-ZKLinOOD. For the sake of presentation, in this case, we include the encodings of all these random polynomials in a submatrix U_{blind} of U, rather than including them in different submatrices (e.g. $(U^x)'$, $(U^y)'$, $(U^z)'$, $(U^w)'$ in Test-ZKMultOOD).

Let C be an arithmetic circuit over $\mathbb{F}$ with ninp inputs and ngates gates (addition and multiplication). The statement to be proved is that the prover knows $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{\mathsf{ninp}}) \in \mathbb{F}^{\mathsf{ninp}}$ such that $C(\boldsymbol{\nu}) = 1$.

We fix positive integers m, ℓ, k, t, n, e such that $m\ell > \mathsf{ninp} + \mathsf{ngates}$, $n > k \ge \ell + t + 1$, $k \ge 2t+1$ and $t \le e \le k - t - 1$. We fix $\zeta_1, \ldots, \zeta_\ell, \eta_1, \ldots, \eta_n$ a set of $n+\ell$ pairwise distinct elements in $\mathbb{F}$ and we fix the Reed Solomon code $\mathsf{RS}_{\mathbb{F},n,k,\boldsymbol{\eta}}$.

- **Circuit satisfiability encoding**

  $\mathcal{P}$ constructs the extended witness $\mathbf{w} = (\nu_1, \ldots, \nu_{\mathsf{ninp}}, \beta_1, \ldots, \beta_{\mathsf{ngates}}, 0, \ldots, 0) \in \mathbb{F}^{m\ell}$ where β_i is the output of the i-th gate of C. The statement $C(\boldsymbol{\nu}) = 1$ amounts to $\beta_{\mathsf{ngates}} = 1$.

  The prover also constructs $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{F}^{m\ell}$ with $\mathbf{x}[j], \mathbf{y}[j], \mathbf{z}[j]$ being the coordinates in $\mathbf{w}$ which are the left and right inputs and the output of the j-th multiplication gate, for

$j \in [\mathsf{ngates}]$, and lets P_x, P_y, $P_z \in \mathbb{F}^{m\ell \times m\ell}$ such that $P_x\mathbf{w} = \mathbf{x}$, $P_y\mathbf{w} = \mathbf{y}$, $P_z\mathbf{w} = \mathbf{z}$ - note the first $\mathsf{nmultgates}$ rows (where $\mathsf{nmultgates}$ is the number of multiplication gates) are unit vectors, the rest of them are 0. The prover also constructs $P_{add} \in \mathbb{F}^{m\ell \times m\ell}$ such that $P_{add}\mathbf{w}$ equals the vector containing $\beta_a + \beta_b - \beta_c$ in its j-th coordinate, where $\beta_a, \beta_b, \beta_c$ are respectively the left, right inputs and outputs of the j-th addition gate. Moreover, if n_{add} denotes the number of addition gates, we set the $(n_{add}+1)$-th row of P_{add} to be the unit vector selecting the coordinate of $\mathbf{w}$ that contains β_{ngates}, and we let $\mathbf{b} \in \mathbb{F}^{4m\ell}$ be the (public) unit vector whose only nonzero coordinate is $b_{3m\ell+n_{add}+1} = 1$. Now let

$$A = \left[\begin{array}{c|c} I_{3m\ell \times 3m\ell} & \begin{matrix} -P_x \\ -P_y \\ -P_z \end{matrix} \\ \hline 0_{m\ell \times m\ell} & P_{add} \end{array}\right]$$

The claims are

$$\mathbf{x} \circ \mathbf{y} - \mathbf{z} = \mathbf{0}_{m\ell}, \quad \text{and} \quad A(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})^\top = \mathbf{b} \tag{1}$$

- **Witness encoding**

  The prover constructs U^x, U^y, U^z, U^w encoding $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{w}$. Namely, split $\mathbf{x}$ (resp. $\mathbf{y}$, $\mathbf{z}$, $\mathbf{w}$) into m blocks of length ℓ; for $i = 1, \ldots, m$, if $(x_{i1}, \ldots, x_{i\ell})$ denotes the i-th block, $\mathcal{P}$ chooses a polynomial p_i^x of degree $\leq k-1$ at random such that $p_i^x(\zeta_c) = x_{ic}$ for $c \in [\ell]$ and defines the i-th row of U^x is $(p_i^x(\eta_1), \ldots, p_i^x(\eta_n))$, $i \in [m]$. $\mathcal{P}$ proceeds analogously for U^y, U^z, U^w.

  The prover also selects uniformly random polynomials p_{iblind}, $p^{(0)}_{\mathsf{mblind}}$, $p^{(1)}_{\mathsf{mblind}}$, $p^{(2)}_{\mathsf{mblind}}$, $p^{(0)}_{\mathsf{lblind}}$, $p^{(1)}_{\mathsf{lblind}}$ of degree $\leq k-1$ with the restrictions $p^{(0)}_{\mathsf{mblind}}(\zeta_c) + \zeta_c^e p^{(1)}_{\mathsf{mblind}}(\zeta_c) + \zeta_c^{k-1} p^{(2)}_{\mathsf{mblind}}(\zeta_c) = 0 \ \forall c \in [\ell]$, and $\sum_{c=1}^{\ell} p^{(0)}_{\mathsf{lblind}}(\zeta_c) + \zeta_c^{\ell-1} p^{(1)}_{\mathsf{lblind}}(\zeta_c) = 0$ (and no restriction for p_{iblind}).

  We define $p_{\mathsf{mblind}}(X) = p^{(0)}_{\mathsf{mblind}}(X) + X^e p^{(1)}_{\mathsf{mblind}}(X) + X^{k-1} p^{(2)}_{\mathsf{mblind}}(X)$,and $p_{\mathsf{lblind}}(X) = p^{(0)}_{\mathsf{lblind}}(X) + X^{\ell-1} p^{(1)}_{\mathsf{lblind}}(X)$.
  $\mathcal{P}$ constructs

$$U_{\mathsf{blind}} = \begin{bmatrix} p^{(0)}_{\mathsf{mblind}}(\eta_1) & \cdots & p^{(0)}_{\mathsf{mblind}}(\eta_n) \\ p^{(1)}_{\mathsf{mblind}}(\eta_1) & \cdots & p^{(1)}_{\mathsf{mblind}}(\eta_n) \\ p^{(2)}_{\mathsf{mblind}}(\eta_1) & \cdots & p^{(2)}_{\mathsf{mblind}}(\eta_n) \\ p^{(0)}_{\mathsf{lblind}}(\eta_1) & \cdots & p^{(0)}_{\mathsf{lblind}}(\eta_n) \\ p^{(1)}_{\mathsf{lblind}}(\eta_1) & \cdots & p^{(1)}_{\mathsf{lblind}}(\eta_n) \\ p_{\mathsf{iblind}}(\eta_1) & \cdots & p_{\mathsf{iblind}}(\eta_n) \end{bmatrix}$$

  Finally, $\mathcal{P}$ constructs the matrix $U = \left[U_x | U_y | U_z | U_w | U_{\mathsf{blind}}\right]^\top$

- **First round:** $\mathcal{P}$ **sends** an oracle to the matrix U.

- **Second round:** The verifier $\mathcal{V}$ samples randomness $\mathbf{r}^{mult} \in \mathbb{F}^{m+1}$, $\mathbf{r}^{lin} \in \mathbb{F}^{4m\ell+1}$ according to $\mathsf{Gen}_{\mathtt{rand}}(m+1)$, $\mathsf{Gen}_{\mathtt{rand}}(4m\ell+1)$, **sends** them to the prover.

- **Third round:**

  - (Multiplication test) $\mathcal{P}$ and $\mathcal{V}$ parse $\mathbf{r}^{mult}$ as $(r_1^{mult}, \ldots, r_m^{mult}, 1)$.
    $\mathcal{P}$ computes and **sends** (the $2k-1$ coefficients of) the polynomial

$$p_{\mathsf{mult}}(X) = \sum_{i=1}^{m} r_i^{mult} p_i(X) + p_{\mathsf{mblind}}(X)$$

where $p_i(X) = p_i^x(X)p_i^y(X) - p_i^z(X)$. [9]

– (Linear test) First parse $\mathbf{r}^{lin} \in \mathbb{F}^{4m\ell+1}$ as $(\mathbf{r}_*^{lin}, 1)$ where $\mathbf{r}_*^{lin} \in \mathbb{F}^{4m\ell}$. $\mathcal{P}$ and $\mathcal{V}$ compute $(\mathbf{r}_*^{lin})^\top A \in \mathbb{F}^{4m\ell}$ which is parsed as

$$(\mathbf{r}_*^{lin})^\top A = ((r_j^{x,lin})_{j=1}^{m\ell}, (r_j^{y,lin})_{j=1}^{m\ell}, (r_j^{z,lin})_{j=1}^{m\ell}, (r_j^{w,lin})_{j=1}^{m\ell}).$$

$\mathcal{P}$ and $\mathcal{V}$ then compute, for $i \in [m]$, the unique polynomials $r_i^{x,lin}(X)$ of degree $\leq \ell - 1$ with $r_i^{x,lin}(\zeta_c) = r_{\ell(i-1)+c}^{x,lin}$ for $c \in [\ell]$, and analogously with y, z, w.
$\mathcal{P}$ **sends** (the $k + \ell - 1$ coefficients of) the polynomial

$$p_{\mathsf{lin}}(X) = \sum_{i=1}^{m}(r_i^{x,lin}p_i^x + r_i^{y,lin}p_i^y + r_i^{z,lin}p_i^z + r_i^{w,lin}p_i^w) + p_{\mathsf{lblind}}(X)$$

