Polylogarithmic Proofs for Multilinears over Binary Towers

Benjamin E. Diamond
Ulvetanna
bdiamond@ulvetanna.io

Jim Posen
Ulvetanna
jposen@ulvetanna.io

Abstract

We introduce a *polylogarithmic-verifier* polynomial commitment scheme for multilinears over towers of binary fields. To achieve this, we adapt an idea of Zeilberger, Chen and Fisch's *BaseFold* ('23) to the setting of binary towers, using FRI (ICALP '18)'s binary-field variant. In the process, we reinterpret Lin, Chung and Han (FOCS '14)'s *novel polynomial basis* so as to make apparent its compatibility with FRI. We moreover introduce a "packed" version of our protocol, which supports—with no embedding overhead during its commitment phase—multilinears over tiny fields (including that with just two elements). Our protocol leverages a new *multilinear* FRI-folding technique, and exploits the recent tensor proximity gap of Diamond and Posen (Commun. Cryptol. '24). We achieve concretely small proofs for enormous binary multilinears, shrinking the proofs of Diamond and Posen ('23) by an order of magnitude.

1 Introduction

In recent work, Diamond and Posen [DP23b] introduce a sublinear argument designed to capture certain efficiencies available in towers of binary fields. Using a "block-level encoding" technique, that work evades, at least during its commitment phase, the *embedding overhead* prone to arise whenever tiny fields are used, especially in those protocols which critically utilize Reed–Solomon codes. That work's key polynomial commitment scheme features opening proofs whose size and verifier complexity both grow on the order of the square root of the size (i.e., measured in total data bits) of the committed polynomial.

In this work, adapting an idea of Zeilberger, Chen and Fisch's *BaseFold* [ZCF23], we present a multilinear polynomial commitment scheme—designed for polynomials over binary tower fields—whose proof size and verifier complexity grow *polylogarithmically* in the size of the committed polynomial. Our scheme again evades embedding overhead, even when applied on polynomials over tiny fields (like $\mathbb{F}_2$). BaseFold's polynomial commitment scheme [ZCF23, § 5], very roughly, identifies a new connection between Ben-Sasson, Bentov, Horesh and Riabzev's [BBHR18] celebrated *FRI* IOP of proximity and *multilinear* polynomials. Specifically, that work observes that, in the setting of prime-field FRI—and when the FRI folding arity is fixed at 2—the constant *value* of the prover's final FRI oracle relates to the *univariate* coefficients of its FRI message just as a multilinear's *evaluation* relates to its *multilinear* coefficients in the monomial basis. This fact underlies BaseFold's use of FRI within its multilinear polynomial commitment scheme. On the other hand, since the query point might, in general, be known in advance to the prover, whereas FRI's folding challenges of course must not be, BaseFold moreover interleaves into the FRI folding process an execution of the *sumcheck* protocol, thereby reducing the evaluation of the multilinear at the *known* query point to its evaluation at the *random* point sampled during FRI. We describe BaseFold further in Subsection 1.1 below.

We note that FRI has figured in commitment schemes—both univariate and multilinear—previously. All prior such uses of FRI, however—that is, with the exception of [ZCF23, § 5]—invoke "quotienting", and so suffer from embedding overhead, a phenomenon described at length in [DP23b]. We refer to Haböck [Hab22] for a description of FRI's use as a *univariate* commitment scheme. In the multilinear setting, we note briefly an approach proposed by Chen, Bünz, Boneh and Zhang [CBBZ23, § B], which itself makes blackbox of a univariate commitment scheme (presumably FRI). Interestingly, that scheme—assuming the FRI-based univariate scheme—resembles [ZCF23, § 5], at least during its commitment phase. Its query phase, however, generically invokes, logarithmically many times, the underlying univariate scheme's evaluation protocol.

1.1 Technical Overview

Each honest FRI prover begins with the evaluation of some polynomial $P(X) \coloneqq \sum_{j=0}^{2^\ell-1} a_j \cdot X^j$ over the initial FRI domain $S^{(0)}$. Under certain mild conditions—specifically, if the folding factor η divides ℓ, and the recursion is carried out to its end—the prover's final oracle will be *identically constant* over its domain (and in fact, the prover will rather send the verifier this latter constant in the clear). What will the *value* of this constant be, as a function of $P(X)$ and of the verifier's folding challenges?

In the setting of *prime field* multiplicative FRI, the folding maps $q^{(i)}$ all take the especially simple form $X \mapsto X^{2^\eta}$. BaseFold [ZCF23, § 5] makes the interesting observation whereby—again, in the prime field setting, for η now moreover set to 1, and for $q^{(0)}, \ldots, q^{(\ell-1)}$ defined in just this way—the prover's final FRI response will be nothing other than $a_0 + a_1 \cdot r_0 + a_2 \cdot r_1 + \cdots + a_{2^\ell-1} \cdot r_0 \cdots r_{\ell-1}$, where $r_0, \ldots, r_{\ell-1}$ are the verifier's FRI folding challenges. That is, it will be exactly the evaluation of the multilinear polynomial $a_0 + a_1 \cdot X_0 + a_2 \cdot X_1 + \cdots + a_{2^\ell-1} \cdot X_0 \cdots X_{\ell-1}$ at the point $(r_0, \ldots, r_{\ell-1})$.

What about in the binary field setting? In this setting, firstly, the simple folding maps $X \mapsto X^{2^\eta}$ no longer suffice, as [BBHR18, § 2.1] already remarks; rather, we must choose for the maps $q^{(i)}$ a certain sequence of *linear subspace polynomials* of degree 2^η. FRI does *not* suggest precise values for these polynomials, beyond merely demanding that they feature the right linear-algebraic syntax (roughly, each $q^{(i)}$'s kernel should reside entirely inside the domain $S^{(i)}$; we discuss this requirement further in Subsection 2.4 below). Given syntactically valid subspace polynomials $q^{(i)}$ chosen otherwise arbitrarily—and, we emphasize, FRI does *not* suggest a choice—the constant value of the prover's final oracle will, in general, relate in a complicated way to the coefficient vector $(a_0, \ldots, a_{2^\ell-1})$ and to the verifier's folding challenges r_i (the exact relationship will depend on the maps $q^{(i)}$).

The additive NTT and FRI. We recall briefly the "additive NTT" of Lin, Chung, and Han [LCH14] (we refer to Subsection 2.3 below for a more thorough description). We fix a binary field K, of degree more than ℓ. The work [LCH14] defines, first of all, a "novel polynomial basis" $(X_j(X))_{j=0}^{2^\ell-1}$ of the K-vector space consisting of polynomials over K of degree less than 2^ℓ (which, of course, differs from the standard monomial K-basis $\left(X^j\right)_{i=0}^{2^\ell-1}$). The essential idea of [LCH14] is that, for a polynomial $P(X) \coloneqq \sum_{j=0}^{2^\ell-1} a_j \cdot X_j(X)$ expressed *with respect to this basis*, as opposed to in standard monomial form, the "additive NTT" of $P(X)$—that is, the set of $P(X)$'s evaluations over any appropriately chosen affine $\mathbb{F}_2$-vector subspace $S \subset K$—can be computed from $P(X)$'s coefficient vector $(a_0, \ldots, a_{2^\ell-1})$ in quasilinear time (in the size of S).

We recover using the following technique, in the binary-field setting, the "classical" FRI folding pattern identified above. For expository purposes, we fix $\eta = 1$ (though cf. Subsection 3.3 below). We stipulate first of all that the prover use the coefficients $(a_0, \ldots, a_{2^\ell-1})$ of its input multilinear as the coefficients in Lin, Chung and Han [LCH14]'s *novel polynomial basis*—as opposed in the standard univariate monomial basis—of its initial univariate FRI polynomial $P(X) \coloneqq \sum_{j=0}^{2^\ell-1} a_j \cdot X_j(X)$. (This choice of basis has the crucial additional effect of making the prover's evaluation of $P(X)$ over $S^{(0)}$ computable in quasilinear time.) Essentially, our insight is that, if we choose the FRI subspace maps $q^{(0)}, \ldots, q^{(\ell-1)}$ appropriately, then the prover's final FRI oracle becomes, once again, meaningfully related to $P(X)$'s initial coefficient vector $(a_0, \ldots, a_{2^\ell-1})$; that is, it becomes once again $a_0 + a_1 \cdot r_0 + a_2 \cdot r_1 + \cdots + a_{2^\ell-1} \cdot r_0 \cdots r_{\ell-1}$. Specifically, our construction—which we explain in detail in Subsection 3.2 below—opts to define the maps $q^{(0)}, \ldots, q^{(\ell-1)}$ precisely so that they *factor* Lin, Chung and Han [LCH14, § II. C.]'s "normalized subspace vanishing polynomials" $\left(\widehat{W}_i(X)\right)_{i=0}^{\ell}$, in the sense that $\widehat{W}_i(X) = q^{(i-1)} \circ \cdots \circ q^{(0)}$ holds for each $i \in \{0, \ldots, \ell\}$ (see Corollary 3.11). Upon defining the maps $q^{(0)}, \ldots, q^{(\ell-1)}$ in this way, we recover, first of all, a familiar, Fourier-theoretic characterization of the novel basis polynomials $(X_j(X))_{j=0}^{2^\ell-1}$ (see (2)), as well as a reïnterpretation of the algorithm [LCH14, § III.] along more classical lines (see Remark 3.23).

More importantly, our particular choice of the maps $q^{(0)}, \ldots, q^{(\ell-1)}$ moreover serves to recover the coefficient-folding behavior of prime-field FRI (i.e., which was exploited by [ZCF23, § 5]). Indeed, using the polynomial identity (2)—together with certain "higher-order" analogues of the novel polynomial basis $(X_j(X))_{j=0}^{2^\ell-1}$ (see also Remark 3.22)—we are able, with some work, to establish the required FRI folding pattern. Our treatment of these ideas takes place in Theorem 3.20 (see in particular Lemma 3.21).

A new FRI folding mechanism. As it happens, we opt moreover to modify FRI itself, so as to induce a *Lagrange-style*, as opposed to monomial-style, folding operation. That is, in our FRI variant, the value of the prover's final oracle becomes rather $a_0 \cdot (1 - r_0) \cdot \cdots \cdot (1 - r_{\ell-1}) + \cdots + a_{2^\ell - 1} \cdot r_0 \cdot \cdots r_{\ell-1}$, the evaluation at $(r_0, \ldots, r_{\ell-1})$ of the polynomial whose coefficients in the *multilinear Lagrange basis*—as opposed to in the multilinear monomial basis—are $(a_0, \ldots, a_{2^\ell - 1})$. We moreover introduce a *multilinear* style of many-to-one FRI folding, which contrasts with FRI's univariate approach [BBHR18, § 3.2]. We describe our FRI folding variant in Subsection 3.3 below (see in particular Definitions 3.13 and 3.15). Interestingly, our FRI-folding variant makes necessary a sort of proximity gap different from that invoked by FRI. Indeed, while the soundness proof [Ben+23, § 8.2] of FRI uses the proximity gap result [Ben+23, Thm. 1.5] for *low-degree parameterized curves*, our security treatment below makes use of the recent, *tensor-folding*-based proximity gap of Diamond and Posen [DP23a, Thm. 3.1]. To bring the proximity gap result [DP23a, Thm. 3.1] to bear on our multilinear FRI-folding variant, we must perform a degree of algebraic work (see Lemma 3.16).

Our protocols. In Subsection 3.4 below, we present the simplest version of our protocol. That subsection's Construction 3.19 can be viewed as binary-field adaptation of [ZCF23, § 5], which moreover leverages our custom-built binary-field subspace polynomials $q^{(0)}, \ldots, q^{(\ell-1)}$, as well as our multilinear folding technique.

Since Construction 3.4 uses Reed–Solomon codes, that construction demands that its input polynomial's coefficient field K have degree (i.e., over its subfield $\mathbb{F}_2$) *larger* than the polynomial's number of variables ℓ. In Subsection 3.5, we present a further construction, which eliminates this restriction. That is, Subsection 3.5's Construction 3.24 supports polynomials over arbitrary small coefficient fields (including $\mathbb{F}_2$ itself). Internally, Construction 3.24 uses a "packing" technique first introduced by Diamond and Posen [DP23b, § 3.4], and in particular an algebraic object—called the *tower algebra*—introduced in that work (see Definition 2.2 below). Construction 3.24's commitment phase lacks embedding overhead entirely, in the sense that the cost of that phase depends *only* on the size—in bits—of the committed polynomial (and not on the size of its coefficient field). To prove the security of Construction 3.24 (see Theorem 3.27), we revisit the security proof of FRI [BBHR18, § 4.2.2]. In the process, we simplify, reorganize, and streamline that proof, as well as modernize it, exploiting the recent maturation of the *proximity gap* phenomenon exhibited by error-correcting codes [Ben+23]. (We treat FRI's security only within the unique decoding radius, in contrast with [Ben+23, § 8.2].) As was just discussed, our security proof uses the tensor-folding proximity gap [DP23a, Thm. 3.1]; specifically, we use an adaptation of that result to the setting of Reed–Solomon codes *over the tower algebra*. Our main coding-theoretic theorem appears as Theorem 2.5 below; that result recapitulates a result already proven by Diamond and Posen [DP23b, Thm. 3.10], which in turn extends [DP23a, Thm. 3.1].

The complexity of Construction 3.24's evaluation protocol *does* depend on the coefficient field, and indeed becomes rather more expensive as the polynomial's coefficient field shrinks (i.e., for total data size held constant). We refer to Subsection 3.6 (see also Tables 1 and 2) for a discussion of this phenomenon. In Section 4, we present a final scheme, we which mitigates this phenomenon. Section 4's Construction 4.1 generalizes still further our paper's technique, so as to incorporate into it ideas from the Brakedown-style protocol [DP23b]. Construction 4.1 represents a "hybrid" between Construction 3.24 and [DP23b, Cons. 3.11]; that construction *begins* as does [DP23b, Cons. 3.11], and yet uses Construction 3.24 for its internal proximity test (as opposed to the trivial sort of test—used by [DP23b, Cons. 3.11]—in which which $\mathcal{P}$ simply sends $\mathcal{V}$ the relevant message). Construction 4.1 in fact, represents essentially a sweeping generalization simultaneously of [DP23b, Cons. 3.11] and of Construction 3.24, and makes available a far-wider space of tradeoffs. In Subsection 4.1, we show that Construction 4.1—parameterized appropriately—indeed yields proofs which are simultaneously smaller than those of Construction 3.24 *and* of [DP23b, Cons. 3.11], where the advantage of Construction 4.1 over Construction 3.24 becomes more pronounced as the coefficient field shrinks.

Miscellanea. Throughout Subsection 3.2, we examine in detail various further aspects of binary-field FRI. For example—even in the abstract IOP model—we must necessarily fix $\mathbb{F}_2$-bases of the respective Reed–Solomon domains $S^{(i)}$, in order to interpret committed functions $f^{(i)} : S^{(i)} \to K$ as K-valued *strings* (that is, must implicitly lexicographically "flatten" each domain $S^{(i)}$ using some ordered $\mathbb{F}_2$-basis of it, known to both the prover and the verifier). The choice of these bases matters. Indeed, for $\mathbb{F}_2$-bases of $S^{(i)}$ and $S^{(i+1)}$ chosen arbitrarily, the fundamental operation which associates to each $y \in S^{(i+1)}$ its fiber ${q^{(i)}}^{-1}(\{y\}) \subset S^{(i)}$—which both the prover and the verifier must perform repeatedly—could come to assume

complexity on the order of $\dim\left(S^{(i)}\right)^2$ bit-operations, even after a linear-algebraic preprocessing phase.

Below, we suggest a family of bases for the respective domain $S^{(i)}$ with respect to which the maps $q^{(i)}$ come to act simply by *projecting* away their first η coordinates. In particular, the application of each map $q^{(i)}$—in coordinates—becomes free; the preimage operation ${q^{(i)}}^{-1}(\{y\})$ comes to amount simply to that of prepending η arbitrary boolean coordinates to y's coordinate representation. While bases with these properties can of course be constructed in FRI even for maps $q^{(i)}$ chosen arbitrarily, our procedure—which, we emphasize, depends on our specially chosen maps $q^{(i)}$—*moreover* yields a basis of the initial domain $S^{(0)}$ which coincides with that expected by the additive NTT of [LCH14]. In particular, our prover may use *as is* the output of the additive NTT as its 0^{th} FRI oracle, without first subjecting that output to the permutation induced by an appropriate change-of-basis transformation on $S^{(0)}$. We believe that these observations stand to aid *all* implementers of binary-field FRI.

1.2 Prior Work

The works most relevant to this one are Zeilberger, Chen and Fisch's *BaseFold* [ZCF23] and Diamond and Posen [DP23b]. *BaseFold* [ZCF23, § 5] introduces the connection between FRI folding and multilinear evaluation upon which this work rests (see also Subsection 1.1 above). That work uses only prime-field FRI, and does not attempt to support small fields (with or without embedding overhead).

The work [DP23b] introduces the use of towers of binary fields in SNARKs, and moreover develops several key ideas fundamental to this one, including the extension code construction [DP23b, § 3.1] and the tower algebra [DP23b, Def. 3.8]. That work moreover isolates the phenomenon of *embedding overhead*, and provides a sublinear argument designed to evade it, at least during its commitment phase [DP23b, Cons. 3.11]. We note that [DP23b] supplies not just a multilinear polynomial commitment scheme, but moreover an entire toolbox of "virtual polynomial protocols" [DP23b, § 4] and a high-level SNARK [DP23b, § 5]. This work presents *only* a polynomial commitment scheme (or rather, a sequence of them). The higher-level content of [DP23b] remains perfectly applicable in our setting; indeed, our scheme serves as a drop-in replacement for that of [DP23b, § 3], and serves the purposes of [DP23b, §§ 4–5] exactly as [DP23b, § 3] does. Our scheme equally stands to concretely improve the open-source implementation, called *Binius*, of [DP23b].

During our main security proofs (see Theorems 3.27 and 4.3), we draw variously on the works [BBHR18] and [Ben+23]. Neither of those works, on the other hand, contain results which serve *as stated* to achieve our purposes; rather, we must instead selectively extract and adapt their ideas. Indeed, our setting differs from those of those works in at least three ways. For one, our results use codes over the tower algebra, as opposed to codes over a field. Secondly, we use a different style of FRI folding than those works do, akin more to *multilinear* interpolation than to *univariate* interpolation. Finally, our soundness proof must concern itself not merely with the prover's distance from the code, but moreover with the *consistency* of its oracles. In any case, the essential ideas of our Lemmas 3.35 and 3.36 below are implicit in [BBHR18, § 4.2.2]; moreover, our Proposition 3.30 below can be viewed as an adaptation to our setting of a technique of [Ben+23, § 8.2].

Unfortunately, we are *not* able, in this work, to use the Reed–Solomon-specific proximity gap [Ben+23, Thm. 1.2], in either of its parameter regimes. The list-decoding regime, for its part, seems simply not to be useful to us, at least barring an advance in our proof strategy. Indeed, since—as just discussed—we must demand not just the proximity but the consistency of the prover's oracles, excessive distance from the code comes ultimately not to help but to harm us, in a way made precise in the proof of Lemma 3.35 (a similar phenomenon occurs in [DP23a, Lem. 4.10]). We leave as an open question whether the per-query rejection probability of this work can be brought, even in principle, above the (relative) unique decoding radius.

On the other hand, we are likewise unable even to use the unique-decoding-specific result [Ben+23, Thm. 4.1] in this work, albeit for different reasons. Indeed, that result is stated for Reed–Solomon codes over fields. Despite expending significant effort, we were not able to adapt that result to our setting of Reed–Solomon codes over the *tower algebra* (a commutative ring which fails, in general, even to be an integral domain, though it contains a field). That hypothetical adaptation would serve to shrink our protocol's proof sizes, concretely, by a factor of roughly $\log(3) - 1 \approx 0.585$. Alternatively, a resolution of the general coding-theoretic conjecture [DP23b, Conj. 2.4] would equally serve this end, at least granting its proof's adaptability to the tower algebra. As it stands, we must instead fall back to the generic Theorem 2.5 below, whose proximity parameter is worse (forced to remain beneath a third of the code's distance, as opposed to a half). We discuss these matters thoroughly in Subsection 2.6 below (see also Conjecture 2.6).

2 Background and Notation

We record notation, following Diamond and Posen [DP23b] where possible. We write $\mathbb{N}$ for the nonnegative integers. We write B^A for the set of maps between sets $A \to B$. We fix a binary field K. For each $\ell \in \mathbb{N}$, we write $\mathcal{B}_\ell$ for the ℓ-dimensional *boolean hypercube* $\{0,1\}^\ell \subset K^\ell$. We frequently identify $\mathcal{B}_\ell$ with the integer range $\{0, \dots 2^\ell - 1\}$ by means of the lexicographic identification $v \mapsto \{v\} \coloneqq \sum_{i=0}^{\ell-1} 2^i \cdot v_i$. The *rings* we treat are nonzero and commutative with unit. For our purposes, an *algebra* A over a field K, also called a *K-algebra*, is a commutative ring A together with an embedding of rings $K \hookrightarrow A$. We adopt the notational convention whereby the degree of the 0 polynomial is $\deg(0) = -\infty$. For for $L \,/\, K$ a field extension and $R \subset L^\vartheta$ a subset, we write $\mu(R) \coloneqq \frac{|R|}{|L|^\vartheta}$.

2.1 Lagrange and Monomial Forms

We review various normal forms for multilinear polynomials, following [DP23b, § 2.1]. An ℓ-variate polynomial in $K[X_0, \dots, X_{\ell-1}]$ is *multilinear* if each of its indeterminates appears with individual degree at most 1; we write $K[X_0, \dots, X_{\ell-1}]^{\preceq 1}$ for the set of multilinear polynomials over K in ℓ indeterminates. Clearly, the set of monomials $(1, X_0, X_1, X_0 \cdot X_1, \dots, X_0 \cdot \dots \cdot X_{\ell-1})$ yields a K-basis for $K[X_0, \dots, X_{\ell-1}]^{\preceq 1}$; we call this basis the *multilinear monomial basis* in ℓ variables.

We introduce the $2 \cdot \ell$-variate polynomial

$$\widetilde{\mathtt{eq}}(X_0, \dots, X_{\ell-1}, Y_0, \dots, Y_{\ell-1}) \coloneqq \prod_{i=0}^{\ell-1} (1 - X_i) \cdot (1 - Y_i) + X_i \cdot Y_i.$$

It is essentially the content of Thaler [Tha22, Fact. 3.5]) that the set $\left(\widetilde{\mathtt{eq}}(X_0, \dots, X_{\ell-1}, v_0, \dots, v_{\ell-1})\right)_{v \in \mathcal{B}_\ell}$ yields a K-basis of the space $K[X_0, \dots, X_{\ell-1}]^{\preceq 1}$.

For each fixed $(r_0, \dots, r_{\ell-1}) \in L^\ell$, the vector $\left(\widetilde{\mathtt{eq}}(r_0, \dots, r_{\ell-1}, v_0, \dots, v_{\ell-1})\right)_{v \in \mathcal{B}_\ell}$ takes the form

$$\left(\prod_{i=0}^{\ell-1} r_i \cdot v_i + (1 - r_i) \cdot (1 - v_i) \right)_{v \in \mathcal{B}_\ell} = ((1 - r_0) \cdot \dots \cdot (1 - r_{\ell-1}), \dots, r_0 \cdot \dots \cdot r_{\ell-1}).$$

We call this vector the *tensor product expansion* of the point $(r_0, \dots, r_{\ell-1}) \in L^\ell$, and denote it by $\bigotimes_{i=0}^{\ell-1}(1 - r_i, r_i)$. We note that this latter vector can be computed in $\Theta(2^\ell)$ time (see e.g. [Tha22, Lem. 3.8]).

As a notational device, we introduce the further $2 \cdot \ell$-variate polynomial:

$$\widetilde{\mathtt{mon}}(X_0, \dots, X_{\ell-1}, Y_0, \dots, Y_{\ell-1}) \coloneqq \prod_{i=0}^{\ell-1} 1 + (X_i - 1) \cdot Y_i;$$

we note that $\left(\widetilde{\mathtt{mon}}(X_0, \dots, X_{\ell-1}, v_0, \dots, v_{\ell-1})\right)_{v \in \mathcal{B}_\ell}$ yields precisely the multilinear monomial basis in ℓ indeterminates.

2.2 Error-Correcting Codes

We recall details on codes, referring throughout to Guruswami [Gur06].

A *code* of block length n over the alphabet Σ is a subset of Σ^n. In Σ^n, we write d for the Hamming distance between two vectors (i.e., the number of components at which they differ). We again fix a field K. A linear *$[n, k, d]$-code* over K is a k-dimensional linear subspace $C \subset K^n$ for which $d(v_0, v_1) \geq d$ holds for each unequal pair of elements v_0 and v_1 of C. An $[n, k, d]$-code $C \subset K^n$'s *unique decoding radius* is $\left\lfloor \frac{d-1}{2} \right\rfloor$; indeed, we note that, for each word $u \in K^n$, at most one codeword $v \in C$ satisfies $d(u, v) < \frac{d}{2}$ (this fact is a direct consequence of the triangle inequality). For $u \in K^n$ arbitrary, we write $d(u, C) \coloneqq \min_{v \in C} d(u, v)$ for the *distance* between u and the code C.

Given a linear code $C \subset K^n$ and an integer $m \geq 1$, we have C's *m-fold interleaved code*, defined as the subset $C^m \subset (K^n)^m \cong (K^m)^n$. We understand this latter set as a length-n block code over the alphabet K^m. In particular, its elements are naturally identified with those matrices in $K^{m \times n}$ each of whose rows is

a C-element. We write matrices $(u_i)_{i=0}^{m-1} \in K^{m\times n}$ row-wise. By definition of C^m, two matrices in $K^{m\times n}$ *differ* at a column if they differ at *any* of that column's components. That a matrix $(u_i)_{i=0}^{m-1} \in K^{m\times n}$ is within distance e to the code C^m—in which event we write $d^m\Big((u_i)_{i=0}^{m-1}, C^m\Big) \leq e$—thus entails precisely that there exists a subset $D \coloneqq \Delta^m\Big((u_i)_{i=0}^{m-1}, C^m\Big)$, say, of $\{0,\dots,n-1\}$, of size at most e, for which, for each $i \in \{0,\dots,m-1\}$, the row u_i admits a codeword $v_i \in C$ for which $u_i|_{\{0,\dots,n-1\}\setminus D} = v_i|_{\{0,\dots,n-1\}\setminus D}$.

