Zinc: Succinct Arguments with Small Arithmetization Overheads from IOPs of Proximity to the Integers

Albert Garreta, Hendrik Waldner, Katerina Hristova, and Luca Dall'Ava

Nethermind Research

{albert,hendrik.waldner}@nethermind.io,
{katherina.hristova,lucadallava.work}@gmail.com

Abstract

We introduce Zinc, a hash-based succinct argument for integer arithmetic. Zinc's goal is to provide a practically efficient scheme that bypasses the arithmetization overheads that many succinct arguments present. These overheads can be of orders of magnitude in many applications. By enabling proving statements over the integers, we are able to arithmetize many operations of interest with almost no overhead. This includes modular operations involving any moduli, not necessarily prime, and possibly involving multiple moduli in the same statement. In particular, Zinc allows to prove statements for the ring $\mathbb{Z}/n\mathbb{Z}$ for arbitrary $n \geq 1$. Importantly, and departing from prior work, our schemes are purely code and hash-based, and do not require hidden order groups. In its final form, Zinc operates similarly to other hash-based schemes using Brakedown as their PCS, and at the same time it benefits from the arithmetization perks brought by working over $\mathbb{Z}$ (and $\mathbb{Q}$) natively.

At its core, Zinc is a succinct argument for proving relations over the rational numbers $\mathbb{Q}$, even though when applied to integer statements, an honest prover and verifier will only operate with small integers. Zinc consists of two main components: 1) Zinc-PIOP, a framework for proving algebraic statements over the rationals by reducing modulo a randomly chosen prime q, followed by running a suitable PIOP over $\mathbb{F}_q$ (this is similar to the approach from [CHA24], with the difference that we use localizations of $\mathbb{Q}$ to enable prime modular projection); and 2) Zip, a Brakedown-type polynomial commitment scheme built from an *IOP of proximity to the integers*, a novel primitive that we introduce. The latter primitive guarantees that a prover is using a polynomial with coefficients close to being integral. With these two primitives in place, one can use a lookup argument over the rationals to ensure that the witness contains only integer elements.

Contents

1 Introduction

SNARGs are proof systems allowing to succinctly prove the validity of statements in arbitrary NP relations. However, the most efficient SNARGs (and SNARKs) to date are specifically designed to handle relations $\mathsf{REL}_{\mathbb{F}}$ that are expressed using algebraic operations over a fixed finite field $\mathbb{F}$. On the other hand, there are many relations of interest, say REL, which are naturally independent of the field $\mathbb{F}$, maybe because they are not algebraic in nature (say, e.g. a graph coloring problem), or because they are expressed over a different field $\mathbb{F}'$ (perhaps the field of rational numbers $\mathbb{Q}$ when proving statements concerning ML models), or even a ring such as $\mathbb{Z}/2^n\mathbb{Z}$ (as is the case for CPU-related operations and operations concerning some cryptographic primitives like FHE). Hence, in these cases, when seeking to prove statements from REL, one is forced to rewrite REL into an equivalent relation $\mathsf{REL}_{\mathbb{F}}$ over $\mathbb{F}$. This process is called *arithmetization*, and it can easily lead to an increase in statement or circuit sizes of a factor of 2^5 or more [BFK+24, OKMZ24]. We loosely call such a factor the *arithmetization overhead*.

When considering, for example, state-of-the-art SNARKs used in industry such as Stwo [HLP24], we see that more than 80% of the proving cost is related to computing the trace witnessing the validity of the statement, then computing the Low-Degree Extension of the trace, and then Merkle-committing to it [Eli]. The cost of all these steps is dramatically affected by the statement size, which, as we mentioned above, can incur overheads of a factor more than 2^5 when the relation is not naturally expressed in Stwo's underlying field, i.e. the Mersenne 31 field. On the other hand, less than 20% of the proving cost in Stwo comes from executing the actual SNARK after the witness has been computed, encoded, and committed. This suggests that the next big step towards reducing the cost of proving may be found in reducing arithmetization costs.

In this work, we present Zinc, a framework for building hash-based SNARKs that enable proving statements for relations expressed using integer arithmetic. As we argue below, this type of arithmetic is highly expressive and can be used, among others, to prove statements involving any type of modular arithmetic (possibly, using multiple moduli in the same relation), with essentially no arithmetization overhead. Moreover, for the most part, Zinc operates as a regular hash-based succinct argument executed modulo a random prime, and when it doesn't, we make sure that the prover and verifier always work with integers of $\approx \lambda$ bit-size (assuming the relation REL admits witnesses with entries of $\approx \lambda$ bit-size), where λ is the security parameter. As a result, Zinc provides a scheme that works similarly to other state-of-the-art hash-based arguments, but that is capable of handling statements involving arbitrary moduli with almost no arithmetization overhead.

A key building block of Zinc is Zip, a Brakedown-like [GLS+23] polynomial commitment scheme (PCS) for multilinear polynomials with coefficients being *close* to integers. Indeed, Zip is built from what we call an *IOP of Proximity to the Integers*, which we believe is of independent interest.

The benefits of integer arithmetic Before delving deeper into our contributions, we justify the expressiveness and convenience of integer arithmetic through an example. Suppose we are interested in proving the following R1CS constraint over a finite field $\mathbb{F}_q$ with a prime number q of elements:

$$\forall\, \mathbf{y} \in \{0,1\}^\mu;$$
$$\left(\sum_{\mathbf{x}\in\{0,1\}^\mu} \mathbf{A}(\mathbf{y},\mathbf{x})\cdot z(\mathbf{x})\right)\cdot\left(\sum_{\mathbf{x}\in\{0,1\}^\mu} \mathbf{B}(\mathbf{y},\mathbf{x})\cdot z(\mathbf{x})\right)-\left(\sum_{\mathbf{x}\in\{0,1\}^\mu} \mathbf{C}(\mathbf{y},\mathbf{x})\cdot z(\mathbf{x})\right) =_{\mathbb{F}_q} 0, \tag{1}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are public multilinear polynomials on 2μ variables, z is a witness multilinear polynomial on μ variables, and $\circ$ denotes the Hadamard product (i.e. component-wise multiplication). Here, the prover P claims knowledge of such a polynomial z. All polynomials have coefficients in the field $\mathbb{F}_q$, which we interpret as the subset $[0, q-1]$ of the integers $\mathbb{Z}$. Above, we wrote $=_{\mathbb{F}_q}$ to emphasize that the equality holds modulo q. Similarly, given an arbitrary ring $\mathcal{R}$, we write $=_{\mathcal{R}}$ to denote equality over the ring $\mathcal{R}$.

Denoting the left-hand side of (1) by $Q(\mathbf{y})$, we have that (1) is equivalent to the following constraint over the integers

$$\forall\ \mathbf{y} \in \{0,1\}^{\mu};\ Q(\mathbf{y}) =_{\mathbb{Z}} q \cdot u(\mathbf{y}), \tag{2}$$

where $u(\mathbf{y})$ is a multilinear polynomial on μ variables with coefficients in $\mathbb{Z}$. Thus, instead of proving knowledge of z satisfying (1), P can prove knowledge of integral (i.e. with integer coefficients) polynomials z and u satisfying (2).

This technique can be extended so as to handle R1CS statements involving any moduli, with possible multiple moduli per statement. Indeed, let m be a vector of size 2^{μ} with integer entries indexed by elements from $\{0,1\}^{\mu}$, i.e. $\mathsf{m} = (\mathsf{m}_{\mathbf{y}} \mid \mathbf{y} \in \{0,1\}^{\mu}) \in \mathbb{Z}^{2^{\mu}}$. Now consider, instead of (2), the following constraint, which generalizes (2):

$$\forall\ \mathbf{y} \in \{0,1\}^{\mu};\ Q(\mathbf{y}) =_{\mathbb{Z}} \mathsf{m}_{\mathbf{y}} \cdot u(\mathbf{y}), \tag{3}$$

Each equality $Q(\mathbf{y}) =_{\mathbb{Z}} \mathsf{m}_{\mathbf{y}} \cdot u(\mathbf{y})$ is equivalent to $Q(\mathbf{y}) = \mathsf{m}_{\mathbf{y}} \cdot u(\mathbf{y}) \mod \mathsf{m}_{\mathbf{y}}$. Hence, Eq. (3) is a R1CS statement where each constraint is equivalent to a constraint modulo $\mathsf{m}_{\mathbf{y}}$. So, in particular, (3) captures statements involving multiple moduli. Moreover, *these moduli do not need to be prime numbers*, they could be, for example, 2^{32} or 2^{64} –a typical modulo of interest due to many CPU and cruptographic operations occurring under this arithmetic.

We also remark that, in this specific use-case of integer arithmetic, if the constraint is satisfiable, then one can always find a satisfying integer witness of "small" (i.e. $\approx 2\lambda$) bit-size. Indeed, notice that, if Eq. (3) is satisfiable, then there exists a witness z, u such that all coefficients of z are in the interval $[0, \max_{\mathbf{y}}\{\mathsf{m}_{\mathbf{y}}\} - 1]$. This is because Eq. (3) is equivalent to a system of statements involving modular arithmetic with m as moduli, and so z can always correspond to values reduced mod m. Further, all the coefficients of u can be bounded, roughly, as $2^{2\mu} \cdot \|\mathbf{A}\|_{\infty} \cdot \|\mathbf{B}\|_{\infty} \cdot \max_{\mathbf{y}}\{\mathsf{m}_{\mathbf{y}}\}^2$. Overall, we have that if the bit-size of $\mathsf{m}_{\mathbf{y}}$ is $\approx \lambda$ for all $\mathbf{y} \in \{0,1\}^{\mu}$, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ contain small entries, then Eq. (3) is satisfiable with an integer witness with entries of bit-length at most $\approx 2\lambda$.

Design principles With these observations in mind, we focus on designing a SNARK for all sorts of algebraic relations over the integers $\mathbb{Z}$, including (3), and the CCS relation [STW23a] over $\mathbb{Z}$. To do so, we set a series of design principles, which we explain next.

1. **Using error correcting codes and collision resistant hash functions.** As mentioned earlier, we seek to design a SNARK that operates similarly to other hash-based state-of-the-art SNARKs used in industry [HLP24, Plo, DP23, BG23]. These are SNARKs based on error-correcting codes and collision resitant hash functions. In particular, we seek to depart from [CHA24], where Campanelli and Hall-Andersen build a SNARK for integer arithmetic by relying on hidden order groups, which introduce

high computational overheads and are not plausibly post-quantum secure. We expand upon this point in Section 2.3, where we discuss different ways of constructing a polynomial commitment scheme for polynomials where all coefficients are (close to) being integral.

2. **Avoid working with large integers or rational numbers.** It is relatively simple to design a PIOP for the CCS relation over the integers. Indeed, one can easily adapt SuperSpartan [STW23a] to work over $\mathbb{Z}$. However, doing so would result in a scheme where both prover and verifier would need to operate with integers of thousands of bits. Given that integer multiplication does not have a linear cost, this could outweigh the savings achieved by reducing arithmetization overheads. To address this problem, we draw inspiration from [CHA24] and execute our protocols modulo a random prime of roughly λ bits. As a result, because we use hash-based cryptography and because we work modulo a random prime, for the most part, Zinc *operates similarly to a regular hash-based SNARK over a random* λ*-bit prime.*

Applications We envision Zinc to find a large number of use-cases. Particularly, many scenarios where arithmetization overheads are problematic could benefit from Zinc. Some such scenarios are: statements involving RSA-group arithmetic (i.e. additions and multiplications modulo a product of two primes), which are relevant in standardized cryptographic primitives like RSA signatures used, e.g. in OAuth, ECDSA signatures, etc.; arithmetic modulo 2^n for some n, e.g. CPU operations, operations pertaining FHE schemes, and some treatments of floating point operations [CCKP19]; performing recursive proving for different proof systems at once (this in particular avoids the problem of finding pairs of friendly elliptic curves, which are less efficient and well-understood than standard curves); avoiding wrong-field arithmetizations in the recursive step of incremental verifiable computation (IVC) schemes; etc.

Additionally, as we discuss below, Zinc is also capable of proving statements expressed over the field of rational numbers. As such, Zinc can find use-cases in proving computations involving (approximations of) real numbers, such as ML-related operations, or finance-related computations.

The main components and ideas behind Zinc Next, we provide a high-level outline of the main building blocks of Zinc. We refer to our technical overview (Section 2) for a detailed explanation of these and the main ideas involved.

We present two main protocols: Zinc-PIOP and Zip. Formally speaking, the first is a framework that provides polynomial interactive oracle proofs (PIOPs) for proving algebraic statements over the field of rational numbers $\mathbb{Q}$ (with bounded bit-size). The second is a Brakedown-like polynomial commitment scheme (PCS) [GLS+23] (expressed using IOPs) that allows to commit to multilinear polynomials with rational coefficients. Put together, we obtain an interactive oracle proof (IOP) for computations over bounded rational numbers, which can then be compiled with Merkle trees using standard methods [CY24, COS20].

When seeking to prove statements over the integers, we can enforce provers to actually use integer witnesses by using Zinc-PIOP to build a lookup argument for membership into a bounded subset of $\mathbb{Z}$.

Our Zinc-PIOP framework works similarly as the *mod-AHP* framework from Campanelli and Hall-Andersen [CHA24]. Mainly, V samples a random prime, and then P and V execute a suitable PIOP over the finite field $\mathbb{F}_q$. The main difference is that [CHA24] works entirely with integers, while we work over the rationals. This makes projection onto the finite field $\mathbb{F}_q$ a subtler matter, and introduces some complexity when analyzing the soundness and completeness of the resulting schemes. Mathematically, we use the notion of *rational localization* to treat such projections. Namely, the set $\mathbb{Z}_{(q)} = \{a/b \in \mathbb{Q} \mid q \text{ does not divide } b\}$ is a subring of $\mathbb{Q}$ which admits a projection onto $\mathbb{F}_q$.

As mentioned, Zip is a Brakedown-like polynomial commitment scheme [GLS+23] for multilinear polynomials with (bounded) rational coefficients. Zip is based on what we call an *IOP of proximity (IOPP) to the integers*. Intuitively, Zip is meant to be used to commit to polynomials with (bounded) integer coefficients, but it only guarantees that the coefficients are rational numbers of a slightly large bit-size (i.e. rationals which are "close" to being integer). This scenario is analogous to what one has with IOP of proximity to a linear code (see e.g. [BSBHR18]). Such a primitive guarantees that a prover knows a word that is close to being a codeword, but it is meant to be used by provers that actually know the codeword.

We emphasize that, despite being schemes that formally work over the rationals, when used honestly, Zinc-PIOP and Zip only require the prover and the verifier to operate with small sized integers, and with field elements.

Related work Arguably, the closest work to ours is Campanelli and Hall-Andersen's [CHA24]. As we mentioned, [CHA24] presents a SNARK for proving statements over the integers, relying on hidden order groups. Precisely, [CHA24] crucially uses a polynomial commitment scheme for polynomials with integer coefficients of Block et al. [BHR+21] which, in turn, relies on [BFS20]. While in our work we depart from hidden order groups, we adopt the idea from [CHA24] of working modulo a random prime (and we extend it so that the idea can be used when operating with rational numbers).

In Rinnochio [GNSV23], the authors construct a delegated verifier scheme for proving statements over Galois rings, which, among others, have the potential of allowing to prove computations modulo 2^n with a small arithmetization overhead. The main drawback of [GNSV23] is that the Galois ring needs to be very large, namely with elements of bit-size around $n \cdot \lambda$, in order to guarantee λ bits of security, creating a large embedding overhead [DP23]. The reason this occurs is that [GNSV23] requires working over a ring with a large, so-called, exceptional set. Further works that build schemes over rings with large exceptional sets (and thus incur embedding overheads) are [BCS21, ACC+22, SV22]. Unlike Zinc, none of these approaches rely solely on collision-resistant hash functions. Other schemes supporting non-prime arithmetics are lattice-based ones such as LaBRADOR [BS23], though they often suffer from embedding overheads and are not hash-based.

Two recent concurrent works [HMZ25, WZD25] propose SNARKs for proving statements over certain cyclotomic rings and over Galois rings, respectively. Ultimately, both schemes end up creating proofs for statements over Galois rings. Interestingly, both protocols use a hash-based and code-based PCS. Precisely, the authors of both works [HMZ25, WZD25] propose versions of the Brakedown PCS [GLS+23] for polynomials whose coefficients are Galois rings elements. The first reference uses an underlying linear code based on Reed-Solomon codes, while the second uses an expand-accumulate code, as we do in Zip. [HMZ25] proposes a technique for reducing the size of the rings used during the creation of the proof, though we note that the prover is still required to commit to polynomials with coefficients in the initial Galois ring.

Other approaches for proving statements modulo 2^n are based on VOLE techniques (see, e.g. [LXY24, BBMH+21, BBMHS22]) but yield non-succinct schemes.

Binius and FRI-binius [DP23, DP24] can be seen as contributing to the effort of reducing arithmetization overheads, since, by working over binary fields, the schemes can handle bit-wise operations with little overhead. Similarly, recent improvements in industry-used SNARKs have sought to work with prime fields $\mathbb{F}_q$ that allow for faster arithmetization of certain operations, e.g. with q being the Mersenne 31 prime [HLP24] or the Babybear prime [Plo] (there are other reasons why working with such primes is beneficial). Finally, all efforts around designing *field-agnostic SNARKs* [BCG+17, BCG20, GLS+23, ZCF23, BFK+24] (i.e. SNARKs that can be instantiated over any field) can be understood as contributing to reducing arithmetization overheads, since one can always instantiate them over the most convenient field, depending on the relation being proved. Besides also being capable of such, Zinc can handle multiple arithmetics at the same time and in the same instantation, and is not restricted to prime arithmetic.

1.1 Acknowledgements

This work was supported by the Ethereum Foundation ZK grant FY24-1490 [Fou]. We thank Ahmet Ramazan Ağırtaş, Alexander R. Block, Remco Bloemen, Matteo Campanelli, Zahra Javar, Ignacio Manzur, Justin Thaler, Michał Zając, and Yevgeny Zaytman for insightful conversations.

2 Technical overview

As mentioned earlier, one of our primary goals is to efficiently prove CCS instances over the integers. For simplicity, let us restrict to R1CS-like constraints such as (2). We emphasize that our techniques support any relation whose constraints are algebraic –a class or relations that we call algebraic indexed relation (cf. Definition 4.1). Concretely, for the purposes of this technical overview, we informally define the following relation, which we call *R1CSℓ* as in *R1CS with ℓifted modules*, and which captures the constraints (3). Fix a size bound $B \geq 1$, and let $\mathbb{Z}_B$ be the set of integers with bit-size less than B. Let $\mathbf{A}, \mathbf{B}, \mathbf{C}$ be three $n \times n$ matrices with entries from $\mathbb{Z}_B$, and let $\mathbf{m} \in \mathbb{Z}_B^n$. Then we define $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$ as

$$\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}} = \left\{ (\mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathbb{x} = (\mathbf{x}),\ \mathbf{x} \in \mathbb{Z}_B^k, \\ \mathbb{w} = (\mathbf{w}, \mathbf{u}),\ \mathbf{w} \in \mathbb{Z}_B^{n-k-1}, \mathbf{u} \in \mathbb{Z}_B^n, \\ \mathbf{z} = (\mathbf{w}, \mathbf{x}, 1), \\ \left(\mathbf{A} \cdot \mathbf{z}^\intercal\right) \circ \left(\mathbf{B} \cdot \mathbf{z}^\intercal\right) = \left(\mathbf{C} \cdot \mathbf{z}^\intercal\right) + \mathbf{m}^\intercal \circ \mathbf{u}^\intercal \end{array} \right\}$$

where $\circ$ is the Hadamard product (i.e. component-wise multiplication), $\intercal$ denotes transposition, and $k \geq 0$ is a parameter.

In practice, we think of B as $\mathsf{poly}(\lambda)$. One may wonder if this places too much of a restriction on the expressiveness of the relation $\mathsf{REL}_{\mathsf{R1CS},\mathbb{Q}_B}$. We argue that, in practice, this is not the case: first, when it comes to the constraints of the form (3), which lift modular arithmetic onto $\mathbb{Z}$, one can easily see that, as long as $\mathbf{x}$ and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ have entries of size, say $\approx \lambda$, any satisfying witness has size at least $\approx \lambda^2$. Further, in general, if we think of an R1CS constraint as a depth-d layered arithmetic circuit, then, if the circuit admits a valid witness, there is always one such witness of size $\approx d \cdot \lambda$.