- **Fourth round:** $\mathcal{V}$ chooses $\alpha \in \mathbb{F}_{\mathrm{OOD}}$ and **sends** α to $\mathcal{P}$.
- **Fifth round** $\mathcal{P}$ **sends**

$$\mathbf{h} = ((h_i^x)_{i=1}^m, (h_i^y)_{i=1}^m, (h_i^z)_{i=1}^m, (h_i^w)_{i=1}^m, (h_{\mathsf{mblind}}^{(b)})_{b=0}^2, (h_{\mathsf{lblind}}^{(b)})_{b=0}^1, h_{\mathsf{iblind}}) \in \mathbb{F}^{4m+6}$$

where respectively h_i^x, h_i^y, h_i^z, h_i^w, $h_{\mathsf{mblind}}^{(0)}$, $h_{\mathsf{mblind}}^{(1)}$, $h_{\mathsf{mblind}}^{(2)}$, $h_{\mathsf{lblind}}^{(0)}$, $h_{\mathsf{lblind}}^{(1)}$, h_{iblind} are the evaluations of p_i^x, p_i^y, p_i^z, p_i^w, $p_{\mathsf{mblind}}^{(0)}$, $p_{\mathsf{mblind}}^{(1)}$, $p_{\mathsf{mblind}}^{(2)}$, $p_{\mathsf{lblind}}^{(0)}$, $p_{\mathsf{lblind}}^{(1)}$, p_{iblind} on α.

- **Sixth round** $\mathcal{V}$ samples randomness $\mathbf{r}^{int} \in \mathbb{F}^{4m+6}$ according to $\mathsf{Gen}_{\mathtt{rand}}(4m+6)$ and **sends** it to the prover.
- **Seventh round** (Interleaved code test) $\mathcal{P}$ and $\mathcal{V}$ parse $\mathbf{r}^{int}$ as

$$((r_{x,i}^{int})_{i=1}^m, (r_{y,i}^{int})_{i=1}^m, (r_{z,i}^{int})_{i=1}^m, (r_{w,i}^{int})_{i=1}^m, (r_{\mathsf{mblind},b}^{int})_{b=0}^2, (r_{\mathsf{lblind},b}^{int})_{b=0}^1, r_{\mathsf{iblind}}^{int})$$

$\mathcal{P}$ computes and **sends** (the k coefficients of) the polynomial

$$v = \sum_{i=1}^{m}(r_{x,i}^{int}\cdot p_i^x + r_{y,i}^{int}\cdot p_i^y + r_{z,i}^{int}\cdot p_i^z + r_{w,i}^{int}\cdot p_i^w) + \sum_{b=0}^{2} r_{\mathsf{mblind},b}^{int}\cdot p_{\mathsf{mblind}}^b + \sum_{b=0}^{1} r_{\mathsf{lblind},b}^{int}\cdot p_{\mathsf{lblind}}^b + r_{\mathsf{iblind}}^{int}\cdot p_{\mathsf{iblind}}$$

- **Query phase** $\mathcal{V}$ selects uniformly at random $Q \subseteq [n]$ of size t, obtaining $U[j]$, $j \in Q$
- **Checks:**

1. Degree checks: $\mathcal{V}$ checks $\deg p_{\mathsf{mult}} \leq 2k-2$, $\deg p_{\mathsf{lin}} \leq k+\ell-2$, $\deg v \leq k-1$
2. Out-of-domain checks:
   (a) $p_{\mathsf{mult}}(\alpha) =^? \sum_{i=1}^m r_i^{mult} h_{mult,i} + h_{\mathsf{mblind}}$,
   where $h_{mult,i} := h_i^x h_i^y - h_i^z$; and $h_{\mathsf{mblind}} := h_{\mathsf{mblind}}^{(0)} + \alpha^e \cdot h_{\mathsf{mblind}}^{(1)} + \alpha^{k-1} \cdot h_{\mathsf{mblind}}^{(2)}$
   (b) $p_{\mathsf{lin}}(\alpha) =^? \sum_{i=1}^m (r_i^{x,lin}(\alpha)h_i^x + r_i^{y,lin}(\alpha)h_i^y + r_i^{z,lin}(\alpha)h_i^z + r_i^{w,lin}(\alpha)h_i^w) + h_{\mathsf{lblind}}$,
   where $h_{\mathsf{lblind}} := h_{\mathsf{lblind}}^{(0)} + h_{\mathsf{lblind}}^{(1)} \cdot \alpha^{\ell-1}$
3. In-domain checks: For each $j \in Q$, $v(\eta_j) =^? (\mathbf{r}^{int})^\top U[j]$
4. In-domain/Out-domain consistency check: $v(\alpha) =^? (\mathbf{r}^{int})^\top \mathbf{h}$
5. Input multiplication condition check: $p_{\mathsf{mult}}(\zeta_c) =^? 0 \quad \forall c \in [\ell]$

[9] Note that here we are testing $\mathbf{x} \circ \mathbf{y} - \mathbf{z} = \mathbf{0}$. This is a special case of the statement in Test-MultOOD where $\mathbf{a} = (-1, -1, \ldots, -1)$ and $\mathbf{b} = \mathbf{0}$, so we can assume the polynomials p_i^a, p_i^b appearing in Test-MultOOD to be the constant polynomials $p_i^a(X) = -1$, $p_i^b(X) = 0$ for all $i \in [m]$

6. Input linear condition check: $\sum_{c=1}^{\ell} p_{\mathsf{lin}}(\zeta_c) =^{?} (\mathbf{r}_*^{lin})^\top \mathbf{b}$

We now state the security properties of this protocol. The proofs are similar to the ones in previous sections with some slight variations in the proof zero-knowledge, due to the fact that in this protocol we are not assuming an oracle to the inputs as an input to the protocol, but this oracle is constructed during its course.

Theorem 26. *Let $U^* \in \mathbb{F}^{(4m+6)\times n}$ be the (possibly malformed) oracle sent by $\mathcal{P}^*$ in the first round, let $\mathsf{Gen}_{\mathsf{rand}}(4m+6)$, B', ε', γ be as in Theorem 22 with $4m+6$ in place of $3m+m'+4$, and let $\Lambda := \Lambda(L^{4m+6}, U^*, \gamma)$. Then the probability that $\mathcal{V}$ accepts and no $U \in \Lambda$ encodes in its first $4m$ rows a solution of (1) (and hence, through $\mathbf{w}$, an input $\boldsymbol{\nu}$ with $C(\boldsymbol{\nu}) = 1$) is at most*

$$\mathsf{Error}_{\mathsf{arith}} := \varepsilon' + (1-\gamma)^t + |\Lambda| \left(\frac{c_{\mathsf{rand}}(m+1) + c_{\mathsf{rand}}(4m\ell+1) + c_{\mathsf{rand}}(4m+6)}{|\mathbb{F}|} + \frac{3k+\ell-4}{|\mathbb{F}| - (n+\ell)} \right).$$

Proof of Theorem 26. The protocol runs Test-ZKMultOOD (with $\mathbf{a} = (-1, \ldots, -1)$, $\mathbf{b} = 0$) and Test-ZKLinOOD (with the A and $\mathbf{b}$ of Equation (1)) with shared α, $\mathbf{h}$ and interleaved test (consisting of the sixth, seventh and the query rounds, the degree check on $\mathbf{v}$ and checks 3 and 4, with $r^{\mathsf{int}} \leftarrow \mathsf{Gen}_{\mathsf{rand}}(4m+6)$). For $U \in L^{4m+6}$ let $p^U_{\mathsf{mult}}, p^U_{\mathsf{lin}}, \mathbf{h}^U, v^U$ be the honest values, defined as in the proofs of Theorems 22 and 25. Then U is *mult-invalid* if $\mathbf{x}^U \circ \mathbf{y}^U - \mathbf{z}^U \neq 0$, *lin-invalid* if $A\tilde{\mathbf{x}}^U \neq \mathbf{b}$, where $\tilde{\mathbf{x}}^U = (\mathbf{x}^U, \mathbf{y}^U, \mathbf{z}^U, \mathbf{w}^U)$, and valid otherwise. The bad events are:

- B_1^{mult} (resp. B_1^{lin}): some mult-invalid (resp. lin-invalid) $U \in \Lambda$ satisfies $p^U_{\mathsf{mult}}(\zeta_c) = 0$ for all $c \in [\ell]$ (resp. $\sum_{c=1}^{\ell} p^U_{\mathsf{lin}}(\zeta_c) = (\mathbf{r}_*^{lin})^\top \mathbf{b}$). As in Theorems 22 and 25 (the latter with $m' = 4m\ell$), these have probability at most $|\Lambda|\, c_{\mathsf{rand}}(m+1)/|\mathbb{F}|$ and $|\Lambda|\, c_{\mathsf{rand}}(4m\ell+1)/|\mathbb{F}|$.
- B_2: the union of the events B_2 of both theorems, of probability at most $|\Lambda|(3k+\ell-4)/(|\mathbb{F}| - (n+\ell))$.
- B_3, B_4: as in both theorems, with $\mathbf{r}^{\mathsf{int}}$; probabilities at most $|\Lambda|\, c_{\mathsf{rand}}(4m+6)/|\mathbb{F}|$ and ε'.