We recall Reed–Solomon codes (see [Gur06, Def. 2.3]). For notational convenience, we consider only Reed–Solomon codes whose message and block lengths are powers of two. We fix nonnegative *message length* and *rate* parameters ℓ and $\mathcal{R}$, as well as a subset $S \subset K$ of size $2^{\ell+\mathcal{R}}$. We write $C \subset K^{2^{\ell+\mathcal{R}}}$ for the *Reed–Solomon* code $\mathsf{RS}_{K,S}[2^{\ell+\mathcal{R}}, 2^\ell]$ is defined to be the set $\Big\{(P(x))_{x\in S} \;\Big|\; P(X) \in K[X]^{\prec 2^\ell}\Big\}$; that is, $\mathsf{RS}_{K,S}[2^{\ell+\mathcal{R}}, 2^\ell]$ is the set of those $2^{\ell+\mathcal{R}}$-tuples which arise as the evaluations of some polynomial of degree less than 2^ℓ over S. The distance of $\mathsf{RS}_{K,S}[n,k]$ is $d = 2^{\ell+\mathcal{R}} - 2^\ell + 1$. We write $\mathsf{Enc} : K[X]^{\prec 2^\ell} \to K^S$ for the *encoding function* which maps a polynomial $P(X)$ of degree less than 2^ℓ to its tuple of evaluations over S.

We recall the *Berlekamp–Welch* algorithm for Reed–Solomon decoding within the unique decoding radius (see [Gur06, Rem. 4]). For self-containedness, we record a slight extension of that algorithm in which the received word $f : S \to K$ is *not* assumed to reside within the code's unique decoding radius. The key point is to study the behavior of the algorithm even in the case $d(f,C) \geq \frac{d}{2}$.

Algorithm 1 (Berlekamp–Welch [Gur06, Rem. 4].)

procedure DecodeReedSolomon$\big((f(x))_{x\in S}\big)$
  allocate $A(X)$ and $B(X)$ of degrees $\lfloor\frac{d-1}{2}\rfloor$ and $2^{\ell+\mathcal{R}} - \lfloor\frac{d-1}{2}\rfloor - 1$; write $Q(X,Y) \coloneqq A(X)\cdot Y + B(X)$.
  interpret the equalities $Q(x, f(x)) = 0$, for $x \in S$, as a system of $2^{\ell+\mathcal{R}}$ equations in $2^{\ell+\mathcal{R}}+1$ unknowns.
  by finding a nonzero solution of this linear system, obtain values for the polynomials $A(X)$ and $B(X)$.
  if $A(X) \nmid B(X)$ **then return** $\perp$.
  write $P(X) \coloneqq -B(X)/A(X)$.
  if $\deg(P(X)) \geq 2^\ell$ **then return** $\perp$.
  return $P(X)$.

We note that the unknown polynomial $Q(X,Y)$ above indeed has $\lfloor\frac{d-1}{2}\rfloor + 1 + 2^{\ell+\mathcal{R}} - \lfloor\frac{d-1}{2}\rfloor = 2^{\ell+\mathcal{R}} + 1$ coefficients, as required.

When $d(f,C) < \frac{d}{2}$, Algorithm 1 necessarily returns the unique polynomial $P(X)$ of degree less than 2^ℓ for which $d(f, \mathsf{Enc}(P(X))) < \frac{d}{2}$ holds. Indeed, this is simply the correctness of Berlekamp–Welch on input assumed to reside within the unique decoding radius; we refer to [Gur06, Rem. 4] for a thorough treatment. (We note that the $(1, 2^\ell - 1)$-weighted degree of $Q(X,Y)$ is at most $D = 2^{\ell+\mathcal{R}} - \lfloor\frac{d-1}{2}\rfloor - 1$, while $t = 2^{\ell+\mathcal{R}} - \lfloor\frac{d-1}{2}\rfloor$; the hypothesis of [Gur06, Lem. 4.3] is therefore fulfilled. We conclude that $Q(X, P(X))$, of degree at most D with at least t zeros, is in fact identically zero, so that $Y - P(X) \mid Q(X,Y)$.) We record the following converse:

Lemma 2.1. *If $d(f,C) \geq \frac{d}{2}$, then Algorithm 1 outputs $\perp$.*

Proof. We fix a map $f : S \to K$ for which $d(f,C) \geq \frac{d}{2}$; we suppose for contradiction that Algorithm 1, on the input f, nonetheless successfully outputs a polynomial $P(X)$ (necessarily of degree less than 2^ℓ). We first note that the relation $P(X) = -B(X)/A(X)$ implies the factorization $Q(X,Y) = A(X)\cdot(Y - P(X))$. Separately, since $\deg(P(X)) < 2^\ell$, $\mathsf{Enc}(P(X))$ is a codeword; our hypothesis on f thus implies that $d(f, \mathsf{Enc}(P(X))) \geq \frac{d}{2}$. On the other hand, by its degree, $A(X)$ can have at most $\lfloor\frac{d-1}{2}\rfloor < \frac{d}{2}$ roots. We conclude that there necessarily exists some element $x^* \in S$ for which $P(x^*) \neq f(x^*)$ *and* $A(x^*) \neq 0$ simultaneously hold. Finally, by its construction, $Q(x, f(x)) = 0$ necessarily holds for each $x \in S$. Putting these facts together, we see that $0 = Q(x^*, f(x^*)) = A(x^*)\cdot(f(x^*) - P(x^*)) \neq 0$, a contradiction. □

Most analyses of the Berlekamp–Welch algorithm *assume* inputs guaranteed to reside within the unique decoding radius, and implicitly leave as undefined the algorithm's behavior on arbitrary words. It is interesting to note that the algorithm (with the aid of a simple additional degree check) serves moreover to *detect* whether its input is in the unique decoding radius.

2.3 Novel Polynomial Basis

We recall in detail the *novel polynomial basis* of Lin, Chung and Han [LCH14, § II.]. We fix again a binary field K, of degree r, say, over $\mathbb{F}_2$. For our purposes, a *subspace polynomial* over K is a polynomial $W(X) \in K[X]$ which splits completely over K, and whose roots, each of multiplicity 1, form an $\mathbb{F}_2$-linear subspace of K. For a detailed treatment of subspace polynomials, we refer to Berlekamp [Ber15, § 11]. We recall that, for each subspace polynomial $W(X) \in K[X]$, the evaluation map $W : K \to K$ is $\mathbb{F}_2$-*linear*.

For each fixed $\ell \in \{0, \ldots, r-1\}$, the set $K[X]^{\prec 2^\ell}$ of polynomials of degree less than 2^ℓ is a K-vector space of dimension 2^ℓ. Of course, the set $(1, X, X^2, \ldots, X^{2^\ell - 1})$ yields an obvious K-basis of $K[X]^{\prec 2^\ell}$. Lin, Chung and Han define an alternate K-basis of $K[X]^{\prec 2^\ell}$—called the *novel polynomial basis*—in the following way. We fix once and for all an $\mathbb{F}_2$-basis $(\beta_0, \ldots, \beta_{r-1})$ of the ground field K. For each $i \in \{0, \ldots, \ell - 1\}$, we write $U_i \coloneqq \langle \beta_0, \ldots, \beta_{i-1} \rangle$ for the $\mathbb{F}_2$-linear span of the prefix $(\beta_0, \ldots, \beta_{i-1})$, and define the *subspace vanishing polynomial* $W_i(X) \coloneqq \prod_{u \in U_i} X - u$, as well as its *normalized* variant $\widehat{W}_i(X) \coloneqq \frac{W_i(X)}{W_i(\beta_i)}$ (we note that $\beta_i \notin U_i$, so that $W_i(\beta_i) \neq 0$). In words, for each $i \in \{0, \ldots, \ell - 1\}$, $W_i(X)$ vanishes precisely on $U_i \subset K$; $\widehat{W}_i(X)$ moreover satisfies $\widehat{W}_i(X)(\beta_i) = 1$. Finally, for each $j \in \{0, \ldots, 2^\ell - 1\}$, we write $X_j(X) \coloneqq \prod_{i=0}^{\ell-1} \widehat{W}_i(X)^{j_i}$; here, $(j_0, \ldots, j_{\ell-1})$ are the bits of j, in the sense that $j = \sum_{k=0}^{\ell-1} 2^k \cdot j_k$ holds. We note that, for each $j \in \{0, \ldots, 2^\ell - 1\}$, $X_j(X)$ is of degree j. We conclude that the change-of-basis matrix from $(1, X, \ldots, X^{2^\ell - 1})$ to $(X_0(X), X_1(X), \ldots, X_{2^\ell - 1}(X))$ is triangular (with an everywhere-nonzero diagonal), so that this latter list indeed yields a K-basis of $K[X]^{\prec 2^\ell}$.

As in Subsection 2.4 above, we now fix moreover a rate parameter $\mathcal{R} \in \{1, \ldots, r - \ell\}$ and an $\mathbb{F}_2$-subspace $S \subset K$ of dimension $\ell + \mathcal{R}$; now, we require moreover that S *contain* the $\mathbb{F}_2$-subspace $U_\ell \coloneqq \langle \beta_0, \ldots, \beta_{\ell-1} \rangle$. Lin, Chung and Han [LCH14, § III.] show that, for $S \subset K$ defined in this way, and for $P(X) \coloneqq \sum_{j=0}^{2^\ell - 1} a_j \cdot X_j(X)$ given in coordinates with respect to the novel polynomial basis defined above, the encoding $(P(x))_{x \in S}$ can be computed in time $\Theta(\ell \cdot 2^{\ell + \mathcal{R}})$.

In Remark 3.23 below, we suggest a new interpretation of Lin, Chung and Han's algorithm [LCH14, § III.], based on the techniques of this paper. For now, for self-containedness, we record here in full their key algorithm, in our notation. We note that Algorithm 2's equivalence with [LCH14, § III.] is not obvious; we explain the correctness of this description in Remark 3.23 below. In what follows, we fix as above the degree and rate parameters ℓ and $\mathcal{R}$. We finally fix a polynomial $P(X) = \sum_{j=0}^{2^\ell - 1} a_j \cdot X_j(X)$; we write $b : \mathcal{B}_{v+\mathcal{R}} \to K$ for $(a_j)_{j=0}^{2^\ell - 1}$'s $2^\mathcal{R}$-fold tiling, so that, for each $v \in \mathcal{B}_{\ell+\mathcal{R}}$, $b(v) \coloneqq a_{\{v\} \pmod{2^\ell}}$ holds.

Algorithm 2 (Lin–Chung–Han [LCH14, § III.].)

procedure AdditiveNTT$\Big((b(v))_{v \in \mathcal{B}_{\ell+\mathcal{R}}}\Big)$
  for $i \in \{\ell - 1, \ldots, 0\}$ (i.e., in downward order) **do**
    for $(u, v) \in \mathcal{B}_{\ell+\mathcal{R}-i-1} \times \mathcal{B}_i$ **do**
      define the twiddle factor $t \coloneqq \sum_{k=0}^{\ell+\mathcal{R}-i-2} u_k \cdot \widehat{W}_i(\beta_{i+1+k})$.
      overwrite both $b(u \parallel 0 \parallel v)$ $+\!\!=$ $t \cdot b(u \parallel 1 \parallel v)$ and $b(u \parallel 1 \parallel v)$ $+\!\!=$ $b(u \parallel 0 \parallel v)$.
  return $(b(v))_{v \in \mathcal{B}_{\ell+\mathcal{R}}}$.

We note that the twiddle factor t above depends only on u, and not on v, and can be reused accordingly. Finally, in the final return statement above, we implicitly identify $\mathcal{B}_{\ell+\mathcal{R}} \cong S$ using the standard basis $\beta_0, \ldots, \beta_{\ell+\mathcal{R}-1}$ of the latter space (see also Subsection 3.2 below).

2.4 FRI

We recall Ben-Sasson, Bentov, Horesh and Riabzev's [BBHR18] *Fast Reed–Solomon Interactive Oracle Proof of Proximity* (FRI). For K a binary field, and size and rate parameters ℓ and $\mathcal{R}$ fixed, FRI yields an *IOP of proximity* for the Reed–Solomon code $\mathsf{RS}_{K,S}[2^{\ell+\mathcal{R}}, 2^\ell]$; here, we require that $S \subset K$ be an $\mathbb{F}_2$-*linear* subspace (of dimension $\ell + \mathcal{R}$, of course). That is, FRI yields an IOP for the claim whereby some oracle $[f]$—i.e., representing a function $f : S \to K$—is close to a codeword $(P(x))_{x \in S}$ (here, $P(X) \in K[X]^{\prec 2^\ell}$ represents

a polynomial of degree less than 2^ℓ). FRI's verifier complexity is polylogarithmic in 2^ℓ. We abbreviate $\rho \coloneqq 2^{-\mathcal{R}}$, so that $\mathsf{RS}_{K,S}[2^{\ell+\mathcal{R}}, 2^\ell]$ is of rate ρ.

Internally, FRI makes use of a folding constant η—which we fix once and for all to be 1—as well as a fixed, global *sequence* of subspaces and maps of the form:

$$S = S^{(0)} \xrightarrow{q^{(0)}} S^{(1)} \xrightarrow{q^{(1)}} S^{(2)} \xrightarrow{q^{(2)}} \cdots \xrightarrow{q^{(\ell-1)}} S^{(\ell)}. \tag{1}$$

Here, for each $i \in \{0, \ldots, \ell-1\}$, $q^{(i)}$ is a subspace polynomial of degree $\eta \coloneqq 1$, whose kernel moreover is contained in $S^{(i)}$. By linear-algebraic considerations, we conclude that $S^{(i+1)}$'s $\mathbb{F}_2$-dimension is 1 less than $S^{(i)}$'s is; inductively, we conclude that each $S^{(i)}$ is of dimension $\ell + \mathcal{R} - i$.

2.5 The Tower Algebra

We recall towers of binary fields, referring throughout to [DP23b, § 2.3]. For simplicity, we present only Wiedemann's tower [Wie88]; on the other hand, our results go through without change on other binary towers (cf. e.g. the *Cantor tower* given in Li et al. [Li+18, § 2.1]). That is, we set $\mathcal{T}_0 \coloneqq \mathbb{F}_2$ and $\mathcal{T}_1 \coloneqq \mathbb{F}_2[X_0]/(X_0^2 + X_0 + 1)$, and, for each $\iota > 1$, $\mathcal{T}_\iota \coloneqq \mathcal{T}_{\iota-1}/(X_{\iota-1}^2 + X_{\iota-2} \cdot X_{\iota-1} + 1)$. Fan and Paar [FP97] observe that the basic arithmetic operations in Wiedemann's tower admit efficient—that is, $O\big(2^{\log(3)\cdot\iota}\big)$-time—algorithms.

The *monomial* $\mathbb{F}_2$*-basis* of the binary tower $\mathcal{T}_\iota$ is $(\beta_v)_{v\in\mathcal{B}_\iota} \coloneqq (\widetilde{\mathtt{mon}}(X_0, \ldots, X_{\iota-1}, v_0, \ldots, v_{\iota-1}))_{v\in\mathcal{B}_\iota}$. More generally, for each pair of integers $\iota \geq 0$ and $\kappa \geq 0$, the set $(\widetilde{\mathtt{mon}}(X_\iota, \ldots, X_{\iota+\kappa-1}, v_0, \ldots, v_{\iota-1})_{v\in\mathcal{B}_\iota}$ likewise yields a $\mathcal{T}_\iota$-basis of $\mathcal{T}_{\iota+\kappa}$; we again write $(\beta_v)_{v\in\mathcal{B}_\kappa}$ for this basis.

Fixing again an element $\ell \in \{0, \ldots, 2^{\iota+\kappa} - 1\}$, and applying the results of Subsection 2.3 to $\mathcal{T}_{\iota+\kappa}$—using now the *multilinear* $\mathbb{F}_2$-basis $(\beta_0, \ldots, \beta_{2^{\iota+\kappa}-1})$ of $\mathcal{T}_{\iota+\kappa}$—we obtain as before a corresponding novel polynomial basis $(X_0(X), X_1(X), \ldots, X_{2^\ell-1}(X))$ of the $\mathcal{T}_{\iota+\kappa}$-vector space $\mathcal{T}_{\iota+\kappa}[X]^{\prec 2^\ell}$, as well as, for each $S \subset \mathcal{T}_{\iota+\kappa}$ of the form given above, an efficient encoding algorithm for the Reed–Solomon code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa},S}[2^{\ell+\mathcal{R}}, 2^\ell]$.

We recall the *tower algebra* data structure of Diamond and Posen [DP23b, § 3.4]. The tower algebra is a mathematical object which algebraically captures our key "packing" technique. Informally, for integers $\iota \geq 0$ and $\kappa \geq 0$ fixed, the multilinear basis $\mathcal{T}_\iota$-basis $(\beta_v)_{v\in\mathcal{B}_\kappa}$ of $\mathcal{T}_{\iota+\kappa}$ allows us to associate, to each vector $(a_v)_{v\in\mathcal{B}_\kappa}$ of $\mathcal{T}_\iota$-elements, a $\mathcal{T}_{\iota+\kappa}$-element, say α. The tower algebra makes this association algebraic, in such a way as to give meaning to the "multiplication" of α by a $\mathcal{T}_\tau$-element (here, $\mathcal{T}_\tau \,/\, \mathcal{T}_\iota$ is a cryptographically large extension). This latter multiplication, on the other hand, does not proceed simply by embedding $\mathcal{T}_{\iota+\kappa} \subset \mathcal{T}_\tau$; rather, it operates "independently" on α's 2^κ components. This special sort of multiplication will prove crucial in our packing-based scheme, which we present in Subsection 3.5 below (see Construction 3.24).

We recall the key definition verbatim. We fix parameters ι, κ, and τ in $\mathbb{N}$, where $\tau \geq \iota$; here, $\mathcal{T}_\iota$ represents our coefficient field, κ is our packing factor, and $\mathcal{T}_\tau$ yields a cryptographically sized extension of $\mathcal{T}_\iota$.

Definition 2.2 (Diamond–Posen [DP23b, Def. 3.8])**.** We define the *tower algebra* $A_{\iota,\kappa,\tau}$ as:

$$A_{\iota,\kappa,\tau} \coloneqq \mathcal{T}_\tau[Y_0, \ldots, Y_{\kappa-1}]/\big(Y_0^2 + X_{\iota-1} \cdot Y_0 + 1, Y_1^2 + Y_0 \cdot Y_1 + 1, \ldots, Y_{\kappa-1}^2 + Y_{\kappa-2} \cdot Y_{\kappa-1} + 1\big),$$

where we understand $X_{\iota-1}$ as a $\mathcal{T}_\tau$-element (and slightly abuse notation by letting $X_{-1} \coloneqq 1$ in case $\iota = 0$).

Concretely, each $A_{\iota,\kappa,\tau}$-element may be represented as a $2^{\tau-\iota} \times 2^\kappa$ array of $\mathcal{T}_\iota$-elements (we refer to [DP23b, Fig. 1]). The left-most column of this array represents the subring consisting of the constant polynomials in $Y_0, \ldots, Y_{\kappa-1}$; we call this subring, which is isomorphic as a ring to the field $\mathcal{T}_\tau$, the *constant subring*. The top-most row of the array represents the subring consisting of polynomials in the indeterminates $Y_0, \ldots, Y_{\kappa-1}$ whose coefficients all reside in $\mathcal{T}_\iota$. We call this latter subring, which is isomorphic as a ring to $\mathcal{T}_{\iota+\kappa}$, the *synthetic subring*. These two subrings define $\mathcal{T}_\tau$ and $\mathcal{T}_{\iota+\kappa}$ (respectively) vector-space structures on the ring $A_{\iota,\kappa,\tau}$; we call these the *constant* and *synthetic* vector space structures (respectively). Similarly, we refer to $A_{\iota,\kappa,\tau}$ as a $\mathcal{T}_\tau$*-algebra* and as a $\mathcal{T}_{\iota+\kappa}$*-algebra* with these vector space structures, respectively, in mind. We finally recall the $\mathcal{T}_\tau$-vector space isomorphism $a_{\iota,\kappa,\tau} : \mathcal{T}_\tau^{2^\kappa} \to A_{\iota,\kappa,\tau}$, which sends the coefficient vector $a_{\iota,\kappa,\tau} : (\varsigma_u)_{u\in\mathcal{B}_\kappa} \mapsto \sum_{u\in\mathcal{B}_\kappa} \varsigma_u \cdot \widetilde{\mathtt{mon}}(Y_0, \ldots, Y_{\kappa-1}, u_0, \ldots, u_{\kappa-1})$; following [DP23b, § 3.4], we call this map the *natural embedding*.

We recall the *extension code* construction of [DP23b, § 3.1]. For notational convenience, we specialize that construction to our setting of interest (namely, to that characterized by a Reed–Solomon code with

symbols in the tower algebra). Below, we fix tower height parameters ι, κ, and τ as above, size and rate parameters $\ell \geq 0$ and $\mathcal{R} \geq 0$, and finally a domain $S \subset \mathcal{T}_{\iota+\kappa}$ constructed as in Subsection 2.5.

Definition 2.3 (Diamond–Posen [DP23b, Def. 3.1])**.** For $C \subset \mathcal{T}_{\iota+\kappa}^{2^{\ell+\mathcal{R}}}$ the code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa},S}[2^{\ell+\mathcal{R}}, 2^{\ell}]$, and $A_{\iota,\kappa,\tau}$ a tower algebra, we define C's *extension code* $\widehat{C} \subset A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}}}$ by reusing C's generator matrix. Equivalently, we set as $\widehat{C} \coloneqq \left\{ (P(x))_{x \in S} \;\middle|\; P(X) \in A_{\iota,\kappa,\tau}[X]^{\prec 2^{\ell}} \right\}$ the set of $A_{\iota,\kappa,\tau}$-valued $2^{\ell+\mathcal{R}}$-tuples which arise as the evaluations of some polynomial $P(X)$ of degree less than 2^{ℓ}, *with coefficients in* $A_{\iota,\kappa,\tau}$, over the domain S.

In Definition 2.3, we give meaning to the expression $P(x)$, for $x \in S$, by embedding $S \subset \mathcal{T}_{\iota+\kappa} \subset A_{\iota,\kappa,\tau}$ via the *synthetic* ring inclusion. We note that the extension code $\widehat{C} \subset A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}}}$ *also* has distance $d \coloneqq 2^{\ell+\mathcal{R}} - 2^{\ell} + 1$ (see [DP23b, Thm. 3.2]).

The following Schwartz–Zippel variant will prove useful below.

Lemma 2.4. *Fix a ϑ-variate polynomial $s(X_0, \ldots, X_{\vartheta-1}) \in A_{\iota,\kappa,\tau}[X_0, \ldots, X_{\vartheta-1}]$, of total degree at most d say, with coefficients in the tower algebra $A_{\iota,\kappa,\tau}$. If $s(X_0, \ldots, X_{\vartheta-1})$ is not the zero polynomial, then the locus $E \subset \mathcal{T}_{\tau}^{\vartheta}$ consisting of tuples $(r_0, \ldots, r_{\vartheta-1}) \in \mathcal{T}_{\tau}^{\vartheta}$ for which $s(r_0, \ldots, r_{\vartheta-1}) = 0$ is of mass $\mu(E) \leq \frac{d}{|\mathcal{T}_{\tau}|}$.*

Proof. We fix a $\mathcal{T}_{\tau}$-basis of $A_{\iota,\kappa,\tau}$ (for example, the monomial basis $1, Y_0, Y_1, \ldots, Y_0 \cdot \cdots Y_{\kappa-1}$ suffices). Writing each of s's coefficients in coordinates with respect to this basis, we obtain a collection of polynomials $(s_v(X_0, \ldots, X_{\vartheta-1}))_{v \in \mathcal{B}_{\kappa}}$, say, in $\mathcal{T}_{\tau}[X_0, \ldots, X_{\vartheta-1}]$—each likewise of total degree at most d—such that, for each input $(r_0, \ldots, r_{\vartheta-1}) \in \mathcal{T}_{\tau}^{\vartheta}$, $s(r_0, \ldots, r_{\vartheta-1}) = 0$ if and only if $s_v(r_0, \ldots, r_{\vartheta-1}) = 0$ holds for *each* $v \in \mathcal{B}_{\kappa}$. Moreover, our hypothesis whereby s is nonzero implies that at least one of these polynomials—say, $s_{v^*}(X_0, \ldots, X_{\vartheta-1})$—is not zero; we see that the standard Schwartz–Zippel lemma applies to s_{v^*}. Since s's vanishing locus in $\mathcal{T}_{\tau}^{\vartheta}$ is the intersection of those of the respective polynomials $(s_v(X_0, \ldots, X_{\vartheta-1}))_{v \in \mathcal{B}_{\kappa}}$, we conclude that the conclusion of the Schwartz–Zippel lemma applies equally to s. □

We note that, in Lemma 2.4, we evaluate $s(X_0, \ldots, X_{\vartheta-1})$ *only* on $\mathcal{T}_{\tau}$-valued inputs $(r_0, \ldots, r_{\vartheta-1}) \in \mathcal{T}_{\tau}^{\vartheta}$.

2.6 Proximity Gaps

We turn to proximity gaps, following [DP23a] and [DP23b]. As above, we fix a Reed–Solomon code $C \coloneqq \mathsf{RS}_{\mathcal{T}_{\iota+\kappa},S}[2^{\ell+\mathcal{R}}, 2^{\ell}]$; we moreover write $d \coloneqq 2^{\ell+\mathcal{R}} - 2^{\ell} + 1$ for C's distance, as well as $\widehat{C} \subset A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}}}$ for C's extension code. We record below the following specialization of a result of Diamond and Posen [DP23b, Thm. 3.10]. We fix a folding parameter ϑ. Below, we understand the matrix action of $\bigotimes_{i=0}^{\vartheta-1}(1 - r_i, r_i)$ via the *constant* vector space structure on $A_{\iota,\kappa,\tau}$.

Theorem 2.5. *Fix a proximity parameter $e \in \left\{0, \ldots, \left\lfloor \frac{d-1}{3} \right\rfloor\right\}$. If elements $u_0, \ldots, u_{2^{\vartheta}-1}$ of $A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}}}$ satisfy*

$$\Pr_{(r_0,\ldots,r_{\vartheta-1}) \in \mathcal{T}_{\tau}^{\vartheta}} \left[d\left(\left[\bigotimes_{i=0}^{\vartheta-1}(1 - r_i, r_i) \right] \cdot \begin{bmatrix} | & & | \\ u_0 & \cdots & | \\ | & & u_{2^{\vartheta}-1} \end{bmatrix}, \widehat{C} \right) \leq e \right] > 2 \cdot \vartheta \cdot \frac{e+1}{|\mathcal{T}_{\tau}|},$$

then $d^m\left((u_i)_{i=0}^{2^{\vartheta}-1}, \widehat{C}^{2^{\vartheta}}\right) \leq e$.