A first attempt A relatively naïve attempt towards designing a SNARK for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$ is to start with a suitable polynomial interactive oracle proof (PIOP) for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$, in which the prover (even a malicious one) is guaranteed to send polynomials whose coefficients are integers from $\mathbb{Z}_B$. Let us call such a PIOP a *PIOP over* $\mathbb{Z}_B$. It is relatively straightforward to adapt the original Spartan PIOP [Set19] over a finite field $\mathbb{F}$ into a PIOP over $\mathbb{Z}_B$ for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$. The scheme can be obtained by simply replacing all oracle polynomials from $\mathbb{F}[\mathbf{X}]$, i.e. the ring of polynomials with coefficients in the field $\mathbb{F}$ sent by the prover, with polynomials from $\mathbb{Z}_B[\mathbf{X}]$, i.e. the ring of polynomials whose coefficients are integers with encoding at most B bits, and then making some simple modifications to the verifier so its challenges are integers. Since in a PIOP over $\mathbb{Z}_B$ we are allowed to make the very strong assumption that provers, even malicious ones, always send polynomials from $\mathbb{Z}_B$, the knowledge soundness of the scheme follows with standard arguments involving Scwhartz-Zippel lemma (which holds over any integral domains such as $\mathbb{Z}$, since integral domains are always contained in a field). We note that completeness is also simple to establish, perhaps with the only point worth commenting upon being that multilinear extensions over the ring $\mathbb{Z}$ work in the same way as multilinear extensions over fields.

We emphasize that such a PIOP over $\mathbb{Z}_B$ is not described, nor used, in this paper. The only reason we are discussing it is to illustrate what goes wrong when naïvely trying to design a SNARK for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$.

One initial objection to this approach is in regards to its efficiency: the PIOP we have outlined is, essentially, "Spartan running over integers". This means that, as is, the sumchecks used in the PIOP will require the prover and verifier to operate with increasingly large integers. For example, say we set $B = 256$ and $n = 2^{20}$ (which makes sumchecks have around 20 rounds), and we have the verifier sample ≤ 128-bit integer challenges. Then the integers used in the PIOP can grow up to over thousands of bits. In [CHA24], Campanelli and Hall-Andersen address this issue by, essentially, running the PIOP over $\mathbb{Z}_B$ modulo a random prime sampled by the verifier, with the restriction that the initial oracle polynomials sent by the prover are from $\mathbb{Z}_B[\mathbf{X}]$. The authors call this a *mod-algebraic holographic proof.* As we will see later, we adopt the same idea, but instead of restricting ourselves to an integral setting, we will work over the field of rational numbers and its local subrings, which admit modding out by all but one prime (see Section 2.1 below).

As the next step of our naïve attempt, one would then compile the PIOP over $\mathbb{Z}_B$ into a succinct argument by leveraging a PCS, adapting some of the compilers from, e.g. [CHM+19, BFS20]. Crucially, since the security of PIOPs over $\mathbb{Z}_B$ only holds if provers are guaranteed to send polynomials from $\mathbb{Z}_B[\mathbf{X}]$, here one is required to use a PCS that guarantees this property, i.e. a PCS that allows to extract bounded integral polynomials. However, this is not a trivial task at all. We note that [CHA24] also requires this.

To our knowledge, the only PCS for integral polynomials that guarantees extraction of integral polynomials is due to Block et al. [BHR+21], cf. also [CHA24] for an improvement in regards to communication complexity. The PCS in [BHR+21] extends some of the ideas from Bünz et al. in [BFS20], where, in [BFS20], a PCS for polynomials of the form f/N is presented, with $f \in \mathbb{Z}_B[\mathbf{X}]$ and $N \geq 1$ is a bounded integer. We refer to Section 5.2 from [CHA24] for an informative overview of the two references [BFS20, BHR+21]. The resulting PCSs (both in [BHR+21] and [CHA24]) rely on hidden order groups, which introduce significant efficiency and technical overheads. Further, these groups require the RSA assumption which makes the resulting constructions insecure against quantum adversaries.

Key design choice: moving to the field of rational numbers One of our main objectives is to design a SNARK that relies solely on the random oracle heuristic, and so using hidden order groups is something we seek to avoid. Since there is no apparent way to design a PCS for integer polynomials without these, in this work, we choose to depart from the above attempt where one starts with a PIOP over the integers and compiles it with a PCS for integral polynomials. **Instead, we set up to work over the rational numbers $\mathbb{Q}$ and to require extraction of bounded rational polynomials** (i.e. polynomials whose coefficients belong to $\mathbb{Q}$), rather than integral polynomials. With $\mathbb{Q}$ being a field, extraction becomes much simpler, and this unlocks a plethora of design directions. Indeed, we build our PCS as a variation of Brakedown [GLS+23], as we discuss later on.

Our techniques are highly general. We use them to design both a SNARK for the CCS relation over $\mathbb{Q}$, and a lookup argument which we use to prove set membership of subsets of $\mathbb{Q}$. Using the latter to prove membership to the ring $\mathbb{Z}$, we obtain a SNARK for the CCS relation over $\mathbb{Z}$.

Next, we present an overview of the main components of Zinc.

2.1 Constructing a PIOP over $\mathbb{Q}$ from a collection of PIOPs over finite fields

We begin by reformulating the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$ into the field of rational numbers $\mathbb{Q}$. For technical reasons, we additionally add an oracle to the witness as part of the public instance (this is a usual technicality occurring in many schemes in general). Further, since we are interested in working with polynomials, instead of having the witness be two vectors $(\mathbf{w}, \mathbf{u})$ with entries in $\mathbb{Q}_B$ (where $\mathbb{Q}_B$ is the set of rational numbers with bit-size less than B), we instead have the witness consist of multilinear polynomials with coefficients in $\mathbb{Q}_B$. Namely, we use the multilinear extensions of $\mathbf{w}, \mathbf{u}$, which we denote by $\tilde{\mathbf{w}}, \tilde{\mathbf{u}}$, respectively.

Below, given a multilinear polynomial $\tilde{\mathbf{v}}$, we use $[[\tilde{\mathbf{v}}]]$ to denote an oracle to such vector. This is an idealized object that allows querying $\tilde{\mathbf{v}}$ at arbitrary points, but does not require reading or storing $\tilde{\mathbf{v}}$ in its entirety. In practice, these oracles are replaced by commitments computed using a polynomial commitment scheme.

We now define

$$\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}} = \left\{ (\mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathbb{x} = (\mathbf{x}, [[\tilde{\mathbf{w}}]], [[\tilde{\mathbf{u}}]]),\ \mathbf{x} \in \mathbb{Z}_B^k, \\ \mathbb{w} = (\tilde{\mathbf{w}}, \tilde{\mathbf{u}}),\ \mathbf{w} \in \mathbb{Q}_B^{n-k-1}, \mathbf{u} \in \mathbb{Q}_B^n, \\ \mathbf{z} = (\mathbf{w}, \mathbf{x}, 1), \\ \left(\mathbf{A} \cdot \mathbf{z}^\top\right) \circ \left(\mathbf{B} \cdot \mathbf{z}^\top\right) = \left(\mathbf{C} \cdot \mathbf{z}^\top\right) + \mathbf{m}^\top \circ \mathbf{u}^\top \end{array} \right\}$$

where $\mathbb{Q}_B$ is the set of rational numbers with bit-size less than B, $\mathbf{m} \in \mathbb{Z}_B^n$ is a vector with entries in $\mathbb{Z}_B$, and $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are three $n \times n$ matrices with entries in $\mathbb{Z}_B$. For simplicity, in this technical overview we omit all technicalities regarding the usage of indexes, indexers, and prover and verifier parameters (cf. Section 3.4).

Our goal is still to allow a prover to prove statements over the integers. As such, we expect the statements being proved to be over the integers, and this is why $\mathbf{x}$ is defined as an integer vector above, and why the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are integral as well. Similarly, and this will be very relevant for technical reasons later, **an honest prover is expected to be using a witness $(\tilde{\mathbf{w}}, \tilde{\mathbf{u}})$ such that $(\mathbf{w}, \mathbf{u}) \in \mathbb{Z}_{B'}^{2n-k-1}$**, rather than $(\mathbf{w}, \mathbf{u}) \in \mathbb{Q}_B^{2n-k-1}$, for certain $B' \leq B$. This is a similar scenario to how, in interactive oracle proofs of proximity (IOPP) to a code, the honest prover is expected to be using a codeword, but the IOPP allows to prove only that the prover is using words that are close to the codeword (cf. Section 2.3 for more details on this matter regarding rationals vs integers). We remark that, in any case, our techniques can be extended to work purely over the rational numbers.

We next describe a PIOP over $\mathbb{Q}$ for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$, in which we assume that provers (possibly malicious) are bound to use polynomials whose coefficients (in Lagrange basis) belong to $\mathbb{Q}_B$ (again, an honest prover however will want to use polynomials with coefficients in $\mathbb{Z}_{B'}$) or in a suitable finite field. When this is the case, we informally say that the PIOP is *over* $\mathbb{Q}_B$. In a nutshell, the PIOP over $\mathbb{Q}_B$ simply has the verifier sample a random $\Omega(\lambda)$-bit prime q, and then P and V execute a PIOP for the R1CSℓ relation over finite fields, modding out the entries in $(\mathbb{x}; \mathbb{w})$ by q. In what follows, we unpack how this is done.

First, let q be a prime number, and let $\mathbb{Z}_{(q)}$ denote the set of rational numbers $a/b \in \mathbb{Q}$ such that q does not divide b. This set is called the *localization of* $\mathbb{Q}$ *at the ideal* (q), and forms a subring of $\mathbb{Q}$. Additionally, $\mathbb{Z}_{(q)}$ admits a natural projection (an exhaustive ring homomorphism) onto $\mathbb{F}_q$:

$$\begin{aligned} \phi_q : \mathbb{Z}_{(q)} &\to \mathbb{F}_q \\ a/b &\mapsto a \cdot b^{-1} \mod q, \end{aligned} \tag{4}$$

where b^{-1} denotes a multiplicative inverse of b modulo q. We extend the notation ϕ_q so that it applies component-wise to vectors, tuples, and matrices. For example, if $\mathbf{r} = (r_1, \ldots, r_n)$ is a vector of elements from $\mathbb{Z}_{(q)}$, then $\phi_q(\mathbf{r}) = (\phi_q(r_1), \ldots, \phi_q(r_n))$.

Given a prime q, we define a relation $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ which is exactly the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}}$, with the difference that the equality $(\mathbf{A} \cdot \mathbf{z}^\mathsf{T}) \circ (\mathbf{B} \cdot \mathbf{z}^\mathsf{T}) = (\mathbf{C} \cdot \mathbf{z}^\mathsf{T}) + \mathbf{m}^\mathsf{T} \circ \mathbf{u}^\mathsf{T}$ is replaced with the equality $(\phi_q(\mathbf{A}) \cdot \phi_q(\mathbf{z})^\mathsf{T}) \circ (\phi_q(\mathbf{B}) \cdot \phi_q(\mathbf{z})^\mathsf{T}) =_{\mathbb{F}_q} (\phi_q(\mathbf{C}) \cdot \phi_q(\mathbf{z})^\mathsf{T}) + \phi_q(\mathbf{m})^\mathsf{T} \circ \phi_q(\mathbf{u})^\mathsf{T}$. More precisely,

$$\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}}) = \left\{ (\mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathbb{x} = (\mathbf{x}, [[\tilde{\mathbf{w}}]], [[\tilde{\mathbf{u}}]]),\ \mathbf{x} \in \mathbb{Z}_B^k, \\ \mathbb{w} = (\tilde{\mathbf{w}}, \tilde{\mathbf{u}}),\ \mathbf{w} \in (\mathbb{Z}_{(q)})_B^{n-k-1}, \mathbf{u} \in (\mathbb{Z}_{(q)})_B^n, \\ \mathbf{z} = (\mathbf{w}, \mathbf{x}, 1), \\ \left(\phi_q(\mathbf{A}) \cdot \phi_q(\mathbf{z})^\mathsf{T}\right) \circ \left(\phi_q(\mathbf{B}) \cdot \phi_q(\mathbf{z})^\mathsf{T}\right) \\ \qquad\qquad =_{\mathbb{F}_q} \left(\phi_q(\mathbf{C}) \cdot \phi_q(\mathbf{z})^\mathsf{T}\right) + \phi_q(\mathbf{m})^\mathsf{T} \circ \phi_q(\mathbf{u})^\mathsf{T} \end{array} \right\},$$

where $(\mathbb{Z}_{(q)})_B$ denotes the set of rational numbers with bit-size less than B that belong to $\mathbb{Z}_{(q)}$. Note that the last constraint is well-defined, since $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{x}$ are integral, and $\mathbb{w}$

contains entries in $\mathbb{Z}_{(q)}$, and so all elements in such constraint are suitable arguments for the homomorphism ϕ_q. This relation is similar to the so-calle *associated fingerprinting relation* from [CHA24].

For large enough B (i.e. $B \geq \log(q)$), the relation $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ is essentially an R1CS relation (with the extra term $\phi_q(\mathbf{m})^\mathsf{T} \circ \phi_q(\mathbf{u})^\mathsf{T}$) over the finite field $\mathbb{F}_q$, with the difference that the instance and the witnesses are specified as elements of the ring $\mathbb{Z}_{(q)}$ rather than elements from $\mathbb{F}_q = \phi_q(\mathbb{Z}_{(q)})$. For the purpose of proving instances from $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$, this is an irrelevant distinction, since, one can prove, $(\mathbb{x}; \mathbb{w}) \in \phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ if and only if $(\phi_q(\mathbb{x}); \phi_q(\mathbb{w}))$ satisfies the R1CS relation (with the extra term) over $\mathbb{F}_q$. With this in mind, it is not difficult to turn a PIOP over $\mathbb{F}_q$ for the latter relation, e.g. Spartan [Set19], into a PIOP over $\mathbb{Z}_{(q)}$ for $(\mathbb{x}; \mathbb{w}) \in \phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$.

We emphasize again the following aspect regarding the security of PIOPs. When we speak of a PIOP over $\mathbb{Q}_B$, or over $\mathbb{Z}_{(q)}$, for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ or $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}}$, we only consider security against provers that send polynomials of appropriate degree and number of variables, and whose coefficients belong to $\mathbb{Q}_B$ or $(\mathbb{Z}_{(q)})_B$ (or some prescribed subset of these or a quotient of these). Further, we only consider security for instances $\mathbb{x} = (\mathbf{x}, [[\tilde{\mathbf{w}}]], [[\tilde{\mathbf{u}}]])$ such that $(\mathbf{w}, \mathbf{u})$ belongs to $\mathbb{Q}_B^{2n-k-1}$ or to $\mathbb{Z}_B^{2n-k-1}$, respectively. In that case we say that $\mathbb{x}$ is *well-formed.*

Our PIOP over $\mathbb{Q}_B$ for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ is built from PIOPs over $\mathbb{Z}_{(q)}$ for $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$, for different primes q. The scheme is described informally in Protocol 1 below.

Protocol 1 Zinc-PIOP: A PIOP over $\mathbb{Q}_B$ for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ from PIOPs over $\mathbb{Z}_{(q)}$ for $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ (Informal).

Input: P and V receive $(\mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ and $(\mathbb{x})$ as input, respectively. Write $\mathbb{w} = (\tilde{\mathbf{w}}, \tilde{\mathbf{u}})$ with $(\mathbf{w}, \mathbf{u}) \in \mathbb{Q}_B^{2n-k-1}$.

1: V uniformly samples a prime q of $\Omega(\lambda)$ bits.

2: If $(\mathbf{w}, \mathbf{u})$ contains a rational number whose denominator is divisible by q, i.e. if $(\mathbf{w}, \mathbf{u}) \notin \mathbb{Z}_{(q)}^{n-1-k}$, P indicates V which entry of $\mathbb{w}$ contains such an entry. Then V checks (by querying an appropriate oracle) that this is indeed the case, and if so, it accepts the proof and the protocol terminates.

3: Otherwise, P and V execute a PIOP over $\mathbb{Z}_{(q)}$ for the relation $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ with prover input $(\mathbb{x}; \mathbb{w})$ and with verifier input $(\mathbb{x})$.

We next argue that Protocol 1 is perfectly complete and knowledge sound if the PIOPs for $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ are.

Knowledge soundness error We provide an intuitive explanation of why Protocol 1 has negligible knowledge soundness error, if configured appropriately.

Assume a malicious prover P^*, given input $\mathbb{x}$, is able to convince the verifier V with probability ε. Let q be the prime sampled by V at Step 1 of Protocol 1. Let $\mathcal{E}_{\mathsf{wf}}$ be the event that $(\mathbf{w}, \mathbf{u}) \in (\mathbb{Z}_{(q)})_B^{2n-k-1}$. In this event, $\mathbb{x}$ is a well-formed instance for $\phi(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$, and P^* is able to convince the verifier V_q of the PIOP for the relation $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ with probability ε_q, where ε_q is the soundness error of this PIOP. Since we assume the latter PIOP is knowledge sound, there is an extractor that is able to extract a witness $\mathbb{w}^* = (\tilde{\mathbf{w}}^*, \tilde{\mathbf{u}}^*)$

with $(\mathbf{w}^*, \mathbf{u}^*) \in (\mathbb{Z}_{(q)})^{2n-k-1}$ such that $(\mathbb{x}; \mathbb{w}^*) \in \phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$ with probability at least $\geq \varepsilon_q - \mathsf{negl}(\lambda)$. In particular, since $\mathbb{x}$ contains oracles $[[\mathbf{w}]], [[\mathbf{u}]]$ to the witness, we have $\mathbf{w}^* = \mathbf{w}, \mathbf{u}^* = \mathbf{u}$. Let $\mathbb{w} = (\tilde{\mathbf{w}}, \tilde{\mathbf{u}})$. Now, if $(\mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$, then we are done. Otherwise, since $(\mathbb{x}; \mathbb{w}) = (\mathbb{x}; \mathbb{w}^*) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}}$, we have

$$\mathbf{A} \cdot \mathbf{z}^\mathsf{T} \circ \mathbf{B} \cdot \mathbf{z}^\mathsf{T} - \mathbf{C} \cdot \mathbf{z}^\mathsf{T} - \mathbf{m}^\mathsf{T} \circ \mathbf{u}^\mathsf{T} = q \cdot \mathbf{v}_q \tag{5}$$

for some nonzero vector $\mathbf{v}_q \in \mathbb{Z}^n_{(q)}$ (it is nonzero because, otherwise, we would have $(\mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$), where $\mathbf{z} = (\mathbf{w}, \mathbf{x}, 1)$. Let I be the set of primes that can be sampled by V and that satisfy such property, and let $Q(\mathbf{z})$ be the left-hand side of Eq. (5). Then we have that for each prime $q \in I$, there exists a nonzero vector $\mathbf{v}_q \in \mathbb{Z}_{(q)}$ such that $Q(\mathbf{z}) = q \cdot \mathbf{v}_q$.

Assume for a moment that the vectors $\mathbf{v}_q$, as well as $\mathbf{z}$ and $\mathbf{u}$, are integral (but note that in general they are rational). Then we have

$$Q(\mathbf{z}) = \mathbf{A} \cdot \mathbf{z}^\mathsf{T} \circ \mathbf{B} \cdot \mathbf{z}^\mathsf{T} - \mathbf{C} \cdot \mathbf{z}^\mathsf{T} - \mathbf{m}^\mathsf{T} \circ \mathbf{u}^\mathsf{T} = \left(\prod_{q \in I} q\right) \cdot \mathbf{v}$$

for some nonzero integral vector $\mathbf{v}$. This means that some entry, say $Q(\mathbf{z})_i$ of the vector $Q(\mathbf{z})$ has bit-size at least $\log(\prod_{q \in I} q)$. Assuming, say, that $\log(q) = \lambda$, we have that $Q(\mathbf{z})_i$ has bit-size at least $\lambda \cdot |I|$. Since the entries of $Q(\mathbf{z})$ are polynomial expressions on the entries of $A, B, C, \mathbf{z}, \mathbf{m}$, and $\mathbf{u}$, and we assumed that all of these contain entries in $\mathbb{Z}_B$, this places a polynomial $\mathsf{poly}(\lambda)$ upper bound on $|I|$, assuming $B, n = \mathsf{poly}(\lambda)$.

Hence the probability that V samples q such that $(\mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}}$ but $(\mathbb{x}; \mathbb{w}) \notin \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ is at most $\mathsf{poly}(\lambda)/|\mathcal{P}|$, where $\mathcal{P}$ is the set of primes that can be sampled by V. Choosing $|\mathcal{P}| = O(2^\lambda)$ we obtain that this probability is negligible.

More precisely, and in the general case where $\mathbf{v}, \mathbf{z}, \mathbf{u}$ contain rational entries, we obtain a similar bound using the following general result.