Assume none of them happens and no $U \in \Lambda$ is valid. Case 1 is as in all proofs before. In Case 2, $\mathbf{v}^* = \mathbf{v}^{U_0}$ for some (invalid) U_0 in Λ. Exactly as in the proofs of Theorems 22 and 25, check 4 forces $\mathbf{h}^* = \mathbf{h}^{U_0}$, after which checks 2(a) and 2(b) force $p^*_{\mathsf{mult}} = p^{U_0}_{\mathsf{mult}}$ and $p^*_{\mathsf{lin}} = p^{U_0}_{\mathsf{lin}}$. Since U_0 is mult-invalid or lin-invalid, check 5 or check 6 rejects. ∎

For zero-knowledge, the difference with respect to Section 5.4 is that the oracle to $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, $\mathbf{w}$ is not an input to the protocol but sent by the prover. Hence the simulator does not query it, but samples directly the corresponding evaluations of the witness rows.

Theorem 27. *The protocol above is perfectly honest-verifier zero-knowledge.*

Proof. The view of $\mathcal{V}$ consists of its randomness, p_{mult}, p_{lin}, v, and the evaluations of all $4m+6$ row polynomials on T (as part of $\mathbf{h}$ and $U[Q]$). We call the $4m$ rows of U^x, U^y, U^z, U^w the witness rows. Let

$$S_{\mathrm{mult}} = \sum_{i=1}^{m} r_i^{mult}\big(p_i^x p_i^y - p_i^z\big), \qquad S_{\mathrm{lin}} = \sum_{i=1}^{m} \big(r_i^{x,lin} p_i^x + r_i^{y,lin} p_i^y + r_i^{z,lin} p_i^z + r_i^{w,lin} p_i^w\big).$$

Let V_0, W, Φ be as in Lemma 21, V_0', W', Φ' as in Lemma 24, and let $V'_{\mathbf{b}}$ be the coset of V_0'

$$V'_{\mathbf{b}} := \left\{ q \in \mathbb{F}[X] :\ \deg q \le k+\ell-2,\ \sum_{c=1}^{\ell} q(\zeta_c) = (\mathbf{r}_*^{lin})^\top \mathbf{b} \right\}$$

The simulator samples $\mathbf{r}^{mult} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(m+1)$, $\mathbf{r}^{lin} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(4m\ell+1)$, $\alpha \leftarrow \mathbb{F}_{\mathrm{OOD}}$, $\mathbf{r}^{int} \leftarrow \mathsf{Gen}_{\mathtt{rand}}(4m+6)$ and $Q \subseteq [n]$ with $|Q| = t$, computes the polynomials $r_i^{x,lin}, r_i^{y,lin}, r_i^{z,lin}, r_i^{w,lin}$ as in the protocol, and then:

1. samples the evaluations of the $4m$ witness rows on $T = \{\eta_j : j \in Q\} \cup \{\alpha\}$ uniformly and independently in $\mathbb{F}^{4m(t+1)}$; from here it can compute S_{mult} and S_{lin} on T.

2. samples p_{mult} uniformly in V_0 and p_{lin} uniformly in $V'_{\mathbf{b}}$, independently, and sets $g_{\mathrm{mult}}(\theta) = p_{\mathsf{mult}}(\theta) - S_{\mathrm{mult}}(\theta)$ and $g_{\mathrm{lin}}(\theta) = p_{\mathsf{lin}}(\theta) - S_{\mathrm{lin}}(\theta)$ for $\theta \in T$;

3. samples $(p^{(1)}_{\mathsf{mblind}}(\theta), p^{(2)}_{\mathsf{mblind}}(\theta))_{\theta\in T}$ and $(p^{(1)}_{\mathsf{lblind}}(\theta))_{\theta\in T}$ uniformly and independently, and sets $p^{(0)}_{\mathsf{mblind}}(\theta) = g_{\mathrm{mult}}(\theta) - \theta^e p^{(1)}_{\mathsf{mblind}}(\theta) - \theta^{k-1} p^{(2)}_{\mathsf{mblind}}(\theta)$, and $p^{(0)}_{\mathsf{lblind}}(\theta) = g_{\mathrm{lin}}(\theta) - \theta^{\ell-1} p^{(1)}_{\mathsf{lblind}}(\theta)$;

4. samples v uniformly of degree $\leq k-1$ and sets $p_{\mathsf{iblind}}(\theta) = v(\theta) - \sum_R r_R^{int} p_R(\theta)$ for $\theta \in T$, where R ranges in $[4m+5]$, i.e. all rows R of U except the last one;

5. outputs the values corresponding to $\mathcal{V}$'s view (its randomness, p_{mult}, p_{lin}, v, and the evaluations on all θ in T of all $4m+6$ rows)

We show that the real view has the same distribution, arguing as in items (a)–(d) of the proof of Theorem 20. Since the verifier's randomness has the same distribution in both views, we condition on it.

(a) In the real view, each witness-row polynomial is uniform of degree $\leq k-1$ subject to its values at $\zeta_1, \ldots, \zeta_\ell$, which encode the witness, independently across rows. Since $\ell + t + 1 \leq k$ and $T \cap \{\zeta_1, \ldots, \zeta_\ell\} = \emptyset$, their evaluations on T are uniform in $\mathbb{F}^{4m(t+1)}$, independently of the witness, as in step 1.

(b) In the real view, $p_{\mathsf{mult}} = S_{\mathrm{mult}} + \Phi(\mathbf{f})$ and $p_{\mathsf{lin}} = S_{\mathrm{lin}} + \Phi'(\mathbf{f}')$, where $\mathbf{f} = (p^{(0)}_{\mathsf{mblind}}, p^{(1)}_{\mathsf{mblind}}, p^{(2)}_{\mathsf{mblind}})$ and $\mathbf{f}' = (p^{(0)}_{\mathsf{lblind}}, p^{(1)}_{\mathsf{lblind}})$ are uniform in W and W', independent of each other and of the witness rows. Since the prover is honest, the witness satisfies (1), so $S_{\mathrm{mult}} \in V_0$ and $\sum_c S_{\mathrm{lin}}(\zeta_c) = (\mathbf{r}^{lin}_*)^\top \mathbf{b}$, i.e. $S_{\mathrm{lin}} + V'_0 = V'_{\mathbf{b}}$. By part 1 of Lemmas 21 and 24, p_{mult} and p_{lin} are uniform in V_0 and $V'_{\mathbf{b}}$, independent of each other and of (a), as in step 2.

(c) Conditioned on (a) and (b), $\mathbf{f}$ and $\mathbf{f}'$ are independent and uniform in $W \cap \Phi^{-1}(\{g_{\mathrm{mult}}\})$ and $W' \cap \Phi'^{-1}(\{g_{\mathrm{lin}}\})$, where $g_{\mathrm{mult}} = p_{\mathsf{mult}} - S_{\mathrm{mult}} \in V_0$ and $g_{\mathrm{lin}} = p_{\mathsf{lin}} - S_{\mathrm{lin}} \in V'_0$; their values on T are given by (a), (b) and the verifier's randomness exactly as in step 2. By part 2 of the same lemmas, $(p^{(1)}_{\mathsf{mblind}}(\theta), p^{(2)}_{\mathsf{mblind}}(\theta))_{\theta\in T}$ and $(p^{(1)}_{\mathsf{lblind}}(\theta))_{\theta\in T}$ are uniform and independent, and $p^{(0)}_{\mathsf{mblind}}(\theta)$, $p^{(0)}_{\mathsf{lblind}}(\theta)$ are determined from them and from $g_{\mathrm{mult}}(\theta), g_{\mathrm{lin}}(\theta)$ as in step 3.

(d) In the real view, p_{iblind} is uniform of degree $\leq k-1$, independent of everything in (a)–(c), and has coefficient 1 in v (the last coordinate of $\mathbf{r}^{int}$). Hence v is uniform of degree $\leq k-1$ given (a)–(c), and $p_{\mathsf{iblind}}(\theta)$, $\theta \in T$, is determined by v and the other rows as in step 4.

Hence the real and simulated views are identically distributed.

■

6.1 Round-by-Round (RBR) Knowledge Soundness

We will now analyze the round-by-round soundness error of the protocol above. The concept of round-by-round soundness was introduced in [CCH+19] as a method for evaluating the soundness of a proof system after it is transformed into a non-interactive protocol using the Fiat-Shamir heuristic. The final proof system indeed utilizes the BCS-transformation, which, in turn, relies on the Fiat-Shamir heuristic, to convert the interactive scheme

into a non-interactive protocol. As shown in [BCS16], the soundness of the protocol resulting from the BCS-transformation can be analyzed by identifying the round-by-round knowledge soundness of the IOP. We repeat the definition of round-by-round knowledge from [BGK+23] (which in turn is a variant of [CMS19]).

Definition 28. *A $2r$-round protocol Π for a language L has round-by-round knowledge error $\varepsilon_K(\cdot)$ if there exists a polynomial time extractor E and "doomed set" $\mathcal{D}$ of partial transcripts of interaction using Π such that the following hold:*

1. *For all x, $(x, \emptyset) \in \mathcal{D}$, where $\emptyset$ denotes the empty transcript.*
2. *For any input x and complete transcript τ, if $(x, \tau) \in \mathcal{D}$ then $V(x, \tau) = \mathsf{reject}$.*
3. *If for $i \in [r]$, (x, τ) is a $2(i-1)$-round partial transcript such that $(x, \tau) \in \mathcal{D}$, then for every possible prover message $m_{\mathcal{P}}$ if*

$$\Pr[m_{\mathcal{V}} \leftarrow V(x, (\tau, m_{\mathcal{P}})) : (x, (\tau, m_{\mathcal{P}}, m_{\mathcal{V}})) \notin \mathcal{D}] > \varepsilon_K(x),$$

then $E(x, \tau, m_{\mathcal{P}})$ outputs a valid witness for x.