Proof. This is exactly the specialization of [DP23b, Thm. 3.10] to the Reed–Solomon code $C \subset \mathcal{T}_{\iota+\kappa}^{2^{\ell+\mathcal{R}}}$. □

It is an important open question to improve the proximity parameter range in Theorem 2.5's hypothesis to $e \in \left\{0, \ldots, \left\lfloor \frac{d-1}{2} \right\rfloor\right\}$. It is essentially the content of [DP23a, Thm. 3.1] that—for *any* linear code, and for any proximity parameter $e \in \left\{0, \ldots, \left\lfloor \frac{d-1}{2} \right\rfloor\right\}$—the case $\vartheta = 1$ of Theorem 2.5 implies the general case $\vartheta > 1$ (and this reduction goes through *even* for codes over algebras). Our question would thus follow from an algebra-theoretic variant of [DP23a, Conj. 2.4]. The conjecture [DP23a, Conj. 2.4] itself appears implicitly in Ames, Hazay, Ishai, and Venkitasubramaniam's *Ligero* [AHIV23, § 4.1.1]. In the Reed–Solomon setting in particular—and, crucially, over a field—Ben-Sasson et al. [Ben+23, Thm. 4.1] achieve a result essentially equivalent to [DP23a, Conj. 2.4] (albeit with a slightly worse false witness probability).

The prospect of adapting the proof technique of [Ben+23, Thm. 4.1] to the algebra setting is discussed explicitly in [DP23b, Rem. 3.18]. Unfortunately, while the proof of [DP23a, Thm. 2.1] (which treats only the range $e \in \{0, \ldots, \lfloor \frac{d-1}{3} \rfloor\}$) adapts almost immediately to the algebra setting, that of [Ben+23, Thm. 4.1] does not. Rather, that latter result invokes in a central way the algebraic properties of the code's coefficient field K, and relies heavily on unique factorization and cancellability in the univariate polynomial ring $K[Z]$. Despite expending significant effort, we were not able to adapt the proof of [Ben+23, Thm. 4.1] to the algebra setting; we pose that adaptation as a possible future avenue. On the other hand, the algebraic technicalities raised by that adaptation make plausible the prospect that a *direct* attack on the general conjecture [DP23a, Conj. 2.4]—rather than an adaptation of the Reed–Solomon-specific result [Ben+23, Thm. 4.1]—might most directly serve to advance our goal. Of course, again, this latter endeavor must moreover resolve [DP23a, Conj. 2.4] by means which adapt to the algebra setting (i.e., akin to those used by [DP23a, Thm. 2.1]).

We record the relevant conjecture as follows.

Conjecture 2.6. *We wonder whether Theorem 2.5 holds even for proximity parameters* $e \in \{0, \ldots, \lfloor \frac{d-1}{2} \rfloor\}$.

3 Polynomial Commitment Schemes

We now present our results. Following the approach of [DP23b, § 3], we present two schemes. Our first, presented in Subsection 3.4, is a binary-field adaptation of a technique due to Zeilberger, Chen and Fisch [ZCF23, § 5], which, moreover, introduces a multilinear folding technique. Our second scheme, given in Subsection 3.5, adjusts our first, in such a way as to make it support even tiny fields with no embedding overhead. That is, our second scheme is a "packed version" of our first.

3.1 Security Definitions and Notions

We record security definitions. Departing slightly from previous works, we treat polynomial commitment *in the IOP model*; that is, for our purposes, a "polynomial commitment scheme" is an IOP (i.e., a protocol in which a string oracle is available to both parties) which captures the commitment, and subsequent evaluation, of a polynomial. Our key security results asserts that a secure "IOPCS", upon being inlined into a secure PIOP, yields a secure IOP.

Our approach contrasts with that taken by previous works (we note e.g. Diamond and Posen [DP23b] and Setty [Set20]). These works opt to define polynomial commitment schemes in the *plain* (random oracle) model; these works then argue that a plain PCS, upon being inlined into a secure PIOP, yields a sound argument. Of course, this latter approach absorbs the Merklization process both into the PCS and into the composition theorem. Our approach bypasses this technicality, and separates the relevant concerns; indeed, upon bootstrapping a secure PIOP into a secure IOP (using our composition procedure), we may finally, by invoking generically the compiler of Ben-Sasson, Chiesa and Spooner [BCS16] from IOPs to secure arguments, obtain a secure argument.

We begin by defining various oracle models, following [DP23b].

Definition 3.1. An *IOP* $\Pi = (\mathcal{P}, \mathcal{V})$ is an interactive protocol in which the parties may freely use a certain *vector oracle*, which operates as follows, on the security parameter $\lambda \in \mathbb{N}$:

FUNCTIONALITY 3.2 (vector oracle)**.**
An alphabet A (allowed to depend on λ) is fixed.

- Upon receiving $(\texttt{submit}, A, m, f)$ from $\mathcal{P}$, where $m \in \mathbb{N}$ and $f \in A^{\mathcal{B}_m}$, output $(\texttt{receipt}, A, m, [f])$ to all parties, where $[f]$ is some unique handle onto the vector f.
- Upon receiving $(\texttt{query}, [f], v)$ from $\mathcal{V}$, where $v \in \mathcal{B}_m$, send $\mathcal{V}$ $(\texttt{result}, f(v))$.

Definition 3.3. A *polynomial IOP* $\Pi = (\mathcal{P}, \mathcal{V})$ is an interactive protocol in which the parties may freely use a certain *multilinear polynomial oracle*, which operates as follows, on the security parameter $\lambda \in \mathbb{N}$:

FUNCTIONALITY 3.4 (polynomial oracle)**.**
A field K and a field extension $L \,/\, K$ (allowed to depend on λ) are fixed.

- Upon receiving $(\texttt{submit}, K, \ell, t)$ from $\mathcal{P}$, where $\ell \in \mathbb{N}$ and $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, output $(\texttt{receipt}, K, \ell, [t])$ to all parties, where $[t]$ is some unique handle onto the polynomial t.
- On input $(\texttt{query}, [t], r)$ from $\mathcal{V}$, where $r \in L^\ell$, send $\mathcal{V}$ $(\texttt{result}, t(r_0, \ldots, r_{\ell-1}))$.

Definition 3.5. We say that the IOP or a polynomial IOP (as the case may be) $\Pi = (\mathcal{P}, \mathcal{V})$ is *secure* with respect to the relation R if, for each PPT adversary $\mathcal{A}$, there is an expected PPT emulator $\mathcal{E}$ and a negligible function negl, such that, for each security parameter $\lambda \in \mathbb{N}$ and each pair $(\mathbb{i}, \mathbb{x})$, provided that the protocol is run on the security parameter λ, writing $\mathbb{w} \leftarrow \mathcal{E}^{\mathcal{A}}(\mathbb{i}, \mathbb{x})$, we have $|\Pr[\langle \mathcal{A}(\mathbb{i}, \mathbb{x}), \mathcal{V}(\mathbb{i}, \mathbb{x})\rangle = 1] - \Pr[R(\mathbb{i}, \mathbb{x}, \mathbb{w}) = 1]| \leq \mathsf{negl}(\lambda)$.

We informally interpret the above definitions in the following way. In Definition 3.1, the oracle queries at a verifier-supplied point *in the cube* a prover-supplied mapping defined on the cube; in Definition 3.3, the oracle queries at an *arbitrary* verifier-supplied point the *multilinear extension* of just such a mapping. The difference between these two models, perhaps superficially minor, is in fact enormous; the primary purpose of this paper is to construct various *IOPCS*s—see Definition 3.6 below—each of which, by definition, serves to bootstrap a vector oracle into a polynomial oracle (and consequently, a PIOP into an IOP).

Definition 3.6. A *interactive oracle polynomial commitment scheme* (IOPCS) is a tuple of algorithms $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$, each allowed access to the vector oracle, with the following syntax:

- $\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \ell, K)$. On input a number-of-variables parameter ℓ and a field K, outputs params, which includes, among other things, a field extension $L \,/\, K$.
- $[f] \leftarrow \Pi.\mathsf{Commit}(\mathsf{params}, t)$. On input a multilinear polynomial $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, outputs a handle $[f]$ to a vector.
- $b \leftarrow \langle \mathcal{P}([f], s, r; t), \mathcal{V}([f], s, r)\rangle$ is an IOP, with common input a vector handle $[f]$, an evaluation point $(r_0, \ldots, r_{\ell-1}) \in L^\ell$, and a claimed evaluation $s \in L$, where $\mathcal{P}$ has as further input a multilinear polynomial $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, and where $\mathcal{V}$ outputs a success bit b.

In order to ensure that elements $t(X_0, \ldots, X_{\ell-1})$ of $K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$ are representable using polynomially many bits, as well as that the opening IOP $(\mathcal{P}, \mathcal{V})$ is efficient for both parties, we impose without further comment the mild assumption whereby both $\log(|K|)$ and $\log(|L|)$ grow polynomially in λ.

The IOPCS Π is *complete* if the obvious correctness property holds. That is, for each multilinear polynomial $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$ and each honestly generated commitment $[f] \leftarrow \Pi.\mathsf{Commit}(\mathsf{params}, t)$, it should hold that, for each $r \in L^\ell$, setting $s \coloneqq t(r_0, \ldots, r_{\ell-1})$, the honest prover algorithm induces the verifier to accept with probability 1, so that $\langle \mathcal{P}([f], s, r; t), \mathcal{V}([f], s, r)\rangle = 1$.

We now define the security of IOPCSs.

Definition 3.7. For each interactive oracle polynomial commitment scheme Π, security parameter λ, values ℓ and K, PPT query sampler $\mathcal{Q}$, PPT adversary $\mathcal{A}$, and PPT emulator $\mathcal{E}$, we define the following experiment:

- The experimenter samples $\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \ell, K)$, and gives params to $\mathcal{A}$ and $\mathcal{E}$.
- The adversary, after interacting arbitrarily with the vector oracle, outputs a handle $[f] \leftarrow \mathcal{A}(\mathsf{params})$.
- On input $\mathcal{A}$'s record of interactions with the oracle, $\mathcal{E}$ outputs $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$.
- The query sampler outputs $(r_0, \ldots, r_{\ell-1}) \leftarrow \mathcal{Q}(\mathsf{params})$; $\mathcal{A}$ responds with an evaluation claim $s \leftarrow \mathcal{A}(r)$.
- The experimenter defines the following two random bits:
  - By running the evaluation IOP with $\mathcal{A}$ as $\mathcal{V}$, obtain the bit $b \leftarrow \langle \mathcal{A}(s, r), \mathcal{V}([f], s, r)\rangle$.
  - Obtain the further bit $b' \coloneqq t(r_0, \ldots, r_{\ell-1}) \stackrel{?}{=} s$.

The IOPCS Π is *secure* if, for each PPT adversary $\mathcal{A}$, there is a PPT emulator $\mathcal{E}$ and a negligible function negl such that, for each $\lambda \in \mathbb{N}$, each ℓ and K, and each PPT query sampler $\mathcal{Q}$, $\Pr[b = 1 \wedge b' = 0] \leq \mathsf{negl}(\lambda)$.

3.2 Using FRI in Novel Polynomial Basis

We begin by proposing a *specific* construction of those subspace polynomials $q^{(0)}, \ldots, q^{(\ell-1)}$ invoked internally by FRI. They key is to do so in such a way as to induce compatibility with the novel polynomial basis.

Throughout this section, we fix a binary field K, with $\mathbb{F}_2$-basis $(\beta_0, \ldots, \beta_{r-1})$, say, as well as a size parameter $\ell \in \{0, \ldots, r-1\}$ and a rate parameter $\mathcal{R} \in \{1, \ldots, r-\ell\}$. We finally recall the (non-normalized) subspace vanishing polynomials $W_i(X) \in K[X]$, for $i \in \{0, \ldots, \ell-1\}$, which we now view as $\mathbb{F}_2$-*linear maps* $W_i : K \to K$ (see Subsection 2.3).

Definition 3.8. We initialize $S^{(0)} \coloneqq \langle \beta_0, \ldots, \beta_{\ell+\mathcal{R}-1} \rangle$. Moreover, for each $i \in \{0, \ldots, \ell-1\}$, we set

$$q^{(i)} \coloneqq \frac{W_i(\beta_i)^2}{W_{i+1}(\beta_{i+1})} \cdot X \cdot (X+1),$$

as well as, inductively, $S^{(i+1)} \coloneqq \mathrm{im}\big(q^{(i)}\big|_{S^{(i)}}\big)$.

The following lemma demonstrates that this construction fulfills the template demanded by (1).

Lemma 3.9. *For each $i \in \{0, \ldots, \ell-1\}$, $\ker\big(q^{(i)}\big) \subset S^{(i)}$ holds.*

Proof. We note that, trivially, $\ker\big(q^{(i)}\big) = \{0, 1\}$ for each $i \in \{0, \ldots, \ell-1\}$. For each $i \in \{0, \ldots, \ell-1\}$, we claim in fact that the inductive invariant $S^{(i)} = \mathrm{im}\Big(\widehat{W}_i\Big|_{S^{(0)}}\Big)$ holds. Assuming this invariant, the conclusion of the lemma certainly follows; indeed, we see immediately that $1 = \widehat{W}_i(\beta_i)$, while of course $\beta_i \in S^{(0)}$.

It thus suffices to argue that the inductive invariant holds throughout. In the base case $i = 0$, the claim is a triviality, since $\widehat{W}_0(X) = X$ is the identity. We thus fix an index $i \in \{0, \ldots, \ell-1\}$, and show that the assignment $S^{(i+1)} \coloneqq \mathrm{im}\big(q^{(i)}\big|_{S^{(i)}}\big)$ preserves the inductive invariant; in other words, we must show that $S^{(i+1)} \coloneqq \mathrm{im}\big(q^{(i)}\big|_{S^{(i)}}\big) \stackrel{?}{=} \mathrm{im}\Big(\widehat{W}_{i+1}\Big|_{S^{(0)}}\Big)$. Unrolling the assumed inductive invariant on this equality's left-hand side, we reduce it in turn to the equality $\mathrm{im}\Big(q^{(i)} \circ \widehat{W}_i\Big|_{S^{(0)}}\Big) \stackrel{?}{=} \mathrm{im}\Big(\widehat{W}_{i+1}\Big|_{S^{(0)}}\Big)$. This latter equality itself follows from the following direct calculation:

$$\begin{aligned}
\Big(q^{(i)} \circ \widehat{W}_i\Big)(X) &= \frac{W_i(\beta_i)^2}{W_{i+1}(\beta_i)} \cdot \widehat{W}_i(X) \cdot \Big(\widehat{W}_i(X) + 1\Big) && \text{(by definition of } q^{(i)}.) \\
&= \frac{W_i(\beta_i)^2}{W_{i+1}(\beta_{i+1})} \cdot \frac{W_i(X)}{W_i(\beta_i)} \cdot \frac{W_i(X) + W_i(\beta_i)}{W_i(\beta_i)} && \text{(by definition of } \widehat{W}_i.) \\
&= \frac{W_i(X) \cdot (W_i(X) + W_i(\beta_i))}{W_{i+1}(\beta_{i+1})} && \text{(cancellation of } W_i(\beta_i)^2.) \\
&= \frac{W_{i+1}(X)}{W_{i+1}(\beta_{i+1})} && \text{(recursive characterization of } W_{i+1}(X).) \\
&= \widehat{W}_{i+1}(X) && \text{(by definition of } \widehat{W}_{i+1}(X).)
\end{aligned}$$

in the second-to-last step, we exploit the recursive identity $W_{i+1}(X) = W_i(X) \cdot (W_i(X) + W_i(\beta_i))$, itself a basic consequence of the definitions of W_{i+1} and W_i and of the linearity of W_i. □

Lemma 3.9 shows that the maps $q^{(0)}, \ldots, q^{(\ell-1)}$ and the spaces $S^{(0)}, \ldots, S^{(\ell)}$ yield a *valid* global parameterization, suitable for use in FRI.

We extract and state separately a few corollaries of the proof of Lemma 3.9.

Corollary 3.10. *For each $i \in \{0, \ldots, \ell\}$, $S^{(i)} = \mathrm{im}\Big(\widehat{W}_i\Big|_{S^{(0)}}\Big)$.*

Proof. This fact is shown explicitly in the course of Lemma 3.9. □

As a further side effect, Lemma 3.9 shows that the polynomials $q^{(0)}, \ldots, q^{(\ell-1)}$ collectively "factor" the normalized subspace polynomials $\widehat{W}_0, \ldots, \widehat{W}_{\ell-1}$, in the following sense:

Corollary 3.11. *For each $i \in \{0, \ldots, \ell\}$, $\widehat{W}_i = q^{(i-1)} \circ \cdots \circ q^{(0)}$.*

Proof. This fact admits a simple inductive proof. In the base case $i = 0$, there's nothing to prove (the empty composition is the identity). Letting $i \in \{0, \ldots, \ell - 1\}$ be arbitrary, the proof of Lemma 3.9 shows that $\widehat{W}_{i+1} = q^{(i)} \circ \widehat{W}_i$. Applying induction, we conclude that this latter map in turn equals $q^{(i)} \circ \cdots \circ q^{(0)}$. □

We note finally the following result.

Corollary 3.12. *For each $i \in \{0, \ldots, \ell\}$, the set $\left(\widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{\ell+\mathcal{R}-1})\right)$ is an $\mathbb{F}_2$-basis of the space $S^{(i)}$.*

Proof. Indeed, the subspace $V_i \coloneqq \langle \beta_i, \ldots, \beta_{\ell+\mathcal{R}-1} \rangle$ clearly satisfies $V_i \subset S^{(0)}$, so that $\widehat{W}_i(V_i) \subset \widehat{W}_i(S^{(0)})$, which itself equals $S^{(i)}$ (by Corollary 3.10). On the other hand, the restriction of $\widehat{W}_i$ to V_i is necessarily injective, since $\widehat{W}_i$'s kernel $\langle \beta_0, \ldots, \beta_{i-1} \rangle$ intersects V_i trivially. Since $S^{(i)}$ is $\ell + \mathcal{R} - i$-dimensional, we conclude by a dimension count that $\left(\widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{\ell+\mathcal{R}-1})\right)$ spans $S^{(i)}$. □

The bases $\left\langle \widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{\ell+\mathcal{R}-1}) \right\rangle = S^{(i)}$, for $i \in \{0, \ldots, \ell\}$, allow us to simplify various aspects of our protocol's implementation. For example, expressed in coordinates with respect to these bases, each map $q^{(i)} : S^{(i)} \to S^{(i+1)}$ acts simply by projecting away its 0^{th}-indexed component (indeed, for each $i \in \{0, \ldots, \ell\}$, $q^{(i)}$ maps the basis $(\widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{\ell+\mathcal{R}-1}))$ to $(0, \widehat{W}_{i+1}(\beta_{i+1}), \ldots, \widehat{W}_{i+1}(\beta_{\ell+\mathcal{R}-1}))$). Similarly, for each $i \in \{0, \ldots, \ell - 1\}$ and each $y \in S^{(i+1)}$, the two K-elements $x \in S^{(i)}$ for which $q^{(i)}(x) = y$ differ precisely at their 0^{th} components, and elsewhere agree with y's coordinate representation. Below, we often identify $S^{(i)} \cong \mathcal{B}_{\ell+\mathcal{R}-i}$ as sets, using these bases; moreover, where possible, we eliminate altogether the maps $q^{(0)}, \ldots, q^{(\ell-1)}$ from our descriptions. These measures make our protocol's description (and in particular, its implementation) more transparent.

3.3 FRI Folding, Revisited

We now introduce a new FRI-like folding mechanism. We recall that FRI [BBHR18, § 3.2] makes use of a folding arity constant η; FRI stipulates that, to fold a given oracle, the prover interpolate a *univariate* polynomial of degree less than 2^η on each coset of the given oracle, and finally evaluate the resulting polynomials collectively at the verifier's challenge point. We introduce a new, *multilinear* folding mechanism as follows. Informally, we stipulate that the verifier send a fixed and positive—and yet arbitrary—number ϑ of folding challenges, and that the prover fold its oracle, again coset-wise, using a length-2^ϑ *tensor combination* (in the sense of Subsection 2.1) of the verifier's challenges over each coset. Below, we write $L \,/\, K$ for a field extension.

Definition 3.13. We fix an index $i \in \{0, \ldots, \ell - 1\}$ and a map $f^{(i)} : S^{(i)} \to L$. For each $r \in L$, we define the map $\mathsf{fold}\big(f^{(i)}, r\big) : S^{(i+1)} \to L$ by setting, for each $y \in S^{(i+1)}$:

$$\mathsf{fold}\Big(f^{(i)}, r\Big) : y \mapsto \begin{bmatrix} 1 - r & r \end{bmatrix} \cdot \begin{bmatrix} x_1 & -x_0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} f^{(i)}(x_0) \\ f^{(i)}(x_1) \end{bmatrix},$$

where we write $(x_0, x_1) \coloneqq {q^{(i)}}^{-1}(\{y\})$ for the fiber of $q^{(i)}$ over $y \in S^{(i)}$.

Remark 3.14. Definition 3.13's quantity $\mathsf{fold}\big(f^{(i)}, r\big)(y)$ is closely related—and yet *not* equivalent—to FRI's expression $\mathsf{interpolant}\Big(f^{(i)}\big|_{{q^{(i)}}^{-1}(\{y\})}\Big)(r)$. (FRI's variant, however, admits a similar matrix expression.) The essential point is that FRI's variant induces a *monomial* fold, as opposed to a Lagrange fold; that is, if we were to use FRI's variant instead of our own, then our Lemma 3.21 below would remain true, albeit with the alternate conclusion $P^{(i+1)}(X) = \sum_{j=0}^{2^{\ell-i-1}-1} (a_{2j} + r'_i \cdot a_{2j+1}) \cdot X_j^{(i+1)}(X)$. Our entire theory admits a parallel variant in this latter setting, though that variant introduces further complications.

We finally record the following iterated extension of Definition 3.14.

Definition 3.15. We fix a positive folding factor ϑ, an index $i \in \{0, \ldots, \ell - \vartheta\}$, and a map $f^{(i)} : S^{(i)} \to L$. For each tuple $(r_0, \ldots, r_{\vartheta-1}) \in L^\vartheta$, we abbreviate $\mathsf{fold}\big(f^{(i)}, r_0, \ldots, r_{\vartheta-1}\big) \coloneqq \mathsf{fold}\big(\cdots \mathsf{fold}\big(f^{(i)}, r_0\big), \cdots, r_{\vartheta-1}\big)$.

We have the following mathematical characterization of this iterated folding operation:

Lemma 3.16. *For each positive folding factor ϑ, each index $i \in \{0, \ldots, \ell - \vartheta\}$, and each $y \in S^{(i+\vartheta)}$, there is a $2^\vartheta \times 2^\vartheta$ invertible matrix M_y with entries in K, which depends only on $y \in S^{(i+\vartheta)}$, such that, for each map $f^{(i)} : S^{(i)} \to L$ and each tuple $(r_0 \ldots, r_{\vartheta-1}) \in L^\vartheta$ of folding challenges, we have the matrix representation:*

$$\mathsf{fold}\Big(f^{(i)}, r_0, \ldots, r_{\vartheta-1}\Big)(y) = \begin{bmatrix} \bigotimes_{j=0}^{\vartheta-1}(1 - r_j, r_j) \end{bmatrix} \cdot \begin{bmatrix} \\ M_y \\ \\ \end{bmatrix} \cdot \begin{bmatrix} f^{(i)}(x_0) \\ \vdots \\ f^{(i)}(x_{2^\vartheta-1}) \end{bmatrix},$$

where the right-hand vector's values $(x_0, \ldots, x_{2^\vartheta-1})$ represent the fiber $\big(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{y\}) \subset S^{(i)}$.

Proof. We prove the result by induction on ϑ. In the base case $\vartheta = 1$, the claim is a tautology, in view of Definition 3.14. We note that that definition's matrix $\begin{bmatrix} x_1 & -x_0 \\ -1 & 1 \end{bmatrix}$ is invertible, since its determinant $x_1 - x_0$ is nonzero (and in fact equals 1, a fact we shall use below).

We thus fix a folding factor $\vartheta > 1$, and suppose that the claim holds for $\vartheta - 1$. We write $(z_0, z_1) \coloneqq q^{(i+\vartheta-1)^{-1}}(\{y\})$, as well as $(x_0, \ldots, x_{2^\vartheta-1}) \coloneqq \big(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{y\})$. Unwinding Definition 3.15, we recursively express the relevant quantity $\mathsf{fold}\big(f^{(i)}, r_0, \ldots, r_{\vartheta-1}\big)(y)$—which, for typographical reasons, we call $\mathfrak{f}$—in the following way:

$$\begin{aligned}
\mathfrak{f} &= \begin{bmatrix} 1 - r_{\vartheta-1} & r_{\vartheta-1} \end{bmatrix} \cdot \begin{bmatrix} z_1 & -z_0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathsf{fold}\big(f^{(i)}, r_0, \ldots, r_{\vartheta-2}\big)(z_0) \\ \mathsf{fold}\big(f^{(i)}, r_0, \ldots, r_{\vartheta-2}\big)(z_1) \end{bmatrix} \\
&= \begin{bmatrix} 1 - r_{\vartheta-1} & r_{\vartheta-1} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} z_1 & -z_0 \\ -1 & 1 \end{bmatrix} \cdot \left[\begin{array}{c|c} \bigotimes_{j=0}^{\vartheta-2}(1 - r_j, r_j) & \\ \hline & \bigotimes_{j=0}^{\vartheta-2}(1 - r_j, r_j) \end{array}\right]}_{\text{these matrices may be interchanged.}} \cdot \left[\begin{array}{c|c} M_{z_0} & \\ \hline & M_{z_1} \end{array}\right] \cdot \begin{bmatrix} f^{(i)}(x_0) \\ \vdots \\ f^{(i)}(x_{2^\vartheta-1}) \end{bmatrix}.
\end{aligned}$$

In the second step above, we apply the inductive hypothesis on both z_0 and z_1. That hypothesis furnishes the nonsingular, $2^{\vartheta-1} \times 2^{\vartheta-1}$ matrices M_{z_0} and M_{z_1}; we note moreover that the union of the fibers $\big(q^{(i+\vartheta-2)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{z_0\})$ and $\big(q^{(i+\vartheta-2)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{z_1\})$ is precisely $\big(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{y\})$. Interchanging the two matrices bracketed above, we further reëxpress this quantity as:

$$= \begin{bmatrix} 1 - r_{\vartheta-1} & r_{\vartheta-1} \end{bmatrix} \cdot \left[\begin{array}{c|c} \bigotimes_{j=0}^{\vartheta-2}(1 - r_j, r_j) & \\ \hline & \bigotimes_{j=0}^{\vartheta-2}(1 - r_j, r_j) \end{array}\right] \cdot \left[\begin{array}{c|c} \mathsf{diag}(z_1) & \mathsf{diag}(-z_0) \\ \hline \mathsf{diag}(-1) & \mathsf{diag}(1) \end{array}\right] \cdot \left[\begin{array}{c|c} M_{z_0} & \\ \hline & M_{z_1} \end{array}\right] \cdot \begin{bmatrix} f^{(i)}(x_0) \\ \vdots \\ f^{(i)}(x_{2^\vartheta-1}) \end{bmatrix}.$$

By the standard recursive substructure of the tensor product, the product of the left-hand two matrices equals exactly $\bigotimes_{j=0}^{\vartheta-1}(1 - r_j, r_j)$. On the other hand, the product of the two $2^\vartheta \times 2^\vartheta$ nonsingular matrices above is itself nonsingular, and supplies the required $2^\vartheta \times 2^\vartheta$ matrix M_y. □

We emphasize that, in Lemma 3.16, the matrix M_y depends *only* on $y \in S^{(i+\vartheta)}$—and of course on ϑ and $i \in \{0, \ldots, \ell - \vartheta\}$—but *not* on the map $f^{(i)}$ or the folding challenges $(r_0, \ldots, r_{\vartheta-1}) \in L^\vartheta$.