Lemma 2.1 (Informal, cf. Definition 4.7 and Proposition 4.6)**.** *Let $P(\mathbf{Y}) = P(Y_1, \ldots, Y_\mu)$ be a polynomial with coefficients in $\mathbb{Z}_B$. Let I be a set of primes of bit-size at least λ, and let $\mathbf{y} \in \mathbb{Q}_B^\mu$ be such that $\mathbf{y} \in \mathbb{Z}^\mu_{(q)}$ for all $q \in I$. Note that then $P(\mathbf{y}) \in \mathbb{Z}_{(q)}$. Assume that $\phi_q(P(\mathbf{y})) = 0$ for all primes q in I. Then one of the entries of $\mathbf{y}$ has bit-size at least*

$$\frac{\lambda \cdot |I| - (B + 2^{\mathsf{degp}(P)})}{\mathsf{degp}(P)},$$

where $\mathsf{degp}(P) = \sum_{i \in [\mu]} \deg_{Y_i}(P)$ is the sum of the partial degrees of P.

Hence, if we know that the entries in $\mathbf{y}$ have bit-size less than B, one concludes that $|I| < \lambda^{-1} \cdot (B \cdot \mathsf{degp}(P) + B + 2^\mu) = \mathsf{poly}(\lambda)$ if $B, 2^\mu, \mathsf{degp}(P) = \mathsf{poly}(\lambda)$.

We now argue that $\mathcal{E}_{\mathsf{wf}}$ holds except with negligible probability. This will conclude our proof outline. Indeed, if $\mathcal{E}_{\mathsf{wf}}$ does not hold for a prime q, then q divides some denominator in the vector $(\mathbf{w}, \mathbf{u})$. Using the pigeonhole principle, it is clear that if $\Pr[\neg\mathcal{E}_{\mathsf{wf}}]$ is large, then there is an entry in $(\mathbf{w}, \mathbf{u})$ whose denominator is divisible by many primes. This makes such entry have a large bit-size, but we have assumeed that the entries in $(\mathbf{w}, \mathbf{u})$ have bit-size less than $B = \mathsf{poly}(\lambda)$.

Remark 2.2. We emphasize the importance in the above analysis of having assumed that $(\mathbf{w}, \mathbf{u})$ are rational numbers of bit-size less than B. Without this guarantee, none of the error bounds we found can be made negligible. When compiling the PIOP into an IOP or into a succinct argument, it will be up to the PCS to make sure this bound holds.

Lookup relations over $\mathbb{Q}$, and forcing provers to use integral witnesses. Above, we constructed a PIOP over $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ from PIOPs for the standard R1CS relation (with an extra term) over finite fields. In many occasions, we can use the same ideas to "lift" PIOPs for other relations over finite fields, into a PIOP for the corresponding relation over $\mathbb{Q}_B$. In particular, we are able to do this for the lookup relation, i.e. a relation that constraints all entries in a witness vector $\mathbf{a}$ to appear as entries in a public vector $\mathbf{t}$. This ultimately provides us with a PIOP for proving membership to any subset S of $\mathbb{Q}_B$. In particular, taking $S = [-2^B + 1, 2^B - 1]$, we are able to force PIOP provers to use integral witness within the range $[-2^B + 1, 2^B - 1]$. Throughout the paper, all intervals of the form $[n, m]$, $n, m \in \mathbb{Z}$, denote the set of integers $\{n, n+1, \ldots, m\}$.

More precisely, consider the following relation, which we call *lookup relation*. Let $n, m \geq 0$ be two vector sizes, and $B \geq 1$ a bit-size bound. Then we define[1]

$$\mathsf{REL}_{\mathsf{Look},\mathbb{Q}} := \left\{ \begin{pmatrix} \mathbb{x} = ([[\tilde{\mathbf{a}}]], [[\tilde{\mathbf{t}}]]); \\ \mathbb{w} = (\tilde{\mathbf{a}}, \tilde{\mathbf{t}}) \end{pmatrix} \middle| \begin{array}{l} \mathbf{a} \in \mathbb{Q}_B^{n_a},\ \mathbf{t} \in \mathbb{Q}_B^{n_t}, \\ \{\mathbf{a}_i \mid i \in [n_a]\} \subseteq \{\mathbf{t}_j \mid j \in [n_t]\} \end{array} \right\},$$

and similarly as before, for q a prime,

$$\phi_q(\mathsf{REL}_{\mathsf{Look},\mathbb{Z}_{(q)}}) := \left\{ \begin{pmatrix} \mathbb{x} = ([[\tilde{\mathbf{a}}]], [[\tilde{\mathbf{t}}]]); \\ \mathbb{w} = (\tilde{\mathbf{a}}, \tilde{\mathbf{t}}) \end{pmatrix} \middle| \begin{array}{l} \mathbf{a} \in (\mathbb{Z}_{(q)})_B^{n_a},\ \mathbf{t} \in (\mathbb{Z}_{(q)})_B^{n_t}, \\ \{\phi_q(\mathbf{a}_i) \mid i \in [n_a]\} \subseteq \{\phi_q(\mathbf{t}_j) \mid j \in [n_t]\} \end{array} \right\},$$

As was the case with $\phi_q(\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}_{(q)}})$, it is relatively simple to construct a PIOP over $\mathbb{Z}_{(q)}$ for the relation $\phi_q(\mathsf{REL}_{\mathsf{Look},\mathbb{Z}_{(q)}})$ from a PIOP for the standard lookup relation over the field $\mathbb{F}_q$. Then, a similar construction as Protocol 1 provides a PIOP over $\mathbb{Q}$ for $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$ with perfect completeness and negligible knowledge soundness error.

Achieving generality We emphasize that our formal treatment (cf. Section 4) of these techniques is fully general. We work over abstract rings and with any relation REL whose constraints can be described algebraically. The abstract rings we can support must satisfy certain technical conditions regarding existence of suitable projection morphisms (cf. Definition 4.7), which are satisfied when working with $\mathbb{Q}, \mathbb{Z}$, $\mathbb{Z}_{(q)}$, and $\mathbb{F}_q$. An informal idea of what these requirements are may be intuited from our previous informal explanation of why Protocol 1 has negligible knowledge soundness error.

2.2 Obtaining PIOP's for integral relations

We now put together the constructions from Section 2.1 to build a PIOP *over* $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$, which is the relation we wanted to treat initially. Essentially, the PIOP

[1] In full formality, the polynomial $\tilde{\mathbf{t}}$ and its oracle would be placed in an index, rather than in $\mathbb{x}$ and $\mathbb{w}$, cf. Section 3.5.

simply has the prover and the verifier execute two PIOPs over $\mathbb{Q}_B$, one for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ and one for $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$. In the latter case, the vector $\mathbf{a}$ is taken to be the witness $\mathbb{w}$ used in $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$, and $\mathbf{t}$ is the vector of integers in the interval $[-2^B+1, 2^B-1]$. As vector sizes for the relation $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$, we set n_a to be $2n-k-1$, and n_t to be $2^{B+1}+1$.

Protocol 2 A PIOP over $\mathbb{Q}_B$ for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$.

Input: P, V receive inputs $(\mathbb{x};\mathbb{w}) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$ and $(\mathbb{x})$. Write $\mathbb{x} = (\mathbf{x}, [[\tilde{\mathbf{w}}]], [[\tilde{\mathbf{u}}]])$, with $(\mathbf{w}, \mathbf{u}) \in \mathbb{Z}_B^{2n-k-1}$, $\mathbf{x} \in \mathbb{Z}_B^k$.

1: P and V execute a PIOP over $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ with inputs $(\mathbb{x};\mathbb{w})$ and $\mathbb{x}$, respectively.

2: Then, P and V execute a PIOP over $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$ with inputs $(\mathbb{x}_{\mathsf{Look}};\mathbb{w}_{\mathsf{Look}})$ and $\mathbb{x}_{\mathsf{Look}}$, respectively, where $\mathbb{x}_{\mathsf{Look}} = ([[(\tilde{\mathbf{w}}, \tilde{\mathbf{u}})]], [[\tilde{\mathbf{t}}]])$, $\mathbb{w}_{\mathsf{Look}} = ((\tilde{\mathbf{w}}, \tilde{\mathbf{u}}), \tilde{\mathbf{t}})$, and where $\mathbf{t}$ is a vector containing all integers in the range $[-2^B+1, 2^B-1]$.

Remark 2.3. As we justify below, Protocol 2 is a PIOP *over* $\mathbb{Q}_B$, meaning that it is knowledge sound against malicious provers that are guaranteed to send polynomials with coefficients in $\mathbb{Q}_B$, or in a suitable finite field, and for well-formed instances $\mathbb{x}$.

In particular, knowledge soundness is not restricted to malicious provers sending polynomials with coefficients in $\mathbb{Z}_B$ (or a finite field). This is crucial, since, $\mathbb{Q}$ being a field, places a much lighter burden on the PCS when compiling the PIOP into a succinct argument.

We next oultine why Protocol 2 has negligible completeness and soundness error, assuming the PIOPs for $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ and $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$ have negligible such errors. Let us denote these two PIOPs by $\Pi_{\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}}, \Pi_{\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}}$. It is clear that, if P is honest, then it will convince V as long as the honest provers of $\Pi_{\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}}$ and of $\Pi_{\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}}$ convince their respective verifiers. Hence, Protocol 2 has negligible completeness error under our assumptions.

Next, we address knowledge soundness. As mentioned above, we want to show that Protocol 2 is knowledge sound against malicious provers that send polynomials with coefficients in $\mathbb{Q}_B$ (as opposed to only in $\mathbb{Z}_B$), or in some suitable finite field. To prove this, informally speaking, we have to show that if such a malicious prover P^* manages to convince the verifier with non-negligible probability, then it is possible to extract a witness $\mathbb{w}^* = (\tilde{\mathbf{w}}^*, \tilde{\mathbf{u}}^*)$ with $(\mathbf{w}^*, \mathbf{u}^*) \in \mathbb{Z}_B^{2n-k-1}$, such that $(\mathbb{x};\mathbb{w}^*) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$. For P^* to be able to convince V reliably, it has to be able to convince the verifiers of both PIOPs $\Pi_{\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}}$ and $\Pi_{\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}}$. Since we assume that these two PIOPs *over* $\mathbb{Q}_B$ have negligible knowledge soundness, it is possible, except with negligible probability, to extract from P^* witnesses $\mathbb{w}^* = (\tilde{\mathbf{w}}^*, \tilde{\mathbf{u}}^*)$ with $(\mathbf{w}^*, \mathbf{u}^*) \in \mathbb{Q}_B^{2n-k-1}$ and $\mathbb{w}^{**}_{\mathsf{Look}} = ((\tilde{\mathbf{w}}^{**}, \tilde{\mathbf{u}}^{**}), \tilde{\mathbf{t}}^{**})$ with $(\tilde{\mathbf{w}}^{**}, \tilde{\mathbf{u}}^{**}) \in \mathbb{Q}_B^{2n-k-1}$, such that $(\mathbb{x};\mathbb{w}^*) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ and $(\mathbb{x}_{\mathsf{Look}};\mathbb{w}^{**}) \in \mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$. In particular, since both $\mathbb{x}$ and $\mathbb{x}_{\mathsf{Look}}$ contain oracles $(\tilde{\mathbf{w}}, \tilde{\mathbf{u}})$ and $[[\tilde{\mathbf{t}}]]$ to the witness polynomials, we have $\mathbf{w}^* = \mathbf{w}^{**} = \mathbf{w}$, $\mathbf{u}^* = \mathbf{u}^{**} = \mathbf{u}$, and $\mathbf{t}^{**} = \mathbf{t}$. Now, since $(\mathbb{x}_{\mathsf{Look}};\mathbb{w}^{**}_{\mathsf{Look}}) = (\mathbb{x}_{\mathsf{Look}}; ((\tilde{\mathbf{w}}, \tilde{\mathbf{u}}), \tilde{\mathbf{t}})) \in \mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$, we have that $(\mathbf{w}, \mathbf{u}) \in \mathbb{Z}_B^{2n-k-1}$, i.e. $(\mathbf{w}, \mathbf{u})$ consists of integer entries. Since we have $(\mathbb{x};\mathbb{w}^*) = (\mathbb{x}; (\tilde{\mathbf{w}}, \tilde{\mathbf{u}})) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$, we obtain that $(\mathbb{x};\mathbb{w}^*) \in \mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Z}}$, as needed.

2.3 Zip: A PCS over $\mathbb{Q}_B$ from an IOP of Proximity (IOPP) to the Integers

So far, we outlined the construction of a PIOP over $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ (and for $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$), see Protocol 1. Our next goal is to turn this PIOP into a succinct argument by compiling it with a polynomial commitment scheme (PCS)[2]. As emphasized previously, the security of these PIOPs is only guaranteed against malicious provers that send polynomials of appropriate degrees, number of variables, and with coefficients either in $\mathbb{Q}_B$ or in a suitable finite field. This applies as well to the polynomials in the instance and witnesses of $\mathsf{REL}_{\mathsf{R1CS}\ell,\mathbb{Q}}$ and $\mathsf{REL}_{\mathsf{Look},\mathbb{Q}}$. Hence, our PCS, which we call Zip, must ensure that provers are committing to such polynomials, i.e. we must make sure that it is possible to extract such polynomials.

For technical reasons that will become apparent later, another important aspect to consider is that we expect honest provers to actually use polynomials with coefficients in $\mathbb{Z}_{B'}$ for some $B' < B$, instead of in $\mathbb{Q}_B \setminus \mathbb{Z}_{B'}$. As such, we allow ourselves the freedom to design a PCS whose completeness is only guaranteed when P is using polynomials with coefficients in $\mathbb{Z}_{B'}$ (including the witness polynomials), but whose extractor is only guaranteed to output polynomials with coefficients in $\mathbb{Q}_B$. In other words, Zip is a PCS for polynomials with coefficients in $\mathbb{Q}_B$ whose completeness is only guaranteed when provers use polynomials with coefficients in $\mathbb{Z}_{B'}$ for some suitable $B' < B$, with B being polynomial on B' and other parameters. Precisely, $B = B' + (2\mathsf{dim} + 1) \cdot (\lambda + \log(\mathsf{dim}))$, where dim is, roughly, the square root of the length of witnesses.

This situation is very similar to what occurs with interactive oracle proofs of proximity (IOPPs) to a code $\mathcal{C}$, see e.g. [BSBHR18]. In such an IOPP, the honest prover P is expected to use a codeword as a witness, but the scheme can only guarantee that P is using a word that is close to a codeword. On the other hand, if P does not use a codeword as a witness, then the IOPP does not guarantee completeness. In our scenario, Zip is a PCS for polynomials whose coefficients belong to $\mathbb{Q}_B$ –in that case, in a sense, they are close to being integers– but the honest prover is expected to actually use integral polynomials, of a bit-size smaller than B. Because of this, we say that Zip is built from an ***Interactive Oracle Proof of Proximity to the Integers***.

The IOP underlying Zip is not only an IOPP to the Integers in the above sense, it is also an IOPP to certain linear codes, as we see next. Indeed, Zip is based on the Brakedown PCS [GLS+23] and its instantiation from [BFK+24] via Expand-Accumulate (EA) codes (for reasons we will explain later). The variation works at times over the field of rational numbers, and modulo a random prime. We put special care to make sure that, in the former case, the honest prover and the verifier only ever handle integers of small bit-size. Our variation of Brakedown includes an additional crucial check from the verifier which forces provers to use witnesses with rational entries of bit-size at most B. Completeness is only guaranteed for honest provers using witnesses with integer entries of bit-size at most B', where B' is described above. Additionally, Zip allows to prove polynomial evaluations at integral points.

We remark that Zip can be extended to allow honest provers to actually use polynomials with coefficients in $\mathbb{Q}_{B'}$ and with bounded denominators, instead of polynomials with coefficients in $\mathbb{Z}_{B'}$.

[2] In fact, our compilation proceeds in a slightly different way: namely, our PCS is in "IOP form". Then we compile our PIOP with this PCS, which results in an IOP, and then we compile this IOP using the iCOS/iBCS transformation [CY24]. This is similar as how compilation is achieved in works such as [COS20].

2.3.1 The commitment and testing phase

As we mentioned, Zip is a variation of Brakedown [GLS+23]. The PCS relies on a linear code $\mathcal{C}$ over $\mathbb{Q}$ of dimension dim and length n (i.e. a $\mathbb{Q}$-vector subspace of $\mathbb{Q}^{\mathsf{n}}$ of dimension dim). In Section 2.3.3 we discuss how we construct such a code (as well as in Section 6).

Let $f(\mathbf{X}) = f(X_1, \ldots, X_\mu)$ be a multilinear polynomial with coefficients in $\mathbb{Q}_B$, with an even number of variables μ. Say P wishes to commit to f. To do so, P proceeds analogously as in the original Brakedown scheme. Namely, P organizes the coefficients of f into a $\mathsf{dim} \times \mathsf{dim}$ matrix $\mathbf{v}^f$, where $\mathsf{dim} = 2^{\mu/2}$. Then, P computes the encoding $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i^f)$ with respect to the code $\mathcal{C}$ of each row $\mathbf{v}_i^f$ ($i \in [\mathsf{dim}]$) of $\mathbf{v}^f$. The commitment to f is then the vector $(\mathsf{Commit}(\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i^f)))_{i\in[\mathsf{dim}]}$, where Commit denotes a vector commitment, such as the Merkle tree commitment scheme.

The commitment is supposed to be followed by an interactive testing phase, which we describe in simplified form in Protocol 3.

Protocol 3 Zip's testing procedure (informal).

Input: Let $\mathbb{x} = (\mathsf{cm}_i)_{i\in[\mathsf{dim}]}$ with each cm_i being supposedly a vector commitment to a word $\hat{\mathbf{u}}_i$ which is close to the code $\mathcal{C}$, i.e. so that $\hat{\mathbf{u}}_i$ agrees with a unique codeword $\hat{\mathbf{v}}_i = \mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i) \in \mathcal{C}$ on "many" positions. P and V receive $(\mathbb{x})$ as input. P additionally receives $\mathbb{w} = (\mathbf{v}_i)_{i\in[\mathsf{dim}]}$, the coefficient matrix of a polynomial f with coefficients in $\mathbb{Q}_B$, as input. The words $\hat{\mathbf{u}}_i$ are given to P as part of its prover parameters. // Unlike when designing our PIOP in Section 2.1, on this occasion we place no assumption on the bit-sizes of the entries in $\hat{\mathbf{u}}_i$ and $\mathbf{v}_i$, $i \in [\mathsf{dim}]$.

1: V sends P uniformly sampled elements $r_1, \ldots, r_{\mathsf{dim}} \in [0, 2^\lambda - 1]$[3].
2: P sends V the vector $\bar{\mathbf{v}} = \sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i \in \mathbb{Q}^{\mathsf{dim}}$.
3: If for some $j \in [\mathsf{dim}]$, the absolute value of $\bar{\mathbf{v}}_j$ (i.e. the j-th component of the vector $\bar{\mathbf{v}}_j$) is "too large", i.e. if $|\bar{\mathbf{v}}_j| > \mathsf{dim} \cdot 2^{B'+\lambda}$, or if $\bar{\mathbf{v}}_j$ is not an integer, V rejects.
4: V randomly chooses a subset $J \subseteq [\mathsf{n}]$ with $|J| = \Theta(\lambda)$. For each $j \in J$:

- V interacts with P to open the commitments cm_i at position j, for each $i \in [\mathsf{dim}]$. Let $\hat{\mathbf{u}}_{1,j}, \ldots, \hat{\mathbf{u}}_{\mathsf{dim},j}$ be the received values. If these are not integers of a certain bit-size bound, V rejects.
- V checks whether $\mathsf{Enc}_{\mathcal{C}}(\bar{\mathbf{v}})_j = \sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_{i,j}$.