In our case r=4 (we have 8 rounds). Towards proving round-by-round soundness, let us first fix some instance x, consisting of the circuit C together with the public parameters of the protocol of Section 6. We keep the assumptions and notations of Theorem 26. For an invalid $U \in \Lambda$, we say that U is caught at $\boldsymbol{\zeta}$ if either U is mult-invalid and $p^U_{\mathsf{mult}}(\zeta_c) \neq 0$ for some $c \in [\ell]$, or U is lin-invalid and $\sum_{c=1}^{\ell} p^U_{\mathsf{lin}}(\zeta_c) \neq (\mathbf{r}^{lin}_*)^\top \mathbf{b}$. In order to prove the round-by-round soundness, we first need to define the extractor E and the doomed set $\mathcal{D}$.

The extractor E is defined as follows: On input $(x, \tau, m_{\mathcal{P}})$, where τ or $m_{\mathcal{P}}$ contains the oracle U^* (sent in the first round), E computes Λ by list-decoding U^*, decodes every $U \in \Lambda$, and outputs a witness $\boldsymbol{\nu}$ using $\mathbf{w}^U$ for some valid $U \in \Lambda$, if there is one; if Λ contains no valid U, it fails. Since the first $4m$ rows of a valid U encode a solution of (1), the corresponding $\boldsymbol{\nu}$ satisfies $C(\boldsymbol{\nu}) = 1$. E runs in polynomial time: within the Johnson bound $\gamma < 1 - \sqrt{\rho}$ by the Guruswami-Sudan algorithm [GS98], and, in the regime $\gamma < 1 - a_1(\rho)$ of Lemma 12, by the first-order decoder of [DKT26]. In particular, $E(x, \tau, m_{\mathcal{P}})$ succeeds if and only if the list Λ of the oracle U^* contains a valid U.

The doomed set $\mathcal{D}$ consists of the following pairs (x, τ), according to the number of rounds of τ:

Initially: $(x, \emptyset) \in \mathcal{D}$.

After the 2nd round ($\tau_2 = ([U^*], \mathbf{r}^{mult}, \mathbf{r}^{lin})$): $(x, \tau_2) \in \mathcal{D}$ iff no $U \in \Lambda$ is valid and every $U \in \Lambda$ is caught at $\boldsymbol{\zeta}$.

After the 4th round ($\tau_4 = (\tau_2, p^*_{\mathsf{mult}}, p^*_{\mathsf{lin}}, \alpha)$): $(x, \tau_4) \in \mathcal{D}$ iff no $U \in \Lambda$ is valid and either $p^*_{\mathsf{mult}}, p^*_{\mathsf{lin}}$ fail check 1, 5 or 6, or $(x, \tau_2) \in \mathcal{D}$ and every $U \in \Lambda$ satisfies $p^U_{\mathsf{mult}}(\alpha) \neq p^*_{\mathsf{mult}}(\alpha)$ or $p^U_{\mathsf{lin}}(\alpha) \neq p^*_{\mathsf{lin}}(\alpha)$.

After the 6th round ($\tau_6 = (\tau_4, \mathbf{h}^*, \mathbf{r}^{int})$): $(x, \tau_6) \in \mathcal{D}$ iff no $U \in \Lambda$ is valid and either the messages sent so far fail check 1, 2, 5 or 6, or $(x, \tau_4) \in \mathcal{D}$, every $U \in \Lambda$ satisfies $(\mathbf{r}^{int})^\top \mathbf{h}^U \neq (\mathbf{r}^{int})^\top \mathbf{h}^*$, and $\Lambda(L, (\mathbf{r}^{int})^\top U^*, \gamma) = \{(\mathbf{r}^{int})^\top U : U \in \Lambda\}$.

After the 8th round ($\tau_8 = (\tau_6, v^*, Q)$): $(x, \tau_8) \in \mathcal{D}$ iff $V(x, \tau_8) = \mathsf{reject}$.

Intuitively, a partial transcript is doomed if the oracle U^* does not allow to extract a witness, and the verifier's randomness so far has not produced any of the bad events in the proof of Theorem 26. Therefore, a transcript can leave $\mathcal{D}$ in two ways: at the first verifier

message, because Λ contains a valid U, in which case the extractor succeeds; or, at any verifier message, because one of these bad events happens. Next, we show that the latter has small probability.

Theorem 29. *Under the assumptions of Theorem 26 and the extractor E and the doomed set $\mathcal{D}$ defined above, the protocol of Section 6 has round-by-round knowledge error*

$$\varepsilon_K = \max\Bigg\{ |\Lambda|\frac{\max\{c_{\mathtt{rand}}(m+1), c_{\mathtt{rand}}(4m\ell+1)\}}{|\mathbb{F}|}, \qquad |\Lambda|\frac{2k-2}{|\mathbb{F}|-(n+\ell)}, \\ \varepsilon' + |\Lambda|\frac{c_{\mathtt{rand}}(4m+6)}{|\mathbb{F}|}, \qquad (1-\gamma)^t \Bigg\}.$$

Proof. Conditions 1 and 2 in Definition 28 hold by our definition of $\mathcal{D}$.

We verify Condition 3 for $i = 1, 2, 3, 4$. Let $(x, \tau) \in \mathcal{D}$ be a $2(i-1)$-round partial transcript and $m_{\mathcal{P}}$ a prover message. We show that either $E(x, \tau, m_{\mathcal{P}})$ outputs a valid witness, or $\Pr_{m_{\mathcal{V}}}[(x, (\tau, m_{\mathcal{P}}, m_{\mathcal{V}})) \notin \mathcal{D}] \leq \varepsilon_K$.

$i = 1$ ($m_{\mathcal{P}} = [U^*]$, $m_{\mathcal{V}} = (\mathbf{r}^{mult}, \mathbf{r}^{lin})$). If Λ contains a valid U, then $E(x, \emptyset, [U^*])$ outputs a valid witness. Otherwise, $(x, \tau_2) \notin \mathcal{D}$ if and only if some, necessarily invalid, $U \in \Lambda$ is not caught at $\boldsymbol{\zeta}$. For a fixed mult-invalid U, this happens with probability at most $c_{\mathtt{rand}}(m+1)/|\mathbb{F}|$, and for a fixed lin-invalid U with probability at most $c_{\mathtt{rand}}(4m\ell+1)/|\mathbb{F}|$, by the argument for B_1^{mult} and B_1^{lin} resp. in the proof of Theorem 26.

By the union bound, $\Pr_{(\mathbf{r}^{mult}, \mathbf{r}^{lin})}[(x, \tau_2) \notin \mathcal{D}] \leq |\Lambda| \max\{c_{\mathtt{rand}}(m+1), c_{\mathtt{rand}}(4m\ell + 1)\}/|\mathbb{F}| \leq \varepsilon_K$.

For $i \geq 2$, note that $(x, \tau) \in \mathcal{D}$ implies that no $U \in \Lambda$ is valid, so E fails on $(x, \tau, m_{\mathcal{P}})$ for every $m_{\mathcal{P}}$ and therefore what we need to prove is that the probability of leaving $\mathcal{D}$ is upper-bounded by ε_K. Moreover, the property "no $U \in \Lambda$ is valid" only depends on U^*, so it cannot change after round 1, so a transcript can leave $\mathcal{D}$ only through the other conditions defining $\mathcal{D}$.

$i = 2$ ($m_{\mathcal{P}} = (p^*_{\mathsf{mult}}, p^*_{\mathsf{lin}})$, $m_{\mathcal{V}} = \alpha$). If $p^*_{\mathsf{mult}}, p^*_{\mathsf{lin}}$ fail check 1, 5 or 6, then $(x, \tau_4) \in \mathcal{D}$ for every α. Otherwise, they pass checks 5 and 6, while every $U \in \Lambda$ is caught at $\boldsymbol{\zeta}$ since $(x, \tau_2) \in \mathcal{D}$; hence every $U \in \Lambda$ satisfies $p^U_{\mathsf{mult}} \neq p^*_{\mathsf{mult}}$ or $p^U_{\mathsf{lin}} \neq p^*_{\mathsf{lin}}$. These polynomials are fixed before α is sampled and have degree at most $2k-2$, so for each fixed U the probability that both evaluations at α coincide is at most $\frac{2k-2}{|\mathbb{F}_{\mathrm{OOD}}|}$.

Applying the union bound, $\Pr_\alpha[(x, \tau_4) \notin \mathcal{D}] \leq |\Lambda| \frac{2k-2}{|\mathbb{F}|-(n+\ell)}$.