Remark 3.17. Interestingly, the matrix M_y of Lemma 3.16 is nothing other than that of the *inverse* additive NTT [LCH14, § III. C.] on the coset $(x_0, \ldots, x_{2^\vartheta-1})$; i.e., it's the matrix which, given the evaluations of some polynomial of degree less than 2^ϑ on $(x_0, \ldots, x_{2^\vartheta-1})$, computes the coefficients, with respect to the i^{th}-order novel basis (see Remark 3.22 below), of that polynomial. We currently lack a clean explanation of this fact.

Remark 3.18. For each given map $f^{(i)} : S^{(i)} \to L$—expressed as a table of values, via the identification $S^{(i)} \cong \mathcal{B}_{\ell+\mathcal{R}-i}$, say—the table of values of $\mathsf{fold}\big(f^{(i)}, r_0, \ldots, r_{\vartheta-1}\big) : S^{(i+\vartheta)} \to L$ may be computed efficiently, given the tuple $(r_0, \ldots, r_{\vartheta-1})$. Indeed, Definitions 3.13 and 3.15 directly suggests a ϑ-pass, $\Theta(|S^{(i)}|)$-time algorithm for this task. Lemma 3.16 is not interesting algorithmically, but rather mathematically; indeed, it appears repeatedly in our security proofs below (see Theorem 3.27, and in particular Proposition 3.30).

3.4 Our Protocol

We begin by introducing our simple small-field IOPCS, which doesn't use packing. In order to present a notationally simpler version of our protocol, we assume below that $\vartheta \mid \ell$; this requirement is not necessary.

CONSTRUCTION 3.19 (Simple IOPCS)**.**
We define $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ as follows.

- $\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \ell, K)$. On input 1^λ, ℓ, and K, where the degree r (say) of K over $\mathbb{F}_2$ is more than ℓ, return an extension field $L \,/\, K$ for which $|L| \geq 2^{\omega(\log \lambda)}$, a constant Reed–Solomon rate parameter $\mathcal{R} \in \{1, \ldots, r - \ell\}$, a folding factor $\vartheta \mid \ell$, and a repetition parameter $\gamma = \omega(\log(\lambda))$. Fix an arbitrary $\mathbb{F}_2$-basis $(\beta_0, \ldots, \beta_{r-1})$ of K; writing $(X_0(X), \ldots, X_{2^\ell-1}(X))$ for the resulting novel K-basis of $K[X]^{\prec 2^\ell}$, fix the domains $S^{(0)}, \ldots, S^{(\ell)}$ and the polynomials $q^{(0)}, \ldots, q^{(\ell-1)}$ as prescribed by Subsection 3.2. Write $C^{(0)} \subset K^{2^{\ell+\mathcal{R}}}$ for the Reed–Solomon code $\mathsf{RS}_{K,S^{(0)}}[2^{\ell+\mathcal{R}}, 2^\ell]$.
- $[f] \leftarrow \Pi.\mathsf{Commit}(\mathsf{params}, t)$. On input $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, use t's Lagrange coefficients $(t(v))_{v \in \mathcal{B}_\ell}$ as the coefficients, in the novel polynomial basis, of a univariate polynomial $P(X) \coloneqq \sum_{v \in \mathcal{B}_\ell} t(v) \cdot X_{\{v\}}(X)$, say. Using Algorithm 2, compute the Reed–Solomon codeword $f : S^{(0)} \to K$ defined by $f : x \mapsto P(x)$. Submit $(\mathtt{submit}, K, \ell + \mathcal{R}, f)$ to the vector oracle. Upon receiving $(\mathtt{receipt}, K, \ell + \mathcal{R}, [f])$ from the oracle, output the vector handle $[f]$.

We define $(\mathcal{P}, \mathcal{V})$ as the following IOP, in which both parties have the common input $[f]$, $s \in L$, and $(r_0, \ldots, r_{\ell-1}) \in L^\ell$, and $\mathcal{P}$ has the further input $t(X_0, \ldots, X_{\ell-1}) \in K[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$.

- $\mathcal{P}$ writes $h(X_0, \ldots, X_{\ell-1}) \coloneqq t(X_0, \ldots, X_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, X_0, \ldots, X_{\ell-1})$.
- $\mathcal{P}$ and $\mathcal{V}$ both abbreviate $f^{(0)} \coloneqq f$ and $s_0 \coloneqq s$, and execute the following loop:
  1: **for** $i \in \{0, \ldots, \ell - 1\}$ **do**
  2: $\quad\mathcal{P}$ sends $\mathcal{V}$ the univariate polynomial $h_i(X) \coloneqq \sum_{v \in \mathcal{B}_{\ell-i-1}} h(r'_0, \ldots, r'_{i-1}, X, v_0, \ldots, v_{\ell-i-2})$.
  3: $\quad\mathcal{V}$ requires $s_i \overset{?}{=} h_i(0) + h_i(1)$. $\mathcal{V}$ samples $r'_i \leftarrow L$, sets $s_{i+1} \coloneqq h_i(r'_i)$, and sends $\mathcal{P}$ r'_i.
  4: $\quad\mathcal{P}$ defines $f^{(i+1)} : S^{(i+1)} \to L$ as the function $\mathsf{fold}\big(f^{(i)}, r'_i\big)$ of Definition 3.13.
  5: $\quad$**if** $i + 1 = \ell$ **then** $\mathcal{P}$ sends $c \coloneqq f^{(\ell)}(0, \ldots, 0)$ to $\mathcal{V}$.
  6: $\quad$**else if** $\vartheta \mid i + 1$ **then** $\mathcal{P}$ submits $(\mathtt{submit}, L, \ell + \mathcal{R} - i - 1, f^{(i+1)})$ to the oracle.
- $\mathcal{V}$ requires $s_\ell \overset{?}{=} c \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-1})$.
- $\mathcal{V}$ assigns $c_\ell \coloneqq c$, and executes the following querying procedure:
  1: **for** γ repetitions **do**
  2: $\quad\mathcal{V}$ samples $u \leftarrow \mathcal{B}_\mathcal{R}$ randomly.
  3: $\quad$**for** $i \in \{\ell - \vartheta, \ell - 2 \cdot \vartheta, \ldots, 0\}$ (i.e., in downward order, taking ϑ-sized steps) **do**
  4: $\quad\quad$for each $v \in \mathcal{B}_\vartheta$, $\mathcal{V}$ submits $(\mathtt{query}, [f^{(i)}], v \parallel u)$ to the vector oracle.
  5: $\quad\quad\mathcal{V}$ requires $c_{i+\vartheta} \overset{?}{=} \mathsf{fold}\big(f^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)(u)$.
  6: $\quad\quad\mathcal{V}$ samples $v \leftarrow \mathcal{B}_\vartheta$, sets $c_i \coloneqq f^{(i)}(v \parallel u)$, and overwrites $u \coloneqq v \parallel u$.

In our commitment procedure above, we give meaning to the commitment of f by implicitly identifying $S^{(0)} \cong \mathcal{B}_{\ell+\mathcal{R}}$ as sets (as discussed above); similarly, in the prover's line 6 above, we identify $\mathcal{B}_{\ell+\mathcal{R}-i-1} \cong S^{(i+1)}$. Conversely, in line 5 of the verifier's querying procedure above, the verifier must implicitly identify the $\mathcal{B}_{\ell+\mathcal{R}-i-\vartheta}$-element u with an $S^{(i+\vartheta)}$-element—and the $\mathcal{B}_{\ell+\mathcal{R}-i}$-elements $(v \parallel u)_{v \in \mathcal{B}_\vartheta}$ with $S^{(i)}$-elements—in order to appropriately apply Definition 3.15. We note that, in line 5, $\mathcal{V}$ has precisely the information it requires in order to compute $\mathsf{fold}\big(f^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)(u)$ (namely, the values of $f^{(i)}$ on the fiber $(v \parallel u)_{v \in \mathcal{B}_\vartheta} \cong \big(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\big)^{-1}(\{u\})$).

The completness of Construction 3.19's evaluation IOP is not straightforward. For instance, it is simply not obvious what the folding operation of line 4 does to the *coefficients* of the low-degree polynomial $P^{(i)}(X)$ underlying $f^{(i)}$. (Though our folding operation departs slightly from FRI's—we refer to Remark 3.14 for

a discussion of this fact—the conceptual obstacle is essentially the same.) Indeed, the completeness proof of generic FRI [BBHR18, § 4.1.1] tells us that the folded function $f^{(i+1)}$ represents the evaluations of *some* polynomial $P^{(i+1)}(X)$ of appropriate degree on the domain $S^{(i+1)}$. But which one? The proof of [BBHR18, § 4.1.1] fails to *constructively* answer this question, in that it invokes the generic characteristics of the multivariate reduction—called $Q^{(i)}(X,Y)$—of $P^{(i)}(X)$ by $Y - q^{(i)}(X)$. (We refer to e.g. von zur Gathen and Gerhard [GG13, Alg. 21.11] for a thorough treatment of multivariate division.) It seems simply infeasible to analyze by hand the execution of the multivariate division algorithm with sufficient fidelity as to determine with any precision the result $P^{(i+1)}(Y) = Q^{(i)}(r'_i, Y)$ (though we don't rule out that a proof could in principle be achieved by this means).

Instead, we introduce certain, carefully-selected K-bases of the spaces $K[X]^{\prec 2^{\ell-i}}$, for $i \in \{0,\ldots,\ell\}$ (essentially, "higher-order" variants of the novel polynomial basis). As it turns out, the respective coefficients of $P^{(i)}(X)$ and $P^{(i+1)}(X)$ *with respect to these bases* are tractably related; in fact, their relationship amounts to an even–odd tensor-fold by the FRI challenge r'_i. Proceeding by induction, we obtain the desired characterization of c.

Theorem 3.20. *The IOPCS* $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ *of Construction 3.19 is complete.*

Proof. We fix $\mathcal{P}$'s input $t(X_0,\ldots,X_{\ell-1}) \in K[X_0,\ldots,X_{\ell-1}]^{\preceq 1}$, given in Lagrange coefficients as $\sum_{v\in\mathcal{B}_\ell} t(v)\cdot \widetilde{\mathsf{eq}}(X_0,\ldots,X_{\ell-1},v_0,\ldots,v_{\ell-1})$. Our primary task is to argue that, provided $\mathcal{P}$ follows the protocol, its final FRI message c is nothing other than $t(r'_0,\ldots,r'_{\ell-1})$.

We introduce a family of further polynomial bases. For each $i \in \{0,\ldots,\ell-1\}$, we define the *$i^{th}$-order subspace vanishing polynomials* $\widehat{W}_0^{(i)},\ldots,\widehat{W}_{\ell-i-1}^{(i)}$ as the polynomials $X, q^{(i)}, q^{(i+1)}\circ q^{(i)},\ldots,q^{(\ell-2)}\circ\cdots\circ q^{(i)}$, respectively (that is, $\widehat{W}_k^{(i)} \coloneqq q^{(i+k-1)}\circ\cdots\circ q^{(i)}$, for each $k\in\{0,\ldots,\ell-i-1\}$). Finally, we define the *i^{th}-order novel polynomial basis* by setting $X_j^{(i)} \coloneqq \prod_{k=0}^{\ell-i-1}\widehat{W}_k^{(i)^{j_k}}$, for each $j\in\{0,\ldots,2^{\ell-i}-1\}$ (here, again, we write $(j_0,\ldots,j_{\ell-i-1})$ for the bits of j). We adopt the notational convention whereby the ℓ^{th} basis consists simply of the constant polynomial $X_0^{(\ell)}(X) = 1$. Our proof below relies on the following inductive relationship between the bases $\left(X_j^{(i)}(X)\right)_{j=0}^{2^{\ell-i}-1}$ and $\left(X_j^{(i+1)}(X)\right)_{j=0}^{2^{\ell-i-1}-1}$. Indeed, for each $j\in\{0,\ldots,2^{\ell-i-1}-1\}$, the polynomials $X_{2j}^{(i)}(X)$ and $X_{2j+1}^{(i)}(X)$ are precisely $X_j^{(i+1)}\big(q^{(i)}(X)\big)$ and $X\cdot X_j^{(i+1)}\big(q^{(i)}(X)\big)$, respectively.

We now pose the following inductive claim:

Lemma 3.21. *Fix an index $i\in\{0,\ldots,\ell-1\}$. If $f^{(i)}: S^{(i)}\to L$ is exactly the evaluation over $S^{(i)}$ of the polynomial $P^{(i)}(X) = \sum_{j=0}^{2^{\ell-i}-1} a_j\cdot X_j^{(i)}(X)$, then, under honest prover behavior, $f^{(i+1)}: S^{(i+1)}\to L$ is exactly the evaluation over $S^{(i+1)}$ of the polynomial $P^{(i+1)}(X) = \sum_{j=0}^{2^{\ell-i-1}-1}((1-r'_i)\cdot a_{2j} + r'_i\cdot a_{2j+1})\cdot X_j^{(i+1)}(X)$.*

Proof. Given $P^{(i)}(X)$ as in the hypothesis of the lemma, we introduce the *even and odd refinements* $P_0^{(i+1)}(X) \coloneqq \sum_{j=0}^{2^{\ell-i-1}-1} a_{2j}\cdot X_j^{(i+1)}(X)$ and $P_1^{(i+1)}(X) \coloneqq \sum_{j=0}^{2^{\ell-i-1}-1} a_{2j+1}\cdot X_j^{(i+1)}(X)$ of $P^{(i)}(X)$. We note the following key polynomial identity:

$$P^{(i)}(X) = P_0^{(i+1)}(q^{(i)}(X)) + X\cdot P_1^{(i+1)}(q^{(i)}(X)); \tag{2}$$

This identity is a direct consequence of the definitions of the higher-order novel polynomial bases.

We turn to the proof of the lemma. We claim that $f^{(i+1)}(y) = P^{(i+1)}(y)$ holds for each $y\in S^{(i+1)}$. To this end, we let $y\in S^{(i+1)}$ be arbitrary; we moreover write $(x_0,x_1) \coloneqq {q^{(i)}}^{-1}(\{y\})$ for the fiber of $q^{(i)}$ over y. We begin by examining the values $P^{(i)}(x_0)$ and $P^{(i)}(x_1)$. For each $b\in\{0,1\}$ we have:

$$\begin{aligned}
P^{(i)}(x_b) &= \sum_{j=0}^{2^{\ell-i}-1} a_j\cdot X_j^{(i)}(x_b) && \text{(by definition of } P^{(i)}.)\\
&= \sum_{j=0}^{2^{\ell-i-1}-1} a_{2j}\cdot X_j^{(i+1)}\Big(q^{(i)}(x_b)\Big) + x_b\cdot\sum_{j=0}^{2^{\ell-i-1}-1} a_{2j+1}\cdot X_j^{(i+1)}\Big(q^{(i)}(x_b)\Big) && \text{(by the identity (2).)}\\
&= P_0^{(i+1)}(y) + x_b\cdot P_1^{(i+1)}(y). && \text{(using } q^{(i)}(x_b)=y \text{ and the definitions of } P_0^{(i+1)} \text{ and } P_1^{(i+1)}.)
\end{aligned}$$

Using now our assumption whereby $f^{(i)}(x_b) = P^{(i)}(x_b)$ for each $b \in \{0,1\}$, and unwinding the prescription of Definition 3.13, we obtain:

$$\begin{aligned}
f^{(i+1)}(y) &= \begin{bmatrix} 1-r'_i & r'_i \end{bmatrix} \cdot \begin{bmatrix} x_1 & -x_0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} P^{(i)}(x_0) \\ P^{(i)}(x_1) \end{bmatrix} && \text{(by our hypothesis on } f^{(i)}\text{, and by Definition 3.13.)} \\
&= \begin{bmatrix} 1-r'_i & r'_i \end{bmatrix} \cdot \begin{bmatrix} x_1 & -x_0 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \cdot \begin{bmatrix} P_0^{(i+1)}(y) \\ P_1^{(i+1)}(y) \end{bmatrix} && \text{(by the calculation just performed above.)} \\
&= \begin{bmatrix} 1-r'_i & r'_i \end{bmatrix} \cdot \begin{bmatrix} P_0^{(i+1)}(y) \\ P_1^{(i+1)}(y) \end{bmatrix} && \text{(cancellation of inverse matrices.)} \\
&= P^{(i+1)}(y). && \text{(by the definitions of } P_0^{(i+1)}(X)\text{, } P_1^{(i+1)}(X)\text{, and } P^{(i+1)}(X)\text{.)}
\end{aligned}$$

To achieve the third equality above, we note that the matrices $\begin{bmatrix} x_1 & -x_0 \\ -1 & 1 \end{bmatrix}$ and $\begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix}$ are inverses; here, we use the guarantee $x_1 - x_0 = 1$, a basic consequence of Definition 3.8 (or rather of $\ker\left(q^{(i)}\right) = \{0,1\}$). $\square$

Applying Corollary 3.11, we note finally that $\left(\widehat{W}_k^{(0)}\right)_{k=0}^{\ell-1}$ and $\left(X_j^{(0)}\right)_{i=0}^{2^\ell-1}$ themselves yield precisely the *standard* subspace vanishing and novel basis polynomials, respectively. It follows that in the base case $i = 0$ of Lemma 3.21—and assuming honest behavior by the prover—we have that $f^{(0)}$ is exactly the evaluation over $S^{(0)}$ of $P^{(0)}(X) \coloneqq P(X) = \sum_{v \in \mathcal{B}_\ell} t(v) \cdot X_{\{v\}}^{(0)}(X)$. Applying Lemma 3.21 repeatedly, we conclude by induction that $f^{(\ell)}$ is the evaluation over $S^{(\ell)}$ of the constant polynomial $\sum_{v \in \mathcal{B}_\ell} t(v) \cdot \widetilde{\mathsf{eq}}(r'_0, \ldots, r'_{\ell-1}, v_0, \ldots, v_{\ell-1}) = t(r'_0, \ldots, r'_{\ell-1})$, and that c equals exactly this latter constant value.

We are now prepared to examine the verifier's check $s_\ell \stackrel{?}{=} c \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-1})$. By the analysis just performed, under honest behavior by the prover, this latter quantity itself equals exactly $t(r'_0, \ldots, r'_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-1})$. Our claim reduces to the correctness of a certain sumcheck, as we presently argue. Indeed, for $h(X_0, \ldots, X_{\ell-1}) \coloneqq t(X_0, \ldots, X_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, X_0, \ldots, X_{\ell-1})$ defined exactly as above, we note that $\sum_{v \in \mathcal{B}_\ell} h(v) = \sum_{v \in \mathcal{B}_\ell} t(v_0, \ldots, v_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, v_0, \ldots, v_{\ell-1}) = t(r_0, \ldots, r_{\ell-1})$. Assuming now that the prover's claim is true, we see that $s = t(r_0, \ldots, r_{\ell-1}) = \sum_{v \in \mathcal{B}_\ell} h(v)$. The completeness of the sumcheck thus implies that the verifier will accept its checks $s_i \stackrel{?}{=} h_i(0) + h_i(1)$. Finally, by the prover's description (really the sumcheck's), we have $h_i(X) = \sum_{v \in \mathcal{B}_{\ell-i-1}} h(r'_0, \ldots, r'_{i-1}, X, v_0, \ldots, v_{\ell-i-2})$ in each instance of line 3 above. In particular, $s_\ell = h(r'_0, \ldots, r'_{\ell-1})$ holds. Our analysis above thus shows that $c \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-1}) = t(r'_0, \ldots, r'_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-1}) = h(r'_0, \ldots, r'_{\ell-1}) = s_\ell$.

The completeness of the verifier's query phase is essentially self-evident; we note that $\mathcal{V}$ applies to each oracle $f^{(i)}$ the same folding procedure as $\mathcal{P}$ does. This completes the proof of completeness. $\square$

Remark 3.22. Though it seems inessential to the proof of Theorem 3.20, it is interesting to note that, for each $i \in \{0, \ldots, \ell-1\}$, the i^{th}-order basis $\left(X_j^{(i)}\right)_{i=0}^{2^{\ell-i}-1}$ is *itself* a novel polynomial basis in its own right, namely that attached to the set of vectors $\left(\widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{\ell-1})\right)$. Equivalently, the i^{th}-order subspace vanishing polynomials $\left(\widehat{W}_k^{(i)}\right)_{k=0}^{\ell-i-1}$ are simply the subspace vanishing polynomials attached to this latter set of vectors. Indeed, for each $k \in \{0, \ldots, \ell-i-1\}$, $\left\langle \widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{i+k-1}) \right\rangle \subset \ker\left(\widehat{W}_k^{(i)}\right)$ certainly holds, since $\widehat{W}_k^{(i)} \circ \widehat{W}_i = q^{(i+k-1)} \circ \cdots \circ q^{(i)} \circ \widehat{W}_i = \widehat{W}_{i+k}$, which annihilates $\langle \beta_0, \ldots, \beta_{i+k-1} \rangle$ (here, we use the definition of $\widehat{W}_k^{(i)}$ and Corollary 3.11). On the other hand, $\widehat{W}_k^{(i)} = q^{(i+k-1)} \circ \cdots \circ q^{(i)}$'s kernel can be of dimension at most k (say by degree considerations), while the vectors $\widehat{W}_i(\beta_i), \ldots, \widehat{W}_i(\beta_{i+k-1})$ are linearly independent (a consequence of Corollary 3.12). We conclude that the above containment is an equality. Finally, the subspace polynomials $\left(\widehat{W}_k^{(i)}\right)_{k=0}^{\ell-i-1}$ are normalized. Indeed, using Corollary 3.11 again, we see that, for each $k \in \{0, \ldots, \ell-i-1\}$, $\widehat{W}_k^{(i)}\left(\widehat{W}_i(\beta_{i+k})\right) = \left(q^{(i+k-1)} \circ \cdots \circ q^{(i)} \circ \widehat{W}_i\right)(\beta_{i+k}) = \widehat{W}_{i+k}(\beta_{i+k}) = 1$.

Remark 3.23. Using the techniques of Subsection 3.2 and of Theorem 3.20 above, we are able to suggest a new explanation of the additive NTT algorithm of Lin, Chung and Han [LCH14, § III.], and of its correctness; we note also our Algorithm 2 above. (We refer finally to Li, et al. [Li+18, Alg. 2] for a further perspective.) We fix an index $i \in \{0, \ldots, \ell-1\}$ and a polynomial $P^{(i)}(X) \coloneqq \sum_{j=0}^{2^{\ell-i}-1} a_j \cdot X_j^{(i)}(X)$, expressed with respect to the i^{th}*-order* novel basis. The key idea is that the values of $P^{(i)}(X)$ on the domain $S^{(i)}$ can be derived—using only $\Theta\big(2^{\ell+\mathcal{R}-i}\big)$ K-operations—given the values of $P^{(i)}(X)$'s even and odd refinements $P_0^{(i+1)}(X)$ and $P_1^{(i+1)}(X)$ (as in the proof of Lemma 3.21) over the domain $S^{(i+1)}$. This is a direct consequence of the identity (2) above. Indeed, applying that identity, we see that, for $y \in S^{(i+1)}$ arbitrary, with fiber $(x_0, x_1) \coloneqq {q^{(i)}}^{-1}(\{y\})$, say, we have the equalities $P^{(i)}(x_0) \coloneqq P_0^{(i+1)}(y) + x_0 \cdot P_1^{(i+1)}(y)$ and $P^{(i)}(x_1) \coloneqq P_0^{(i+1)}(y) + x_1 \cdot P_1^{(i+1)}(y)$. Since x_0 and x_1 in fact differ by exactly 1, we see that $P^{(i)}(x_1)$ can be computed from $P^{(i)}(x_0)$ using a single further K-addition. We recover the key butterfly diagram of [LCH14, Fig. 1. (a)] (see also Algorithm 2 above) upon carrying out this procedure recursively, with the convention whereby we flatten (using the space's canonical basis) and *interleave* the two copies of $S^{(i+1)}$ at each instance. The base case of the recursion consists of the 2^ℓ-fold interleaving of the domain $S^{(\ell)}$, into which $P^{(0)}$'s coefficients are tiled $2^{\mathcal{R}}$ times. The final stage of the butterfly diagram yields the desired evaluation of $P^{(0)}(X)$ on $S^{(0)}$. Algorithm 2's twiddle factors in its i^{th} stage, then, are nothing other than the respective first lifts x_0 of y, as the image $y = q^{(i)}(x_0)$ *varies* throughout $S^{(i+1)}$. These latter elements x_0, in turn, take precisely the form $\sum_{k=0}^{\ell+\mathcal{R}-i-2} u_k \cdot \widehat{W}_i(\beta_{i+1+k})$, for $u \in \mathcal{B}_{\ell+\mathcal{R}-i-1} \cong S^{(i+1)}$ arbitrary; indeed, we suppress throughout the 0^{th} canonical basis element $\widehat{W}_i(\beta_i) = 1$ of $S^{(i)}$, since that element is subsumed into each butterfly. We find it interesting that the *same* polynomial identity underlies both the correctness of [LCH14, § III.] and our above analysis of FRI's folding.

We refrain from proving the security of Construction 3.19; rather, we defer instead to the security proof of Construction 3.24 below. The proof of the former construction can be derived from that of the latter, upon specializing that construction's packing factor $\kappa \coloneqq 0$.