This testing phase is very similar to the testing phase of the original Brakedown scheme, but, unlike in the original one, serves two main purposes:

1. First, as in the original Brakedown scheme, it guarantees that, except with negligible probability (e.w.n.p.), P sent a vector of commitments to words $(\hat{\mathbf{u}}_i)_{i\in[\mathsf{dim}]}$ with each $\hat{\mathbf{u}}_i$ being close to a unique codeword $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$.
2. Second, by having V check (at Step 3 of Protocol 3) that the vector $\bar{\mathbf{v}}$ – supposedly the vector $\sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i$ – contains only integer entries, and that these entries are not "too

[3]For technical reasons, in Zip we actually sample the $r_1, \ldots, r_{\mathsf{dim}}$ in an interval of the form $[0, q_0 - 1]$ for q_0 a fixed prime.

large", we guarantee that, e.w.n.p., the witness vectors $\mathbf{v}_i$ belong to $\mathbb{Q}_B$. Intuitively speaking, **this guarantees that the vectors $\mathbf{v}_i$ are "close" to being integral of bit-size B'.**

When it comes to proving that Protocol 3 provides the guarantees mentioned in Item 1 above, our analysis is very similar to the one in the original Brakedown paper [GLS+23]. However, the corresponding arguments only allow to prove the existence of unique codewords $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i) \in \mathbb{Q}^{\mathsf{dim}}$ close to $\hat{\mathbf{u}}_i$, for all $i \in [\mathsf{dim}]$, because we simply apply Brakedown's arguments to a certain linear code over $\mathbb{Q}$ (cf. Section 2.3.3). In particular, we have no restriction whatsoever on the bit-size of the codewords $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$ and their underlying messages $\mathbf{v}_i$.

We next explain why Item 2 above indeed guarantees that $\mathbf{v}_i \in \mathbb{Q}_B^{\mathsf{dim}}$ for all $i \in [\mathsf{dim}]$. To begin with, as in Brakedown's original soundness analysis, it is possible to use the correlated agreement properties of linear codes to guarantee that, e.w.n.p., the vector $\bar{\mathbf{v}}$ sent by P at Step 2 of Protocol 3 is indeed the random linear combination $\sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i$. We call such an event $\mathcal{E}_0$.

Now, assume that in this scenario some entry in the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ has very large bit-size. Say it is the first entry $\mathbf{v}_{11}$ of $\mathbf{v}_1$. Then, if $\mathcal{E}_0$ holds and Item 2 of Protocol 3 passes, we have that the "random linear combination"

$$\sum_{j\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_{1j}$$

is an integer with absolute value less than $\mathsf{dim} \cdot 2^{B'+\lambda}$, where

$$B = B' + (2\mathsf{dim} + 1) \cdot (\lambda + \log(\mathsf{dim})).$$

In particular, we have that a random linear combination of rational numbers, some of them with very large bit-size, resulted in an integer of small bit-size.

Considering this, we prove two results regarding random linear combinations of rational numbers with large bit-sizes, cf. Lemmas 2.4 and 2.5. Combined, these results amount to say that if a random linear combination of rational numbers (using integers for the random coefficients) results in an integer, then the rational numbers had bit-size similar to the bit-size of the random integers.

Applied to our setting we obtain that, e.w.n.p., V accepts at Step 3 of Protocol 3 with probability at most

$$\min\left\{1, \frac{\mathsf{dim} \cdot 2^{B'}}{\|\mathbf{v}\|_\infty}\right\}, \tag{6}$$

where $\|\mathbf{v}\|_\infty$ is the largest absolute value of an entry in the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{\mathsf{dim}}$. This probability is 1 if $\|\mathbf{v}\|_\infty$ is small (as we want it to be), and becomes smaller as $\|\mathbf{v}\|_\infty$ increases. In particular, it is 1 if the bit-size of all entries in $\mathbf{v}$ is at most B'. On the other hand, if it is up to B, then the probability is negligible.

This is essentially why Zip guarantees that P is using a polynomial with coefficients in $\mathbb{Q}_B$, but is supposed to be used for polynomials with coefficients in $\mathbb{Q}_{B'}$.

The two aforementioned results regarding random linear combinations of rational numbers are the following:

Lemma 2.4 (A random linear combination of large rational numbers is large (Lemma 5.7))**.** *Let $v_1, \dots, v_n \in \mathbb{Q}$ be n rational numbers, not all of them zero. Let $S = [0, 2^s - 1]$ for some $s \geq 1$, and let $b \geq 1$. Then*

$$\Pr\left[|\sum_{i \in [n]} r_i \cdot v_i| < 2^b \;\middle|\; r_i \leftarrow S \text{ for all } i \in [n] \right] \leq \min\left\{1, \frac{2^b}{\max_{i \in [n]}\{|v_i|\} \cdot 2^s}\right\},$$

where $r_i \leftarrow S$ means uniformly sampling from S.

Lemma 2.5 (A random linear combination of rational numbers with large denominators rarely results in an integer (Lemma 5.8))**.** *Let $\mathbf{v} = (v_1, \dots, v_n) \in \mathbb{Q}^n$ be a vector of $n \geq 1$ rational numbers, not all of them zero. For each $i \in [n]$, write $v_i = a_i/b_i$ with $a_i, b_i \in \mathbb{Z}$ and* $\gcd(a_i, b_i) = 1$*. Let $M \geq 1$ be a positive integer, and assume there exists $i \in [n]$ such that $|b_i| > M$ and $v_i \neq 0$. Then the probability of uniformly sampling n integers $r_1, \dots, r_n$ in the interval $[0, M-1]$ such that $\sum_{i \in [n]} r_i v_i$ is an integer is at most $1/M$. More formally,*

$$P = \Pr\left[\left(\sum_{i \in [n]} r_i \cdot v_i\right) \in \mathbb{Z} \;\middle|\; r_i \leftarrow [0, M-1] \text{ for all } i \in [n] \right] \leq \frac{1}{M}.$$

Some technical challenges around Zip's extractor Recall that, in Zip, the prover commits to words $\hat{\mathbf{u}}_i \in \mathbb{Q}^{\mathsf{dim}}$, $i \in [\mathsf{dim}]$, which are supposedly close to being codewords $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$ and which, in most cases, will indeed be the actual codewords $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$. However, as with most code-based PCS, the prover can potentially take $\hat{\mathbf{u}}_i$ to be $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$ after replacing a few of its values by arbitrary strings of bits. In this work, we fix an encoding of $\mathbb{Q}$ as bit-strings (cf. Remark 3.1), and hence we interpret any string as a rational number. In any case, a prover may use an arbitrarily large string of bits in an entry of $\hat{\mathbf{u}}_i$. A complication that results from this phenomenon is the following: because this piece of data has large bit-size, a polynomial time extractor cannot read it entirely. This would not occur when working over a finite field $\mathbb{F}$, because the extractor knows that all field elements have at most $\log(|\mathbb{F}|) = O(\lambda)$ bits. Nevertheless, in Section 5.4 we describe an expected polynomial time extractor that can handle this situation. In particular, since the extractor is expected PPT, it does not necessarily read some of the entries in the vectors $\hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{\mathsf{dim}}$ entirely.

2.3.2 The evaluation protocol

The evaluation protocol of Zip is essentially the same as in the original Brakedown scheme, except that we execute it modulo a random prime. We describe it in simplified form in Protocol 4. This protocol is supposed to be run only after the testing protocol (Protocol 3) has been successfully executed and accepted by the verifier. The scheme exploits the well-known fact [Tha22] that, given a μ-variate multilinear polynomial $f(X_1, \dots, X_\mu)$ and an evaluation point $\mathbf{q} \in \mathcal{R}^\mu$ (for $\mathcal{R}$ a ring), there exist $\mathbf{q}_1, \mathbf{q}_2 \in \mathcal{R}^{\mathsf{dim}}$, where $\mathsf{dim} = 2^{\mu/2}$, such that

$$f(\mathbf{q}) = \mathbf{q}_1^\intercal \cdot \mathbf{v} \cdot \mathbf{q}_2,$$

where $\mathbf{v}$ is the $\mathsf{dim} \times \mathsf{dim}$ matrix containing the coefficients of f. Below, we let ϕ_q be the canonical projection of $\mathbb{Z}_{(q)}$ onto $\mathbb{F}_q$, cf. Eq. (4).

Protocol 4 Zip's evaluation protocol (informal).

Input: Let $\mathbb{x} = ((\mathsf{cm}_i)_{i\in[\mathsf{dim}]}, \mathbf{q}, y)$ with cm_i being a Merkle commitment to a word $\hat{\mathbf{u}}_i$ which is close to a codeword $\mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$, for all $i \in [\mathsf{dim}]$, where $\mathbf{v}_i \in \mathbb{Q}_B^{\mathsf{dim}}$; $\mathbf{q} \in \mathbb{Z}_B^{\mu}$, and y is an integer of appropriate size. P claims that the μ-variate polynomial f whose coefficients are given by $(\mathbf{v}_i)_{i\in[\mathsf{dim}]}$ satisfies $f(\mathbf{q}) = y$. P and V receive $\mathbb{x}$ as input. P additionally receives $\mathbb{w} = (\mathbf{v}_i)_{i\in[\mathsf{dim}]}$.

1: V samples a random $\Omega(\lambda)$-bit prime q.
2: If $\mathbf{v}_i \notin \mathbb{Z}_{(q)}^{\mathsf{dim}}$, or $\hat{\mathbf{u}}_i \notin \mathbb{Z}_{(q)}^{\mathsf{n}}$ for some $i \in [\mathsf{dim}]$, then P aborts.
3: Let $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{Z}^{\mathsf{dim}}$ be such that $f(\mathbf{q}) = \mathbf{q}_1^{\mathsf{T}} \cdot \mathbf{v} \cdot \mathbf{q}_2$. P sends V the vector

$$\bar{\mathbf{v}}_q = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\mathbf{v}_i) \in \mathbb{F}_q^{\mathsf{dim}}.$$

4: V randomly chooses a subset $J \subseteq [\mathsf{dim}]$ of size $|J| = \Theta(\lambda)$. For each $j \in J$:

- V interacts with P to open the commitments cm_i at position j, for each $i \in [\mathsf{dim}]$. Let $\hat{\mathbf{u}}_{1,j}, \dots, \hat{\mathbf{u}}_{\mathsf{dim},j}$ be the received values.
- V checks whether

$$\mathsf{Enc}_{\mathcal{C}_q}(\bar{\mathbf{v}}_q)_j = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\hat{\mathbf{u}}_{i,j}), \qquad \phi_q(y) = \sum_{i\in[\mathsf{dim}]} \bar{\mathbf{v}}_{q,i} \cdot \phi_q(\mathbf{q}_{2,i}),$$

where $\mathsf{Enc}_{\mathcal{C}_q}(\bar{\mathbf{v}}_q)$ denotes multiplying the matrix $\phi_q(M_{\mathcal{C}})$ by the vector $\bar{\mathbf{v}}_q$, where $M_{\mathcal{C}}$ is a generator matrix of $\mathcal{C}$.

The purpose of this evaluation protocol is to guarantee that indeed, $f(\mathbf{q}) = y$. The reason why the protocol is knowledge sound comes from two arguments. First, one can show that the check $\mathsf{Enc}_{\mathcal{C}_q}(\bar{\mathbf{v}}_q)_j = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\hat{\mathbf{u}}_{i,j})$ guarantees that, e.w.n.p., P indeed sent the vector $\bar{\mathbf{v}}_q = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\mathbf{v}_i) \in \mathbb{F}_q^{\mathsf{dim}}$ at Step 3 of Protocol 4. This can be shown using the properties of the linear code $\phi_q(\mathcal{C})$ in a similar way as done in the original Brakedown security proof. Below, in Section 2.3.3, we explain what our code is and what properties it has when projected with ϕ_q. Note that all applications of the map ϕ_q above are well defined.

Second, once the correctness of the vector $\bar{\mathbf{v}}_q$ is established, one has that the equality $\phi_q(y) = \sum_{i\in[\mathsf{dim}]} \bar{\mathbf{v}}_{q,i} \cdot \phi_q(\mathbf{q}_{2,i})$ is equivalent to $\phi_q(f(\mathbf{q})) = \phi_q(y)$. Now, we can argue that $f(\mathbf{q}) = y$ over the rational numbers in a similar way as to how we proved that the PIOP in Protocol 1 is knowledge sound, with the crucial difference that, in the latter, we were allowed to assume that provers, even malicious ones, were bound to use polynomials with coefficients of bounded bit-size, including the polynomials in the instance oracles. On this occasion, we cannot place such constraint because one of the goals of Zip is to enforce malicious provers to meet this requirement.

However, as we saw above in Section 2.3.1, Step 3 of Zip's testing phase (Protocol 3) guarantees that the vectors $\mathbf{v}_i$ belong to $\mathbb{Q}_B^{\mathsf{dim}}$, for all $i \in [\mathsf{dim}]$, e.w.n.p. In this setting, one can use again Lemma 2.1 in a similar way as it was used when analyzing the knowledge soundness of Protocol 1, to prove that indeed $f(\mathbf{q}) = y$ over the rationals, e.w.n.p.

2.3.3 The underlying linear code and the cost of committing and testing

As underlying linear code for our Brakedown-like scheme, we use the juxtaposed expand-accumulate (JEA) codes from Block et al. [BFK+24], with the difference that the randomness used when instantiating the code is drawn, approximately, from the interval of integers $[0, 2^{2\lambda} - 1]$ rather than from a finite field $\mathbb{F}$. This linear code, which we denote $\mathcal{C}_{\mathsf{JEA}}$, is constructed as follows: start by randomly sampling expander matrices E_1 and E_2 with entries in $[0, 2^{2\lambda} - 1]$ with respect to two different distributions (cf. Section 6 for the details on these). The parameters defining these distributions are chosen in a way to ensure that both E_1 and E_2 are sparse matrices. Next, we form generator matrices M_1 and M_2 by taking a matrix product of E_1 and E_2, respectively, with an accumulator matrix A (i.e. a square matrix such that every entry on and above the main diagonal is one). The matrices M_1 and M_2 give rise to two distinct linear codes $\mathcal{C}_1$ and $\mathcal{C}_2$. The codewords in our JEA linear code $\mathcal{C}_{\mathsf{JEA}}$ are concatenations of codewords of these two codes.

As a result, we obtain a code with a sparse generating matrix, and with nonzero entries having around 2λ bits[4]. This respects one of our design principles, namely, to not operate with very large integers. Indeed, encoding integer vectors with entries of bit-size B' requires operating with integers of size at most, roughly, $B' + 2 \cdot \lambda$ bits. As outlined at the beginning ot this technical overview, $B' \leq 2\lambda$ is a reasonable choice for some applications relevant to Zinc.

We remark that using other codes may not immediately provide the same properties: indeed, for example, Brakedown's code [GLS+23], or Basefold's code [ZCF23] are defined recursively through iterative matrix multiplications, with each matrix having some of their entries filled with random field elements. Naïvely executing this procedure over the integers would result in encoding matrices whose nonzero entries have thousands of bits. We leave the task of finding other suitable linear codes over $\mathbb{Q}$ as an interesting line of further research.

An additional property that our code $\mathcal{C}$ must satisfy is that, when projected onto a finite field $\mathbb{F}_q$ via the homomorphism ϕ_q, where q is random, the resulting code should have good distance properties, e.w.n.p. over the choice of q. This is because Protocol 4 operates with codewords that are reduced modulo a random prime.

We show that our JEA code satisfies this properties. The way we prove this, roughly, is by showing that, except for a negligible number of $\approx \lambda$-bit primes q sampleable by V, the generator matrix M of $\mathcal{C}$ has all entries in an interval of the form $[0, k \cdot q - 1]$ for some $k \geq 0$, assuming the entries of M where sampled in the interval $[0, 2^{2\lambda} - 1]$. Essentially, this is true because, in such setting, if $k_{\max}$ is the largest k such that $k \cdot q \leq 2^{2\lambda}$, then there are less than q elements in $[0, 2^{2\lambda} - 1]$ that do not belong to $[0, k_{\max} \cdot q - 1]$. Hence, the probability of sampling one such element is at most $q/2^{2\lambda} \approx 1/2^{\lambda}$.

This means that, e.w.n.p., when projected onto $\mathbb{F}_q$, the entries of M are uniformly sampled elements from $\mathbb{F}_q$. As such, we are able to argue that when projected onto $\mathbb{F}_q$, M is a matrix formed in the same way one would form a matrix for a JEA code over $\mathbb{F}_q$. Then, we can

[4] We conjecture that it is possible to replace the interval $[0, 2^{2\lambda} - 1]$ with an interval closer to $[0, 2^{\lambda} - 1]$, cf. Section 2.3.3.

apply the results from [BFK+24] to the projected JEA code. We refer to Section 6 for more details.

In the above analysis, it is crucial that the entries of the matrix M are sampled in an interval of roughly 2λ bits, if the primes samplable by V have roughly λ bits. This is so because we sought to reduce our analysis to the setting in [BFK+24]. We conjecture that, by studying M directly over $\mathbb{Z}$, and avoiding using [BFK+24], it is possible to reduce the sampling interval to integers of roughly λ bits.

2.4 Further work

Ongoing work of ours is to implement and benchmark Zinc and Zip. We leave as further work the task of analyzing the state-restoration soundness of our schemes, so that we can reliably compile our succinct arguments into a SNARK using the Fiat-Shamir transform.

3 Preliminaries

3.1 General notation

Given an integer $k \geq 1$ we let $[k] := \{1, \ldots, k\}$. We let $\{0,1\}^k := \{(b_1, \ldots, b_k) \mid b_i \in \{0,1\}, \text{ for all } i \in [k]\}$ be the hypercube of dimension k, or, in other words, the set of all sequences of k bits. We denote vectors and tuples with lowercase boldface roman letters, e.g. $\mathbf{v}, \mathbf{u}, \ldots$ By $\mathbf{v}_i$ we denote the i-th entry of a vector $\mathbf{v}$. Sometimes (in Section 5), we use symbols like $\mathbf{v}$ or $\mathbf{u}$ to denote matrices. In this case, $\mathbf{v}_i$ denotes the i-th row of $\mathbf{v}$, and $\mathbf{v}_{i,j}$ denotes the (i,j)-th entry of $\mathbf{v}$.

For any rational number $v \in \mathbb{Q}$, we denote its absolute value by $|v|$, and, given $\mathbf{v}$ a vector (or a matrix) as above, we denote by $\|\mathbf{v}\|_\infty = \max_i\{|\mathbf{v}_i|\}$ its norm. Whenever we consider a finite set S, we denote by $|S|$ the cardinality of S. We denote by gcd and lcm, respectively, the greatest common divisor and the lowest common multiple of a tuple of integers. Given two integers a and b with $a \leq b$, we denote by $[a,b] \subset \mathbb{Z}$ the interval of *integer elements* greater than or equal to a and smaller than or equal to b.

Throughout, we use $\mathbb{K}$ to denote a possibly infinite countable field and $\mathbb{F}$ a finite field. By $\mathcal{R}$ we denote a (possibly countably infinite) associative commutative ring with multiplicative identity. Note that *a field is a ring*, and hence when working over $\mathcal{R}$, we are also including the scenario where we work over a field.

Given non-zero vectors $\mathbf{v}_1, \ldots, \mathbf{v}_t \in \mathcal{R}^m$, for some $t, m \geq 1$, we say that the vectors are linearly independent over $\mathcal{R}$, if any linear combination $\sum_{i \in [t]} r_i \cdot \mathbf{v}_i \neq \mathbf{0} := (0, \ldots, 0)^\mathsf{T} \in \mathcal{R}^m$, for any $(r_1, \ldots, r_t)^\mathsf{T} \neq \mathbf{0}$ in $\mathcal{R}^t$; otherwise, we say that the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_t$ are linearly dependent over $\mathcal{R}$.

Let $\mathbf{X} = (X_1, \ldots, X_\mu)$ be a tuple of variables. We let $\mathcal{R}[\mathbf{X}]$ denote the ring of multivariate polynomials on variables $\mathbf{X}$ and with coefficients in $\mathcal{R}$. By $\mathcal{R}^{\mathsf{multilin}}[\mathbf{X}]$ we denote the set of polynomials from $\mathcal{R}[\mathbf{X}]$ all whose variables have individual degree at most 1, i.e. the set of multilinear polynomials over $\mathcal{R}$.

We use λ to denote the security parameter. A function $f(\lambda)$ is in $\mathsf{poly}(\lambda)$ if there exists $c \in \mathbb{N}$ such that $f(\lambda) = O(\lambda^c)$. If for all $c \in \mathbb{N}$, $f(\lambda) = o(\lambda^{-c})$, then $f(\lambda)$ is said to be negligible in λ, i.e. $\mathsf{negl}(\lambda)$.

3.2 Rings, localizations of rationals at a prime, and multilinear polynomials

Rings and ring homomorphisms All rings in this paper are associative, commutative, have a multiplicative identity element, and are possibly infinite, but countable. A ring homomorphism $\phi : \mathcal{R} \to \mathcal{R}'$ between two rings $\mathcal{R}, \mathcal{R}'$ is a map such that $\phi(a+b) = \phi(a)+\phi(b)$, $\phi(ab) = \phi(a)\phi(b)$ for all $a, b \in \mathcal{R}$, and $\phi(1) = 1$. The *kernel* of ϕ, denoted $\ker\phi$, is the subset of $\mathcal{R}$ formed by all elements $a \in \mathcal{R}$ such that $\phi(a) = 0$. Such subset forms an ideal of $\mathcal{R}$. We say ϕ is *onto* if for each $b \in \mathcal{R}'$ there exists $a \in \mathcal{R}$ such that $\phi(a) = b$.