$i = 3$ ($m_{\mathcal{P}} = \mathbf{h}^*$, $m_{\mathcal{V}} = \mathbf{r}^{int}$). If check 2 fails, or if (x, τ_4) is doomed because check 1, 5 or 6 fails, then $(x, \tau_6) \in \mathcal{D}$ for every $\mathbf{r}^{int}$. Otherwise check 2 passes, and every $U \in \Lambda$ satisfies $p^U_{\mathsf{mult}}(\alpha) \neq p^*_{\mathsf{mult}}(\alpha)$ or $p^U_{\mathsf{lin}}(\alpha) \neq p^*_{\mathsf{lin}}(\alpha)$. If $\mathbf{h}^* = \mathbf{h}^U$ for some $U \in \Lambda$, the right-hand sides of checks 2(a) and 2(b) would equal $p^U_{\mathsf{mult}}(\alpha)$ and $p^U_{\mathsf{lin}}(\alpha)$ respectively, as computed in the proof of Theorem 26, contradicting that check 2 passes. Hence $\mathbf{h}^U \neq \mathbf{h}^*$ for every $U \in \Lambda$. By Lemma 10 and a union bound, some $U \in \Lambda$ satisfies $(\mathbf{r}^{int})^\top \mathbf{h}^U = (\mathbf{r}^{int})^\top \mathbf{h}^*$ with probability at most $|\Lambda| c_{\mathtt{rand}}(4m+6)/|\mathbb{F}|$, and by Lemma 7 the equality of lists fails with probability at most ε'. Hence $\Pr_{\mathbf{r}^{int}}[(x, \tau_6) \notin \mathcal{D}] \leq \varepsilon' + |\Lambda| c_{\mathtt{rand}}(4m+6)/|\mathbb{F}|$.

$i = 4$ ($m_{\mathcal{P}} = v^*$, $m_{\mathcal{V}} = Q$). If (x, τ_6) is doomed because a check among 1, 2, 5, 6 fails, then $\mathcal{V}$ rejects for every Q, so $(x, \tau_8) \in \mathcal{D}$. Otherwise, every $U \in \Lambda$ satisfies $(\mathbf{r}^{int})^\top \mathbf{h}^U \neq (\mathbf{r}^{int})^\top \mathbf{h}^*$, and $\Lambda(L, (\mathbf{r}^{int})^\top U^*, \gamma) = \{(\mathbf{r}^{int})^\top U : U \in \Lambda\}$. If check 4 passes, then $v^*(\alpha) = (\mathbf{r}^{int})^\top \mathbf{h}^* \neq (\mathbf{r}^{int})^\top \mathbf{h}^U = v^U(\alpha)$, so $\mathbf{v}^* \neq \mathbf{v}^U = (\mathbf{r}^{int})^\top U$ for every $U \in \Lambda$. By the equality of lists, $\mathbf{v}^* \notin \Lambda(L, (\mathbf{r}^{int})^\top U^*, \gamma)$, i.e. $d(\mathbf{v}^*, (\mathbf{r}^{int})^\top U^*) > \gamma n$, so check 3 passes with probability at most $\binom{n-\gamma n-1}{t}/\binom{n}{t} \leq (1-\gamma)^t$ over Q. Hence $\Pr_Q[(x, \tau_8) \notin \mathcal{D}] = \Pr_Q[V(x, \tau_8) = \mathsf{accept}] \leq (1-\gamma)^t$.

We have proved that either $E(x, \tau, m_{\mathcal{P}})$ outputs a valid witness or the probability of leaving $\mathcal{D}$ at each interaction (i.e. at every two rounds) is at most ε_K, which proves condition 3. ∎

6.2 Non-interactive protocol via BCS compiler.

Once we have determined the round-by-round soundness of the IOP, we can apply the BCS transformation [BCS16] to the protocol to turn it into a non-interactive proof of knowledge. In the BCS transformation we replace the oracle to the matrix U with a Merkle-commitment, where each of the leaves of the Merkle tree is a column of the matrix U. The queries to the columns of U are answered by decommitting the corresponding leaf of the Merkle tree. Finally, the random challenges from the receiver are computed using the Fiat-Shamir heuristic, namely via a random oracle (in practice, a hash function) applied to the string formed by the public input and transcript so far. Let λ be the output size of this random oracle.

Applying the arguments from [Goo25], we obtain the following.

Theorem 30. *Let ε_K be the round-by-round knowledge error from Theorem 29. The scheme after applying the BCS transformation with a random oracle that outputs λ bits has an adaptive knowledge error against T-size adversaries which is at most $T \cdot \varepsilon_K + \frac{9}{2} \cdot \frac{T^2}{2^\lambda}$.*

6.3 A variant using extension fields

We consider a generalization of the protocol we have just presented, where the circuit and the witness are defined over a finite field $\mathbb{F}$, but now we will be using an extension field $\mathbb{K}$ for some elements of the protocol. Namely, $\mathbb{F}$ is a subfield of $\mathbb{K}$ with $|\mathbb{K}| = |\mathbb{F}|^r$, for some $r \geq 1$ (the previous protocol will be the case $r = 1$). The goal of this generalization is to be able to use it in settings where $\mathbb{F}$ can be taken relatively small and still have good soundness guarantees due to the use of $\mathbb{K}$, but without needing to embed the whole proof in $\mathbb{K}$ so that we can have some gains in proof size. We will proceed as follows:

- We will need to set $k \geq \ell + t + r$, where r is the degree of the extension.
- The evaluation points ζ_c, η_j will be in the smaller field $\mathbb{F}$. Consider the Reed-Solomon codes $L = \mathsf{RS}_{\mathbb{F},n,k,\boldsymbol{\eta}}$ and $\overline{L} = \mathsf{RS}_{\mathbb{K},n,k,\boldsymbol{\eta}}$. Clearly $L \subseteq \overline{L}$.
- For the sake of notation let $U_0 = [U^x|U^y|U^z|U^w]^\top$ the part of U encoding inputs (recall the committed U is U_0 stacked with U_{blind}). We will use the code L to encode the witness, thereby creating $U_0 \in L^{4m}$. For reasons related to soundness we will need to force that a possibly malicious prover can only commit to U_0^* with entries in $\mathbb{F}$, namely $U_0^* \in \mathbb{F}^{4m \times n}$ (while we need to allow $U_{\mathsf{blind}}^* \in \mathbb{K}^{6 \times n}$ as we see below). In the non-interactive version this can be done by having every leaf of the Merkle tree encode, and be parsed as, an element of $\mathbb{F}^{4m} \times \mathbb{K}^6$
- The $\mathsf{Gen}_{\mathtt{rand}}$ from which $\mathcal{V}$ samples randomness are over the larger field $\mathbb{K}$.
- The out-of-domain point α is sampled from $\mathbb{K}_{OOD}$ (defined as $\mathbb{K}$ excluding the points ζ_c, η_j). Consequently $\mathbf{h}$ is defined in $\mathbb{K}$
- The blinding polynomials $p^{(b)}_{\mathsf{mblind}}$, $b = 0, 1, 2$, $p^{(b)}_{\mathsf{lblind}}$, $b = 0, 1$ and p_{iblind} are all taken in $\mathbb{K}[X]$ and consequently define codewords in the larger code $\overline{L}$.
- Note the sent polynomials p_{mult}, p_{lin}, v are all in $\mathbb{K}[X]$, both because the blinding polynomials are in $\mathbb{K}[X]$ and because the randomness sent by the verifier is over $\mathbb{K}$

Let us consider the security properties of this protocol:

Soundness Using exactly the proof of Theorem 26 with $\overline{L}$ and $\mathbb{K}$ in place of L and $\mathbb{F}$, we would already obtain a guarantee that there is a witness $\boldsymbol{\nu}$ such that $C(\boldsymbol{\nu}) = 1$, with the soundness error where we replace $|\mathbb{F}|$ by $|\mathbb{K}|$. However, in principle this ν might be defined over $\mathbb{K}$, so we need an additional argument to guarantee that the extracted witness is defined over $\mathbb{F}$.

Lemma 31. *Given $\overline{L}$ defined as above ($\overline{L} = \mathsf{RS}_{\mathbb{K},n,k,\boldsymbol{\eta}}$ with all η_j in $\mathbb{F}$), if a codeword $\mathbf{v} \in \overline{L}$ has at least k coordinates defined in $\mathbb{F}$, then $\mathbf{v}$ is in the subcode L.*

Proof. The proof is very simple: by the usual interpolation argument, given any set $\{j_1, \dots, j_k\} \subseteq [n]$ of k coordinates, for every possible vector $\mathbf{z} \in \mathbb{F}^k$ there exists a polynomial $f \in \mathbb{F}[X]$ of degree $k-1$ with $f(\eta_{j_i}) = z_i$, so there is a (unique) codeword $\mathbf{v}'$ in L matching this vector. On the other hand, no two codewords in $\overline{L}$ can coincide in k positions or more, so this means any codeword of $\overline{L}$ with k coordinates in $\mathbb{F}$ must be in L. ∎

Since we are always considering $\gamma \leq 1 - \rho$, and hence $(1-\gamma)n \geq k$, this implies the following: given any $U^* \in \mathbb{F}^{4m \times n} \times \mathbb{K}^{6 \times n}$ committed by the prover (recall we are forcing the prover to commit to values from $\mathbb{F}$ in the input encoding part of the matrix), if $U \in \overline{L}^{4m+6}$ is at distance $\leq \gamma n$ of U^*, then in particular $(1-\gamma)n \geq k$ of the columns in U_0 (its input-encoding part) must be defined in $\mathbb{F}$, and therefore $U_0 \in L^{4m}$ by the lemma. Plugging this in the soundness proof from Theorem 26 applied to this variant, we get that whenever we can extract an input, this must be defined over $\mathbb{F}$.

The conclusion is that the soundness in Theorem 26 applies to this variant replacing $|\mathbb{F}|$ by $|\mathbb{K}|$. The round-by-round soundness analysis from Section 6.1 holds in the same way.