3.5 Packing-Based Scheme

In this subsection, we describe a variant of Construction 3.19, designed for the use of *very small* fields (like $\mathbb{F}_2$). We recall throughout the algebraic content of Subsection 2.5. Before presenting our construction, we discuss how it must adapt the preliminary material of Subsection 3.2. In short, we must construct our novel polynomial basis in the tower field $\mathcal{T}_{\iota+\kappa}$; as a consequence, the entire content of Subsection 3.2 must be understood in *this field*. In particular, below, we universally understand the domains $S^{(0)}, \ldots, S^{(\ell-\kappa)}$ as subsets of $\mathcal{T}_{\iota+\kappa}$, and hence as subsets of $A_{\iota,\kappa,\iota} \subset A_{\iota,\kappa,\tau}$ via the *synthetic* ring inclusion. On the other hand, the verifier's folding challenges all r'_i come from $\mathcal{T}_\tau$, and must operate on $A_{\iota,\kappa,\tau}$ via the *constant* vector-space structure. Finally, the maps $f^{(i)} : S^{(i)} \to A_{\iota,\kappa,\tau}$, in general take values in the full tower algebra.

We explain this concretely in the following way. For each $i \in \{0, \ldots, \ell-\kappa-1\}$ and each $y \in S^{(i+1)}$, we study in detail the meaning we must give to the expression $\mathsf{fold}\big(f^{(i)}, r'_i\big)(y)$ of Definition 3.13. Of course, y, as well as its preimages x_0 and x_1, resides in the *synthetic* subring of $A_{\iota,\kappa,\tau}$; on the other hand, we understand $1-r'_i$ and r'_i as elements of $A_{\iota,\kappa,\tau}$'s constant subring. Finally, the values $f^{(i)}(x_0)$ and $f^{(i)}(x_1)$, in general, are general elements of the algebra $A_{\iota,\kappa,\tau}$. Thus, while the expression for $\mathsf{fold}\big(f^{(i)}, r'_i\big)(y)$ remains superficially identical, it involves both of $A_{\iota,\kappa,\tau}$'s named subrings.

Our packing-based polynomial commitment scheme proceeds as follows.

CONSTRUCTION 3.24 (Packed IOPCS)**.**
We define $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ as follows.

- $\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \ell, \iota)$. On input 1^λ, ℓ, and ι, return a constant, positive rate parameter $\mathcal{R} \in \mathbb{N}$, a packing factor $\kappa \geq 0$ for which $2^{\iota+\kappa} \geq \ell - \kappa + \mathcal{R}$, a folding factor $\vartheta \mid \ell - \kappa$, a tower height $\tau \geq \log(\omega(\log\lambda))$, and a repetition parameter $\gamma = \omega(\log(\lambda))$. Write $(X_0(X), \ldots, X_{2^{\ell-\kappa}-1}(X))$ for the novel $\mathcal{T}_{\iota+\kappa}$-basis of $\mathcal{T}_{\iota+\kappa}[X]^{\prec 2^{\ell-\kappa}}$, and fix $S^{(0)}, \ldots, S^{(\ell-\kappa)}$ and $q^{(0)}, \ldots, q^{(\ell-\kappa-1)}$ as in Subsection 3.2. Write $C^{(0)} \subset \mathcal{T}_{\iota+\kappa}^{2^{\ell-\kappa+\mathcal{R}}}$ for the Reed–Solomon code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa}, S^{(0)}}[2^{\ell-\kappa+\mathcal{R}}, 2^{\ell-\kappa}]$.

- $[f] \leftarrow \Pi.\mathsf{Commit}(\mathsf{params}, t)$. On input $t(X_0, \dots, X_{\ell-1}) \in \mathcal{T}_\iota[X_0, \dots, X_{\ell-1}]^{\preceq 1}$, apply the natural embedding chunk-wise to t's Lagrange coefficients $(t(v))_{v \in \mathcal{B}_\ell}$, so obtaining the $\mathcal{T}_{\iota+\kappa}$-vector $\big(\hat{t}(v)\big)_{v \in \mathcal{B}_{\ell-\kappa}}$, say. Write $P(X) = \sum_{v \in \mathcal{B}_{\ell-\kappa}} \hat{t}(v) \cdot X_{\{v\}}(X)$. Using Algorithm 2, compute the Reed–Solomon codeword $f : S^{(0)} \to \mathcal{T}_{\iota+\kappa}$ defined by $f : x \mapsto P(x)$. Submit $(\mathtt{submit}, \mathcal{T}_{\iota+\kappa}, \ell - \kappa + \mathcal{R}, f)$ to the vector oracle. Upon receiving $(\mathtt{receipt}, \mathcal{T}_{\iota+\kappa}, \ell - \kappa + \mathcal{R}, [f])$ from the oracle, output $[f]$.

We define $(\mathcal{P}, \mathcal{V})$ as the following IOP, in which both parties have the common input $[f]$, $s \in \mathcal{T}_\tau$, and $(r_0, \dots, r_{\ell-1}) \in \mathcal{T}_\tau^\ell$, and $\mathcal{P}$ has the further input $t(X_0, \dots, X_{\ell-1}) \in \mathcal{T}_\iota[X_0, \dots, X_{\ell-1}]^{\preceq 1}$.

- $\mathcal{P}$ writes $h(X_0, \dots, X_{\ell-1}) \coloneqq t(X_0, \dots, X_{\ell-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \dots, r_{\ell-1}, X_0, \dots, X_{\ell-1})$; $\mathcal{P}$ moreover abbreviates $h'(X_0, \dots, X_{\ell-\kappa-1}) \coloneqq \sum_{u \in \mathcal{B}_\kappa} h(u_0, \dots, u_{\kappa-1}, X_0, \dots, X_{\ell-\kappa-1})$.
- $\mathcal{P}$ and $\mathcal{V}$ both abbreviate $f^{(0)} \coloneqq f$ and $s_0 \coloneqq s$, and execute the following loop:
  1: **for** $i \in \{0, \dots, \ell - \kappa - 1\}$ **do**
  2: $\quad\mathcal{P}$ sends $\mathcal{V}$ the polynomial $h_i'(X) \coloneqq \sum_{v \in \mathcal{B}_{\ell-\kappa-i-1}} h'(r_0', \dots, r_{i-1}', X, v_0, \dots, v_{\ell-\kappa-i-2})$.
  3: $\quad\mathcal{V}$ requires $s_i \overset{?}{=} h_i'(0) + h_i'(1)$. $\mathcal{V}$ samples $r_i' \leftarrow \mathcal{T}_\tau$, sets $s_{i+1} \coloneqq h_i'(r_i')$, and sends $\mathcal{P}$ r_i'.
  4: $\quad\mathcal{P}$ defines $f^{(i+1)} : S^{(i+1)} \to A_{\iota,\kappa,\tau}$ as the function $\mathsf{fold}\big(f^{(i)}, r_i'\big)$ of Definition 3.13.
  5: $\quad$**if** $i + 1 = \ell$ **then** $\mathcal{P}$ sends $c \coloneqq f^{(\ell-\kappa)}(0, \dots, 0)$ to $\mathcal{V}$.
  6: $\quad$**else if** $\vartheta \mid i + 1$ **then** $\mathcal{P}$ submits $(\mathtt{submit}, A_{\iota,\kappa,\tau}, \ell - \kappa + \mathcal{R} - i - 1, f^{(i+1)})$ to the oracle.
- By reversing the natural embedding, $\mathcal{V}$ destructures $(c_u)_{u \in \mathcal{B}_\kappa} \coloneqq c$ as a $\mathcal{T}_\tau^{2^\kappa}$-element.
- $\mathcal{V}$ requires $s_{\ell-\kappa} \overset{?}{=} \widetilde{\mathsf{eq}}(r_\kappa, \dots, r_{\ell-1}, r_0', \dots, r_{\ell-\kappa-1}') \cdot \sum_{u \in \mathcal{B}_\kappa} c_u \cdot \widetilde{\mathsf{eq}}(r_0, \dots, r_{\kappa-1}, u_0, \dots, u_{\kappa-1})$.
- $\mathcal{V}$ assigns $c_{\ell-\kappa} \coloneqq c$, and executes the following querying procedure:
  1: **for** γ repetitions **do**
  2: $\quad\mathcal{V}$ samples $u \leftarrow \mathcal{B}_\mathcal{R}$ randomly.
  3: $\quad$**for** $i \in \{\ell - \kappa - \vartheta, \ell - \kappa - 2 \cdot \vartheta, \dots, 0\}$ (i.e., in downward order, taking ϑ-sized steps) **do**
  4: $\quad\quad$for each $v \in \mathcal{B}_\vartheta$, $\mathcal{V}$ submits $(\mathtt{query}, [f^{(i)}], v \parallel u)$ to the vector oracle.
  5: $\quad\quad\mathcal{V}$ requires $c_{i+\vartheta} \overset{?}{=} \mathsf{fold}\big(f^{(i)}, r_i', \dots, r_{i+\vartheta-1}'\big)(u)$.
  6: $\quad\quad\mathcal{V}$ samples $v \leftarrow \mathcal{B}_\vartheta$, sets $c_i \coloneqq f^{(i)}(v \parallel u)$, and overwrites $u \coloneqq v \parallel u$.

The polynomial $h'(X_0, \dots, X_{\ell-\kappa-1}) \coloneqq \sum_{u \in \mathcal{B}_\kappa} h(u_0, \dots, u_{\kappa-1}, X_0, \dots, X_{\ell-\kappa-1})$ serves essentially to effect a "partial sumcheck" on $h(X_0, \dots, X_{\ell-1})$. Indeed, the intent of Construction 3.24's sumcheck is essentially to "cut short" that of h at the $\ell - \kappa^{\text{th}}$ round; on the other hand, we must specialize h's *last* $\ell - \kappa$ variables, as opposed to its first. (We could have equally remedied this issue by specializing right-to-left in the sumcheck, instead of left-to-right, and using h directly.)

Theorem 3.25. *The IOPCS $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ of Construction 3.24 is complete.*

Proof. We require first an analogue of Lemma 3.21 in the packed setting. Indeed, that lemma goes through exactly as written; we emphasize, however, that the (higher-order) novel basis polynomials $\Big(X_j^{(i)}(X)\Big)_{j=0}^{2^{\ell-\kappa-i}-1}$ have *coefficients* in the synthetic subring $A_{\iota,\kappa,\iota} \subset A_{\iota,\kappa,\tau}$ of the tower algebra. The elements y, x_0, and x_1 in the proof of that lemma all reside in the synthetic subring, while r_i' resides in the constant subring; finally, the coefficients $a_0, \dots, a_{2^{\ell-\kappa-i}-1}$ of $P^{(i)}(X)$, in general, are *general* members of the tower algebra. The proof of that lemma otherwise goes through without change. For completeness, we state the relevant analogue:

Lemma 3.26. *Fix an index $i \in \{0, \dots, \ell - \kappa - 1\}$. If $f^{(i)} : S^{(i)} \to A_{\iota,\kappa,\tau}$ is exactly the evaluation of the polynomial $P^{(i)}(X) = \sum_{j=0}^{2^{\ell-\kappa-i}-1} a_j \cdot X_j^{(i)}(X)$, then, under honest prover behavior, $f^{(i+1)} : S^{(i+1)} \to A_{\iota,\kappa,\tau}$ is exactly the evaluation of the polynomial $P^{(i+1)}(X) = \sum_{j=0}^{2^{\ell-\kappa-i-1}-1} ((1 - r_i') \cdot a_{2j} + r_i' \cdot a_{2j+1}) \cdot X_j^{(i+1)}(X)$.*

Proof. This lemma's proof is the same as Lemma 3.21's. □

Applying Lemma 3.26 inductively, we conclude as in the proof of Theorem 3.20 that $\mathcal{P}$'s final FRI message $c = \sum_{v\in\mathcal{B}_{\ell-\kappa}} \hat{t}(v)\cdot\widetilde{\mathsf{eq}}\big(r'_0,\ldots,r'_{\ell-\kappa-1},v_0,\ldots,v_{\ell-\kappa-1}\big)$ (here, each coefficient $\hat{t}(v)$ is an *algebra* element in the synthetic subring, on which the scalar on right acts by the constant structure). By definition of the constant vector space structure, the destructuring $(c_u)_{u\in\mathcal{B}_\kappa}$ of this latter quantity has, at each of its indices $u\in\mathcal{B}_u$, the component $c_u = \sum_{v\in\mathcal{B}_{\ell-\kappa}} t(u_0,\ldots,u_{\kappa-1},v_0,\ldots,v_{\ell-\kappa-1})\cdot\widetilde{\mathsf{eq}}(r'_0,\ldots,r'_{\ell-\kappa-1},v_0,\ldots,v_{\ell-\kappa-1}) = t(u_0,\ldots,u_{\kappa-1},r'_0,\ldots r'_{\ell-\kappa-1})$ (in the last equality, we use a standard property of multilinear evaluation).

On the other hand, again as in the proof of Lemma 3.20—and assuming now that the prover's claim is correct—we have the equality $s = t(r_0,\ldots,r_{\ell-1}) = \sum_{v\in\mathcal{B}_\ell} h(v) = \sum_{v\in\mathcal{B}_{\ell-\kappa}} h'(v)$. The correctness of the sumcheck, applied to $h'(X_0,\ldots,X_{\ell-\kappa-1})$, thus implies that the verifier will accept its checks $s_i \stackrel{?}{=} h'_i(0)+h'_i(1)$, as well as that $s_{\ell-\kappa} = h'(r'_0,\ldots,r'_{\ell-\kappa-1})$. We unroll this latter quantity in the following way:

$$\begin{aligned}
h'(r'_0,\ldots,r'_{\ell-\kappa-1}) &= \sum_{u\in\mathcal{B}_\kappa} h(u_0,\ldots,u_{\kappa-1},r'_0,\ldots,r'_{\ell-\kappa-1})\\
&= \sum_{u\in\mathcal{B}_\kappa} t(u_0,\ldots,u_{\kappa-1},r'_0,\ldots,r'_{\ell-\kappa-1})\cdot\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\ell-1},u_0,\ldots,u_{\kappa-1},r'_0,\ldots,r'_{\ell-\kappa-1})\\
&= \sum_{u\in\mathcal{B}_\kappa} c_u\cdot\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\ell-1},u_0,\ldots,u_{\kappa-1},r'_0,\ldots,r'_{\ell-\kappa-1})\\
&= \widetilde{\mathsf{eq}}(r_\kappa,\ldots,r_{\ell-1},r'_0,\ldots,r'_{\ell-\kappa-1})\cdot\sum_{u\in\mathcal{B}_\kappa} c_u\cdot\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\kappa-1},u_0,\ldots,u_{\kappa-1}),
\end{aligned}$$

which is exactly what the verifier compares $s_{\ell-\kappa}$ to. In the third equality above, we use the identity $c_u = t(u_0,\ldots,u_{\kappa-1},r'_0,\ldots r'_{\ell-\kappa-1})$, already explained above. □

We now prove the security of Construction 3.24. Our key technical results below (see Propositions 3.30 and 3.34), essentially, jointly constitute a variant of FRI's soundness statement [BBHR18, § 4.2.2]. Our proofs of these results incorporate—in an attenuated form—various ideas present in [BBHR18, § 4.2.2] and [Ben+23, § 8.2]. We also introduce a number of new ideas, which, by and large, pertain to our new folding technique (see Subsection 3.3).

Theorem 3.27. *The IOPCS* $\Pi = (\mathsf{Setup},\mathsf{Commit},\mathcal{P},\mathcal{V})$ *of Construction 3.24 is secure.*

Proof. We define a straight-line emulator $\mathcal{E}$ as follows.

1. By inspecting $\mathcal{A}$'s messages to the vector oracle, $\mathcal{E}$ immediately recovers the function $f : S^{(0)}\to\mathcal{T}_{\iota+\kappa}$ underlying the handle $[f]$ output by $\mathcal{A}$.
2. $\mathcal{E}$ runs Algorithm 1 on the word $f : S^{(0)}\to\mathcal{T}_{\iota+\kappa}$. If that algorithm outputs $P(X)=\perp$, then $\mathcal{E}$ sets $t(X_0,\ldots,X_{\ell-1}) \coloneqq 0$. Otherwise, $\mathcal{E}$ expresses the output $P(X) = \sum_{v\in\mathcal{B}_{\ell-\kappa}}\hat{t}(v)\cdot X_{\{v\}}(X)$ in coordinates with respect to the novel polynomial basis. By reversing the natural embedding on each of $P(X)$'s coordinates (see Subsection 2.5), $\mathcal{E}$ obtains the $\mathcal{T}_\iota^{\mathcal{B}_\ell}$-element $(t(v))_{v\in\mathcal{B}_\ell}$, say. $\mathcal{E}$ writes $t(X_0,\ldots,X_{\ell-1})\in\mathcal{T}_\iota[X_0,\ldots,X_{\ell-1}]^{\preceq 1}$ for the polynomial given in Lagrange coordinates by $(t(v))_{v\in\mathcal{B}_\ell}$.
3. $\mathcal{E}$ outputs $t(X_0,\ldots,X_{\ell-1})$ and terminates.

We now argue that $\mathcal{E}$ fulfills the requirements of Definition 3.7 with respect to the protocol Π.

We define various notions, adapting [BBHR18, § 4.2.1]. For each $i\in\{0,\vartheta,\ldots,\ell-\kappa\}$ (i.e., ascending in ϑ-sized steps), we write $C^{(i)}\subset\mathcal{T}_{\iota+\kappa}^{2^{\ell-\kappa+\mathcal{R}-i}}$ for the Reed–Solomon code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa},S^{(i)}}[2^{\ell-\kappa+\mathcal{R}-i},2^{\ell-\kappa-i}]$, as well as $\widehat{C}^{(i)}\subset A_{\iota,\kappa,\tau}^{2^{\ell-\kappa+\mathcal{R}-i}}$ for its extension code. We recall that $C^{(i)}$ and $\widehat{C}^{(i)}$ are both of distance $d_i \coloneqq 2^{\ell-\kappa+\mathcal{R}-i}-2^{\ell-\kappa-i}+1$. We write $f^{(0)},f^{(\vartheta)},\ldots,f^{(\ell-\kappa-\vartheta)}$ for the oracles committed by $\mathcal{A}$; we moreover write $f^{(\ell-\kappa)}:S^{(\ell-\kappa)}\to A_{\iota,\kappa,\tau}$ for the identically-c function (here, $c\in A_{\iota,\kappa,\tau}$ is $\mathcal{A}$'s final FRI message). For each $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$, we write $\Delta\big(f^{(i+\vartheta)},g^{(i+\vartheta)}\big)\subset S^{(i+\vartheta)}$ for the *disagreement* set between the elements $f^{(i+\vartheta)}$ and $g^{(i+\vartheta)}$ of $A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}-i-\vartheta}}$; that is, $\Delta\big(f^{(i+\vartheta)},g^{(i+\vartheta)}\big)$ is the set of elements $y\in S^{(i+\vartheta)}$ for which $f^{(i+\vartheta)}(y)\neq g^{(i+\vartheta)}(y)$. We moreover write $\Delta^{(i)}\big(f^{(i)},g^{(i)}\big)\subset S^{(i+\vartheta)}$ for the *fiber-wise disagreement set* of the elements $f^{(i)}$ and $g^{(i)}$ of $A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}-i}}$. That is, $\Delta^{(i)}\big(f^{(i)},g^{(i)}\big)$ denotes the set of elements $y\in S^{(i+\vartheta)}$ for which

the respective restrictions of $f^{(i)}$ and $g^{(i)}$ to the fiber $\big(q^{(i+\vartheta-1)}\circ\cdots\circ q^{(i)}\big)^{-1}(\{y\})\subset S^{(i)}$ are not identically equal. We define $d^{(i)}\Big(f^{(i)},\widehat{C}^{(i)}\Big)\coloneqq\min_{g^{(i)}\in\widehat{C}^{(i)}}\big|\Delta^{(i)}\big(f^{(i)},g^{(i)}\big)\big|$. We note that, if $d^{(i)}\Big(f^{(i)},\widehat{C}^{(i)}\Big)<\frac{d_{i+\vartheta}}{2}$, then $d\Big(f^{(i)},\widehat{C}^{(i)}\Big)<\frac{d_i}{2}$ *a fortiori* holds. (Each offending fiber contributes at most 2^ϑ errors; on the other hand, $2^\vartheta\cdot\left\lfloor\frac{d_{i+\vartheta}-1}{2}\right\rfloor\leq\left\lfloor\frac{d_i-1}{2}\right\rfloor$.) In any case, in case the oracle $f^{(i)}:S^{(i)}\to A_{\iota,\kappa,\tau}$ is such that $d\Big(f^{(i)},\widehat{C}^{(i)}\Big)<\frac{d_i}{2}$ happens to hold, we write $\overline{f}^{(i)}\in\widehat{C}^{(i)}$ for the unique codeword for which $d\big(f^{(i)},\overline{f}^{(i)}\big)<\frac{d_i}{2}$.

We begin with the following elementary fact.

Lemma 3.28. *For each $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$, in case $d\Big(f^{(i)},\widehat{C}^{(i)}\Big)<\frac{d_i}{2}$ holds, then, for each tuple of folding challenges $(r'_i,\ldots,r'_{i+\vartheta-1})\in\mathcal{T}_\tau^\vartheta$, $\Delta\big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)\big)\subset\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)$.*

Proof. For each $y\in S^{(i+\vartheta)}$ for which the restrictions $f^{(i)}\big|_{\left(q^{(i+\vartheta-1)}\circ\cdots\circ q^{(i)}\right)^{-1}(\{y\})}=\overline{f}^{(i)}\big|_{\left(q^{(i+\vartheta-1)}\circ\cdots\circ q^{(i)}\right)^{-1}(\{y\})}$ are identically equal, $\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)(y)=\mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)(y)$ certainly also holds. □

Of course, the conclusion Lemma 3.28 holds for *each* pair of elements $f^{(i)}$ and $g^{(i)}$ of $A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}-i}}$, and not just for $f^{(i)}$ and $\overline{f}^{(i)}$; we will use it only in the form just given.

We now define a sequence of bad folding events. For each $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$ (i.e., ascending in steps of size ϑ), the bad subset $E_i\subset\mathcal{T}_\tau^\vartheta$ depends *only* on the already-committed oracle $f^{(i)}$. Our definition of E_i is case-based, and depends on the status of $f^{(i)}$. If $f^{(i)}$ is within the (fiber-wise) unique decoding radius, then E_i captures the event whereby the generic inclusion of Lemma 3.28 becomes *strict*. Otherwise, E_i captures the "bad batching" event whereby $\mathsf{fold}(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1})$ becomes close to $\widehat{C}^{(i+\vartheta)}$.

Definition 3.29. For each $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$, we define the *bad subset* $E_i\subset\mathcal{T}_\tau^\vartheta$ as the set of tuples $(r'_i,\ldots,r'_{i+\vartheta-1})\in\mathcal{T}_\tau^\vartheta$ for which, as the case may be:

in case $\boldsymbol{d^{(i)}\Big(f^{(i)},\widehat{C}^{(i)}\Big)<\frac{d_{i+\vartheta}}{2}}$: $\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)\not\subset\Delta\big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)\big)$.

in case $\boldsymbol{d^{(i)}\Big(f^{(i)},\widehat{C}^{(i)}\Big)\geq\frac{d_{i+\vartheta}}{2}}$: $d\Big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\widehat{C}^{(i+\vartheta)}\Big)<\frac{d_{i+\vartheta}}{3}$.

We now bound the bad subsets E_i of Definition 3.29. We recall that $\mu(E_i)\coloneqq\frac{|E_i|}{|\mathcal{T}_\tau|^\vartheta}$ signifies the probability mass of the set $E_i\subset\mathcal{T}_\tau^\vartheta$.

Proposition 3.30. *For each $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$, $\mu(E_i)\leq\vartheta\cdot\frac{\left|S^{(i+\vartheta)}\right|}{|\mathcal{T}_\tau|}$ holds.*

Proof. We treat separately the two cases of Definition 3.29.

We begin with the first case. We fix an element $y\in\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)$, we moreover write $E_i^y\subset\mathcal{T}_\tau^\vartheta$ for the set of tuples $(r'_i,\ldots,r'_{i+\vartheta-1})\in\mathcal{T}_\tau^\vartheta$ for which $y\notin\Delta\big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)\big)$. We argue that $\mu(E_i^y)\leq\frac{\vartheta}{|\mathcal{T}_\tau|}$. This latter claim suffices to complete the proof of the first case; indeed, since $E_i=\bigcup_{y\in\Delta^{(i)}\left(f^{(i)},\overline{f}^{(i)}\right)}E_i^y$, assuming the claim, we conclude that $\mu(E_i)\leq\big|\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)\big|\cdot\frac{\vartheta}{|\mathcal{T}_\tau|}\leq|S^{(i+\vartheta)}|\cdot\frac{\vartheta}{|\mathcal{T}_\tau|}$.

For $y\in\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)$ chosen as above, we apply Lemma 3.16 to the words $f^{(i)}$ and $\overline{f}^{(i)}$. Applying that lemma, we see that $(r'_i,\ldots,r'_{i+\vartheta-1})\in E_i^y$ holds if and only if we have the following matrix identity:

$$0=\begin{bmatrix}\bigotimes_{j=0}^{\vartheta-1}(1-r'_{i+j},r'_{i+j})\end{bmatrix}\cdot\begin{bmatrix}M_y\end{bmatrix}\cdot\begin{bmatrix}f^{(i)}(x_0)-\overline{f}^{(i)}(x_0)\\\vdots\\f^{(i)}(x_{2^\vartheta-1})-\overline{f}^{(i)}(x_{2^\vartheta-1})\end{bmatrix},\tag{3}$$

where we again write $(x_0,\ldots,x_{2^\vartheta-1})\coloneqq\big(q^{(i+\vartheta-1)}\circ\cdots\circ q^{(i)}\big)^{-1}(\{y\})$. Our hypothesis $y\in\Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)$ entails precisely that the right-hand vector of (3)—whose entries reside in $A_{\iota,\kappa,\tau}$—is not identically zero. Lemma 3.16 guarantees that M_y, whose entries reside in $\mathcal{T}_{\iota+\kappa}$, is nonsingular; we conclude that the image of the right-hand vector of (3) under M_y is likewise not identically zero (the nonsingular $\mathcal{T}_{\iota+\kappa}$-matrix M_y necessarily induces an injective map on the space of length-2^ϑ $A_{\iota,\kappa,\tau}$-vectors). Writing $(a_0,\ldots,a_{2^\vartheta-1})$ for this

latter vector—which, we repeat, is not zero—we conclude that $E_i \subset \mathcal{T}_\tau^\vartheta$ is precisely the vanishing locus of the restriction to $\mathcal{T}_\tau^\vartheta$ of the ϑ-variate polynomial $s(X_0, \ldots, X_{\vartheta-1}) \coloneqq \sum_{v \in \mathcal{B}_\vartheta} a_{\{v\}} \cdot \widetilde{\mathsf{eq}}(X_0, \ldots, X_{\vartheta-1}, v_0, \ldots, v_{\vartheta-1})$. Since $s(X_0, \ldots, X_{\vartheta-1})$'s respective values over the cube $\{0,1\}^\vartheta \subset \mathcal{T}_\tau^\vartheta$ are exactly $(a_0, \ldots, a_{2^\vartheta-1})$, s is certainly not zero. Applying Schwartz–Zippel—precisely, the variant of that lemma given in Lemma 2.4 above—to $s(X_0, \ldots, X_{\vartheta-1})$, we conclude that the relevant locus is of mass at most $\mu(E_i^y) \leq \frac{\vartheta}{|\mathcal{T}_\tau|}$, as required.