The homomorphism ϕ extends naturally to a homomorphism $\phi : \mathcal{R}[\mathbf{X}] \to \mathcal{R}'[\mathbf{X}]$ between the polynomial rings $\mathcal{R}[\mathbf{X}], \mathcal{R}'[\mathbf{X}]$ on variables $\mathbf{X}$. Similarly, given an oracle $[[f]]$ to a polynomial $f \in \mathcal{R}[\mathbf{X}]$, we let $\phi([[f]]) = [[\phi(f)]]$. Given a vector $\mathbf{v}$ containing elements from $\mathcal{R}$, polynomials from $\mathcal{R}[\mathbf{X}]$, and oracles to polynomials from $\mathcal{R}[\mathbf{X}]$, by $\phi(\mathbf{v})$ we denote the vector resulting from applying ϕ at each component of $\mathbf{v}$.

Given a countable ring $\mathcal{R}$, we fix an encoding of the elements of $\mathcal{R}$ as binary strings. Given $B \geq 1$, we let $\mathcal{R}_B$ denote the subset of $\mathcal{R}$ consisting of all ring elements whose encoding has at most B bits.

Rational numbers and their representation We let $\mathbb{Z}$ denote the ring of integers, and $\mathbb{Q}$ the field of rational numbers. We represent uniquely each element from $\mathbb{Q}$ as a pair $(a, b) \in \mathbb{Z}^2$ with $b > 0$ such that the greatest common divisor of a and b is 1. In this case we say that (a, b) are in *lowest form*, and we we denote the pair by a/b. The bitstring representation of $|a/b|$ has size $\log(|a|) + \log(|b|) = \log(|a \cdot b|)$.

When speaking of the *bit-length* of a rational number a/b expressed in lowest form, we mean the quantity $\log_2(|a|) + \log_2(b)$. Note that, technically, one would need to add for possibly one extra bit accounting for the sign of a/b. We omit it for simplicity.

In this work we fix an encoding of the rational numbers as strings of bits, so that the encoding of $a/b \in \mathbb{Q}$ in lowest form has bit-size at most $\log(|a|) + \log(b) + 2$. This encoding can be obtained by using the standard bijection $f : \mathbb{Q} \to \mathbb{N}$, where positive rational numbers are mapped to positive natural numbers as specified by the order

$$1, 2/1, 1/2, 1/3, 2/2, 3/1, 4/1, 3/2, 2/2, 1/4, \ldots$$

In such encoding, bit-strings of size k encode rational numbers of the form a/b satisfying $\log(a) + \log(b) + 1 \leq k$. Extending this to handle negative rational numbers by adding an initial bit indicating the sign, we obtain the desired encoding.

Remark 3.1. We will use this encoding throughout the paper, mostly without referring to it further. Additionally, to ease our exposition, we assume that the encoding of a/b has at most $\log(|a|) + \log(b)$ bits rather than $\log(|a|) + \log(b) + 2$ bits.

Localization of rationals at a prime The *localization of* $\mathbb{Q}$ *at a prime number* q is the set

$$\mathbb{Z}_{(q)} = \left\{ \frac{a}{b} \;\middle|\; \begin{array}{l} \frac{a}{b} \in \mathbb{Q}, \\ b \not\equiv 0 \mod q \end{array} \right\}.$$

It is well-known that $\mathbb{Z}_{(q)}$ is a ring. Further, $\mathbb{Z}_{(q)}$ admits the following ring-homomorphism:

$$\begin{aligned}\phi_q : \mathbb{Z}_{(q)} &\to \mathbb{F}_q \\ a/b &\mapsto a \cdot b^{-1}\end{aligned}$$

where $\mathbb{F}_q$ is the finite field with q elements, and b^{-1} above denotes the inverse of b modulo the prime q. The kernel $\ker(\phi_q)$ of ϕ_q is the subring of elements of $\mathbb{Z}_{(q)}$ that are mapped to 0 by ϕ_q. We have

$$\ker(\phi_q) = q\mathbb{Z}_{(q)} = \{a/b \in \mathbb{Z}_{(q)} \mid a \equiv 0 \mod q\}.$$

An *integral domain* is a ring $\mathcal{D}$ where whenever $a \cdot b = 0$ for some $a, b \in \mathcal{D}$, it must be the case that either a or b are zero. The ring of integers $\mathbb{Z}$ is an integral domain, and any field is an integral domain. If $\mathcal{D}$ is an integral domain, then one can define its *field of fractions*, which is the field comprised of elements of the form a/b, and whose addition and multiplication operations are analogous to how rational addition and multiplication work. Indeed, the field of rational numbers $\mathbb{Q}$ is the field of fractions of $\mathbb{Z}$.

Multilinear polynomials and multilinear extensions (MLE) over a ring Let $\mathcal{R}$ be a ring. Let $\mu \geq 1$ and let $\mathbf{X} = (X_1, \ldots, X_\mu)$ be a tuple of variables. It is well known [Tha22] that a multilinear polynomial $f(\mathbf{X}) \in \mathcal{R}^{\mathsf{multilin}}[\mathbf{X}]$ is uniquely defined by the multiset of the values it takes on $\{0,1\}^\mu$, i.e. $f(\{0,1\}^\mu) := \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{R}^\mu\}$. In other words, any two $f, g \in \mathcal{R}^{\mathsf{multilin}}[\mathbf{X}]$ such that $f(\mathbf{x}) = g(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\mu$ are the same polynomial. Further, given a map $f : \{0,1\}^\mu \to \mathcal{R}$, there always exists a unique multilinear polynomial on μ variables, denoted $\widetilde{f}(\mathbf{X})$, such that $\widetilde{f}(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\mu$. It is given by the expression

$$\widetilde{f}(\mathbf{X}) := \sum_{\mathbf{x} \in \{0,1\}^\mu} f(\mathbf{x}) \cdot \widetilde{\mathsf{eq}}(\mathbf{x}; \mathbf{X}) \tag{7}$$

where $\widetilde{\mathsf{eq}}(\mathbf{x}; X)$ is the unique multilinear polynomial on μ variables that takes the value 0 on all points of the hypercube $\{0,1\}^\mu$, except at $\mathbf{x}$ where it takes the value 1. Precisely,

$$\widetilde{\mathsf{eq}}(\mathbf{x}; \mathbf{X}) := \prod_{i \in [\mu]} \left(x_i X_i - (1 - x_i)(1 - X_i)\right).$$

This unique multilinear polynomial $\widetilde{f}(\mathbf{X})$ is called the *multilinear extension (MLE)* of f. Given a vector $\mathbf{v} = (v_1, \ldots, v_{2^\mu}) \in \mathcal{R}^{2^\mu}$, we define the MLE of $\mathbf{v}$ (denoted by $\widetilde{\mathbf{v}}(\mathbf{X})$) as the MLE of the map $\mathbf{v} : \{0,1\}^\mu \to \mathcal{R}$ assigning to each element $\mathbf{x} \in \{0,1\}^\mu$ the element $v_{\mathbf{x}}$, where here we interpret $\mathbf{x}$ as the natural number whose binary representation is $\mathbf{x}$.

3.3 Linear codes

A linear code $\mathcal{C}$ over a field $\mathbb{K}$ of length n and dimension $\mathsf{dim} \leq \mathsf{n}$ is a vector subspace of $\mathbb{K}^{\mathsf{n}}$ of dimension dim. The rate of $\mathcal{C}$ is $\rho = \mathsf{dim}/\mathsf{n}$. A matrix $M_{\mathcal{C}} \in \mathbb{K}^{\mathsf{dim} \times \mathsf{n}}$ is said to be a *generator matrix* for $\mathcal{C}$ if it has rank dim and its rows span $\mathcal{C}$. Elements in $\mathcal{C}$ are called *codewords*. A vectors of length n is called a *word*. A word may have arbitrary entries, not even from the field $\mathbb{K}$. The *Hamming distance* $\Delta_H(\mathsf{c}_1, \mathsf{c}_2)$ between two words $\mathsf{c}_1, \mathsf{c}_2 \in \mathcal{C}$ is the number of positions where c_1 and c_2 differ. The *relative Hamming distance* $\Delta(\mathsf{c}_1, \mathsf{c}_2)$ between two words c_1 and c_2 is $\Delta(\mathsf{c}_1, \mathsf{c}_2) = \Delta_H(\mathsf{c}_1, \mathsf{c}_2)/\mathsf{n}$. The *relative distance* dist of $\mathcal{C}$ is the minimum relative Hamming distance between two codewords of $\mathcal{C}$, i.e. $\mathsf{dist} = \min_{\mathsf{c}_1,\mathsf{c}_2 \in \mathcal{C}, \mathsf{c}_1 \neq \mathsf{c}_2} \Delta(\mathsf{c}_1, \mathsf{c}_2)$.

Given any word $\mathbf{v}$, we say that $\mathbf{v}$ is δ*-close* to $\mathcal{C}$ if the relative Hamming distance between $\mathbf{v}$ and the closest codeword c from $\mathcal{C}$ is δ, written $\Delta(\mathbf{v}, \mathcal{C}) \leq \delta$.

For a given word $\mathbf{v}$, let $B_{\mathbf{v}}(\delta) = \{\mathsf{c} \in \mathcal{C} \mid \Delta(\mathbf{v}, \mathsf{c}) \leq \delta\}$ be the ball of radius δ centered at $\mathbf{v}$. We say δ is *within the unique decoding radius* if for every word $\mathbf{v}$, $B_{\mathbf{v}}(\delta)$ contains at most one codeword. Any δ satisfying $\delta < (1 - \mathsf{dist}/\mathsf{n})/2$ is within the unique decoding radius.

For any linear code $\mathcal{C}$ over $\mathbb{K}$, the *interleaved code* $\mathcal{C}^k$ over $\mathbb{K}^k$ has codewords $k \times \mathsf{n}$ matrices U, such that every row U_i is a codeword in $\mathcal{C}$. $\mathcal{C}^k$ is a code of length n and dimension $k \cdot \mathsf{dim}$. For any matrix A in $\mathbb{K}^{k \times \mathsf{n}}$, the relative Hamming distance $\Delta(A, \mathcal{C}^k)$ is the number of columns where A and the closest codeword $U \in \mathcal{C}^k$ differ on at least one entry, divided by the code length n.

Definition 3.1 ((δ, α, K)-correlated agreement of a code [BSCI+23, Theorem 1.4])**.** *Let $\mathcal{C}$ be a linear code over a (possibly infinite) field $\mathbb{K}$ with length n and dimension dim. Let $K \subseteq \mathbb{K}$ be a finite subset of $\mathbb{K}$. Let $0 < \delta, \alpha < 1$. Let $k \geq 1$ and let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be k words. We say that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ have* δ-correlated agreement in $\mathcal{C}$ *if there exists a set $E \subseteq [\mathsf{n}]$ with $|E| \geq (1 - \delta) \cdot \mathsf{n}$ and codewords $\mathsf{c}_1, \ldots, \mathsf{c}_k \in \mathcal{C}$ such that $\mathbf{v}_i$ and c_i agree on E, for all $i \in [k]$.*

We say that $\mathcal{C}$ has (δ, α, K)-correlated agreement *if for any k words $\mathbf{v}_1, \ldots, \mathbf{v}_k$ such that*

$$\Pr\left[\Delta\left(\sum_{i \in [k]} r_i \mathbf{v}_i, \mathcal{C}\right) \leq \delta \;\middle|\; r_i \leftarrow K,\ i \in [k]\right] > \alpha,$$

we have that $\mathbf{v}_1, \ldots, \mathbf{v}_k$ have δ-correlated agreement in $\mathcal{C}$.

When no confusion arises, we will just say that the code has δ-correlated agreement and drop the reference to α and the set K.

Lemma 3.2 (Correlated agreement for linear codes over infinite fields, c.f. [AHIV22], Lemma 4.5)**.** *Let $\mathcal{C}$ be a linear code over a field $\mathbb{K}$ with dimension dim, length n and relative distance dist. Let $K \subseteq \mathbb{K}$ be a finite nonempty subset of $\mathbb{K}$. Then $\mathcal{C}$ has (δ, α, K)-correlated agreement with $\delta < \mathsf{dist}/3$, $\alpha = \mathsf{n}/|K|$.*

This lemma is proven in Appendix A.

3.4 Interactive Proofs, PIOPs, and Polynomial Commitments

In this section we provide standard definitions and notations for interactive proofs and arguments, IOPs, PIOPs, and polynomial commitment schemes. We mainly follow [CBBZ22, BFS19, CY24].

Indexed relations An *indexed relation* REL is a set consisting of index-instance-witness triples $(\mathbb{i}, \mathbb{x}; \mathbb{w})$. In this paper, typically, $\mathbb{i}, \mathbb{x}, \mathbb{w}$ are tuples of natural numbers, ring elements, and polynomials with coefficients in a ring. Additionally, $\mathbb{i}$ and $\mathbb{x}$ may contain oracles (see below). Our relations are parameterized by a collection of global parameters, which we denote by gp. Accordingly, we often denote relations by $\mathsf{REL}_{\mathsf{gp}}$. We always include the security parameter in gp, but we omit referring to it.

We let $\mathsf{LANG}(\mathsf{REL}_{\mathsf{gp}})$ denote the *language associated to* $\mathsf{REL}_{\mathsf{gp}}$. This is the set of index-instance pairs $(\mathbb{i}, \mathbb{x})$ for which there exists a witness $\mathbb{w}$ such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp}}$.

An *oracle* is a finite string $(f_i)_{i\in[n]}$ of ring elements indexed over an ordered subset $S = (s_i)_{i\in[n]}$ of a ring $\mathcal{R}$. Oracles receive a special treatment as they are meant to be received by a verifier, but not necessarily read in full. In this paper we think of a string $(f_i)_{i\in[n]}$ as the map $f : S \to \mathcal{R}$ with $f(s_i) = f_i$. We distinguish between the actual map f and its oracle, which we denote by $[[f]]$. The latter can be received by a verifier, and read only at specific points, but not read in full. We distinguish between regular oracles (i.e. oracles to strings as we just defined), and oracles to polynomials. The latter are oracles to a string that consists of evaluations of a polynomial f of a prescribed number of variables, degree, and with coefficients in some prescribed set. These evaluations contain enough information to uniquely determine f. For example, if f is multilinear with μ variables, then the oracle may consists of all evaluations of f in $\{0, 1\}^{\mu}$.

Interactive proofs Let $\mathsf{REL}_{\mathsf{gp}}$ be an indexed relation (without oracles) parameterized by some global parameters gp.

Definition 3.2 (Interactive Proof). *An Interactive Proof for an indexed relation* $\mathsf{REL}_{\mathsf{gp}}$ *with global parameters* gp *is a tuple of algorithms* $(\mathsf{Indexer}, \mathsf{P}, \mathsf{V})$ *where: for all* $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp}}$, $\mathsf{Indexer}$ *is a deterministic algorithm taking* $\mathsf{gp}, \mathbb{i}$ *as input and outputting verifier and prover parameters* $(\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i})$*. The pair* (P, V) *is a pair of interactive algorithms where* P *receives* $(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ *as input, and* V *receives* $(\mathsf{vp}, \mathbb{x})$ *as input. The interactive proof cna have the following properties:*

- Completeness with error $\varepsilon_{\mathsf{comp}}$. *For all* $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp}}$,

$$\Pr\big[\langle \mathsf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \;\;\big|\;\; (\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i})\big] \geq 1 - \varepsilon_{\mathsf{comp}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}).$$

- Soundness with error $\varepsilon_{\mathsf{sound}}$ (adaptive). *Let* $\mathcal{L}(\mathsf{REL}_{\mathsf{gp}})$ *be the language corresponding to the indexed relation* $\mathsf{REL}_{\mathsf{gp}}$*. The protocol* Π *has* soundness error $\varepsilon_{\mathsf{sound}}$ *if for any unbounded adversarial prover* P^**, for any* $(\mathbb{i}, \mathbb{x})$*, the following holds:*

$$\Pr\left[\begin{array}{r} \langle \mathsf{P}^*(\mathsf{pp}, \mathbb{i}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \\ \wedge\ (\mathbb{i}, \mathbb{x}) \notin \mathcal{L}(\mathsf{REL}_{\mathsf{gp}}) \end{array} \;\middle|\; (\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i}) \right] \leq \varepsilon_{\mathsf{sound}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}).$$

- Knowledge soundness with error $\varepsilon_{\mathsf{ks}}$. *There exists a probabilistic oracle machine* Ext *(called the* extractor*) such that, given oracle access to any unbounded adversarial prover* P^**, and any* $(\mathbb{i}, \mathbb{x})$ *the following holds:*

$$\Pr\left[\begin{array}{r} \langle \mathsf{P}^*(\mathsf{pp}, \mathbb{i}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \\ \wedge\ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \notin \mathsf{REL}_{\mathsf{gp}} \end{array} \;\middle|\; \begin{array}{l} (\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i}) \\ \mathbb{w} \leftarrow \mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) \end{array} \right] \leq \varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}).$$

  We require $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ *to run in expected polynomial time on* $(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}^{-1})$ *(we do not count the running time of the oracle to* P^**).*

- Public coin. Π *is public coin if the verifier messages are random coins and if the final verification procedure can be executed publicly.*

Interactive Oracle Proofs (IOP)

Definition 3.3 (Interactive Oracle Proof (IOP))**.** *An* interactive oracle proof (IOP) over a ring $\mathcal{R}$ *is a public-coin interactive proof for an indexed relation* $\mathsf{REL}_{\mathsf{gp}} = \{(\mathbb{i}, \mathbb{x}, \mathbb{w})\}$*. In this relation,* $\mathbb{i}$ *and* $\mathbb{x}$ *can contain oracles to strings of elements from* $\mathcal{R}$ *or a prescribed subset of* $\mathcal{R}$*. The full strings behind the oracles from* $\mathbb{i}$ *and* $\mathbb{x}$ *are contained in the prover's parameters* pp *and in the witness* $\mathbb{w}$*, respectively. We denote an oracle to a string* π *by* $[[\pi]]$*.*

In every protocol message, the prover $\mathcal{P}$ *sends oracles to strings of elements from* $\mathcal{R}$*, a prescribed subset of* $\mathcal{R}$*, or a prescribed subset of some other ring. In every round, the verifier* $\mathcal{V}$ *sends a random challenge* ρ_i*.*

We define completeness, soundness, and knowledge soundness in an analogous way as to how they were defined for interactive proofs and arguments in Definition 3.2.

Definition 3.4 (Polynomial Interactive Oracle Proof (PIOP))**.** *A* polynomial interactive oracle proof (PIOP) over a ring $\mathcal{R}$ *is an IOP with the difference that all oracles contain polynomials of prescribed number of variables* μ*, degrees, and with coefficient in some subset of* $\mathcal{R}$ *or some other ring* $\mathcal{R}'$*. The oracle polynomials can be queried at any point of* $\mathcal{R}^{\mu}$ *(or* $\mathcal{R}^{\mu'}$*).*

We define completeness, soundness, and knowledge soundness in an analogous way as to how they were defined for interactive proofs and arguments in Definition 3.2.

Remark 3.3 (On having well-formed global parameters, indices, instances, and oracles)**.** Soundness and knowledge soundness notions of interactive proofs/arguments, IOPs, and PIOP all assume that the data in the parameters $\mathsf{gp}, \mathsf{pp}, \mathsf{vp}$ and in the index $\mathbb{i}$ and instance $\mathbb{x}$ have the correct syntactic form. In particular, it is assumed that all strings or polynomials behind the oracles have the correct length, have entries in the correct sets, or correspond to polynomials of the prescribed number of variables, degrees, and with coefficients in a prescribed set. In this case we say that $(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ is *well-formed*.

This applies as well to the oracles sent by the prover during the interactive phase of the protocol, in which case we say that P *sends well-formed oracles*. In particular, *it applies also to malicious provers*. For example, in a PIOP, the knowledge error only applies in the case that the malicious provers are given well-formed $(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ and send well-formed oracles.