Zero Knowledge The main difference in the zero-knowledge statement is that we replace the condition $k \geq \ell + t + 1$ by the more general $k \geq \ell + t + r$. We use the following interpolation result, which can for example be derived from [CDN15, Theorem 11.96]

Lemma 32. *Let $\mathbb{F}$ be a field and $\mathbb{K}$ be a extension of degree r. Let $Z \subseteq \mathbb{F}$ of size $\ell + t$, and let $\alpha \in \mathbb{K} \setminus Z$. Let $\mathbb{K}'$ be the smallest extension $\mathbb{F} \subseteq \mathbb{K}' \subseteq \mathbb{K}$ to which α belongs, i.e. $\mathbb{K}' = \mathbb{F}(\alpha)$. Let $\mathbb{F}_{\leq k-1}[X]$ the set of polynomials of degree $\leq k-1$. If $k \geq \ell + t + r$, $\mathcal{E} : \mathbb{F}_{\leq k-1}[X] \to \mathbb{F}^{\ell+t} \times \mathbb{K}'$ given by $f \mapsto ((f(z))_{z \in Z}, f(\alpha))$ is surjective.*[10]

Intuitively, the proposition guarantees that the evaluations of a polynomial in $\mathbb{F}_{\leq k-1}[X]$ at the points $\eta_j, j \in Q$ and α give no information about the evaluations at the points ζ_c, which we clearly need because we will reveal the input encoding polynomials at those points. Moreover, given any prescribed evaluations at $\zeta_1, \dots, \zeta_\ell$, the distribution of $((f(\eta_j))_{j \in Q}, f(\alpha))$ conditioned to those evaluations is uniform in $\mathbb{F}^t \times \mathbb{K}'$, which we will use in the simulation. Now we can prove Theorem 33

Theorem 33. *Suppose $k \geq \ell + t + r$, $k \geq 2t + 1$ and $t \leq e \leq k - t - 1$. Then the protocol of this section is perfectly honest-verifier zero-knowledge.*

Proof. The proof is the same as that of Theorem 27, with Lemmas 21 and 24 taken over $\mathbb{K}$ (both hold over any field), and with p_{mult}, p_{lin}, v and the blinding evaluations sampled over $\mathbb{K}$. The only change is in step 1 of the simulator and in item (a): by Lemma 32, since $k \geq \ell + t + r$, in the real view the evaluations of each witness row at η_j, $j \in Q$, and at α are uniform in $\mathbb{F}^t \times \mathbb{F}(\alpha)$, independently across rows and of the witness, so the simulator samples them from this distribution. ∎

[10] The theorem from [CDN15] would in particular guarantee that if $k = \ell + t + r'$, where r' is the degree of $\mathbb{K}'$ over $\mathbb{F}$ (the degree of the minimal polynomial of α over $\mathbb{F}$) then the map is an isomorphism

7 Proof Sizes

We consider the proof size of the non-interactive version of the combined protocol to prove circuit satisfiability, described in detail in Section 6. The considered circuit has N arithmetic gates over a field of size $|\mathbb{F}|$ and we need to set m, ℓ with $N \leq m\ell$. The prover sends $4k + 4m + \ell + 4$ elements from $\mathbb{F}$ as part of the elements p_{mult}, p_{lin}, v, $\mathbf{h}$ that are communicated directly in the IOP. Moreover, the prover sends t queried columns of U, which amounts to $(4m + 6)t$ elements of $\mathbb{F}$, and the corresponding Merkle opening proofs, which amount to (at most) $t(\log n)$ hash outputs. We set $k = \ell + t + 1$, which is minimal for achieving zero-knowledge, and assume the hash domain is $\{0,1\}^\lambda$. We can write the proof size as $((4m + 10)(t + 1) + 5\ell - 2)\log|\mathbb{F}| + t\lambda \log n$ bits.

The choice for m and ℓ in terms of N that minimizes the proof length is $m \approx \sqrt{\frac{5}{4(t+1)}N}$, $\ell \approx \sqrt{\frac{4(t+1)}{5}N}$, which leads to $\left(4\sqrt{5(t+1)} \cdot N^{1/2} + (10t + 8)\right)\log|\mathbb{F}| + t\lambda \log n$ bits.

In comparison the original Ligero (without the out-of-domain strategy and with $k = \ell + t$) would communicate a total of $((4m + 10)t + 5\ell - 2)\log|\mathbb{F}| + t\lambda \log n$ bits, which represents $(4m + 10)\log|\mathbb{F}|$ fewer bits. In this case we set $m \approx \sqrt{\frac{5}{4t}N}$, $\ell \approx \sqrt{\frac{4t}{5}N}$ and obtain $\left(4\sqrt{5t} \cdot N^{1/2} + (10t - 2)\right)\log|\mathbb{F}| + t\lambda \log n$ bits in terms of N. As discussed before, the increased proof size is in exchange for a better soundness for the same rate ρ and query complexity t.

Table 1 in Appendix A fixes $N = 2^{20}$ and a 256-bit field and varies the rate ρ and target κ (with $\lambda = 2\kappa$). It compares original Ligero under unique-decoding and Johnson analyses with OOD Ligero under the Johnson analysis. The tables use the leading query-error approximation and the size accounting specified in the appendix; they do not certify total compiled security.

Sizes for the variant using extension fields For the variant of Section 6.3, where the witness is in a small field $\mathbb{F}$ but the randomness sent by the tester (and other elements like the masking polynomials) is over a larger field $\mathbb{K}$ the proof size is $(4mt + (4m + 5\ell + 10t + 4r + 4)r)\log|\mathbb{F}| + t\lambda \log n$ bits, gaining around $4mt(r - 1)\log|\mathbb{F}|$ bits with respect to the version where we embed the entire protocol in $\mathbb{K}$, with the same soundness error.

Setting $m \approx \sqrt{\frac{5rN}{4(t+r)}}$, $\ell \approx N/m \approx \sqrt{\frac{4(t+r)N}{5r}}$, we can express this proof size as approximately $(4\sqrt{5(t + r)r}N^{1/2} + (10t + 4r + 4)r)\log|\mathbb{F}| + t\lambda \log n$ bits. Assuming $r \ll t$, the first term is then around $\sqrt{r}$ times lower than if we run the proof entirely over $\mathbb{K}$.

Table 2 fixes $\rho = 1/32$ and gives OOD proof-size estimates for $N = 2^{16}, \ldots, 2^{22}$ and 32-, 64-, and 256-bit base fields, with 256-bit challenge storage throughout ($r = 8, 4, 1$, respectively).

7.1 Observations

Table 1 separates the benefit of a stronger soundness analysis from that of modifying the protocol. At $\rho = 1/32$ and the 80-bit query-error target, the Johnson analysis reduces the query count of original Ligero from 84 to 32 and the estimated proof size from 2.743 to 1.679 MB, a reduction of approximately 39%, without changing the protocol. The corresponding 100-bit figures are 105 versus 40 queries and 3.085 versus 1.884 MB. At these low rates, OOD uses the same query count and incurs a small additional communication cost (1.705 and 1.907 MB, respectively).

The benefit of OOD appears at higher rates. At $\rho = 1/3$, the unique-decoding and Johnson bounds for original Ligero coincide, whereas OOD reduces the 80-bit query count from 137 to 101 and the size from 3.515 to 3.023 MB (approximately 14%). Moreover, OOD admits rates 1/2 and 5/8, where the displayed bounds for original Ligero are vacuous.

Higher rates trade more queries and larger proofs for reduced code expansion; these estimates do not measure prover running time.

Table 2 illustrates the approximately $\sqrt{N}$ growth of proof size at a fixed rate and target, together with the savings from small-field witnesses. At $N = 2^{20}$ and the 80-bit target, the 32-, 64-, and 256-bit base-field estimates are 0.677, 0.901, and 1.705 MB; at 100 bits they are 0.748, 1.002, and 1.907 MB. Thus 32-bit witnesses reduce the estimate by approximately 60% relative to 256-bit witnesses, while keeping challenge storage at 256 bits. Only witness-column entries become smaller: masks, aggregate responses, and OOD values retain the larger-field cost. All figures use no grinding and the appendix's leading query-error approximation, with packing chosen separately for each configuration.

References

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: reed-solomon proximity testing with fewer queries. In *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, pages 380–413, 2024.

[ACFY25] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: reed-solomon proximity testing with super-fast verification. In *EUROCRYPT*, pages 214–243, 2025.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *CCS*, pages 2087–2104, 2017.

[AHIV23] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: lightweight sublinear arguments without a trusted setup (full version of [AHIV17]). *Des. Codes Cryptogr.*, 91(11):3379–3424, 2023. URL: https://doi.org/10.1007/s10623-023-01222-8.

[AST24] Arasu Arun, Srinath T. V. Setty, and Justin Thaler. Jolt: Snarks for virtual machines via lookups. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 3–33. Springer, 2024.

[Azt26] Aztec Protocol. Barretenberg. https://github.com/AZTECProtocol/barretenberg, 2026.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In *ICALP*, pages 14:1–14:17, 2018.

[BBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO*, pages 701–732, 2019.

[BC12] Nir Bitansky and Alessandro Chiesa. Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In *CRYPTO*, pages 255–272, 2012.

[BC25a] Dan Boneh and Binyi Chen. Latticefold: A lattice-based folding scheme and its applications to succinct proof systems. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part III*, Lecture Notes in Computer Science, pages 330–362. Springer, 2025.

[BC25b] Dan Boneh and Binyi Chen. Latticefold+: Faster, simpler, shorter lattice-based folding for succinct proof systems. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VII*, Lecture Notes in Computer Science, pages 327–361. Springer, 2025.