We turn to the second case of Definition 3.29; in particular, we assume that $d^{(i)}\Big(f^{(i)}, \widehat{C}^{(i)}\Big) \geq \frac{d_{i+\vartheta}}{2}$. We define an interleaved word $\left(f_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$—i.e., a $2^\vartheta \times 2^{\ell-\kappa+\mathcal{R}-i-\vartheta}$ matrix, with entries in $A_{\iota,\kappa,\tau}$—in the following way. For each $y \in S^{(i+\vartheta)}$, writing M_y for the matrix guaranteed to exist by Lemma 3.16, we define:

$$\begin{bmatrix} f_0^{(i+\vartheta)}(y) \\ \vdots \\ f_{2^\vartheta-1}^{(i+\vartheta)}(y) \end{bmatrix} \coloneqq \begin{bmatrix} \\ M_y \\ \\ \end{bmatrix} \cdot \begin{bmatrix} f^{(i)}(x_0) \\ \vdots \\ f^{(i)}(x_{2^\vartheta-1}) \end{bmatrix}. \tag{4}$$

We note that the resulting $2^\vartheta \times 2^{\ell-\kappa+\mathcal{R}-i-\vartheta}$ matrix $\left(f_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$—i.e., that whose columns are given by the respective left-hand sides of (4), for $y \in S^{(i+\vartheta)}$—satisfies, for each $(r'_i, \ldots, r'_{i+\vartheta-1}) \in \mathcal{T}_\tau^\vartheta$,

$$\mathsf{fold}\Big(f^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\Big) = \begin{bmatrix} \bigotimes_{j=i}^{i+\vartheta-1}(1-r'_j, r'_j) \end{bmatrix} \cdot \begin{bmatrix} — & f_0^{(i+\vartheta)} & — \\ & \vdots & \\ — & f_{2^\vartheta-1}^{(i+\vartheta)} & — \end{bmatrix}. \tag{5}$$

Indeed, this is essentially the content of Lemma 3.16, which we apply here jointly to *all* elements $y \in S^{(i+\vartheta)}$.

We claim that the interleaved word $\left(f_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$ constructed in this way is far from the interleaved code $\widehat{C}^{(i+\vartheta)^{2^\vartheta}}$.

Lemma 3.31. *Under our hypothesis $d^{(i)}\Big(f^{(i)}, \widehat{C}^{(i)}\Big) \geq \frac{d_{i+\vartheta}}{2}$, we have $d^{2^\vartheta}\left(\left(f_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}, \widehat{C}^{(i+\vartheta)^{2^\vartheta}}\right) \geq \frac{d_{i+\vartheta}}{2}$.*

Proof. We fix an arbitrary interleaved *codeword* $\left(g_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1} \in \widehat{C}^{(i+\vartheta)^{2^\vartheta}}$. We define a "lift" $g^{(i)} \in \widehat{C}^{(i)}$ of $\left(g_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$ in the following way. Writing, for each $j \in \{0, \ldots, 2^\vartheta - 1\}$, $P_j^{(i+\vartheta)}(X) \coloneqq \sum_{k=0}^{2^{\ell-\kappa-i-\vartheta}-1} a_{j,k} \cdot X_k^{(i+\vartheta)}(X)$ for the polynomial—expressed in coordinates with respect to the $i+\vartheta^{\text{th}}$-order novel polynomial basis—for which $g_j^{(i+\vartheta)} = \mathsf{Enc}(P_j^{(i+\vartheta)})$ holds, we define

$$P^{(i)}(X) \coloneqq \sum_{j=0}^{2^\vartheta-1} \sum_{k=0}^{2^{\ell-\kappa-i-\vartheta}-1} a_{j,k} \cdot X_{k \cdot 2^\vartheta + j}^{(i)};$$

that is, $P^{(i)}$'s list of i^{th}*-order* coefficients is precisely the 2^ϑ-fold interleaving of the polynomials $P_0^{(i+\vartheta)}(X), \ldots, P_{2^\vartheta-1}^{(i+\vartheta)}(X)$'s respective lists of $i+\vartheta^{th}$*-order* coefficients. Finally, we define $g^{(i)} \coloneqq \mathsf{Enc}(P^{(i)})$.

We argue that the codeword $g^{(i)} \in \widehat{C}^{(i)}$ constructed in this way stands in relation to $\left(g_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$ just as $f^{(i)}$ does to $\left(f_j^{(i+\vartheta)}\right)_{j=0}^{2^\vartheta-1}$ (i.e., it also satisfies a matrix identity analogous to (4) for each $y \in S^{(i+\vartheta)}$). To prove this, we fix an arbitrary element $y \in S^{(i+\vartheta)}$; we moreover fix a row-index $j \in \{0, \ldots, 2^\vartheta - 1\}$. We write $(j_0, \ldots, j_{\vartheta-1})$ for the bits of j (i.e., so that $j = \sum_{k=0}^{\vartheta-1} 2^k \cdot j_k$ holds). We first note that the functions $g_j^{(i+\vartheta)}$ and $\mathsf{fold}\big(g^{(i)}, j_0, \ldots, j_{\vartheta-1}\big)$ agree identically over the domain $S^{(i+\vartheta)}$. Indeed, this is a direct consequence of Lemma

3.26 and of the construction of $g^{(i)}$ ($g_j^{(i+\vartheta)}(y)$'s underlying polynomial's coefficients are the j^{th} refinement of $g^{(i)}$'s underlying polynomial's). On the other hand, applying Lemma 3.16 to $y \in S^{(i+\vartheta)}$ and $g^{(i)}$, with the folding tuple $(j_0, \ldots, j_{\vartheta-1})$, we see that the dot product between M_y's j^{th} row and $\big(g^{(i)}(x_0), \ldots, g^{(i)}(x_{2^\vartheta-1})\big)$ is exactly $\mathsf{fold}\big(g^{(i)}, j_0, \ldots, j_{\vartheta-1}\big)(y) = g_j^{(i+\vartheta)}(y)$, where the latter equality was just argued.

Since $g^{(i)} \in \widehat{C}^{(i)}$ is a codeword, our hypothesis $d^{(i)}\Big(f^{(i)}, \widehat{C}^{(i)}\Big) \geq \frac{d_{i+\vartheta}}{2}$ applies to it. That hypothesis entails precisely that, for *at least* $\frac{d_{i+\vartheta}}{2}$ elements $y \in S^{(i+\vartheta)}$, the restrictions $f^{(i)}\big|_{\left(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\right)^{-1}(\{y\})}$ and $g^{(i)}\big|_{\left(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\right)^{-1}(\{y\})}$ are not identically equal. For each such $y \in S^{(i+\vartheta)}$, since M_y is invertible (and since both $f^{(i)}$ and $g^{(i)}$ satisfy (4)), we conclude that the columns $\Big(f_j^{(i+\vartheta)}(y)\Big)_{j=0}^{2^\vartheta-1}$ and $\Big(g_j^{(i+\vartheta)}(y)\Big)_{j=0}^{2^\vartheta-1}$ are in turn unequal. Since $\Big(g_j^{(i+\vartheta)}\Big)_{j=0}^{2^\vartheta-1}$ was arbitrary, we conclude that $d^{2^\vartheta}\left(\Big(f_j^{(i+\vartheta)}\Big)_{j=0}^{2^\vartheta-1}, \widehat{C}^{(i+\vartheta)^{2^\vartheta}}\right) \geq \frac{d_{i+\vartheta}}{2}$. □

Applying Lemma 3.31, we conclude directly that the contraposition of Theorem 2.5 is fulfilled with respect to the code $\widehat{C}^{(i+\vartheta)} \subset A_{\iota,\kappa,\tau}^{2^{\ell+\mathcal{R}-i-\vartheta}}$, the proximity parameter $e \coloneqq \left\lfloor \frac{d_{i+\vartheta}-1}{3} \right\rfloor < \frac{d_{i+\vartheta}}{2}$, and the interleaved word $\Big(f_j^{(i+\vartheta)}\Big)_{j=0}^{2^\vartheta-1}$. That theorem's contraposition immediately implies that the set $E_i \subset \mathcal{T}_\tau^\vartheta$ consisting of those tuples $(r'_i, \ldots, r'_{i+\vartheta-1}) \in \mathcal{T}_\tau^\vartheta$ for which $d\Big(\mathsf{fold}\big(f^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big), \widehat{C}^{(i+\vartheta)}\Big) < \frac{d_{i+\vartheta}}{3}$ holds—and here, we use (5)—is of mass at most $\mu(E_i) \leq 2 \cdot \vartheta \cdot \frac{e+1}{|\mathcal{T}_\tau|} \leq \vartheta \cdot \frac{2^{\ell-\kappa+\mathcal{R}-i-\vartheta}}{|\mathcal{T}_\tau|} = \vartheta \cdot \frac{\left|S^{(i+\vartheta)}\right|}{|\mathcal{T}_\tau|}$, as required. □

Proposition 3.32. *The probability that any of the bad events $E_0, E_\vartheta, \ldots, E_{\ell-\kappa-\vartheta}$ occurs is at most $\frac{2^{\ell-\kappa+\mathcal{R}}}{|\mathcal{T}_\tau|}$.*

Proof. Applying Proposition 3.30, we upper-bound the quantity of interest as:

$$\frac{\vartheta}{|\mathcal{T}_\tau|} \cdot \left(|S_\vartheta| + \cdots + |S_{\ell-\kappa}|\right) = \frac{\vartheta}{|\mathcal{T}_\tau|} \cdot \left(2^{\ell-\kappa+\mathcal{R}-\vartheta} + \cdots + 2^{\mathcal{R}}\right) \leq \frac{\vartheta}{|\mathcal{T}_\tau|} \cdot \frac{2^\vartheta}{2^\vartheta - 1} \cdot 2^{\ell-\kappa+\mathcal{R}-\vartheta} \leq \frac{2^{\ell-\kappa+\mathcal{R}}}{|\mathcal{T}_\tau|},$$

which completes the proof. In the last two steps, we use the geometric series formula and the inequality $\frac{\vartheta}{2^\vartheta-1} \leq 1$ (which holds for each $\vartheta \geq 1$), respectively. □

In light of Proposition 3.32, we freely assume that none of the events $E_0, E_\vartheta, \ldots, E_{\ell-\kappa-\vartheta}$ occur. Under this assumption, we proceed with the proof.

We record the following key compliance condition:

Definition 3.33. For each index $i \in \{0, \vartheta, \ldots, \ell-\kappa-\vartheta\}$, we say that $\mathcal{A}$'s i^{th} oracle $f^{(i)}$ is *compliant* if the conditions $d^{(i)}\Big(f^{(i)}, \widehat{C}^{(i)}\Big) < \frac{d_i}{2}$, $d\Big(f^{(i+\vartheta)}, \widehat{C}^{(i+\vartheta)}\Big) < \frac{d_{i+\vartheta}}{2}$, and $\overline{f}^{(i+\vartheta)} = \mathsf{fold}\big(\overline{f}^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)$ all hold.

Proposition 3.34. *If* any *of $\mathcal{A}$'s oracles is not compliant, then $\mathcal{V}$ accepts with at most negligible probability.*

Proof. We suppose that at least one of $\mathcal{A}$'s oracles is not compliant; we write $i^* \in \{0, \vartheta, \ldots, \ell-\kappa-\vartheta\}$ for $\mathcal{A}$'s *highest-indexed* noncompliant oracle. As a trivial consequence of Definition 3.33 and of the definition of i^*, we note immediately that, for each $i \in \{i^*+\vartheta, \ldots, \ell-\kappa-\vartheta\}$, the conditions $d^{(i)}\Big(f^{(i)}, \widehat{C}^{(i)}\Big) < \frac{d_i}{2}$, $d\Big(f^{(i+\vartheta)}, \widehat{C}^{(i+\vartheta)}\Big) < \frac{d_{i+\vartheta}}{2}$, and $\overline{f}^{(i+\vartheta)} = \mathsf{fold}\big(\overline{f}^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)$ all hold.

Lemma 3.35. *For $i^* \in \{0, \vartheta, \ldots, \ell-\kappa-\vartheta\}$ as above, we have $d\big(\mathsf{fold}\big(f^{(i^*)}, r'_{i^*}, \ldots, r'_{i^*+\vartheta-1}\big), \overline{f}^{(i^*+\vartheta)}\big) \geq \frac{d_{i^*+\vartheta}}{3}$.*

Proof. We again treat two cases, corresponding to those isolated by Definition 3.29.

Assuming first that $d^{(i^*)}\Big(f^{(i^*)}, \widehat{C}^{(i^*)}\Big) < \frac{d_{i^*+\vartheta}}{2}$, we write $\overline{f}^{(i^*)} \in \widehat{C}^{(i^*)}$ for the codeword for which $\left|\Delta^{(i^*)}\big(f^{(i^*)}, \overline{f}^{(i^*)}\big)\right| < \frac{d_{i^*+\vartheta}}{2}$ holds. Since $d\big(f^{(i^*)}, \overline{f}^{(i^*)}\big) < \frac{d_{i^*}}{2}$ *a fortiori* holds, by Definition 3.33 and our choice of i^*, we necessarily have $\overline{f}^{(i^*+\vartheta)} \neq \mathsf{fold}\big(\overline{f}^{(i^*)}, r'_{i^*}, \ldots, r'_{i^*+\vartheta-1}\big)$. On the other hand, by Lemma 3.28,

$\big|\Delta^{(i^*)}\big(f^{(i^*)},\overline{f}^{(i^*)}\big)\big| < \frac{d_{i^*+\vartheta}}{2}$ implies that $d\big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big)\big) < \frac{d_{i^*+\vartheta}}{2}$. Finally, by the reverse triangle inequality, $d\big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big),\overline{f}^{(i^*+\vartheta)}\big)$ is at least:

$$d\Big(\overline{f}^{(i^*+\vartheta)},\mathsf{fold}\Big(\overline{f}^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\Big)\Big) - d\Big(\mathsf{fold}\Big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\Big),\mathsf{fold}\Big(\overline{f}^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\Big)\Big).$$

Since $\overline{f}^{(i^*+\vartheta)}$ and $\mathsf{fold}\big(\overline{f}^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big)$ are unequal codewords in $\widehat{C}^{(i^*+\vartheta)}$, this quantity in turn is greater than $d_{i^*+\vartheta} - \frac{d_{i^*+\vartheta}}{2} \geq \frac{d_{i^*+\vartheta}}{2}$, and the proof of the first case is complete.

In the case $d^{(i^*)}\Big(f^{(i^*)},\widehat{C}^{(i^*)}\Big) \geq \frac{d_{i^*+\vartheta}}{2}$, our assumption whereby E_{i^*} didn't occur implies, by definition, that $d\Big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big),\widehat{C}^{(i^*+\vartheta)}\Big) \geq \frac{d_{i^*+\vartheta}}{3}$. Since $\overline{f}^{(i^*+\vartheta)} \in \widehat{C}^{(i^*+\vartheta)}$ is a codeword, $d\big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big),\overline{f}^{(i^*+\vartheta)}\big) \geq \frac{d_{i^*+\vartheta}}{3}$ in particular holds, and the proof is again complete. □

In order to simplify the statement of the following lemma, we tacitly replace the query phase in Construction 3.19 above by a variant in which $\mathcal{V}$, in each iteration of its main query loop, first collects its equalities, and *only then* checks them, aborting upon encountering a failure (as opposed to progressing through them one by one, and aborting immediately upon encountering a failure). Clearly, this latter variant is functionally equivalent to the written one.

Lemma 3.36. *If its level-$i^*+\vartheta$ point $c_{i^*+\vartheta} \in \Delta\big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i+\vartheta-1}\big),\overline{f}^{(i^*+\vartheta)}\big)$, then the verifier rejects.*

Proof. We initialize $i \coloneqq i^* + \vartheta$, and argue inductively as follows. Our hypothesis entails exactly that $\mathsf{fold}\big(f^{(i-\vartheta)},r'_{i-\vartheta},\ldots,r'_{i-1}\big)(c_i) \neq \overline{f}^{(i)}(c_i)$. We first claim that if $\overline{f}^{(i)}(c_i) = f^{(i)}(c_i)$, then $\mathcal{V}$ rejects. Indeed, in this case, we see immediately that $\mathsf{fold}\big(f^{(i-\vartheta)},r'_{i-\vartheta},\ldots,r'_{i-1}\big)(c_i) \neq \overline{f}^{(i)}(c_i) = f^{(i)}(c_i)$; this is precisely the equality checked by the verifier in its line 5 above. If on the other hand $\overline{f}^{(i)}(c_i) \neq f^{(i)}(c_i)$ instead holds, then $c_{i+\vartheta} = \big(q^{(i+\vartheta-1)} \circ \cdots \circ q^{(i)}\big)(c_i)$ certainly satisfies $c_{i+\vartheta} \in \Delta^{(i)}\big(f^{(i)},\overline{f}^{(i)}\big)$, by definition of this latter set. Using our assumption whereby the event E_i *didn't* occur, we conclude in turn that $c_{i+\vartheta} \in \Delta\big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)\big)$. This latter set itself equals $\Delta\big(\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big),\overline{f}^{(i+\vartheta)}\big)$, by our guarantee whereby $\overline{f}^{(i+\vartheta)} = \mathsf{fold}\big(\overline{f}^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)$ (a consequence of the maximality of i^*). We see that $\mathsf{fold}\big(f^{(i)},r'_i,\ldots,r'_{i+\vartheta-1}\big)(c_{i+\vartheta}) \neq \overline{f}^{(i+\vartheta)}(c_{i+\vartheta})$. We have thus "pushed" our hypothesis on i to the same hypothesis on $i+\vartheta$. Since $\overline{f}^{(\ell-\kappa)}$ and $f^{(\ell-\kappa)}$ agree identically on $S^{(\ell-\kappa)}$, this process must eventually terminate in a rejection by the verifier. □

We are now ready to prove the proposition. Lemma 3.35 guarantees (i.e., assuming E_{i^*} doesn't occur) that $c_{i^*+\vartheta} \in \Delta\big(\mathsf{fold}\big(f^{(i^*)},r'_{i^*},\ldots,r'_{i^*+\vartheta-1}\big),\overline{f}^{(i^*+\vartheta)}\big)$ holds with probability at least $\frac{1}{|S^{(i^*+\vartheta)}|} \cdot \frac{d_{i^*+\vartheta}}{3} \geq \frac{1}{3} - \frac{1}{3\cdot 2^{\mathcal{R}}}$ in each of the verifier's query iterations. By Lemma 3.36, the verifier rejects in each such iteration (i.e., assuming none of the events $E_{i^*+\vartheta},\ldots,E_{\ell-\kappa-\vartheta}$ occur). We see that $\mathcal{V}$ accepts with probability at most $\big(\frac{2}{3} + \frac{1}{3\cdot 2^{\mathcal{R}}}\big)^{\gamma}$, which is negligible (we recall that $\mathcal{R}$ is a positive constant). This completes the proof. □

If any of $\mathcal{A}$'s oracles is noncompliant, then Proposition 3.34 (together with our assumption whereby none of the events $E_0,E_\vartheta,\ldots,E_{\ell-\kappa-\vartheta}$ occur, itself justified by Proposition 3.32) completes the proof. In particular, we may ignore the case in which Algorithm 1 outputs $P(X) = \bot$ in step 2 above, as we presently argue. Indeed, if $P(X) = \bot$, then certainly $d\big(f^{(0)},C^{(0)}\big) \geq \frac{d_0}{2}$ holds (by the correctness of the Berlekamp–Welch algorithm). Under this guarantee, we note that $d\Big(f^{(0)},\widehat{C}^{(0)}\Big) \geq \frac{d_0}{2}$ too necessarily holds; indeed, each codeword in $\widehat{C}^{(0)}$ whose entries *don't* reside exclusively in the subring $A_{\iota,\kappa,\iota} \subset A_{\iota,\kappa,\tau}$ must have at least d_0 entries not in this subring—essentially by the argument of [DP23b, Thm. 3.2]—and so must be of distance at least d_0 from $f^{(0)}$ (whose entries *do* reside exclusively in $A_{\iota,\kappa,\iota} \subset A_{\iota,\kappa,\tau}$). We thus conclude as claimed that $d\Big(f^{(0)},\widehat{C}^{(0)}\Big) \geq \frac{d_0}{2}$, so that $\mathcal{A}$'s 0^{th} oracle is bad, and the proof is complete by Proposition 3.34.

We're thus left with the case in which Algorithm 1 terminates successfully in step 2 above. We write $t(X_0,\ldots,X_{\ell-1}) \in \mathcal{T}_\iota[X_0,\ldots,X_{\ell-1}]^{\preceq 1}$ for the polynomial output by $\mathcal{E}$ in that step. We write $(r_0,\ldots,r_{\ell-1}) \in \mathcal{T}_\tau^\ell$ for the evaluation point output by $\mathcal{Q}$, and $s \in \mathcal{T}_\tau$ for $\mathcal{A}$'s response. We must show that the probability with which $s \neq t(r_0,\ldots,r_{\ell-1})$ *and* $\mathcal{V}$ accepts is negligible. It suffices to assume that $s \neq t(r_0,\ldots,r_{\ell-1})$, and to argue conditionally that $\mathcal{V}$ accepts with negligible probability. We thus assume that $s \neq t(r_0,\ldots,r_{\ell-1})$.

Under our assumption, justified above, whereby all of $\mathcal{A}$'s oracles are compliant, we inspect the value of $\mathcal{A}$'s final FRI message c. We apply Definition 3.33 inductively. In the base case $i = 0$, we note that $\overline{f}^{(0)}$ is the encoding of $P^{(0)}(X) = \sum_{v \in \mathcal{B}_{\ell-\kappa}} \hat{t}(v) \cdot X^{(0)}_{\{v\}}(X)$, precisely by $\mathcal{E}$'s construction of $\big(\hat{t}(v)\big)_{v \in \mathcal{B}_{\ell-\kappa}}$. On the other hand, for each $i \in \{0, \vartheta, \ldots, \ell - \kappa - \vartheta\}$, writing $P^{(i)}(X) \in A_{\iota,\kappa,\tau}[X]^{\prec 2^{\ell-\kappa-i}}$ for the polynomial for which $\mathsf{Enc}(P^{(i)}) = \overline{f}^{(i)}$ holds, the condition $\overline{f}^{(i+\vartheta)} = \mathsf{fold}\big(\overline{f}^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)$ implies that $\overline{f}^{(i+\vartheta)}$ is exactly the encoding of the polynomial $P^{(i+\vartheta)}(X) \in A_{\iota,\kappa,\tau}[X]^{\prec 2^{\ell-\kappa-i-\vartheta}}$ obtained by repeatedly applying the conclusion of Lemma 3.26 to $P^{(i)}(X)$ (with the folding challenges $r'_i, \ldots, r'_{i+\vartheta-1}$, respectively). Carrying out the induction, we see that $\overline{f}^{(\ell-\kappa)}$ is identically equal to $\sum_{v \in \mathcal{B}_{\ell-\kappa}} \hat{t}(v) \cdot \widetilde{\mathsf{eq}}\big(r'_0, \ldots, r'_{\ell-\kappa-1}, v_0, \ldots, v_{\ell-\kappa-1}\big)$, so that $c = \sum_{v \in \mathcal{B}_{\ell-\kappa}} \hat{t}(v) \cdot \widetilde{\mathsf{eq}}\big(r'_0, \ldots, r'_{\ell-\kappa-1}, v_0, \ldots, v_{\ell-\kappa-1}\big)$ holds.

We turn to the sumcheck. As in Construction 3.24, we write $h(X_0, \ldots, X_{\ell-1}) \coloneqq t(X_0, \ldots, X_{\ell-1}) \cdot \mathsf{eq}(r_0, \ldots, r_{\ell-1}, X_0, \ldots, X_{\ell-1})$ and $h'(X_0, \ldots, X_{\ell-\kappa-1}) \coloneqq \sum_{u \in \mathcal{B}_\kappa} h(u_0, \ldots, u_{\kappa-1}, X_0, \ldots, X_{\ell-\kappa-1})$. Since, as before, $t(r_0, \ldots, r_{\ell-1}) = \sum_{v \in \mathcal{B}_\ell} h(v)$, we see that our hypothesis $s \neq t(r_0, \ldots, r_{\ell-1})$ amounts to the condition $s \neq \sum_{v \in \mathcal{B}_\ell} h(v) = \sum_{v \in \mathcal{B}_{\ell-\kappa}} h'(v)$. The soundness analysis of the sumcheck (we refer to Thaler [Tha22, § 4.1]) states that, under this very assumption, the probability that the verifier accepts *and* $s_{\ell-\kappa} = h'(r'_0, \ldots, r'_{\ell-\kappa-1})$ holds is at most $\frac{2 \cdot (\ell-\kappa)}{|\mathcal{T}_\tau|}$ over $\mathcal{V}$'s choice of its folding challenges $(r'_0, \ldots, r'_{\ell-\kappa-1})$. We thus assume finally that $s_{\ell-\kappa} \neq h'(r'_0, \ldots, r'_{\ell-\kappa-1})$.

Putting the pieces together, we see that, under our various assumptions introduced above, we have that $s_{\ell-\kappa} \neq h'(r'_0, \ldots, r'_{\ell-\kappa-1}) = \widetilde{\mathsf{eq}}(r_\kappa, \ldots, r_{\ell-1}, r'_0, \ldots, r'_{\ell-\kappa-1}) \cdot \sum_{u \in \mathcal{B}_\kappa} c_u \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\kappa-1}, u_0, \ldots, u_{\kappa-1})$, so that $\mathcal{V}$ rejects. The final equality above is justified—i.e., under precisely the assumption whereby $c = \sum_{v \in \mathcal{B}_{\ell-\kappa}} \hat{t}(v) \cdot \mathsf{eq}\big(r'_0, \ldots, r'_{\ell-\kappa-1}, v_0, \ldots, v_{\ell-\kappa-1}\big)$—in the proof of Theorem 3.25. This completes the proof of the theorem. □

Remark 3.37. If Conjecture 2.6 were true, then we could replace the expression $\frac{d_{i+1}}{3}$ with $\frac{d_{i+1}}{2}$ throughout the proof of Theorem 3.27 (including in the second case of Definition 3.29, in the final steps of Proposition 3.30, and in the conclusion of Lemma 3.35). Moreover, we could improve the concrete soundness error at the end of that theorem's proof from $\big(\frac{2}{3} + \frac{1}{3 \cdot 2^{\mathcal{R}}}\big)^\gamma$ to $\big(\frac{1}{2} + \frac{1}{2 \cdot 2^{\mathcal{R}}}\big)^\gamma$.