Lemma 3.4 (Sound PIOPs are knowledge sound – Adaptation of [CBBZ22, Lemma 2.3])**.** *Let* Π *be a PIOP over a ring* $\mathcal{R}$ *for an indexed relation* $\mathsf{REL}_{\mathsf{gp}}$ *with soundness error* ε*. Assume that, for all* $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp}}$*,* $\mathbb{w}$ *consists only of multilinear polynomials such that* $\mathbb{x}$ *contains oracles to these polynomials. Then the PIOP has knowledge soundness error* ε*, and the extractor runs in time* $O(|\mathbb{w}|)$*.*

Proof. We translate the proof of [CBBZ22, Lemma 2.3] to our formalism. We construct an extractor Ext that can produce a witness $\mathbb{w}^*$ such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}^*) \in \mathsf{REL}_{\mathsf{gp}}$ if and only if $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp}})$, for any index-instance pair $(\mathbb{i}, \mathbb{x})$. Therefore, on every input $(\mathbb{i}, \mathbb{x})$ in the language of $\mathsf{REL}_{\mathsf{gp}}$, the soundness error is exactly the knowledge-soundness error. Let $\tilde{\mathsf{P}}$ be a PPT adversary for Π. By definition of extractor, the algorithm can query oracles. Thus, for each oracle of a multilinear polynomial in μ variables in $\mathbb{x}$, $\mathsf{Ext}^{\tilde{\mathsf{P}}}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ can query it at the points $\{0,1\}^{\mu}$ distinct points to extract the polynomial inside the oracle and thus extract the whole $\mathbb{w}^*$. Note that such extractor works in $\mathsf{poly}(\mathbb{i}, \mathbb{x})$ time. Now, if

$(\mathbb{i}, \mathbb{x}; \mathbb{w}^*) \in \mathsf{REL}_{\mathsf{gp}}$, then $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp}})$ by definition. Let us prove that also the other implication is true. Assume that $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp}})$. By definition, there exists a witness $\mathbb{w}$ (a priori possibly different from $\mathbb{w}^*$) such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp}}$. Since the extractor evaluates the oracles at $\{0,1\}^\mu$, and any polynomial in $\mathbb{w}$ and $\mathbb{w}^*$ is multilinear in μ-variables, it follows that the queried polynomials must coincide, hence proving that $\mathbb{w} = \mathbb{w}^*$. Hence $(\mathbb{i}, \mathbb{x}; \mathbb{w} \coloneqq \mathbb{w}^*) \in \mathsf{REL}_{\mathsf{gp}}$. $\mathsf{Ext}^{\tilde{\mathsf{P}}}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ then outputs the unique valid witness associated with $(\mathbb{i}, \mathbb{x})$, for any $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp}})$. $\square$

Polynomial Commitment Schemes We next recall the notion of Polynomial Commitment Scheme. Our formulation allows for handling polynomials with coefficients in some subset of a ring (possibly infinite). We also allow for the possibility that the commitments produced by a PCS contain oracles to strings, and that the evaluation protocol is an IOP rather than an interactive argument. We do this because we later describe Zip's evaluation procedure by means of an IOP. The oracles can then be replaced later with Merkle tree commitments, or some other vector commitment.

Definition 3.5 (Multilinear PCS over a ring $\mathcal{R}$ – with oracles allowed)**.** *A multilinear Polynomial Commitment Scheme* PCS *consists of a tuple of PPT algorithms and protocols* (Commit, Open, Eval) *and global parameters* gp *with the following properties.*

The global parameters gp *have the form* $\mathsf{gp} = (\mu, S, \mathcal{R}, \mathsf{gp}')$ *where* μ *is a number of variables,* $\mathcal{R}$ *is a ring,* S *is a subset of* $\mathcal{R}$ *(possibly* $S = \mathcal{R}$*), and* gp' *are other parameters required to describe the scheme.*

$\mathsf{Commit}(\mathsf{gp}, f, \mathsf{aux})$: *Takes as input the parameters* $\mathsf{gp} = (\mu, S, \mathcal{R}, \mathsf{gp}')$*, a multilinear polynomial* $f \in S^{\mathsf{multilin}}[\mathbf{X}]$*, where* $\mathbf{X} = (X_1, \ldots, X_\mu)$*, and some auxiliary data* aux*. The* Commit *algorithm outputs a commitment* cm*, which may contain oracles to strings, and an opening hint* hint*, which may or may not be randomness used in the computation of the commitment.*

$\mathsf{Open}(\mathsf{gp}, \mathsf{cm}, f, \mathsf{hint})$**:** *Takes as input the parameters* $\mathsf{gp} = (\mu, S, \mathcal{R}, \mathsf{gp}')$*, a commitment* cm*, an opening hint* hint*, and a multilinear polynomial* $f \in S^{\mathsf{multilin}}[\mathbf{X}]$*, and outputs a bit* $b \in \{0,1\}$*.*

The scheme PCS *has* binding error $\varepsilon_{\mathsf{bind}}$ *if for every adversary* P^**,*

$$\Pr\left[b_0 = b_1 = 1 \ \wedge \ f_1 \neq f_2 \;\middle|\; \begin{array}{l} (\mathsf{cm}, f_1, f_2, \mathsf{hint}_1, \mathsf{hint}_2) \leftarrow \mathsf{P}^*(\mathsf{gp}) \\ b_1 \leftarrow \mathsf{Op}(\mathsf{gp}, \mathsf{cm}, f_1, \mathsf{hint}_1) \\ b_2 \leftarrow \mathsf{Op}(\mathsf{gp}, \mathsf{cm}, f_2, \mathsf{hint}_2) \end{array} \right] \leq \varepsilon_{\mathsf{bind}}(\mathsf{gp}, \mathsf{cm}).$$

$\mathsf{Eval} = (\mathsf{Indexer}_{\mathsf{Eval}}, \mathsf{P}_{\mathsf{Eval}}, \mathsf{V}_{\mathsf{Eval}})$**:** *Is an interactive public-coin protocol, possibly an IOP, for* *the following relation:*

$$\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}} = \left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} = (\mu, S \subseteq \mathcal{R}, \mathsf{gp}'), \\ \mathbb{i} = (\mathsf{cm}, \mathsf{hint})^{5}, \\ \mathbb{x} = (\mathbf{x}, y),\ \ \mathbf{x} \in \mathcal{R}^{\mu},\ \ y \in \mathcal{R} \\ \mathbb{w} = (f),\ \ f \in S^{\mathsf{multilin}}[X_1, \ldots, X_\mu], \\ f(\mathbf{x}) = y, \\ \mathsf{Open}(\mathsf{gp}, c, f, \mathsf{hint}) = 1. \end{array} \right\}.$$

Additionally, the relation implicitly includes constraints requiring that $(\mathsf{cm}, \mathsf{hint})$ *has the correct form and are valid outputs of* Commit*. These constraints may use parameters from* gp'*. Moreover, the relation* $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ *can contain constraints forcing query points* $\mathbf{x}$ *and evaluation values* y *to belong to some sets specified in* gp'*. We refer to Section 5.2 for an example of a concrete evaluation relation* $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$

We say the Polynomial Commitment Scheme PCS *has* knowledge soundness error $\varepsilon_{\mathsf{ks}}$ *if* Eval *has knowledge soundness error* $\varepsilon_{\mathsf{ks}}$*.*

We say that PCS *has* completeness error $\varepsilon_{\mathsf{comp}}$ *if the following two conditions hold:*

- *For every parameters* gp*,* $f \in S^{\mathsf{multilin}}[X_1, \ldots, X_\mu]$*, auxiliary input* aux*, and opening hint* hint*,*

$$\Pr[\mathsf{Open}(\mathsf{gp}, \mathsf{cm}, f, \mathsf{hint}) = 1 \mid (\mathsf{cm}, \mathsf{hint}) \leftarrow \mathsf{Commit}(\mathsf{gp}, f, \mathsf{aux})] = 1 - \varepsilon_{\mathsf{comp}}(\mathsf{gp}, f).$$

- *For every parameters* gp*,* $f \in S^{\mathsf{multilin}}[X_1, \ldots, X_\mu]$*, auxiliary input* aux*, opening hint* hint*, and* $\mathbf{x} \in S^{\mu}, y \in \mathcal{R}$*,*

$$\Pr\left[1 \leftarrow \langle \mathsf{P}_{\mathsf{Eval}}(\mathsf{pp}, \mathbf{x}, y, f), \mathsf{V}_{\mathsf{Eval}}(\mathsf{vp}, \mathbf{x}, y) \rangle \;\middle|\; \begin{array}{l} \mathsf{cm} = \mathsf{Commit}(\mathsf{gp}, f, \mathsf{aux}) \\ y = f(\mathbf{x}) \end{array} \right] = 1 - \varepsilon_{\mathsf{comp}}(\mathsf{gp}, f).$$

A multilinear PCS PC *is said to be* succinct *if the commitment* cm *output by* Commit *is has size sublinear in* 2^{μ}*.*

3.5 CCS and lookup relations

We next recall the definition of two well-known types of constraints. Instead of formulating them over a finite field $\mathbb{F}$, we use a general ring $\mathcal{R}$.

[5] Having the commitment cm and the hint hint in the index $\mathbb{i}$ allows to not place the hint in the witness $\mathbb{w}$. This way, when designing an interative proof for $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, the hint hint can be given to the prover as part of the prover parameters pp output by P, and since hint is not part of the witness, we do not rquire extractors to recover hint, which in some cases is not something one wants to do. For example, in a PCS constructed from IOPs of Proximity, cm is a (collection of) Merkle tree commitment to the encoding of a word hint, which is, supposedly close to a codeword c. We are only interested in having an extractor that recovers c, not hint.

Customizable Constraint System (CCS) over a ring In this section we recall the definition of Customizable Constraint Systems (CCS), which constitute a generalization of R1CS and other constraint systems. We generalize the original definition so that it operates over a general commutative ring $\mathcal{R}$, rather than a field. We follow the polynomial-based version of the definition from [KS23].

We start by fixing global parameters $\mathsf{gp} = (\mathcal{R}, 2^{\mu_1}, 2^{\mu_2}, \ell, n, q, d, \mathbf{c}, \mathbf{S})$ for $m, n, \ell, n, q, d \geq 1$, $\mathbf{c} \in \mathcal{R}^q$, and where $\mathbf{S}$ is a tuple of multisets:

$$\mathbf{S} = (S_1, \ldots, S_q), \quad S_i \text{ is a multiset with} \leq d \text{ elements from } [n]. \tag{8}$$

The *CCS relation* for the parameters gp, which we denote $\mathsf{CCS}_{\mathsf{gp}}$, is defined as:

$$\mathsf{CCS}_{\mathsf{gp}} := \left\{ \begin{pmatrix} \mathbb{i}, \\ \mathbb{x}; \\ \mathbb{w} \end{pmatrix} \middle| \begin{array}{l} \mathsf{gp} = (\mathcal{R}, 2^{\mu_1}, 2^{\mu_2}, \ell, n, q, d, \mathbf{S}, \mathbf{c}), \\ \mathbb{i} = ([[M_1]], \ldots, [[M_t]]), \\ \mu = \mu_1 + \mu_2,\ M_1 \ldots, M_n \in \mathcal{R}^{\mathsf{multilin}}[X_1, \ldots, X_\mu], \\ \mathbb{x} = (\mathbf{y}, [[f]]),\ \mathbf{y} \in \mathcal{R}^{2^\ell - 1},\ f \in \mathcal{R}^{\mathsf{multilin}}[Y_1, \ldots, Y_{\mu_2 - \ell}], \\ \mathbb{w} = [[f_1]], \\ Z(Y_1, \ldots, Y_{\mu_2}) = \mathsf{MLE}((f(\mathbf{x})_{\mathbf{x} \in \{0,1\}^{\mu_2 - \ell}}, \mathbf{y}, 1)) \\ \displaystyle\sum_{i \in [q]} c_i \cdot \left(\prod_{j \in S_i} \left(\sum_{\mathbf{y} \in \{0,1\}^s} M_j(\mathbf{x}, \mathbf{y}) \cdot Z(\mathbf{y}) \right) \right) = 0 \quad \text{for all } \mathbf{x} \in \{0,1\}^{\mu_1}. \end{array} \right\},$$

where by MLE we denote the multilinear extension of a vector.

Lookup relation over a ring Let gp be global parameters of the form $\mathsf{gp} = (\mathcal{R}, n_a, n_t, B)$ where $n_a, n_t, B \geq 1$ and n_a, n_t are powers of two. The *lookup relation* $\mathsf{Look}_{\mathsf{gp}}$ for the parameters gp is defined as:

$$\mathsf{Look}_{\mathsf{gp}} := \left\{ \begin{pmatrix} \mathbb{i} = ([[t]]), \\ \mathbb{x} = ([[a]]); \\ \mathbb{w} = (a(\mathbf{X})) \end{pmatrix} \middle| \begin{array}{l} \mathsf{gp} = (\mathcal{R}, n_a, n_t, B), \\ a(\mathbf{X}) \in \mathcal{R}_B[\mathbf{X}],\ \mathbf{X} = (X_1, \ldots, X_{\log(n_a)}), \\ t(\mathbf{Y}) \in \mathcal{R}_B[\mathbf{Y}],\ \mathbf{Y} = (Y_1, \ldots, Y_{\log(n_t)}), \\ \{a(\mathbf{x}) \mid \mathbf{x} \in \{0,1\}^{\log n_a}\} \subseteq \{t(\mathbf{y}) \mid \mathbf{y} \in \{0,1\}^{\log(n_t)}\}. \end{array} \right\}$$

4 Zinc-PIOP: Building PIOPs over rings from collections of PIOPs over quotients of subrings

In this section, we describe a framework, which we call Zinc-PIOP, for building PIOPs for relations whose constraints are, in nature, algebraic over some ring. We call such relations *algebraic indexed relations* (cf. Definition 4.1). We refer to Section 2.1 for an intuitive but detailed explanation of the ideas used in this section.

4.1 PIOP over a ring, projections, and lifts

We start by outlining our setting and presenting some tools we will work with. Mainly, we define what we mean by a relation to be algebraic over a ring $\mathcal{R}$, and we establish correspondences between PIOPs for a relation over a ring $\mathcal{R}$, and a PIOP for an homomorphic image $\mathcal{R}'$ of $\mathcal{R}$. A reader wishing to tone down the abstractness of the presentation can replace $\mathcal{R}$ by the local ring $\mathbb{Z}_{(q)}$ (or even $\mathbb{Z}$, even though we will never instantiate the schemes in this section on $\mathbb{Z}$), and $\mathcal{R}'$ by $\mathbb{F}_q$, for q a prime.

Recall that, given a ring $\mathcal{R}$, we implicitly fix an encoding of its elements as strings of bits. Given $B \geq 1$, by $\mathcal{R}_B$ we denote the subset of $\mathcal{R}$ consisting of ring elements whose encoding has at most B bits.

Definition 4.1 (Algebraic indexed relation over a ring $\mathcal{R}$)**.** *Let $\mathcal{R}$ be a ring, and let $\mathcal{Q}$ be a set of polynomials with coefficients in $\mathcal{R}$ (possibly multivariate and of arbitraty degree). Let $\mathsf{gp} = (k, m, n, \mu, B)$ be global parameters where k, m, n, μ, B are size parameters. Abusing the language, we set gp to also include the security parameter λ, the ring $\mathcal{R}$, and the polynomials $\mathcal{Q}$, but we do not explicitly display them inside gp. Instead, we refer to $\mathcal{R}, \mathcal{Q}$ in more explicit ways. We do so because in our constructions, gp stays fixed, while $\mathcal{R}$ and $\mathcal{Q}$ often vary.*

An algebraic indexed relation *for the parameters $(\mathsf{gp}, \mathcal{R}, \mathcal{Q})$ is a set $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ of triples $(\mathbb{i}, \mathbb{x}; \mathbb{w})$ with the following properties:*

- *The* index $\mathbb{i}$ *contains n oracles $[[g_1]], \ldots, [[g_n]]$ to multilinear polynomials $g_1, \ldots, g_n \in \mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]$, where $\mathbf{X} = (X_1, \ldots, X_\mu)$.*
- *$\mathcal{Q}$ is a set of polynomials with coefficients in $\mathcal{R}$, each on $(n + k) \cdot 2^\mu + m$ variables.*
- *$\mathbb{w}$ is a vector consisting of k multilinear polynomials $f_1(\mathbf{X}), \ldots, f_k(\mathbf{X})$ from $\mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]$.*
- *$\mathbb{x} = (\mathbf{y}, [[f_1]], \ldots, [[f_k]])$, where $\mathbf{y} \in \mathcal{R}_B^m$.*
- *Each of the polynomials in $\mathcal{Q}$ vanishes when evaluated on the values*
$$((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x}\in\{0,1\}^\mu}, \mathbf{y}).$$

Formally, $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ has the following form:

$$\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}} = \left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} = (k, m, n, \mu, B), \\ \mathbb{i} = ([[g_1]], \ldots, [[g_n]]), \\ \mathbb{x} = (\mathbf{y}, [[f_1]], \ldots, [[f_k]]) \text{ for some } \mathbf{y} \in \mathcal{R}_B^m, \\ \mathbb{w} = (f_1(\mathbf{X}), \ldots, f_k(\mathbf{X})) \in \left(\mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]\right)^n, \ \mathbf{X} = (X_1, \ldots, X_\mu), \\ (g_1(\mathbf{X}), \ldots, g_n(\mathbf{X})) \in \left(\mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]\right)^n, \\ Q((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x}\in\{0,1\}^\mu}, \mathbf{y}) = 0 \text{ for all } Q \in \mathcal{Q} \end{array} \right\}$$

In Section 4.4 we show that both the CCS relation and the lookup relation (Section 3.5) can be rewritten as algebraic indexed relations. In the next definition we generalize the notion of algebraic indexed relation.

Definition 4.2 (Projected algebraic indexed relation). *Let* $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ *be an algebraic indexed relation with parameters* $(\mathsf{gp}, \mathcal{R}, \mathcal{Q})$*, and let* $\phi : \mathcal{R} \to \mathcal{R}'$ *be a ring homomorphism. We define an associated relation, which we call* ϕ-projected algebraic indexed relation*, denoted by* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$*, as the relation* $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$*, with the only difference that in* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ *we require that the image by* ϕ *of the polynomials in* $\mathcal{Q}$ *vanishes. Formally (we highlight the difference between* $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ *and* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ *in blue):*

$$\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}) = \left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} := (k, m, n, \mu, B), \\ \mathbb{i} := ([[g_1]], \dots, [[g_n]]), \\ \mathbb{x} = (\mathbf{y}, [[f_1]], \dots, [[f_k]]) \textit{ for some } \mathbf{y} \in \mathcal{R}_B^m, \\ \mathbb{w} = (f_1(\mathbf{X}), \dots, f_k(\mathbf{X})) \in \left(\mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]\right)^k, \ \mathbf{X} = (X_1, \dots, X_\mu), \\ (g_1(\mathbf{X}), \dots, g_n(\mathbf{X})) \in \left(\mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}]\right)^n, \\ \color{blue}{\phi(Q((g_1(\mathbf{x}), \dots, g_n(\mathbf{x}), f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))_{\mathbf{x} \in \{0,1\}^\mu}, \mathbf{y})) = 0} \\ \hfill \textit{for all } Q \in \mathcal{Q} \end{array} \right\}$$

Importantly, in a projected algebraic indexed relation, the oracles and polynomials in $(\mathbb{i}, \mathbb{x}; \mathbb{w})$ are over the ring $\mathcal{R}$, even though the constraints posed by the polynomials $\mathcal{Q}$ are enforced only under the image of ϕ.

Notice that, by taking $\mathcal{R}' = \mathcal{R}$ and ϕ the identity homomorphism (i.e. $\phi(a) = a$ for all $a \in \mathcal{R}$) we have that $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}) = \mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$. Hence, projected algebraic indexed relations are a generalization of algebraic indexed relations.