[BCF+17] Eli Ben-Sasson, Alessandro Chiesa, Michael A. Forbes, Ariel Gabizon, Michael Riabzev, and Nicholas Spooner. Zero knowledge protocols from succinct constraint detection. In *Theory of Cryptography - 15th International Conference, TCC 2017, Baltimore, MD, USA, November 12-15, 2017, Proceedings, Part II*, pages 172–206, 2017.

[BCF+25] Anubhav Baweja, Alessandro Chiesa, Elisabetta Fedele, Giacomo Fenzi, Pratyush Mishra, Tushar Mopuri, and Andrew Zitek-Estrada. Time-space trade-offs for sumcheck. In Benny Applebaum and Huijia (Rachel) Lin, editors, *Theory of Cryptography - 23rd International Conference, TCC 2025, Aarhus, Denmark, December 1-5, 2025, Proceedings, Part IV*, Lecture Notes in Computer Science, pages 37–70. Springer, 2025.

[BCGM25] Sarah Bordage, Alessandro Chiesa, Ziyi Guan, and Ignacio Manzur. All polynomial generators preserve distance with mutual correlated agreement. *IACR Cryptol. ePrint Arch.*, page 2051, 2025.

[BCH+25] Eli Ben-Sasson, Dan Carmon, Ulrich Haböck, Swastik Kopparty, and Shubhangi Saraf. On proximity gaps for reed–solomon codes. *IACR Cryptol. ePrint Arch.*, page 2055, 2025.

[BCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed-solomon codes. In *FOCS*, 2020.

[BCL22] Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge iops with linear-time prover and polylogarithmic-time verifier. In *EUROCRYPT*, pages 275–304, 2022.

[BCMS20] Benedikt Bünz, Alessandro Chiesa, Pratyush Mishra, and Nicholas Spooner. Recursive proof composition from accumulation schemes. In *TCC*, pages 1–18, 2020.

[BCP+26] Joshua Brakensiek, Yeyuan Chen, Aaron Putterman, Zihan Zhang, and Kai Zhe Zheng. Algorithmic list decoding of reed-solomon codes up to capacity, 2026. URL: https://arxiv.org/abs/2609.08005, arXiv:2609.08005.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT*, pages 103–128, 2019.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC*, pages 31–60, 2016.

[BGH19] Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. *IACR Cryptol. ePrint Arch.*, page 1021, 2019.

[BGK+23] Alexander R. Block, Albert Garreta, Jonathan Katz, Justin Thaler, Pratyush Ranjan Tiwari, and Michal Zajac. Fiat-shamir security of FRI and related snarks. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT*, pages 3–40, 2023.

[BGKS20] Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: sampling outside the box improves soundness. In *ITCS*, pages 5:1–5:32, 2020.

[BHR+20] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Public-coin zero-knowledge arguments with (almost) minimal time and space overheads. In *TCC*, pages 168–197, 2020.

[BHR+21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Time- and space-efficient arguments from groups of unknown order. In *CRYPTO*, pages 123–152, 2021.

[BS21] Eli Ben-Sasson. ethSTARK documentation. Cryptology ePrint Archive, Paper 2021/582, 2021. URL: https://eprint.iacr.org/2021/582.

[BSU24] Alexandre Belling, Azam Soleimanian, and Bogdan Ursu. Vortex: A list polynomial commitment and its application to arguments of knowledge. Cryptology ePrint Archive, Paper 2024/185, 2024. URL: https://eprint.iacr.org/2024/185.

[CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. Hyperplonk: Plonk with linear-time prover and high-degree custom gates. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part II*, Lecture Notes in Computer Science, pages 499–530. Springer, 2023.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-shamir: from practice to theory. In *STOC*, pages 1082–1090, 2019.

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. URL: http://www.cambridge.org/de/academic/subjects/computer-science/cryptography-cryptology-and-coding/secure-multiparty-computation-and-secret-sharing?format=HB&isbn=9781107043053.

[CFW26] Alessandro Chiesa, Giacomo Fenzi, and Guy Weissenberg. Zero-knowledge iopps for constrained interleaved codes. *IACR Cryptol. ePrint Arch.*, 2026:391, 2026. URL: https://eprint.iacr.org/2026/391.

[CHK25] Soham Chatterjee, Prahladh Harsha, and Mrinal Kumar. Deterministic list decoding of reed-solomon codes. *CoRR*, abs/2511.05176, 2025.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In *TCC*, pages 1–29, 2019.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *EUROCRYPT*, pages 769–793, 2020.

[CS25] Elizabeth Crites and Alistair Stewart. On reed–solomon proximity gaps conjectures. *IACR Cryptol. ePrint Arch.*, page 2046, 2025.

[DG25] Benjamin E. Diamond and Angus Gruen. On the distribution of the distances of random words. *IACR Cryptol. ePrint Arch.*, page 2010, 2025.

[DHRR26] Rahul Dalal, Tamir Hemo, Eugene Rabinovich, and Ron Rothblum. VEIL: Lightweight zero-knowledge for hash-based multilinear proof systems. Cryptology ePrint Archive, Paper 2026/683, 2026. URL: https://eprint.iacr.org/2026/683.

[DI06] Ivan Damgård and Yuval Ishai. Scalable secure multiparty computation. In *CRYPTO*, pages 501–520, 2006.

[DKT26] Quang Dao, Scott Duke Kominers, and Justin Thaler. Reed-solomon codes beyond johnson: Efficient decoding and smaller cryptographic proofs. 2026. URL: https://eprint.iacr.org/2026/2056.

[EFKP20] Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. Sparks: Succinct parallelizable arguments of knowledge. In *EUROCRYPT*, pages 707–737, 2020.

[FS24] Matteo Frigo and Abhi Shelat. Anonymous credentials from ECDSA. *IACR Cryptol. ePrint Arch.*, page 2010, 2024.

[FS25] Giacomo Fenzi and Antonio Sanso. Small-field hash-based snargs are less sound than conjectured. *IACR Cryptol. ePrint Arch.*, page 2197, 2025.

[GG25] Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) reed-solomon codes. *IACR Cryptol. ePrint Arch.*, page 2054, 2025.

[GKL24] Yiwen Gao, Haibin Kan, and Yuan Li. Linear proximity gap for linear codes within the 1.5 johnson bound. *IACR Cryptol. ePrint Arch.*, 2024:1810, 2024. URL: https://eprint.iacr.org/2024/1810.

[GLS+23a] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*, Lecture Notes in Computer Science, pages 193–226. Springer, 2023.

[GLS+23b] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In *CRYPTO*, pages 193–226, 2023.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *STOC*, pages 291–304, 1985.

[Goo25] Google. Longfellow security review. https://github.com/google/longfellow-zk/blob/main/docs/static/reviews/Longfellow_security_2025_12_15.pdf, 2025.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, Lecture Notes in Computer Science, pages 305–326. Springer, 2016.

[GS98] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *FOCS*, pages 28–39, 1998.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptol. ePrint Arch.*, 2019:953, 2019.

[Hab25] Ulrich Haböck. A note on mutual correlated agreement for reed-solomon codes. *IACR Cryptol. ePrint Arch.*, 2025:2110, 2025. URL: https://eprint.iacr.org/2025/2110.

[HK24] Ulrich Haböck and Al Kindi. A note on adding zero-knowledge to STARKs. Cryptology ePrint Archive, Paper 2024/1037, 2024. URL: https://eprint.iacr.org/2024/1037.

[HR18] Justin Holmgren and Ron Rothblum. Delegating computations with (almost) minimal time and space overhead. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 124–135. IEEE Computer Society, 2018.

[IKOS07] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, pages 21–30, 2007.

[Jer26] Fernando Granha Jeronimo. Algorithmic list decoding at capacity and optimal proximity gaps for reed-solomon codes, 2026. URL: https://arxiv.org/abs/2609.05870, arXiv:2609.05870.

[Jo26] Sunghyeon Jo. Reed–solomon mutual correlated agreement beyond the johnson radius. Cryptology ePrint Archive, Paper 2026/1432, 2026. URL: https://eprint.iacr.org/2026/1432.

[KS24] Abhiram Kothapalli and Srinath T. V. Setty. Hypernova: Recursive arguments for customizable constraint systems. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, Lecture Notes in Computer Science, pages 345–379. Springer, 2024.

[KST22] Abhiram Kothapalli, Srinath T. V. Setty, and Ioanna Tzialla. Nova: Recursive zero-knowledge arguments from folding schemes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV*, Lecture Notes in Computer Science, pages 359–388. Springer, 2022.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. In *FOCS*, pages 2–10, 1990.

[NA25] Andrija Novakovic and Guillermo Angeris. Ligerito: A small and concretely fast polynomial commitment scheme, May 2025. URL: https://angeris.github.io/papers/ligerito.pdf.

[NS25] Wilson D. Nguyen and Srinath T. V. Setty. Neo: Lattice-based folding scheme for CCS over small fields and pay-per-bit commitments. *IACR Cryptol. ePrint Arch.*, 2025:294, 2025.

[NTZ25] Vineet Nair, Justin Thaler, and Michael Zhu. Proving CPU executions in small space. *IACR Cryptol. ePrint Arch.*, 2025:611, 2025. URL: https://eprint.iacr.org/2025/611.

[RW24] Noga Ron-Zewi and Mor Weiss. Zero-knowledge iops approaching witness length. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, Lecture Notes in Computer Science, pages 105–137. Springer, 2024.

[Set20] Srinath T. V. Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part III*, Lecture Notes in Computer Science, pages 704–737. Springer, 2020.

[STW24] Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Unlocking the lookup singularity with lasso. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*, Lecture Notes in Computer Science, pages 180–209. Springer, 2024.