3.6 Efficiency Analysis

We examine the efficiency of Construction 3.24, both asymptotic and concrete. Throughout our below analysis, we view the coefficient size parameter ι and the Reed–Solomon rate parameter $\mathcal{R}$ as constants, though we note in passing our protocol's various dependencies on these values. For simplicity, we undertake a simplified analysis, in which we allow the dependence—for both parties—on the security parameter to be polynomial. In fact, were we to instantiate the tower degree 2^τ, the hash digest width, and the repetition parameter γ appropriately carefully—specifically, by dictating that these quantities all grow *strictly* polylogarithmically (i.e., with exponent greater than 1) in λ—we could obtain a variant of our protocol with only asymptotically *polylogarithmic* dependence on λ for both the prover and the verifier. As this fact appears to be of essentially theoretical interest, we refrain from developing it (though we refer to [DP23b, Thm. 3.14] for a related treatment).

For the purposes of asymptotic analysis, we thus assume that $2^\tau = \Theta(\lambda)$ (this choice more-than-satisfies the requirement $2^\tau \geq \omega(\log \lambda)$ of $\Pi.\mathsf{Setup}$). Similarly, we assume that $\gamma = \Theta(\lambda)$, as well as that the random oracle's digests are of length λ. We finally assume throughout that ϑ is bounded from above by a constant (increasing ϑ is universally an option, and not a requirement). Though the packing factor κ—for fixed ι and $\mathcal{R}$—must grow as ℓ grows, we may arrange that it does so at most logarithmically in ℓ. Indeed, we recall that $\Pi.\mathsf{Setup}$ may choose κ minimally so that $2^{\iota+\kappa} \geq \ell - \kappa + \mathcal{R}$ holds; in particular, it may safely set $\kappa = \lceil \log(\ell + \mathcal{R}) \rceil - \iota \leq \log(\ell) + C$, where C is some constant depending on $\mathcal{R}$ and ι.

Putting these facts together, we see that each $\mathcal{T}_{\iota+\kappa}$-element takes at most $O(\ell)$ bits to represent, and that each $\mathcal{T}_\tau$-element takes $O(\lambda)$ bits to represent. Similarly, each $\mathcal{T}_{\iota+\kappa}$-operation takes $\mathrm{poly}(\ell)$ time, and each $\mathcal{T}_\tau$-operation $\mathrm{poly}(\lambda)$ time. Finally, each algebra element $A_{\iota,\kappa,\tau}$ takes $2^{\kappa+\tau} = O(\lambda \cdot \ell)$ bits to represent. Similarly, each $A_{\iota,\kappa,\tau}$-operation—and here, we refer to [DP23b, § 2.3]—takes at most $\mathrm{poly}(\lambda \cdot \ell)$ time (in fact, we may take the relevant exponent to be at most $\log(3)$).

The commitment phase of Construction 3.24 amounts to a Reed–Solomon encoding operation in the code $C^{(0)} = \mathsf{RS}_{\mathcal{T}_{\iota+\kappa}, S^{(0)}}[2^{\ell-\kappa+\mathcal{R}}, 2^{\ell-\kappa}]$. By Lin, Chung and Han [LCH14, § III. D.] (see also Algorithm 2), this

operation can be carried out in $\Theta\big((\ell-\kappa)\cdot 2^{\ell-\kappa+\mathcal{R}}\big)$ $\mathcal{T}_{\iota+\kappa}$-operations (specifically, using $(\ell-\kappa)\cdot 2^{\ell-\kappa+\mathcal{R}}$ and $(\ell-\kappa)\cdot 2^{\ell-\kappa+\mathcal{R}-1}$ $\mathcal{T}_{\iota+\kappa}$-additions and $\mathcal{T}_{\iota+\kappa}$-multiplications, respectively). The prover $\mathcal{P}$'s opening protocol entails a (truncated) sumcheck on the polynomial $h(X_0,\ldots,X_{\ell-1})$—whose individual degree in each variable is at most 2—and an execution of our 2^ϑ-ary multilinear FRI variant (see Subsection 3.3) on the $\ell-\kappa$-variate committed word f. By the sumcheck prover analysis of Thaler [Tha22, Lem. 4.5], the first task takes $\Theta(2^\ell)$ $\mathcal{T}_\tau$-operations, which represents $O(2^\ell)\cdot\mathrm{poly}(\lambda)$ total work. It follows essentially by inspection that our prover's FRI-incumbent work amounts to $\Theta(2^{\ell-\kappa+\mathcal{R}})$ $A_{\iota,\kappa,\tau}$-operations. We conclude that our prover takes $O(2^{\ell-\kappa+\mathcal{R}})\cdot\mathrm{poly}(\lambda\cdot\ell)=\widetilde{O}(2^{\ell+\mathcal{R}})\cdot\mathrm{poly}(\lambda)$ time.

In the variant of Construction 3.24 in which, by means of the BCS transform [BCS16], the use of vector oracles is eliminated, the prover must Merkle-hash $f^{(0)}$ during its commitment phase, as well as $f^{(\vartheta)},\ldots,f^{(\ell-\kappa-\vartheta)}$ during its opening proof; these commitments represent total work on the order of $\Theta\big(2^{\iota+\ell+\mathcal{R}}\big)$. During the query phase, the prover must, for each of the γ query repetitions and at each step $i\in\{\ell-\kappa-\vartheta,\ell-\kappa-2\cdot\vartheta,\ldots,0\}$, send a length-$\ell-\kappa+\mathcal{R}-i-\vartheta$ Merkle path, as well as 2^ϑ $A_{\iota,\kappa,\tau}$-elements. The total prover work during the query phase is thus $O\big(\gamma\cdot\big(\lambda\cdot(\ell-\kappa+\mathcal{R})^2+\frac{\ell-\kappa}{\vartheta}\cdot 2^\vartheta\cdot 2^{\kappa+\tau}\big)\big)=O\big(\gamma\cdot\big(\lambda\cdot(\ell-\kappa+\mathcal{R})^2+\ell\cdot O(\lambda\cdot\ell)\big)\big)=O\big(\gamma\cdot(\lambda\cdot\ell^2)\big)$. Using our further assumption $\gamma=\Theta(\lambda)$, we upper-bound the prover's work during the query phase as $O\big(\lambda^2\cdot\ell^2\big)$.

Construction 3.24's verifier complexity is essentially that of the sumcheck verifier plus that of the FRI verifier. These latter tasks entail $\Theta(\ell-\kappa)$ $\mathcal{T}_\tau$-operations and $\Theta(\gamma\cdot 2^\vartheta\cdot\frac{\ell-\kappa}{\vartheta})$ $A_{\iota,\kappa,\tau}$-operations, respectively. (We recall that the individual degree of h is at most 2, a constant.) These latter $A_{\iota,\kappa,\tau}$-operations amount to $O(\gamma\cdot\ell)\cdot\mathrm{poly}(\lambda\cdot\ell)=O(\ell^2)\cdot\mathrm{poly}(\lambda)$ total work. In the non-oracle variant of the protocol, in which the verifier must check Merkle paths, the verifier's FRI cost becomes rather $O\big(\gamma\cdot\big(\lambda\cdot(\ell-\kappa+\mathcal{R})^2+\frac{\ell-\kappa}{\vartheta}\cdot 2^\vartheta\cdot\mathrm{poly}(\lambda\cdot\ell)\big)\big)$, which is again $O(\ell^2)\cdot\mathrm{poly}(\lambda)$. Upon receiving the prover's final FRI message c, the verifier must moreover compute the evaluations $\big(\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\kappa-1},u_0,\ldots,u_{\kappa-1})\big)_{u\in\mathcal{B}_\kappa}$; this task takes $\Theta(2^\kappa)=\Theta(\ell)$ time (we recall from Subsection 2.1 the standard algorithm for this task).

The communication cost of Construction 3.24's main phase amounts to two $\mathcal{T}_\tau$-elements per round of the sumcheck, together with the final $A_{\iota,\kappa,\tau}$-element c. During the query phase—and assuming the BCS-transformed version—we encounter a further cost of $O\big(\gamma\cdot\big(\lambda\cdot(\ell-\kappa+\mathcal{R})^2+\frac{\ell-\kappa}{\vartheta}\cdot 2^\vartheta\cdot 2^{\kappa+\tau}\big)\big)=O\big(\lambda^2\cdot\ell^2\big)$ bits.

Concrete soundness. In our concrete proof size analysis below, we incorporate various fairly straightforward optimizations. For example, for each query repetition, each $i\in\{\ell-\kappa-\vartheta,\ldots,\vartheta,0\}$, and each $u\in\mathcal{B}_{\ell+\mathcal{R}-i-\vartheta}$, the required leaves $\big(f^{(i)}(v\parallel u)\big)_{v\in\mathcal{B}_\vartheta}$ are adjacent in the prover's i^{th} Merkle tree. We thus opt to send only a single shortened Merkle path—of height only $\ell-\kappa+\mathcal{R}-i-\vartheta$—as well as the 2^ϑ relevant algebra elements, at each such query step. Separately, we opt to send the *entire j^{th} layer* of the Merkle tree—as opposed to only its root—in each round $i\in\{0,\vartheta,\ldots,\ell-\kappa-\vartheta\}$, where j is an appropriately chosen constant. (For those i for which $j\geq\ell-\kappa+\mathcal{R}-i-\vartheta$, we simply send the prover's entire oracle in the clear.) Increasing j increases the fixed cost of each Merkle tree, but also causes each among the γ subsequently sent paths to become shorter. The optimal truncation height turns out to be $j\coloneqq\lceil\log_2(\gamma)\rceil$. In our below benchmarks, we use a straightforward generalization of our protocol in which the requirement $\vartheta\mid\ell-\kappa$ is dropped.

In order to appropriately select the query repetition parameter γ, we must examine the concrete security of our protocol (we refer to [DP23b, § 3.5] for an analogous analysis). It follows immediately from the proof of Theorem 3.27 that Construction 3.24's concrete soundness error is bounded from above by

$$\frac{2\cdot(\ell-\kappa)}{|\mathcal{T}_\tau|}+\frac{2^{\ell-\kappa+\mathcal{R}}}{|\mathcal{T}_\tau|}+\left(\frac{2}{3}+\frac{1}{3\cdot 2^{\mathcal{R}}}\right)^\gamma; \tag{6}$$

above, the first summand is sumcheck-specific, whereas the latter two come from Propositions 3.32 and 3.34, respectively. If Conjecture 2.6 were in fact assumed, then (6)'s final summand would become $\big(\frac{1}{2}+\frac{1}{2\cdot 2^{\mathcal{R}}}\big)^\gamma$ instead. In any case, for each desired *concrete security* level Ξ, we thus set γ minimally so that (6) (or its conjectured analogue, as the case may be) becomes bounded from above by Ξ. Clearly, this is possible only when τ is sufficiently large that $\Xi>\frac{2\cdot(\ell-\kappa)}{|\mathcal{T}_\tau|}+\frac{2^{\ell-\kappa+\mathcal{R}}}{|\mathcal{T}_\tau|}$ holds. We say in this case that Construction 3.24 attains $-\log_2(\Xi)$ *bits of security.*

We now record proof sizes. In each example, we set $\tau := 7$, as well as the rate parameter $\mathcal{R} = 2$ (so that the code's rate $\rho = \frac{1}{4}$). Throughout each benchmark, we achieve 90 bits of concrete security. In each case, we set $\kappa \geq 0$ minimally so that $2^{\iota+\kappa} \geq \ell - \kappa + \mathcal{R}$ holds, and moreover manually select the folding factor ϑ. Table 1 below records our provably secure benchmarks; these attain 90 bits of security, using $\gamma = 217$ queries. We use the Merkle tree truncation height $j = 8$.

Total Data Size	Num. Vars ℓ	Coeff. Size ι	Pack. Factor κ	Fold. Factor ϑ	Prf. Size (MiB)
32 MiB (2^{28} bits)	22	6	0	4	0.382
	25	3	2	3	0.806
	28	0	5	2	3.786
512 MiB (2^{32} bits)	26	6	0	4	0.549
	29	3	2	3	1.144
	32	0	5	2	4.889
8.192 GiB (2^{36} bits)	30	6	0	4	0.742
	33	3	3	3	2.182
	36	0	6	2	10.773

Table 1: Proof sizes of the packed Construction 3.24.

In Table 2 below, we record analogous benchmarks, in which we moreover *assume* Conjecture 2.6. In this setting, we need only $\gamma = 133$ queries throughout; we again set the Merkle cap $j = 8$.

Total Data Size	Num. Vars ℓ	Coeff. Size ι	Pack. Factor κ	Fold. Factor ϑ	Prf. Size (MiB)
32 MiB (2^{28} bits)	22	6	0	3	0.264
	25	3	2	3	0.536
	28	0	5	2	2.472
512 MiB (2^{32} bits)	26	6	0	4	0.375
	29	3	2	3	0.775
	32	0	5	2	3.154
8.192 GiB (2^{36} bits)	30	6	0	4	0.496
	33	3	3	2	1.442
	36	0	6	2	7.151

Table 2: Proof sizes of the packed Construction 3.24, *assuming* Conjecture 2.6.

We emphasize that, if Conjecture 2.6 is resolved, then the query parameter $\gamma = 133$ of Table 2 will immediately become sufficient; no further protocol changes will be needed in this setting.

4 Hybrid Brakedown–FRI-Style Protocol

In this section, we describe a further generalization of Construction 3.24. This section's generalization combines Construction 3.24 with ideas from [DP23b, Cons. 3.11]; essentially, it can be viewed as a "hybrid" protocol, which begins as does the Brakedown-style protocol of [DP23b, Cons. 3.11], and yet uses

the construction of Subsection 3.5 above for its key "inner proximity test". The protocol of this section may alternatively be viewed as an instance of the *batched FRI* technique of Ben-Sasson, et al. [Ben+23, § 8.2], where our batching combination is actually a Brakedown-style tensor fold (and where we moreover incorporate packing and the tower algebra).

The idea is, informally, to exploit the fact whereby the prover's initial oracle takes values in the *ground field* $\mathcal{T}_{\iota+\kappa}$ (while its subsequent oracles take values in the larger algebra). To take advantage of this fact, we stipulate that the prover do a *very* high-arity, Brakedown-style fold in its initial round, and proceed as in Construction 3.24 subsequently. We obtain a hybrid protocol which is more efficient for both the prover and the verifier.

We present our protocol below.

CONSTRUCTION 4.1 (Hybrid Brakedown–FRI IOPCS)**.**
We define $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ as follows.

- $\mathsf{params} \leftarrow \Pi.\mathsf{Setup}(1^\lambda, \ell, \iota)$. On input 1^λ, ℓ, and ι, choose integers ℓ_0 and ℓ_1 for which $\ell_0 + \ell_1 = \ell$, a constant, positive rate parameter $\mathcal{R} \in \mathbb{N}$, a packing factor $\kappa \geq 0$ for which $2^{\iota+\kappa} \geq \ell_1 - \kappa + \mathcal{R}$, a folding factor $\vartheta \mid \ell_1 - \kappa$, a tower height $\tau \geq \log(\omega(\log\lambda))$, and a repetition parameter $\gamma = \omega(\log(\lambda))$. Write $(X_0(X), \ldots, X_{2^{\ell_1-\kappa}-1}(X))$ for the novel $\mathcal{T}_{\iota+\kappa}$-basis of $\mathcal{T}_{\iota+\kappa}[X]^{\prec 2^{\ell_1-\kappa}}$, and fix $S^{(0)}, \ldots, S^{(\ell_1-\kappa)}$ and $q^{(0)}, \ldots, q^{(\ell_1-\kappa-1)}$ as in Subsection 3.2. Write $C^{(0)} \subset \mathcal{T}_{\iota+\kappa}^{2^{\ell_1-\kappa+\mathcal{R}}}$ for the Reed–Solomon code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa}, S^{(0)}}[2^{\ell_1-\kappa+\mathcal{R}}, 2^{\ell_1-\kappa}]$.

- $[f] \leftarrow \Pi.\mathsf{Commit}(\mathsf{params}, t)$. On input $t(X_0, \ldots, X_{\ell-1}) \in \mathcal{T}_\iota[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$, reshape t's Lagrange coefficients $(t(v))_{v\in\mathcal{B}_\ell}$ into a $2^{\ell_0} \times 2^{\ell_1}$ matrix $(t_i)_{i=0}^{2^{\ell_0}-1}$. By grouping the column indices $\{0, \ldots, 2^{\ell_1} - 1\}$ into length-2^κ chunks, and applying the natural embedding chunk-wise, express $(t_i)_{i=0}^{2^{\ell_0}-1}$ as a $2^{\ell_0} \times 2^{\ell_1-\kappa}$ matrix $\big(\hat{t}_i\big)_{i=0}^{2^{\ell_0}-1}$ with entries in $\mathcal{T}_{\iota+\kappa}$. Independently encode each row of $\big(\hat{t}_i\big)_{i=0}^{2^{\ell_0}-1}$ using $C^{(0)}$; in this way, obtain the further matrix $(f_i)_{i=0}^{2^{\ell_0}-1}$, say. For each $i \in \{0, \ldots, 2^{\ell_0} - 1\}$, submit $(\mathtt{submit}, \mathcal{T}_{\iota+\kappa}, \ell_1 - \kappa + \mathcal{R}, f_i)$ to the vector oracle. Output the list of handles $[f] \coloneqq ([f_i])_{i=0}^{2^{\ell_0}-1}$.

We define $(\mathcal{P}, \mathcal{V})$ as the following IOP, in which both parties have the common input $[f] = ([f_i])_{i=0}^{2^{\ell_0}-1}$, $s \in \mathcal{T}_\tau$, and $(r_0, \ldots, r_{\ell-1}) \in \mathcal{T}_\tau^\ell$, and $\mathcal{P}$ has the further input $t(X_0, \ldots, X_{\ell-1}) \in \mathcal{T}_\iota[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$.

- $\mathcal{P}$ computes the matrix–vector product $f^{(0)} \coloneqq \bigotimes_{i=\ell_1}^{\ell-1}(1 - r_i, r_i) \cdot (f_i)_{i=0}^{2^{\ell_0}-1}$. Interpreting this latter vector as a map $f^{(0)} : S^{(0)} \to A_{\iota,\kappa,\tau}$, $\mathcal{P}$ sends $(\mathtt{submit}, A_{\iota,\kappa,\tau}, \ell_1 - \kappa + \mathcal{R}, f^{(0)})$ to the vector oracle.

- $\mathcal{P}$ defines $t'(X_0, \ldots, X_{\ell_1-1}) \coloneqq t(X_0, \ldots, X_{\ell_1-1}, r_{\ell_1}, \ldots, r_{\ell-1})$. As in Construction 3.24, $\mathcal{P}$ moreover writes $h(X_0, \ldots, X_{\ell_1-1}) \coloneqq t'(X_0, \ldots, X_{\ell_1-1}) \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\ell_1-1}, X_0, \ldots, X_{\ell_1-1})$, and finally abbreviates $h'(X_0, \ldots, X_{\ell_1-\kappa-1}) \coloneqq \sum_{u\in\mathcal{B}_\kappa} h(u_0, \ldots, u_{\kappa-1}, X_0, \ldots, X_{\ell_1-\kappa-1})$.

- $\mathcal{P}$ and $\mathcal{V}$ run the main loop of Construction 3.24 on the values $f^{(0)}$ and s. That is, they execute:
  1: **for** $i \in \{0, \ldots, \ell_1 - \kappa - 1\}$ **do**
  2: $\quad\mathcal{P}$ sends $\mathcal{V}$ the polynomial $h'_i(X) \coloneqq \sum_{v\in\mathcal{B}_{\ell_1-\kappa-i-1}} h'(r'_0, \ldots, r'_{i-1}, X, v_0, \ldots, v_{\ell_1-\kappa-i-2})$.
  3: $\quad\mathcal{V}$ requires $s_i \overset{?}{=} h'_i(0) + h'_i(1)$. $\mathcal{V}$ samples $r'_i \leftarrow \mathcal{T}_\tau$, sets $s_{i+1} \coloneqq h'_i(r'_i)$, and sends $\mathcal{P}$ r'_i.
  4: $\quad\mathcal{P}$ defines $f^{(i+1)} : S^{(i+1)} \to A_{\iota,\kappa,\tau}$ as the function $\mathsf{fold}\big(f^{(i)}, r'_i\big)$ of Definition 3.13.
  5: $\quad$**if** $i + 1 = \ell_1$ **then** $\mathcal{P}$ sends $c \coloneqq f^{(\ell_1-\kappa)}(0, \ldots, 0)$ to $\mathcal{V}$.
  6: $\quad$**else if** $\vartheta \mid i + 1$ **then** $\mathcal{P}$ submits $(\mathtt{submit}, A_{\iota,\kappa,\tau}, \ell_1 - \kappa + \mathcal{R} - i - 1, f^{(i+1)})$ to the oracle.

- By reversing the natural embedding, $\mathcal{V}$ destructures $(c_u)_{u\in\mathcal{B}_\kappa} \coloneqq c$ as a $\mathcal{T}_\tau^{2^\kappa}$-element.

- $\mathcal{V}$ requires $s_{\ell_1-\kappa} \overset{?}{=} \widetilde{\mathsf{eq}}(r_\kappa, \ldots, r_{\ell_1-1}, r'_0, \ldots, r'_{\ell_1-\kappa-1}) \cdot \sum_{u\in\mathcal{B}_\kappa} c_u \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\kappa-1}, u_0, \ldots, u_{\kappa-1})$.

- $\mathcal{V}$ assigns $c_{\ell_1-\kappa} \coloneqq c$, and executes the following querying procedure:

1: **for** γ repetitions **do**
2: $\quad\mathcal{V}$ samples $u \leftarrow \mathcal{B}_{\mathcal{R}}$ randomly.
3: $\quad$**for** $i \in \{\ell_1 - \kappa - \vartheta, \ell_1 - \kappa - 2\cdot\vartheta, \ldots, 0\}$ (i.e., in downward order, taking ϑ-sized steps) **do**
4: $\quad\quad$for each $v \in \mathcal{B}_\vartheta$, $\mathcal{V}$ submits $(\mathtt{query}, [f^{(i)}], v \parallel u)$ to the oracle.
5: $\quad\quad\mathcal{V}$ requires $c_{i+\vartheta} \overset{?}{=} \mathsf{fold}\big(f^{(i)}, r'_i, \ldots, r'_{i+\vartheta-1}\big)(u)$.
6: $\quad\quad\mathcal{V}$ samples $v \leftarrow \mathcal{B}_\vartheta$, sets $c_i \coloneqq f^{(i)}(v \parallel u)$, and overwrites $u \coloneqq v \parallel u$.
7: $\quad$for each $i \in \{0, \ldots, 2^{\ell_0 - 1}\}$, $\mathcal{V}$ submits $(\mathtt{query}, [f_i], u)$ to the oracle.
8: $\quad\mathcal{V}$ requires $c_0 \overset{?}{=} \bigotimes_{i=\ell_1}^{\ell-1}(1 - r_i, r_i) \cdot (f_i(u))_{i=0}^{2^{\ell_0}-1}$.

We suggest the following interpretation of Construction 4.1. Construction 4.1's evaluation protocol amounts, essentially, to an execution of that of Construction 3.24 on the *partially evaluated* polynomial $t'(X_0, \ldots, X_{\ell_1 - 1}) \coloneqq t(X_0, \ldots, X_{\ell_1 - 1}, r_{\ell_1}, \ldots, r_{\ell-1})$, using moreover the evaluation point $(r_0, \ldots, r_{\ell_1 - 1}) \in \mathcal{T}_\tau^{\ell_1}$. Construction 4.1, finally, augments Construction 3.24's query phase so as to test moreover the consistency between the virtual oracle $\bigotimes_{i=\ell_1}^{\ell-1}(1 - r_i, r_i) \cdot (f_i)_{i=0}^{2^{\ell_0}-1}$ and $f^{(0)}$.

Theorem 4.2. *The IOPCS* $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ *of Construction 4.1 is complete.*

Proof. We write $(t'(v))_{v \in \mathcal{B}_{\ell_1}}$ for the vector of Lagrange coefficients of the partially evaluated polynomial $t'(X_0, \ldots, X_{\ell_1 - 1}) \coloneqq t(X_0, \ldots, X_{\ell_1 - 1}, r_{\ell_1}, \ldots, r_{\ell-1})$. We note the fundamental tensor relationship $(t'(v))_{v \in \mathcal{B}_{\ell_1}} = \bigotimes_{i=\ell_1}^{\ell-1}(1 - r_i, r_i) \cdot (t_i)_{i=0}^{2^{\ell_0}-1}$ (here, we again reshape, in row-major order, $t(X_0, \ldots, X_{\ell-1})$'s Lagrange coefficients into a $2^{\ell_0} \times 2^{\ell_1}$ matrix). Finally, we write $\big(\hat{t}'(v)\big)_{v \in \mathcal{B}_{\ell_1 - \kappa}}$ for the result of applying the natural embedding, chunk-wise, to $(t'(v))_{v \in \mathcal{B}_{\ell_1}}$.

By the linearity of $\widehat{C}^{(0)}$, we note that $f^{(0)} : S^{(\ell_1 - \kappa)} \to A_{\iota,\kappa,\tau}$ is exactly the Reed–Solomon encoding of $\big(\hat{t}'(v)\big)_{v \in \mathcal{B}_{\ell_1}}$. Applying Lemma 3.26 inductively to $f^{(0)}$, we conclude as in the proof of Theorem 3.25 that $c = \sum_{v \in \mathcal{B}_{\ell_1 - \kappa}} \hat{t}'(v) \cdot \widetilde{\mathsf{eq}}\big(r'_0, \ldots, r'_{\ell_1 - \kappa - 1}, v_0, \ldots, v_{\ell_1 - \kappa - 1}\big)$. As in that proof, we note further that the destructuring $(c_u)_{u \in \mathcal{B}_\kappa}$ has, at each of its indices $u \in \mathcal{B}_u$, the component $c_u = \sum_{v \in \mathcal{B}_{\ell_1 - \kappa}} t'(u_0, \ldots, u_{\kappa-1}, v_0, \ldots, v_{\ell_1 - \kappa - 1}) \cdot \widetilde{\mathsf{eq}}(r'_0, \ldots, r'_{\ell_1 - \kappa - 1}, v_0, \ldots, v_{\ell_1 - \kappa - 1}) = t'(u_0, \ldots, u_{\kappa-1}, r'_0, \ldots r'_{\ell_1 - \kappa - 1})$.