Definition 4.3 (Well-formed index-instance pairs). *Let* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ *be a projected algebraic indexed relation. Following Remark 3.3, we say that* $(\mathbb{i}, \mathbb{x})$ *is a* well-formed index-instance pair for the relation $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ *if* $\mathbb{i}$ *and* $\mathbb{x}$ *have the form specified in the definition of* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$*. Namely, if* $\mathbb{i} = (\mathsf{gp}, [[g_1]], \dots, [[g_n]])$ *and* $\mathbb{x} = (\mathbf{y}, [[f_1]], \dots, [[f_k]])$ *where* $g_1, \dots, g_n, f_1, \dots, f_k$ *are all* μ*-variate multilinear polynomials from* $\mathcal{R}_B[\mathbf{X}]$*, and* $\mathbf{y}$ *is a tuple of* m *elements from* $\mathcal{R}_B$*.*

Let $\phi : \mathcal{R} \to \mathcal{R}'$ be a ring homomorphism and let $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ be an algebraic indexed relation. By $\phi(\mathcal{Q})$ we define the set containing the image under ϕ of the polynomials in $\mathcal{Q}$. One can then consider the relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. By definition, this is precisely,

$$\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})} = \left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} = (k, m, n, \mu, B), \ n, m, k, B \geq 1, \\ \mathbb{i} = ([[g_1]], \dots, [[g_n]]), \\ \mathbb{x} = (\mathbf{y}, [[f_1]], \dots, [[f_k]]) \text{ for some } \mathbf{y} \in \left(\mathcal{R}'_B\right)^m, \\ \mathbb{w} = (f_1(\mathbf{X}), \dots, f_k(\mathbf{X})) \in \left({\mathcal{R}'_B}^{\mathsf{multilin}}[\mathbf{X}]\right)^k, \ \mathbf{X} = (X_1, \dots, X_\mu), \\ (g_1(\mathbf{X}), \dots, g_n(\mathbf{X}) \in \left({\mathcal{R}'_B}^{\mathsf{multilin}}[\mathbf{X}]\right)^n, \\ \phi(Q)((g_1(\mathbf{x}), \dots, g_n(\mathbf{x}), f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))_{\mathbf{x} \in \{0,1\}^\mu}, \mathbf{y}) = 0 \\ \hfill \text{for all } Q \in \phi(\mathcal{Q}) \end{array} \right\}$$

The following definition is necessary to make sure many of the constructions in this section are well defined and result in efficient algorithms.

Definition 4.4 (Efficient homomorphism ϕ)**.** *We say that a ring homomorphism $\phi : \mathcal{R} \to \mathcal{R}'$ is* efficient *if 1) $\phi(\mathcal{R}_B) = \mathcal{R}'_B$ for all $B \geq 1$ (in words, the image by ϕ of the elements from $\mathcal{R}$ of bit-size less than B can be written with less than B bits); 2) $\phi(a)$ can be computed in polynomial time on the bit-size of a, for all $a \in \mathcal{R}$; and 3) given $a' \in \mathcal{R}'_B$, it is possible to find $a \in \mathcal{R}_B$ such that $\phi(a) = a'$ in polynomial time on the bit-length of a'.*

Given an index $\mathbb{i} = ([[g_1]], \dots, [[g_n]])$ for $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ and a homomorphism $\phi : \mathcal{R} \to \mathcal{R}'$, we define $\phi(\mathbb{i}) = (\phi([[g_1]]), \dots, \phi([[g_n]]))$.

The next observation follows immediately from Definition 4.4.

Remark 4.1. Suppose $\phi : \mathcal{R} \to \mathcal{R}'$ is an efficient ring homomorphism. Then, given any well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, we have that $(\phi(\mathbb{i}), \phi(\mathbb{x}))$ is a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$ which can be computed efficiently.

Conversely, given a well-formed index-instance pair $(\mathbb{i}', \mathbb{x}')$ for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\mathcal{Q}'}$, there exists a well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ that can be computed efficiently.

Given an oracle $[[f]]$ to a polynomial $f \in \mathcal{R}[\mathbf{X}]$, we let $\phi([[f]]) = [[\phi(f)]]$. In the scenario where V has received an oracle $[[f]]$, V can query the oracle $\phi([[f]])$ as follows: first, it queries $[[f]]$ at the desired position, and then V applies the homomorphism ϕ to the received value.

Lemma 4.2. *Let $\phi : \mathcal{R} \to \mathcal{R}'$ be an efficient ring homomorphism, and let $(\mathbb{i}, \mathbb{x})$ be a well-formed index-instance pair for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$. Then $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}))$ if and only if $(\phi(\mathbb{i}), \phi(\mathbb{x})) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})})$.*

Proof. Write $\mathsf{gp} = (k, m, n, \mu, B)$, $\mathbb{i} = ([[g_1]], \dots, [[g_n]])$, $\mathbb{x} = (\mathbf{y}, [[f_1]], \dots, [[f_k]])$. Since ϕ is efficient, $(\phi(\mathbb{i}), \phi(\mathbb{x}))$ is a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$, due to Remark 4.1.

Assume first that $(\phi(\mathbb{i}), \phi(\mathbb{x})) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})})$. Then there exists

$$\mathbb{w}' = (h_1(\mathbf{X}), \dots, h_k(\mathbf{X})) \in {\mathcal{R}'}_B^{\mathsf{multilin}}[\mathbf{X}]$$

such that $(\phi(\mathbb{i}), \phi(\mathbb{x}); \mathbb{w}') \in \mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. In particular, $[[h_i]] = \phi([[f_i]])$, and so $h_i(\mathbf{x}) = \phi(f_i)(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\mu$ and all $i \in [k]$. For all $Q' \in \phi(\mathcal{Q})$, we have $Q' = \phi(Q)$ for some $Q \in \mathcal{Q}$, and, by definition and because ϕ is a ring homomorphism,

$$\begin{aligned} 0 &= Q'((\phi(g_1)(\mathbf{x}), \dots, \phi(g_n)(\mathbf{x}), h_1(\mathbf{x}), \dots, h_k(\mathbf{x}))_{\mathbf{x} \in \mathbb{B}^\mu}, \phi(\mathbf{y})) \\ &= \phi(Q((g_1(\mathbf{x}), \dots, g_n(\mathbf{x}), f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))_{\mathbf{x} \in \mathbb{B}^\mu}, \mathbf{y})). \end{aligned}$$

Finally, since $(\mathbb{i}, \mathbb{x})$ is a well-formed index-instance pair for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, we have that $g_i(\mathbf{X}) \in \mathcal{R}_B^{\mathsf{multilin}}(\mathbf{X})$ for all $i \in [n]$; $[[f_i]]$ is a string consisting of all the evaluations of a multilinear polynomial $f_i(\mathbf{X}) \in \mathcal{R}_B^{\mathsf{multilin}}(\mathbf{X})$; and $\mathbf{y} \in \mathcal{R}_B^m$. We conclude that, letting $\mathbb{w} = (f_1(\mathbf{X}), \dots, f_k(\mathbf{X}))$, we have $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, and so $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}))$.

Conversely, suppose that $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}))$. Then, there exists $\mathbb{w} = (h_1(\mathbf{X}), \dots, h_k(\mathbf{X})) \in {\mathcal{R}'}_B^{\mathsf{multilin}}[\mathbf{X}]$ such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$. In particular, $h_i(\mathbf{x}) = f_i(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^\mu$. Then, by the definition of $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ and of $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$, and because ϕ is a ring homomorphism, we have that $(\phi(\mathbb{i}), \phi(\mathbb{x}); \phi(\mathbb{w})) \in \mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. $\square$

Now, let $\Pi'_{\mathcal{R}}$ be a PIOP over $\mathcal{R}'$ for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. We introduce the notion of the lift of $\Pi'_{\mathcal{R}}$. Informally, this is a PIOP over $\mathcal{R}$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ where P and V simply apply the map ϕ to all elements from $\mathcal{R}$ and polynomials with coefficients in $\mathcal{R}$, and then execute $\Pi'_{\mathcal{R}}$.

Definition 4.5 (Lift of a PIOP). *Let* $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ *be an algebraic indexed relation with parameters* $(\mathsf{gp},\mathcal{R},\mathcal{Q})$. *Let* $\phi : \mathcal{R} \to \mathcal{R}'$ *be an efficient (cf. Definition 4.4) ring homomorphism. Let* $\Pi_{\mathcal{R}'} = (\mathsf{Indexer}_{\mathcal{R}'}, \mathsf{P}_{\mathcal{R}'}, \mathsf{V}_{\mathcal{R}'})$ *be a PIOP over* $\mathcal{R}'$ *for* $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. *In Protocol 5 we describe a PIOP over* $\mathcal{R}$ *for* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, *denoted* $\Pi^{\mathsf{lift}}_{\mathcal{R}'} = (\mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}, \mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}, \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'})$, *and called the* lift of $\Pi'_{\mathcal{R}}$ onto $\mathcal{R}$.

Protocol 5 A PIOP $\Pi^{\mathsf{lift}}_{\mathcal{R}'} = (\mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}, \mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}, \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'})$ over $\mathcal{R}$ for the relation $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, called the *lift* of $\Pi'_{\mathcal{R}} = (\mathsf{Indexer}_{\mathcal{R}'}, \mathsf{P}_{\mathcal{R}'}, \mathsf{V}_{\mathcal{R}'})$.

Indexer: Given gp and $\mathbb{i} = ([[g_1]], \dots, [[g_n]])$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$, the indexer $\mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}$ runs $\mathsf{Indexer}_{\mathcal{R}'}$ on input gp and $\phi(\mathbb{i})$, and obtains verifier and prover parameters vp, pp as output. By definition (cf. Section 3.4), $\mathsf{vp} = (\mathsf{vp}', ([[\phi(g_i)]])_{i\in[n]})$ and $\mathsf{pp} = (\mathsf{pp}', (\phi(g_i))_{i\in[n]})$ for some $\mathsf{vp}', \mathsf{pp}'$. Then $\mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}$ outputs $\mathsf{vp}^{\mathsf{lift}} = (\mathsf{vp}', ([[g_i]])_{i\in[n]})$ and $\mathsf{pp}^{\mathsf{lift}} = (\mathsf{pp}', (g_i)_{i\in[n]})$.

Input: Let $(\mathbb{i}, \mathbb{x})$ be a well-formed index-instance pair for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$. $\mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}$ and $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$ receive $(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}, \mathbb{w})$ and $(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})$ as input, respectively, where $(\mathsf{vp}^{\mathsf{lift}}, \mathsf{pp}^{\mathsf{lift}}) \leftarrow \mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{gp}, \mathbb{i})$. Let $(\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}_{\mathcal{R}'}(\mathsf{gp}, \phi(\mathbb{i}))$.

Interactive phase: Let k be the number of communication rounds in $\Pi_{\mathcal{R}'}$, and let $\mathcal{M}_1, \dots, \mathcal{M}_{k+1}, \mathcal{C}_1, \dots, \mathcal{C}_k$ be the message and challenge spaces of $\Pi_{\mathcal{R}'}$.

1: Let m_1 be a first message output by $\mathsf{P}_{\mathcal{R}'}(\mathsf{pp}, \phi(\mathbb{x}); \phi(\mathbb{w}))$. Then $\mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}$ sends m_1 to $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$.
2: For $i = 1, \dots, k$,

- $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$ uniformly samples a challenge ρ_i in the challenge space $\mathcal{C}_i$, and sends ρ_i to $\mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}$.
- Let m_{i+1} be output by $\mathsf{P}_{\mathcal{R}'}(\mathsf{pp}, \phi(\mathbb{x}); \phi(\mathbb{w}))$ after having output messages $m_1, \dots, m_i$ and received challenges $\rho_1, \dots, \rho_i$. Then $\mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}$ sends m_{i+1} to $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$.

3: $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$ outputs $\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}, \phi(\mathbb{x}), m_1, \rho_1, \dots, m_k, \rho_k, m_{k+1})$.

Lemma 4.3. *Let* $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ *be an algebraic indexed relation with parameters* $(\mathsf{gp},\mathcal{R},\mathcal{Q})$. *Let* $\phi : \mathcal{R} \to \mathcal{R}'$ *be an efficient ring homomorphism. Let* $\Pi_{\mathcal{R}'} = (\mathsf{Indexer}_{\mathcal{R}'}, \mathsf{P}_{\mathcal{R}'}, \mathsf{V}_{\mathcal{R}'})$ *be a PIOP over* $\mathcal{R}'$ *for an algebraic indexed relation* $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. *Suppose* $\Pi_{\mathcal{R}'}$ *has soundness error* $\varepsilon_{\mathsf{sound}}$ *and completeness error* $\varepsilon_{\mathsf{comp}}$. *Then* $\Pi^{\mathsf{lift}}_{\mathcal{R}'}$ *is a PIOP over* $\mathcal{R}'$ *for the relation* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\mathcal{Q}})$ *with soundness and completeness errors* $\varepsilon^{\mathsf{lift}}_{\mathsf{sound}}, \varepsilon^{\mathsf{lift}}_{\mathsf{comp}}$ *satisfying*

$$\varepsilon^{\mathsf{lift}}_{\mathsf{sound}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \varepsilon_{\mathsf{sound}}(\mathsf{gp}, \phi(\mathbb{i}), \phi(\mathbb{x})), \quad \varepsilon^{\mathsf{lift}}_{\mathsf{comp}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \varepsilon_{\mathsf{comp}}(\mathsf{gp}, \phi(\mathbb{i}), \phi(\mathbb{x}))$$

for all well-formed index-instance pairs $(\mathbb{i}, \mathbb{x})$ *for* $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ *(note that by Remark 4.1,* $(\phi(\mathbb{i}), \phi(\mathbb{x}))$ *is well-formed as well).*

Proof. We begin by proving the soundness error equality. Let $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}$ be a malicious prover for $\Pi^{\mathsf{lift}}_{\mathcal{R}'}$. We define a malicious prover $\mathsf{P}^*_{\mathcal{R}'}$ for $\Pi_{\mathcal{R}'}$ that, for each well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ such that $(\mathbb{i}, \mathbb{x}) \notin \mathsf{LANG}(\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}))$, succeeds in convincing $\mathsf{V}_{\mathcal{R}'}$ on input $(\mathsf{vp}, \phi(\mathbb{x}))$ with at least the same probability as $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}$ convinces $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}$. We describe $\mathsf{P}^*_{\mathcal{R}'}$ in Protocol 6.

Protocol 6 Malicious prover $\mathsf{P}^*_{\mathcal{R}'}$ for $\Pi_{\mathcal{R}'}$ constructed from a malicious prover $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}$ for $\Pi^{\mathsf{lift}}_{\mathcal{R}'}$.

Input: Let $(\mathbb{i}', \mathbb{x}')$ be a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. Let $(\mathsf{vp}', \mathsf{pp}') \leftarrow \mathsf{Indexer}_{\mathcal{R}'}(\mathsf{gp}, \mathbb{i}')$. $\mathsf{P}^*_{\mathcal{R}'}$ receives $(\mathsf{pp}', \mathbb{x}')$. Let k be the number of rounds of $\Pi^{\mathsf{lift}}_{\mathcal{R}'}$ and let $\mathcal{M}_1, \ldots, \mathcal{M}_{k+1}, \mathcal{C}_1, \ldots, \mathcal{C}_k$ be its message and challenge spaces. Note that these are also the number of rounds and message/challenge spaces of $\Pi_{\mathcal{R}'}$.

1: $\mathsf{P}^*_{\mathcal{R}'}$ first finds a well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ for $\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}})$ such that $(\phi(\mathbb{i}), \phi(\mathbb{x})) = (\mathbb{i}', \mathbb{x}')$. Such a pair exists and is efficiently computable due to Remark 4.1. Then it computes $(\mathsf{vp}^{\mathsf{lift}}, \mathsf{pp}^{\mathsf{lift}}) \leftarrow \mathsf{Indexer}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{gp}, \mathbb{i})$.
2: Let m_1 be a first message output by $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x})$. Then $\mathsf{P}^*_{\mathcal{R}'}$ sends m_1 to $\mathsf{V}_{\mathcal{R}'}$.
3: For $i = 1, \ldots, k$,

- $\mathsf{P}^*_{\mathcal{R}'}$ receives a challenge $\rho_i \in \mathcal{C}_i$ from $\mathsf{V}_{\mathcal{R}'}$.
- Let m_{i+1} be output by $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x})$ after having output messages $m_1, \ldots, m_i$ and received challenges $\rho_1, \ldots, \rho_i$. Then $\mathsf{P}^*_{\mathcal{R}'}$ sends m_{i+1} to $\mathsf{V}_{\mathcal{R}'}$.

Let tr be a transcript of the interaction between $\mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}')$ and $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x}')$. Then, by definition of $\Pi^{\mathsf{lift}}_{\mathcal{R}'}$ we have that $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x}, \mathsf{tr}) = \mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \mathbb{x}', \mathsf{tr})$ (where by $\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x}, \mathsf{tr})$ we mean whether the verifier accepts or rejects after an interaction with transcript tr), and moreover, tr is also a transcript of interaction between $\mathsf{P}^*_{\mathcal{R}'}(\mathsf{pp}', \mathbb{x}')$ and $\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \mathbb{x}')$. Moreover, the probability that tr is output during the protocol $\langle \mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}), \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})\rangle$ and during $\langle \mathsf{P}^*_{\mathcal{R}'}(\mathsf{pp}', \phi(\mathbb{x})), \mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}))\rangle$ is the same. Hence

$$\begin{aligned}
&\Pr[\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x}, \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}), \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})\rangle] \\
= \; &\Pr[\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}), \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}^{\mathsf{lift},*}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}), \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})\rangle] \\
= \; &\Pr[\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}), \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}^*_{\mathcal{R}'}(\mathsf{pp}', \phi(\mathbb{x})), \mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}))\rangle],
\end{aligned}$$

where the notation $\mathsf{tr} \leftarrow \langle \mathsf{P}, \mathsf{V}\rangle$ indicates the transcript resulting of an execution of an interactive protocol between a prover P and a verifier V. By Lemma 4.2, since $(\mathbb{i}, \mathbb{x})$ is well-formed, we have that $(\mathbb{i}, \mathbb{x}) \in \mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}$ if and only if $(\phi(\mathbb{i}), \phi(\mathbb{x})) \in \mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})}$. Hence $\varepsilon^{\mathsf{lift}}_{\mathsf{sound}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \varepsilon_{\mathsf{sound}}(\mathsf{gp}, \phi(\mathbb{i}), \phi(\mathbb{x}))$.

Now, we prove the completeness error bound. Let $(\mathbb{i}, \mathbb{x})$ be well-formed, with $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi(\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}}))$. Then, by Lemma 4.2, $(\phi(\mathbb{i}), \phi(\mathbb{x})) \in \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp},\mathcal{R}',\phi(\mathcal{Q})})$. Moreover, by similar reasons as argued above, we have

$$\begin{aligned}
&\Pr[\mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x}, \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}), \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})\rangle] \\
= \; &\Pr[\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}), \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{pp}^{\mathsf{lift}}, \mathbb{x}), \mathsf{V}^{\mathsf{lift}}_{\mathcal{R}'}(\mathsf{vp}^{\mathsf{lift}}, \mathbb{x})\rangle] \\
= \; &\Pr[\mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}), \mathsf{tr}) = 1 \mid \mathsf{tr} \leftarrow \langle \mathsf{P}_{\mathcal{R}'}(\mathsf{pp}', \phi(\mathbb{x})), \mathsf{V}_{\mathcal{R}'}(\mathsf{vp}', \phi(\mathbb{x}))\rangle].
\end{aligned}$$

Hence $\varepsilon^{\mathsf{lift}}_{\mathsf{comp}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \varepsilon_{\mathsf{comp}}(\mathsf{gp}, \phi(\mathbb{i}), \phi(\mathbb{x}))$. □

4.2 Constructing a PIOP over a ring from PIOPs over quotients of subrings

We proceed to describe our Zinc-PIOP framework. We fix $\mathcal{D}$ to be an integral domain and $\mathbb{K}$ its field of fractions. We let $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ be an algebraic indexed relation. For each tuple $\mathsf{gp} = (k, m, n, \mu, B)$, we let $\eta(\mathsf{gp}) = \eta \geq 1$ be a positive integer, and let $\mathfrak{R}_{\mathsf{gp}} = \{\mathcal{R}_i \mid i \in [\eta]\}$ be a finite collection of subrings of $\mathbb{K}$ (we sometimes omit the dependency on gp for the sake of readability). We assume without further reference that, for all $i \in [\eta]$, it is computationally cheap to check whether a given ring element $a \in \mathbb{K}$ belongs to $\mathcal{R}_i$ or not.

The reader may find it helpful to think of $\mathcal{D}$ as the ring of integers $\mathbb{Z}$, of $\mathbb{K}$ as the field of rational numbers $\mathbb{Q}$, and of the subrings $\mathcal{R}_i$ as the local rings $\mathbb{Z}_{(q_i)}$ for different primes q_i. In this setting, later, the morphisms $\phi_i : \mathcal{R}_i \to \mathcal{R}'_i$ would become the canonical projections $\phi_{q_i} : \mathbb{Z}_{(q_i)} \to \mathbb{F}_{q_i}$. Indeed, in Section 4.3, we instantiate Zinc-PIOP under this setting.