[WHV24] Ruihan Wang, Carmit Hazay, and Muthuramakrishnan Venkitasubramaniam. Ligetron: Lightweight scalable end-to-end zero-knowledge proofs post-quantum zk-snarks on a browser. In *IEEE Symposium on Security and Privacy*, pages 1760–1776, 2024.

[Zei24] Hadas Zeilberger. Khatam: Reducing the communication complexity of code-based snarks. *IACR Cryptol. ePrint Arch.*, 2024:1843, 2024. URL: https://eprint.iacr.org/2024/1843.

A Concrete Proof-Size Estimates

Tables 1 and 2 are generated based on the analysis presented in Section 7. Sizes include queried columns (including masking rows), directly transmitted polynomial coefficients and OOD evaluations, and $t\lambda \log_2 n$ bits of Merkle authentication, with $\lambda = 2\kappa$. All sizes are in decimal MB (10^6 bytes), and no grinding is used.

Table 1: Proof sizes for 256-bit field size

Parameters			Ligero (Unique dec.)		Ligero (List dec.)		Ligero (DKT)		Out-of-Domain		Out-of-Domain (DKT)	
# gates	ρ	κ	t	size	t	size	t	size	t	size	t	size
2^{20}	1/32	80	84	2.74 MB	32	1.68 MB	30	1.63 MB	32	1.71 MB	30	1.65 MB
2^{20}	1/32	100	105	3.08 MB	40	1.88 MB	37	1.81 MB	40	1.91 MB	37	1.84 MB
2^{20}	1/16	80	88	2.81 MB	40	1.88 MB	37	1.81 MB	40	1.90 MB	37	1.83 MB
2^{20}	1/16	100	110	3.16 MB	50	2.11 MB	46	2.02 MB	50	2.13 MB	46	2.04 MB
2^{20}	1/8	80	97	2.95 MB	54	2.19 MB	49	2.08 MB	54	2.21 MB	49	2.10 MB
2^{20}	1/8	100	121	3.31 MB	67	2.45 MB	61	2.33 MB	67	2.47 MB	61	2.35 MB
2^{20}	1/4	80	118	3.26 MB	80	2.67 MB	80	2.67 MB	80	2.69 MB	74	2.58 MB
2^{20}	1/4	100	148	3.67 MB	100	3.00 MB	100	3.00 MB	100	3.02 MB	92	2.89 MB
2^{20}	1/3	80	137	3.52 MB	137	3.52 MB	137	3.52 MB	101	3.02 MB	93	2.90 MB
2^{20}	1/3	100	171	3.95 MB	171	3.95 MB	171	3.95 MB	127	3.40 MB	117	3.27 MB
2^{20}	1/2	80	-	-	-	-	-	-	160	3.82 MB	150	3.69 MB
2^{20}	1/2	100	-	-	-	-	-	-	200	4.29 MB	187	4.15 MB
2^{20}	5/8	80	-	-	-	-	-	-	236	4.66 MB	224	4.53 MB
2^{20}	5/8	100	-	-	-	-	-	-	295	5.25 MB	280	5.11 MB

For Table 2, witness entries and codewords use the indicated base field, while challenges, masking rows, polynomial responses, and OOD values use 256-bit storage. The extension degrees in the size model are therefore $r = 8, 4, 1$, respectively.

Table 2: Scaling and small-field savings for OOD Ligero at rate $\rho = 1/32$. Entries are proof sizes in decimal MB. The leading query-error model gives $t = 32$ and $t = 40$ at the 80- and 100-bit targets, respectively. We have opted for using the Johnson bound for the analysis. The bounds from [DKT26] would provide an additional improvement of around 3 %

	32-bit base field		64-bit base field		256-bit base field	
N	$\kappa = 80$	$\kappa = 100$	$\kappa = 80$	$\kappa = 100$	$\kappa = 80$	$\kappa = 100$
2^{16}	0.184	0.208	0.240	0.271	0.441	0.498
2^{17}	0.253	0.283	0.331	0.372	0.616	0.693
2^{18}	0.349	0.388	0.460	0.515	0.863	0.968
2^{19}	0.485	0.537	0.643	0.717	1.212	1.357
2^{20}	0.677	0.748	0.901	1.002	1.705	1.907
2^{21}	0.949	1.046	1.265	1.405	2.403	2.685
2^{22}	1.333	1.467	1.781	1.975	3.390	3.785

B Comparison of Hash-Based Schemes

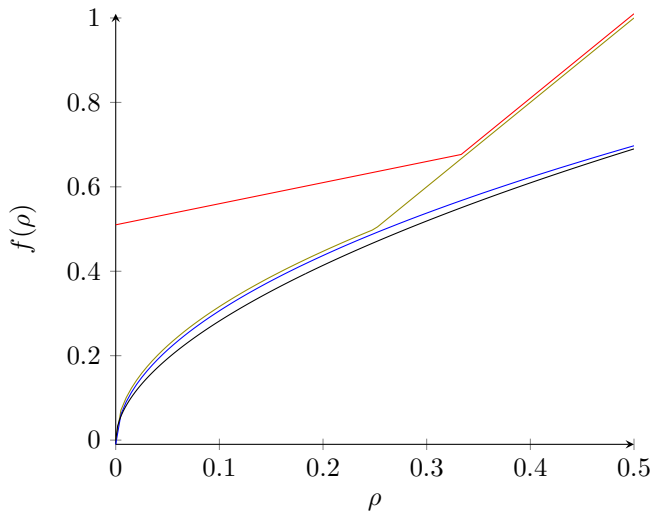
In Table 3, *succinct* means polylogarithmic in computation size N ignoring security-parameter factors. Rounds are counted before Fiat–Shamir compilation, i.e. in the underlying IOP system. Memory efficiency is more qualitative rather than quantitative. Most schemes can be made memory efficient by regenerating the witness. However, the number of times it needs to be regenerated depends on the number of rounds of the underlying IOP. Generating the witness is the most expensive step in proof generation and we mark schemes as memory inefficient if it requires more than constant rounds (and no current implementation are memory efficient). For an implementation to be memory efficient, we require the prover's memory to be sublinear in the circuit size.

Table 3: Qualitative comparison of selected hash-based constructions under their stated assumptions.

Scheme	Succinct proofs	Succinct verification	Number of rounds	Memory efficiency	Zero-knowledge property
Computation arguments					
Our OOD Ligero	$\widetilde{O}(\sqrt{N})$	Batched setting	$O(1)$	Ligetron-compatible	Perfect HVZK
Aurora [BCR+19]	Yes	$O(N)$	$O(\log N)$	No	Yes
STARK + HK masking [BBHR19, HK24]	Yes	Yes	$O(\log N)$	No	Perfect HVZK (HK)
Brakedown (base construction) [GLS+23b]	$\widetilde{O}(\sqrt{N})$	sublinear	$O(\log N)$	No	No, in the base construction
Commitment / proximity components (outer circuit argument excluded)					
Vortex [BSU24]	$\widetilde{O}(\sqrt{N})$	$\widetilde{O}(\sqrt{N})$	$O(1)$ opening	No	No
FRI / STIR [BBHR18, ACFY24]	Yes	Yes	$O(\log N)$	No	No
WHIR [ACFY25]	Yes	Yes	$O(\log N)$	No	No
Ligerito [NA25]	Yes	Yes	$O(\log N)$	No	No
ZK WHIR / Ligerito via CFW [CFW26]	Yes	Yes	$O(\log N)$	No	Perfect HVZK

C Comparison of Soundness Errors

For the different variants of Ligero and their corresponding soundness analyses, we plot the function $f(\rho)$ such that the soundness error is bounded by $f(\rho)^t$ (neglecting much smaller factors). This allows us to visualize the rate regimes in which each variant yields an improvement (to improve visibility, we have not included the version without OOD using the beyond-Johnson bound from [DKT26], which would yield $f(\rho) = \max(a_1(\rho), 2\rho)$ but it is easily deducible from other plots).



— $f(\rho) = \max\{\frac{1+\rho}{2},\ 2\rho\}$ (Original Ligero, Unique Decoding Bound)
— $f(\rho) = \max\{\sqrt{\rho},\ 2\rho\}$ (Original Ligero, Johnson Bound)
— $f(\rho) = \sqrt{\rho}$ (Out-of-Domain Ligero, Johnson Bound)
— $f(\rho) = a_1(\rho)$ (Out-of-Domain Ligero, DKT26 bound)

D Disclosure of AI use

A previous version of this paper was written entirely without the use of generative AI tools. Starting from that version, we have used Generative AI (Claude, GPT) in the following ways:

GenAI was used to help verify correctness of the claims and it detected several problems: the erroneous two-piece mask for the zero knowledge version of the multiplication test (see the explanations in Section 5.4 and the Introduction); some issues with the use of union bounds in the soundness statements of the tests; and the initial circuit encoding did not constrain the output wire, which also required adapting the zero-knowledge simulator. We have since then corrected the results.

We used GenAI to identify which recent results regarding proximity testing beyond the Johnson bound, in particular which results from [DKT26], were applicable in our settings (and in particular to compute the bounds for the parameters after Lemma 12)

We used GenAI to help us compare our work with related recent work (in particular [BSU24, HK24, CFW26]).

Given that several of the proofs in this paper have some overlaps, we used GenAI to quickly identify which parts we could shorten referring to previous proofs, to remove redundancy and make their reading less tedious.

Finally, we used GenAI to generate the tables in the appendices and to find editorial mistakes.

All GenAI-assisted content was checked by the authors.