Assuming now that $s = t(r_0, \ldots, r_{\ell-1})$ holds—i.e., that the prover's claim is correct—we note as in the proof of Theorem 3.25 that $s = t(r_0, \ldots, r_{\ell-1}) = t'(r_0, \ldots, r_{\ell_1 - 1}) = \sum_{v \in \mathcal{B}_{\ell_1}} h(v) = \sum_{v \in \mathcal{B}_{\ell_1 - \kappa}} h'(v)$. We conclude that $\mathcal{V}$ will accept throughout its execution of the sumcheck, as well as that $s_{\ell_1 - \kappa} = h'(r'_0, \ldots, r'_{\ell_1 - \kappa - 1}) = \widetilde{\mathsf{eq}}(r_\kappa, \ldots, r_{\ell_1 - 1}, r'_0, \ldots, r'_{\ell_1 - \kappa - 1}) \cdot \sum_{u \in \mathcal{B}_\kappa} c_u \cdot \widetilde{\mathsf{eq}}(r_0, \ldots, r_{\kappa-1}, u_0, \ldots, u_{\kappa-1})$, where the latter equality is proven—i.e., under the condition whereby $c_u = t'(u_0, \ldots, u_{\kappa-1}, r'_0, \ldots, r'_{\ell_1 - \kappa - 1})$ holds for each $u \in \mathcal{B}_\kappa$—exactly as is the analogous equality in Theorem 3.25. □

In our security theorem below, we assume that the query sampler $\mathcal{Q}$ outputs a *uniformly random* evaluation point $(r_0, \ldots, r_{\ell-1}) \leftarrow \mathcal{Q}(\mathsf{params})$ in $\mathcal{T}_\tau^\ell$ for each params. This assumption is discussed at length in [DP23a, § 4.1], and is fulfilled in the protocol of [DP23b, § 5].

Theorem 4.3. *If* $\mathcal{Q}$ *is uniform, then the IOPCS* $\Pi = (\mathsf{Setup}, \mathsf{Commit}, \mathcal{P}, \mathcal{V})$ *of Construction 3.24 is secure.*

Proof. We define a straight-line emulator $\mathcal{E}$ as follows.

1. By inspecting $\mathcal{A}$'s messages to the vector oracle, $\mathcal{E}$ immediately recovers, for each $i \in \{0, \ldots, 2^{\ell_0} - 1\}$, the function $f_i : S^{(0)} \to \mathcal{T}_{\iota+\kappa}$ underlying the handle $[f_i]$ output by $\mathcal{A}$.
2. For each $i \in \{0, \ldots, 2^{\ell_0} - 1\}$, $\mathcal{E}$ runs Algorithm 1 on the word $f_i : S^{(0)} \to \mathcal{T}_{\iota+\kappa}$. If *any* among that algorithm's executions outputs $P_i(X) = \perp$, then $\mathcal{E}$ sets $t(X_0, \ldots, X_{\ell-1}) \coloneqq 0$. Otherwise, $\mathcal{E}$ expresses, for each $i \in \{0, \ldots, 2^{\ell_0} - 1\}$, the i^{th} output $P_i(X) = \sum_{v \in \mathcal{B}_{\ell_1 - \kappa}} \hat{t}_i(v) \cdot X_{\{v\}}(X)$ in coordinates with respect to the novel polynomial basis. By reversing the natural embedding on each of $P_i(X)$'s coordinates, $\mathcal{E}$ obtains the $\mathcal{T}_\iota^{\mathcal{B}_{\ell_1}}$-element $(t_i(v))_{v \in \mathcal{B}_{\ell_1}}$, say. $\mathcal{E}$ writes $t(X_0, \ldots, X_{\ell-1}) \in \mathcal{T}_\iota[X_0, \ldots, X_{\ell-1}]^{\preceq 1}$ for the polynomial whose Lagrange coordinates are given by the $2^{\ell_0} \times 2^{\ell_1}$ matrix $(t_i)_{i=0}^{2^{\ell_0}-1}$ (in row-major order).
3. $\mathcal{E}$ outputs $t(X_0, \ldots, X_{\ell-1})$ and terminates.

We now argue that $\mathcal{E}$ fulfills the requirements of Definition 3.7 with respect to the protocol Π.

We augment the proof of Theorem 3.27 with a handful of further definitions. We recall that $C^{(0)} \subset \mathcal{T}_{\iota+\kappa}^{2^{\ell_1-\kappa+\mathcal{R}}}$ denotes the Reed–Solomon code $\mathsf{RS}_{\mathcal{T}_{\iota+\kappa},S^{(0)}}[2^{\ell_1-\kappa+\mathcal{R}}, 2^{\ell_1-\kappa}]$. Following the proof of Theorem 3.27, for each $i \in \{0,\ldots,2^{\ell_0}-1\}$, in case $d\big(f_i, C^{(0)}\big) < \frac{d_0}{2}$, we write $\overline{f}_i \in C^{(0)}$ for the codeword for which $d\big(f_i,\overline{f}_i\big) < \frac{d_0}{2}$ holds. The following definition directly extends Definition 3.33:

Definition 4.4. We say that $\mathcal{A}$'s initial committed matrix $(f_i)_{i=0}^{2^{\ell_0}-1}$ is *compliant* if the conditions $d^{2^{\ell_0}}\left((f_i)_{i=0}^{2^{\ell_0}-1}, {C^{(0)}}^{2^{\ell_0}}\right) < \frac{d_0}{2}$, $d\Big(f^{(0)}, \widehat{C}^{(0)}\Big) < \frac{d_0}{2}$, and $\overline{f}^{(0)} = \bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot\big(\overline{f}_i\big)_{i=0}^{2^{\ell_0}-1}$ all hold.

It is useful to think of $\mathcal{A}$'s initial matrix $(f_i)_{i=0}^{2^{\ell_0}-1}$ informally as its "-1^{th} oracle". We define a final bad event, following Definition 3.29. Importantly—and here, we use our hypothesis on $\mathcal{Q}$—the tuple $(r_{\ell_1},\ldots,r_{\ell-1}) \leftarrow \mathcal{Q}(\mathsf{params})$ is uniformly random, and is sampled *after* $\mathcal{A}$ commits $(f_i)_{i=0}^{2^{\ell_0}-1}$.

Definition 4.5. We define the *bad subset* $E_{-1} \subset \mathcal{T}_\tau^{\ell_0}$ as the empty set if $d^{2^{\ell_0}}\left((f_i)_{i=0}^{2^{\ell_0}-1}, {C^{(0)}}^{2^{\ell_0}}\right) < \frac{d_0}{2}$ holds, and otherwise as the set of tuples $(r_{\ell_1},\ldots,r_{\ell-1}) \in \mathcal{T}_\tau^{\ell_0}$ for which $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1}, \widehat{C}^{(0)}\Big) < \frac{d_0}{3}$.

The following proposition is analogous to Proposition 3.30.

Proposition 4.6. *The probability mass* $\mu(E_{-1}) \le \ell_0 \cdot \frac{|S^{(0)}|}{|\mathcal{T}_\tau|}$.

Proof. If $d^{2^{\ell_0}}\left((f_i)_{i=0}^{2^{\ell_0}-1}, {C^{(0)}}^{2^{\ell_0}}\right) < \frac{d_0}{2}$, there's nothing to prove. Otherwise, applying the contraposition of Theorem 2.5 to the interleaved word $(f_i)_{i=0}^{2^{\ell_0}-1}$ and the code $C^{(0)}$, with the proximity parameter $e := \left\lfloor \frac{d_0-1}{3} \right\rfloor < \frac{d_0}{2}$, we conclude immediately that $\mu(E_{-1}) \le 2\cdot\ell_0\cdot\frac{e+1}{|\mathcal{T}_\tau|} \le \ell_0\cdot\frac{|S^{(0)}|}{|\mathcal{T}_\tau|}$, as required. □

We restate this proposition in the following way:

Proposition 4.7. *The probability that the bad event* E_{-1} *occurs is at most* $\ell_0 \cdot \frac{2^{\ell_1-\kappa+\mathcal{R}}}{|\mathcal{T}_\tau|}$.

Proof. This is simply Proposition 4.6. □

Combining Propositions 3.32 and 4.7, we see that the probability that *any* of the bad events $E_{-1}, E_0, E_\vartheta, \ldots, E_{\ell-\kappa-\vartheta}$ occurs is negligible. We thus assume that none of them occur.

Proposition 4.8. *If* $\mathcal{A}$*'s initial matrix is not compliant, then* $\mathcal{V}$ *accepts with at most negligible probability.*

Proof. If any among $\mathcal{A}$'s oracles $f^{(0)}, f^{(\vartheta)}, \ldots, f^{(\ell_1-\kappa-\vartheta)}$ fails to be compliant the sense of Definition 3.33 above, then Proposition 3.34 implies the proposition's conclusion. We thus assume that all of $\mathcal{A}$'s oracles $f^{(0)}, f^{(\vartheta)}, \ldots, f^{(\ell_1-\kappa-\vartheta)}$ are compliant in the sense of Definition 3.33. In particular, we obtain the guarantee $d\Big(f^{(0)}, \widehat{C}^{(0)}\Big) < \frac{d_0}{2}$. The proposition's hypothesis implies, in this case, that *either* $d^{2^{\ell_0}}\left((f_i)_{i=0}^{2^{\ell_0}-1}, {C^{(0)}}^{2^{\ell_0}}\right) \ge \frac{d_0}{2}$ *or* $\overline{f}^{(0)} \neq \bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot\big(\overline{f}_i\big)_{i=0}^{2^{\ell_0}-1}$. In each case, we have the following analogue of Lemma 3.35:

Lemma 4.9. *We have* $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1}, \overline{f}^{(0)}\Big) \ge \frac{d_0}{3}$.

Proof. We first assume that $d^{2^{\ell_0}}\left((f_i)_{i=0}^{2^{\ell_0}-1}, {C^{(0)}}^{2^{\ell_0}}\right) < \frac{d_0}{2}$ and $\overline{f}^{(0)} \neq \bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot\big(\overline{f}_i\big)_{i=0}^{2^{\ell_0}-1}$ both hold (the first of these assumptions implies that the codewords $\big(\overline{f}_i\big)_{i=0}^{2^{\ell_0}-1}$ of the second assumption's right-hand side are well-defined). The first of these assumptions implies that $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1}, \widehat{C}^{(0)}\Big) < \frac{d_0}{2}$, as well as that the codeword closest to this latter word is none other than $\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot\big(\overline{f}_i\big)_{i=0}^{2^{\ell_0}-1}$. Our

second hypothesis, whereby this latter word is unequal to $\overline{f}^{(0)}$, implies—by a reverse triangle calculation analogous to that of Lemma 3.35—that $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1},\overline{f}^{(0)}\Big)\geq d_0-\frac{d_0}{2}\geq\frac{d_0}{3}$, as desired.

In the case $d^{2^{\ell_0}}\Big((f_i)_{i=0}^{2^{\ell_0}-1},{C^{(0)}}^{2^{\ell_0}}\Big)\geq\frac{d_0}{2}$, our assumption whereby E_{-1} didn't occur entails exactly that $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1},\widehat{C}^{(0)}\Big)\geq\frac{d_0}{3}$; in this case, $d\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1},\overline{f}^{(0)}\Big)\geq\frac{d_0}{3}$ in particular certainly holds, since $\overline{f}^{(0)}\in\widehat{C}^{(0)}$ is a codeword. □

We now record an analogue of Lemma 3.36. If *any* among $\mathcal{A}$'s "standard" oracles $f^{(0)},f^{(\vartheta)},\ldots,f^{(\ell_1-\kappa-\vartheta)}$ fails to be compliant, then Lemma 3.36 serves our needs. For the purposes of the following lemma, therefore, we assume that each among $\mathcal{A}$'s oracles $f^{(0)},f^{(\vartheta)},\ldots,f^{(\ell_1-\kappa-\vartheta)}$ is compliant, but its initial matrix is *not*. Under exactly this hypothesis, we have:

Lemma 4.10. *If its level-*0 *point satisfies* $c_0\in\Delta\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1},\overline{f}^{(0)}\Big)$*, then the verifier rejects.*

Proof. The proof is analogous to that of Lemma 3.36. We first claim that if $\overline{f}^{(0)}(c_0)=f^{(0)}(c_0)$, then $\mathcal{V}$ rejects. Indeed, in this case, we see immediately under our hypothesis that $\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i(c_0))_{i=0}^{2^{\ell_0}-1}\neq\overline{f}^{(0)}(c_0)=f^{(0)}(c_0)$, so that the verifier rejects in line 8 above. If on the other hand $\overline{f}^{(0)}(c_0)\neq f^{(0)}(c_0)$ instead holds, then, by definition, $c_\vartheta=q^{(0)}(c_0)$ satisfies $c_\vartheta\in\Delta^{(0)}\big(f^{(0)},\overline{f}^{(0)}\big)$. Using our assumption whereby the event E_0 *didn't* occur, we conclude in turn that $c_\vartheta\in\Delta\big(\mathsf{fold}\big(f^{(0)},r'_0,\ldots,r'_{\vartheta-1}\big),\mathsf{fold}\big(\overline{f}^{(0)},r'_0,\ldots,r'_{\vartheta-1}\big)\big)$. This latter set itself equals $\Delta\big(\mathsf{fold}\big(f^{(0)},r'_0,\ldots,r'_{\vartheta-1}\big),\overline{f}^{(\vartheta)}\big)$, by our guarantee whereby $\overline{f}^{(\vartheta)}=\mathsf{fold}\big(\overline{f}^{(0)},r'_0,\ldots,r'_{\vartheta-1}\big)$ (we've assumed that $\mathcal{A}$'s oracles $f^{(0)},f^{(\vartheta)},\ldots,f^{(\ell_1-\kappa-\vartheta)}$ are compliant). We see that $\mathsf{fold}\big(f^{(0)},r'_0,\ldots,r'_{\vartheta-1}\big)(c_\vartheta)\neq\overline{f}^{(\vartheta)}(c_\vartheta)$. We are thus ready, in this case, to begin the induction of Lemma 3.36 at the index $i\coloneqq\vartheta$. □

The rest of the proof of the proposition proceeds exactly as does that of Proposition 3.34. We continue to assume that all of $\mathcal{A}$'s oracles $f^{(0)},f^{(\vartheta)},\ldots,f^{(\ell_1-\kappa-\vartheta)}$ are compliant (as is justified by Proposition 3.34). In this case—and assuming now the hypothesis of the proposition—Lemma 4.9 implies (i.e., assuming E_{-1} doesn't occur) that $c_0\in\Delta\Big(\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(f_i)_{i=0}^{2^{\ell_0}-1},\overline{f}^{(0)}\Big)$ holds with probability at least $\frac{1}{|S^{(0)}|}\cdot\frac{d_0}{3}\geq\frac{1}{3}-\frac{1}{3\cdot2^{\varkappa}}$ in each of the verifier's query iterations. By Lemma 3.36, the verifier rejects in each such iteration (i.e., assuming none of the events $E_0,E_\vartheta,\ldots,E_{\ell-\kappa-\vartheta}$ occur). We see that $\mathcal{V}$ accepts with probability at most $\big(\frac{2}{3}+\frac{1}{3\cdot2^{\varkappa}}\big)^{\gamma}$, which is negligible. This completes the proof. □

Applying Propositions 4.6 and 4.8 (together with Propositions 3.32 and 3.34), we assume that all of $\mathcal{A}$'s oracles, *including* its initial matrix, are compliant. Under this assumption, we note first of all that $\mathcal{E}$'s invocations of Algorithm 1 all succeed. Exactly as in $\mathcal{E}$'s description, we write, for each $i\in\{0,\ldots,2^{\ell_0}-1\}$, $\big(\hat{t}_i(v)\big)_{v\in\mathcal{B}_{\ell_1-\kappa}}$ for the vector of coefficients—in the novel polynomial basis—of the polynomial $P_i(X)$ extracted by $\mathcal{E}$, and $(t_i(v))_{v\in\mathcal{B}_{\ell_1}}$ for this vector's unpacking. Finally, we write $t(X_0,\ldots,X_{\ell-1})$ for the polynomial ultimately output by $\mathcal{E}$, and $t'(X_0,\ldots,X_{\ell_1-1})\coloneqq t(X_0,\ldots,X_{\ell_1-1},r_{\ell_1},\ldots,r_{\ell-1})$. As in the proof of Theorem 4.2, we note the tensor identity $(t'(v))_{v\in\mathcal{B}_{\ell_1}}=\bigotimes_{i=\ell_1}^{\ell-1}(1-r_i,r_i)\cdot(t_i)_{i=0}^{2^{\ell_0}-1}$. Theorem 4.2's proof shows, in precisely this situation, that $\overline{f}^{(0)}$ is the encoding of the packing $\big(\hat{t}'(v)\big)_{v\in\mathcal{B}_{\ell_1-\kappa}}$ of this latter vector, as well as that $\mathcal{P}$'s final FRI message is exactly $c=\sum_{v\in\mathcal{B}_{\ell_1-\kappa}}\hat{t}'(v)\cdot\widetilde{\mathsf{eq}}\big(r'_0,\ldots,r'_{\ell_1-\kappa-1},v_0,\ldots,v_{\ell_1-\kappa-1}\big)$, an algebra-element whose u^{th} component—for each $u\in\mathcal{B}_\kappa$—is $c_u=t'(u_0,\ldots,u_{\kappa-1},r'_0,\ldots r'_{\ell_1-\kappa-1})$. Finally, that proof shows that, in this setting, $h'(r'_0,\ldots,r'_{\ell_1-\kappa-1})=\widetilde{\mathsf{eq}}(r_\kappa,\ldots,r_{\ell_1-1},r'_0,\ldots,r'_{\ell_1-\kappa-1})\cdot\sum_{u\in\mathcal{B}_\kappa}c_u\cdot\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\kappa-1},u_0,\ldots,u_{\kappa-1})$ moreover holds.

As in the proofs of Theorems 3.27 and 4.2, our hypothesis $s\neq t(r_0,\ldots,r_{\ell-1})$ implies that $s\neq t(r_0,\ldots,r_{\ell-1})=t'(r_0,\ldots,r_{\ell_1-1})=\sum_{v\in\mathcal{B}_{\ell_1}}h(v)=\sum_{v\in\mathcal{B}_{\ell_1-\kappa}}h'(v)$. Under this hypothesis, the soundness of the sumcheck implies that—except with probability at most $\frac{2\cdot(\ell_1-\kappa)}{|\mathcal{T}_\tau|}$ over $\mathcal{V}$'s choice of folding challenges $(r'_0,\ldots,r'_{\ell_1-\kappa-1})$—we have in turn that $s_{\ell_1-\kappa}\neq h'(r'_0,\ldots,r'_{\ell_1-\kappa-1})$; in this latter case, we conclude that $s_{\ell_1-\kappa}\neq\widetilde{\mathsf{eq}}(r_\kappa,\ldots,r_{\ell_1-1},r'_0,\ldots,r'_{\ell_1-\kappa-1})\cdot\sum_{u\in\mathcal{B}_\kappa}c_u\cdot\widetilde{\mathsf{eq}}(r_0,\ldots,r_{\kappa-1},u_0,\ldots,u_{\kappa-1})$, so that $\mathcal{V}$ rejects. □

4.1 Efficiency Analysis

We examine the concrete efficiency of Construction 4.1, closely following Subsection 3.6, and focusing here on concrete efficiency. Analogously to (6), we record here the concrete soundness error of Construction 4.1:

$$\frac{2 \cdot (\ell_1 - \kappa)}{|\mathcal{T}_\tau|} + (l_0 + 1) \cdot \frac{2^{\ell_1 - \kappa + \mathcal{R}}}{|\mathcal{T}_\tau|} + \left(\frac{2}{3} + \frac{1}{3 \cdot 2^{\mathcal{R}}}\right)^{\gamma}. \tag{7}$$

The first summand above is again a sumcheck error. The middle summand corresponds jointly to Propositions 3.32 and 4.7. The final summand corresponds to Propositions 3.34 and 4.8. If Conjecture 2.6 were assumed, we could once again improve (7)'s final summand to $\left(\frac{1}{2} + \frac{1}{2 \cdot 2^{\mathcal{R}}}\right)^{\gamma}$.

We incorporate various concrete proof size optimizations, extending those discussed in Subsection 3.6. For example, we stipulate that the prover, during its commitment phase, send a Merkle root whose leaves are the respective *columns* $(f_i(u))_{i=0}^{2^{\ell_0}-1}$, for $u \in \mathcal{B}_{\ell_1 - \kappa + \mathcal{R}}$. This allows $\mathcal{P}$ to justify each column $(f_i(u))_{i=0}^{2^{\ell_0}-1}$, in line 7 above, using a *single* Merkle path of length $\ell_1 - \kappa + \mathcal{R}$.

In Table 3 below, we benchmark the proof size of Construction 4.1. In each case, we again set $\tau \coloneqq 7$, set $\mathcal{R} = 2$, and obtain 90 bits of concrete security, again using $\gamma = 217$ queries and the Merkle truncation height $j = 8$. We refrain, in the below table, from presenting benchmarks for those cases for which the choice $\kappa = 0$ is available; in each such case, Construction 3.24 beats Construction 4.1 (albeit only slightly). In each benchmark, we choose the matrix shape parameters ℓ_0 and ℓ_1 for which $\ell_0 + \ell_1 = \ell$ holds, as well as the folding factor $\vartheta \geq 1$, optimally.

Total Data Size	$\ell_0 + \ell_1 = \ell$	Coeff. Size ι	Pack. Factor κ	Fold. Factor ϑ	Prf. Size (MiB)
32 MiB (2^{28} bits)	7 + 18 = 25	3	2	3	0.682
	10 + 18 = 28	0	4	2	1.601
512 MiB (2^{32} bits)	7 + 22 = 29	3	2	3	0.950
	8 + 24 = 32	0	5	2	3.509
8.192 GiB (2^{36} bits)	8 + 25 = 33	3	2	3	1.282
	8 + 28 = 36	0	5	2	4.584

Table 3: Proof sizes of Construction 4.1.

In Table 4 below, we record analogous benchmarks, in which we moreover *assume* Conjecture 2.6. In this setting, we need only $\gamma = 133$ queries throughout; we again set the Merkle cap $j = 8$. x

Total Data Size	$\ell_0 + \ell_1 = \ell$	Coeff. Size ι	Pack. Factor κ	Fold. Factor ϑ	Prf. Size (MiB)
32 MiB (2^{28} bits)	7 + 18 = 25	3	2	3	0.441
	10 + 18 = 28	0	4	2	1.119
512 MiB (2^{32} bits)	7 + 22 = 29	3	2	3	0.622
	8 + 24 = 32	0	5	2	2.963
8.192 GiB (2^{36} bits)	6 + 27 = 33	3	2	3	0.822
	8 + 28 = 36	0	5	2	2.962

Table 4: Proof sizes of Construction 4.1.

We note that Construction 4.1's proofs are smaller than those of Construction 3.24 by as much as half, with the gains most prominent in the case $\iota = 0$.

References

[AHIV23] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: lightweight sublinear arguments without a trusted setup". In: *Designs, Codes and Cryptography* (2023). DOI: 10.1007/s10623-023-01222-8.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed–Solomon Interactive Oracle Proofs of Proximity". In: *International Colloquium on Automata, Languages, and Programming*. Ed. by Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella. Vol. 107. Leibniz International Proceedings in Informatics. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 14:1–14:17.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *International Conference on Theory of Cryptography*. Vol. 9986. Berlin, Heidelberg: Springer-Verlag, 2016, pp. 31–60. ISBN: 978-3-662-53644-5. DOI: 10.1007/978-3-662-53644-5_2.

[Ben+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. "Proximity Gaps for Reed–Solomon Codes". In: *Journal of the ACM* 70.5 (Oct. 2023). DOI: 10.1145/3614423.

[Ber15] Elwyn Berlekamp. *Algebraic Coding Theory*. Revised Edition. World Scientific Publishing, 2015.

[CBBZ23] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. "HyperPlonk: Plonk with Linear-Time Prover and High-Degree Custom Gates". In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14005. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023.

[DP23a] Benjamin E. Diamond and Jim Posen. *Proximity Testing with Logarithmic Randomness*. Cryptology ePrint Archive, Paper 2023/630. 2023. URL: https://eprint.iacr.org/2023/630.

[DP23b] Benjamin E. Diamond and Jim Posen. *Succinct Arguments over Towers of Binary Fields*. Cryptology ePrint Archive, Paper 2023/1784. 2023. URL: https://eprint.iacr.org/2023/1784.

[FP97] John L. Fan and Christof Paar. "On efficient inversion in tower fields of characteristic two". In: *Proceedings of IEEE International Symposium on Information Theory*. 1997.

[GG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. 3rd Edition. Cambridge University Press, 2013.

[Gur06] Venkatesan Guruswami. *Algorithmic Results in List Decoding*. Vol. 2. Foundations and Trends in Theoretical Computer Science 2. now publishers, 2006.

[Hab22] Ulrich Haböck. *A summary on the FRI low degree test*. Cryptology ePrint Archive, Paper 2022/1216. 2022. URL: https://eprint.iacr.org/2022/1216.

[LCH14] Sian-Jheng Lin, Wei-Ho Chung, and Yunghsiang S. Han. "Novel Polynomial Basis and Its Application to Reed–Solomon Erasure Codes". In: *IEEE 55th Annual Symposium on Foundations of Computer Science*. 2014, pp. 316–325. DOI: 10.1109/FOCS.2014.41.

[Li+18] Wen-Ding Li, Ming-Shing Chen, Po-Chun Kuo, Chen-Mou Cheng, and Bo-Yin Yang. "Frobenius Additive Fast Fourier Transform". In: *ACM International Symposium on Symbolic and Algebraic Computation*. 2018. ISBN: 9781450355506. DOI: 10.1145/3208976.3208998.

[Set20] Srinath Setty. "Spartan: Efficient and General-Purpose zkSNARKs Without Trusted Setup". In: *Advances in Cryptology – CRYPTO 2020*. Ed. by Daniele Micciancio and Thomas Ristenpart. Cham: Springer International Publishing, 2020, pp. 704–737. ISBN: 978-3-030-56877-1. DOI: 10.1007/978-3-030-56877-1_25.

[Tha22] Justin Thaler. *Proofs, Arguments and Zero-Knowledge*. Vol. 4. Foundations and Trends in Privacy and Security 2–4. now publishers, 2022.

[Wie88] Doug Wiedemann. "An Iterated Quadratic Extension of $GF(2)$". In: *The Fibonacci Quarterly* 26.4 (1988), pp. 290–295.

[ZCF23] Hadas Zeilberger, Binyi Chen, and Ben Fisch. *BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes*. Cryptology ePrint Archive, Paper 2023/1705. 2023. URL: https://eprint.iacr.org/2023/1705.