Definition 4.6 (Compatibility of relations with families of subrings)**.** *We say that* $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ *is* compatible *with* $\mathfrak{R}_{\mathsf{gp}}$ *if all polynomials in* $\mathcal{Q}$ *have coefficients in* $\mathcal{D} \cap \mathcal{R}_B$*, for all* $\mathcal{R} \in \mathfrak{R}_{\mathsf{gp}} = \{\mathcal{R}_i \mid i \in [\eta]\}$*, where* B *is the bit-size bound parameter from* gp*.*

Notice that then the algebraic indexed relations $\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}}$ *are well-defined.*

We further fix ring homomorphisms $\Phi_{\mathsf{gp}} = \{\phi_i : \mathcal{R}_i \to \mathcal{R}'_i \mid i \in [\eta]\}$, and for each $i \in [\eta]$ we let $\Pi_i = (\mathsf{Indexer}_i, \mathsf{P}_i, \mathsf{V}_i)$ be a PIOP for the algebraic indexed relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R}'_i,\mathcal{Q}}$ with soundness error ε_i.

Remark 4.4. Suppose $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ is compatible with $\mathfrak{R}_{\mathsf{gp}}$. Let $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ and $i \in [\eta]$ be such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \notin \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$. Write $\mathbb{i} = (g_1, \ldots, g_n) \in \mathbb{K}_B^{\mathsf{multilin}}[\mathbf{X}]^n$, $\mathbb{w} = (f_1, \ldots, f_k) \in \mathbb{K}_B^{\mathsf{multilin}}[\mathbf{X}]^k$ and $\mathbb{x} = (\mathbf{y}, [[f_1]], \ldots, [[f_k]])$ with $\mathbf{y} \in \mathbb{K}_B^m$, and define $\mathbf{v}(\mathbb{x}, \mathbb{w}) = ((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x} \in \{0,1\}^\mu}, \mathbf{y})$. Then some entry in $\mathbf{v}(\mathbb{x}, \mathbb{w})$ does not belong to $(\mathcal{R}_i)_B$.

This follows directly from the definition of the relations $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ and $\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$. Indeed, assume our claim is false. We know that $Q(\mathbf{v}(\mathbb{x}, \mathbb{w})) = 0$ over $\mathbb{K}$, for all $Q \in \mathcal{Q}$. If our claim does not hold, then all entries in $\mathbf{v}(\mathbb{x}, \mathbb{w})$ belong to $\mathcal{R}_i$, and so $\phi_i(Q(\mathbf{v}(\mathbb{x}, \mathbb{w})))$ is well-defined and zero over $\mathcal{R}'_i$. Hence, $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$. A contradiction.

Definition 4.7 (Expanding family of homomorphisms)**.** *Let* $\mathcal{D}$ *be an integral domain and let* $\mathbb{K}$ *be its field of fractions. Let* $m \geq 1$*, let* $\mathfrak{R} = \{\mathcal{R}_i \mid i \in [m]\}$ *be a collection of subrings of* $\mathbb{K}$*, and let* $\Phi = \{\phi_i : \mathcal{R}_i \to \mathcal{R}'_i \mid i \in [m]\}$ *be a collection of ring homomorphisms. Let* $k \geq 1$ *(one may think* $k = \lambda$*). We say that* $(\mathfrak{R}, \Phi)$ *is* k-expanding *if the following two properties hold:*

1. *For any subset* $I \subseteq [\eta]$*, we have that all elements from*

$$\mathbb{K} \setminus \bigcup_{i \in I} \mathcal{R}_i$$

have an encoding as string of bits of bit-size at least $k \cdot |I|$*.*

2. *For all* $B, n \geq 1$ *and polynomial* $P \in \mathcal{D}_B[Y_1, \ldots, Y_n]$*, the following holds: assume* $\mathbf{y} \in \mathcal{R}^n$ *is such that*

$$P(\mathbf{y}) \in \bigcap_{i \in I} \ker \phi_i \setminus \{0\}$$

for some nonempty subset $I \subseteq [m]$. Then $\mathbf{y}$ contains an entry whose bitstring encoding has more than

$$\frac{k \cdot |I| - (B + \log(n_m(P)))}{\sum_{i \in [n]} \deg_{Y_i}(P)}$$

bits, where $n_m(P)$ denotes the number of nonzero coefficients of P, and $\deg_{Y_i}(P)$ denotes the degree of the variable Y_i in P.

We describe a PIOP for the algebraic indexed relation $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ as follows. Let $(\mathbb{i}, \mathbb{x})$ be a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$, with $\mathbb{i} = ([[g_1]], \ldots, [[g_n]])$. Upon receiving $(\mathsf{gp}, \mathbb{i})$, the indexer outputs $\mathsf{vp} = (\mathsf{gp}, [[g_1]], \ldots, [[g_n]])$, and $\mathsf{pp} = (\mathsf{gp}, g_1, \ldots, g_n)$.

Protocol 7 Zinc-PIOP: A PIOP over $\mathbb{K}$ for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ from $\{\phi_i, \Pi_i \mid i \in [\eta]\}$.

Input: Let $(\mathbb{i}, \mathbb{x})$ be a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ and let $(\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i})$. P receives $(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ and V receives $(\mathsf{vp}, \mathbb{x})$. We assume $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$. Let $\mathfrak{R}_{\mathsf{gp}} = \{\mathcal{R}_i \mid i \in [\eta]\}$ and $\Phi_{\mathsf{gp}} = \{\phi_i \colon \mathcal{R}_i \to \mathcal{R}'_i \mid i \in [\eta]\}$ be collections of η subrings of $\mathbb{K}$ and morphisms, respectively. For each $i \in [\eta]$, let $\Pi_i = (\mathsf{Indexer}_i, \mathsf{P}_i, \mathsf{V}_i)$ be a PIOP for $\mathsf{REL}_{\mathsf{gp},\mathcal{R}'_i,\phi_i(\mathcal{Q})}$.

1: V samples $i \in [\eta]$ uniformly at random, and sends i to P.

2: Suppose $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \notin \mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}}$. Then by Remark 4.4 there exists an entry of $\mathbf{v}(\mathbb{x}, \mathbb{w}) = ((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x} \in \{0,1\}^\mu}, \mathbf{y})$ that does not belong to $(\mathcal{R}_i)_B$. P indicates V what this entry is and V checks[6] that indeed the entry does not belong to $\mathcal{R}_i$. If this is the case, V accepts the proof, and the protocol terminates.

3: Otherwise, P and V execute $\Pi_i^{\mathsf{lift}} = (\mathsf{Indexer}_i^{\mathsf{lift}}, \mathsf{P}_i^{\mathsf{lift}}, \mathsf{V}_i^{\mathsf{lift}})$ (cf. Protocol 5), providing $(\mathsf{pp}', \mathbb{x}; \mathbb{w})$ as input to $\mathsf{P}_i^{\mathsf{lift}}$, and $(\mathsf{vp}', \mathbb{x})$ as input to $\mathsf{V}_i^{\mathsf{lift}}$, where $(\mathsf{vp}', \mathsf{pp}') \leftarrow \mathsf{Indexer}_i^{\mathsf{lift}}(\mathsf{gp}, \mathbb{i})$. V accepts if and only if $\mathsf{V}_i^{\mathsf{lift}}$ accepts at the end of the execution of Π_i^{lift}.

If all the PIOPs Π_i have k rounds of communication, then the PIOP in Protocol 7 has $k+1$ rounds of communication. The prover's message spaces are the empty set (P does not send any message before the verifier's first challenge) and the message spaces of the PIOPs Π_i. Precisely, the j-th message space of the PIOP in Protocol 7 is $\bigcup_{i \in [\eta]} \mathcal{M}_{ij}$ where $\mathcal{M}_{ij}$ is the j-th message space of Π_i. The challenge spaces of Protocol 7 are $[\eta]$ and the challenge spaces of the PIOPs Π_i.

Theorem 4.5. *Let gp be global parameters, let $\mathcal{D}$ be an integral domain, and let $\mathbb{K}$ be its field of fractions. Let $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ be an algebraic indexed relation. Let $\mathfrak{R}_{\mathsf{gp}} = \{\mathcal{R}_i \mid i \in [\eta]\}$ be a finite collection of subrings of $\mathbb{K}$, and let $\Phi_{\mathsf{gp}} = \{\phi_i : \mathcal{R}_i \to \mathcal{R}'_i \mid i \in [\eta]\}$ be efficient ring homomorphisms (cf. Definition 4.4). For each $i \in [\eta]$, let Π_i be a PIOP for the algebraic indexed relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R}'_i,\phi_i(\mathcal{Q})}$ with soundness error ε_i.*

Assume that $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ is compatible with $\mathfrak{R}_{\mathsf{gp}}$ and that $(\mathfrak{R}_{\mathsf{gp}}, \Phi_{\mathsf{gp}})$ is λ-expanding. Then Protocol 7 is a PIOP over $\mathbb{K}$ for the relation $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ with soundness error

$$\varepsilon_{\mathsf{sound}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) \leq \max_{i \in [\eta]} \varepsilon_i(\mathsf{gp}, \mathbb{i}, \mathbb{x}) + \frac{B \cdot (\mathsf{maxdegt}(\mathcal{Q}) + 1) + \log(n_{\max}(\mathcal{Q}))}{\lambda \cdot (\eta - \xi)} + \frac{\xi}{\eta}, \tag{9}$$

[6]The check is made by V by either inspecting an appropriate entry of the vector $\mathbf{y}$, or by querying some of the oracles $f_1, \ldots, f_k, g_1, \ldots, g_n$ at an appropriate point. Note that both P and V have the required access to $g_1, \ldots, g_n$ since they received, respectively, pp and vp.

where

$$\xi = \frac{((n+k)\cdot 2^\mu + m)\cdot B}{\lambda}$$

for all well-formed $(\mathbb{i}, \mathbb{x})$*. Here* $\mathsf{maxdegp}(\mathcal{Q}) = \max_{Q\in\mathcal{Q}}(\sum_i \deg_{Y_i}(Q))$ *is the maximum of the sum of partial degrees of the polynomials in* $\mathcal{Q}$*; and* $\log(n_{\max}(\mathcal{Q})) = \max_{Q\in\mathcal{Q}} \log(n_m(Q))$*, where* $n_m(Q)$ *is the number of nonzero coefficients of* Q*. In particular, if* $\max_{i\in[\eta]} \varepsilon_i, n, k, 2^\mu, m$*, and* B *are polynomial in* λ*, and if* $\eta = O(2^\lambda)$*, then* $\varepsilon_{\mathsf{sound}}$ *is negligible.*

Further, Protocol 7 has knowledge soundness error $\varepsilon_{\mathsf{sound}}$*, and it has completeness error the maximum completeness error of the PIOPs in* $\{\Pi_i \mid i \in [\eta]\}$*. In particular, Protocol 7 is perfectly complete if all* $\{\Pi_i \mid i \in [\eta]\}$ *are perfectly complete.*

Proof. The statement regarding the completeness error of Protocol 7 follows immediately from the fact that ϕ_i is a ring morphism for all i, and so if $Q(\mathbf{v}) = 0$ for $Q \in \mathcal{Q}$ and a vector $\mathbf{v}$ with entries in R_i, then $\phi(Q)(\mathbf{v}) = 0$. We next show that the soundness error of Protocol 7 satisfies inequality (9). Then, by Lemma 3.4, we obtain that the knowledge soundness error of Protocol 7 satisfies the same bound.

Let P^* be a malicious prover for Protocol 7. Let $(\mathbb{i}, \mathbb{x})$ be well-formed for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ and such that $(\mathbb{i}, \mathbb{x}) \notin \mathsf{LANG}(\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}})$. Let $(\mathsf{vp}, \mathsf{pp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i})$. Let $\mathcal{E}_{\mathsf{wf}}$ be the event that V samples $i \in [\eta]$ such that $(\mathbb{i}, \mathbb{x})$ is a well-formed index-instance pair for $\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$, and let $\mathcal{E}$ be the event that V samples $i \in [\eta]$ such that $\mathcal{E}_{\mathsf{wf}}$ holds and $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}}))$. Due to Lemma 4.3, Π_i^{lift} has soundness error ε_i (for well formed index-instance pairs for $\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$). Hence,

$$\Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \neg\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}] \le \max_{i\in[\eta]} \varepsilon_i(\mathsf{gp}, \mathbb{i}, \mathbb{x}).$$

Now, we have

$$\begin{aligned}
&\Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1] \\
&= \Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \mathcal{E}_{\mathsf{wf}}] \cdot \Pr[\mathcal{E}_{\mathsf{wf}}] + \Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \neg\mathcal{E}_{\mathsf{wf}}] \cdot \Pr[\neg\mathcal{E}_{\mathsf{wf}}] \\
&\le \Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \mathcal{E}_{\mathsf{wf}}] + \Pr[\neg\mathcal{E}_{\mathsf{wf}}] \\
&\le \begin{pmatrix} \Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \neg\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}] \cdot \Pr[\neg\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] \\ \qquad + \Pr[\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1 \mid \mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}] \cdot \Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] + \Pr[\neg\mathcal{E}_{\mathsf{wf}}] \end{pmatrix} \\
&\le \max_{i\in[\eta]} \varepsilon_i(\mathbb{i}, \mathbb{x}) + \Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] + \Pr[\neg\mathcal{E}_{\mathsf{wf}}].
\end{aligned}$$

We first bound $\Pr[\neg\mathcal{E}_{\mathsf{wf}}]$. Let $I_0 \subseteq [\eta]$ be the set of indices $i \in I_0$ such that $(\mathbb{i}, \mathbb{x})$ is not a well-formed index-instance pair for the relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}}$. Then $\Pr[\neg\mathcal{E}_{\mathsf{wf}}] = |I_0|/\eta$. As before, let $\mathbf{v} = ((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x}\in\mathbb{B}^\mu}, \mathbf{y})$. Then, since $(\mathbb{i}, \mathbb{x})$ is a well-formed index-instance pair for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$, we have that, for each $i \in I_0$, at least one entry of $\mathbf{v}$ does not belong to $(\mathcal{R}_i)_B$ (cf. Remark 4.4). Let v_0 be the entry of $\mathbf{v}$ that does not belong to the maximum possible number of subrings $\{(\mathcal{R}_i)_B \mid i \in I_0\}$. Let $J \subseteq I_0$ be such that $v_0 \notin (\mathcal{R}_j)_B$ for all $j \in J$. By the pigeonhole principle,

$$|J| \ge \frac{|I_0|}{|\mathbf{v}|} = \frac{|I_0|}{(n+k)\cdot 2^\mu + m}.$$

Since $(\mathbb{i}, \mathbb{x})$ is well-formed for $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$, we have that $v_0 \in \mathbb{K}_B$, and so $v_0 \in \mathbb{K}_B \setminus \bigcup_{j \in J} \mathcal{R}_j$. Then, since we assumed that $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ is compatible with $\mathfrak{R}_{\mathsf{gp}}$, we have that the bitstring encoding of v_0 has size at least $\lambda \cdot |J|$, hence

$$\lambda \cdot \frac{|I_0|}{(n+k) \cdot 2^\mu + m} \leq \lambda \cdot |J| \leq B.$$

It follows that

$$\Pr[\neg \mathcal{E}_{\mathsf{wf}}] = \frac{|I_0|}{\eta} \leq \frac{((n+k) \cdot 2^\mu + m) \cdot B}{\lambda \cdot \eta}.$$

Next, we bound $\Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}]$. Let $\mathcal{E}_L$ be the event that $(\mathbb{i}, \mathbb{x}) \in \mathsf{LANG}(\phi_i(\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}))$. Let I be the set of indices $i \in [\eta]$ for which $\mathcal{E}$ holds. Let $I_{\mathsf{wf}} \subseteq [\eta]$ be the set of all $i \in [\eta]$ such that $(\mathbb{i}, \mathbb{x})$ is well-formed for $\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$. We have $I_{\mathsf{wf}} = [\eta] \setminus I_0$. Then

$$\Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] = \frac{\Pr[\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}]}{\Pr[\mathcal{E}_{\mathsf{wf}}]} = \frac{\Pr[\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}]}{1 - |I_0|/\eta} \leq \frac{\eta \cdot \Pr[\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}]}{\eta - \xi}, \tag{10}$$

where

$$\xi = \frac{((n+k) \cdot 2^\mu + m) \cdot B}{\lambda}.$$

We proceed to bound $\Pr[\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}] = \Pr[\mathcal{E}_L \wedge \mathcal{E}_{\mathsf{wf}}]$. For each $i \in I$, let $\mathbb{w}_i$ be such that $(\mathbb{i}, \mathbb{x}; \mathbb{w}_i) \in \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$ (which exists by definition of $\mathcal{E}$). Write $\mathsf{gp} = (k, m, n, \mu, B)$, $\mathbb{i} = ([[g_1]], \ldots, [[g_n]])$, $\mathbb{x} = (\mathbf{y}, [[f_1]], \ldots, [[f_k]])$, and $\mathbb{w}_i = (f_{i1}(\mathbf{X}), \ldots, f_{ik}(\mathbf{X}))$. By Definition 4.2, since each $(\mathbb{i}, \mathbb{x}; \mathbb{w}_i) \in \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$, we have that for each $i \in I$ and $j \in [k]$, $[[f_j]]$ is the string of evaluations of $f_{ij}(\mathbf{X})$ on $\{0,1\}^\mu$ (or some subset of $\mathbb{K}^\mu$ that allows to fully describe multilinear polynomials). It follows that, for all $j \in [k]$ and $i_1, i_2 \in I$, $f_{i_1 j}(\mathbf{X}) = f_{i_2 j}(\mathbf{X})$ as polynomials. Hence, denoting the polynomials $\{f_{ij} \mid i \in I\}$ by f_j, for all $j \in [k]$, and letting $\mathbb{w} = (f_1(\mathbf{X}), \ldots, f_k(\mathbf{X}))$, we have $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$ for all $i \in I$.

Denote $\mathbf{v} = ((g_1(\mathbf{x}), \ldots, g_n(\mathbf{x}), f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))_{\mathbf{x} \in \mathbb{B}^\mu}, \mathbf{y})$. Since $(\mathbb{i}, \mathbb{x})$ is well-formed for $\phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$ if $i \in I$, we have that $g_j(\mathbf{x}) \in (\mathcal{R}_i)_B$ for all $i \in [\eta], j \in [n]$ and all $\mathbf{x} \in \mathbb{B}^\mu$ (since the oracles $[[g_j]]$ are strings of evaluations of a polynomial in $\{0,1\}^\mu$), and that $\mathbf{y}$ is a tuple of elements from $(\mathcal{R}_i)_B$. By similar reasons, $f_j(\mathbf{x}) \in (\mathcal{R}_i)_B$ for all $j \in [k], i \in [\eta], \mathbf{x} \in \mathbb{B}^\mu$. Hence, $\mathbf{v} \in (\mathcal{R}_i)_B^{(n+k) \cdot 2^\mu + m}$, for all $i \in [\eta]$.

Additionally, because $\mathsf{REL}_{\mathsf{gp},\mathbb{K},\mathcal{Q}}$ is compatible with $\{\mathcal{R}_i \mid i \in [\eta]\}$, we have that all coefficients of all polynomials of $\mathcal{Q}$ belong to $\mathcal{D} \cap \mathcal{R}_i$. In particular, they belong to $\mathcal{R}_i$. We conclude that $Q(\mathbf{v}) \in \mathcal{R}_i$ for all $Q \in \mathcal{Q}$, and so $\phi_i(Q(\mathbf{v}))$ is well-defined. Now, since $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \phi_i(\mathsf{REL}_{\mathsf{gp},\mathcal{R}_i,\mathcal{Q}})$ for all $i \in I$, we have

$$\begin{aligned} \phi_i(Q(\mathbf{v})) &= 0 \text{ for all } i \in [I] \\ \Leftrightarrow Q(\mathbf{v}) &\in \bigcap_{i \in [I]} \ker \phi_i. \end{aligned}$$

Hence, since we assumed $(\{\mathcal{R}_i \mid i \in [\eta]\}, \{\phi_i \colon \mathcal{R}_i \to \mathcal{R}'_i \mid i \in [\eta]\})$ is λ-expanding and each $Q \in \mathcal{Q}$ has coefficients in $\mathcal{D}$ of bit-size at most B, we have that, for all $Q \in \mathcal{Q}$, at least one entry v_0 in $\mathbf{v}$ has an encoding with bit-size larger than

$$\begin{aligned} &\frac{\lambda \cdot |I| - (B + \log(n_m(Q)))}{\mathsf{degp}(Q)} \geq \frac{\lambda \cdot \Pr[\mathcal{E} \wedge \mathcal{E}_{\mathsf{wf}}] \cdot \eta - (B + \log(n_m(Q)))}{\mathsf{degp}(Q)} \\ \geq &\frac{\lambda \cdot \Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] \cdot (\eta - \xi) - (B + \log(n_m(Q)))}{\mathsf{degp}(Q)} \end{aligned}$$

where we have used (10). Above, $n_m(Q)$ denotes the number of nonzero coefficients of Q, and $\mathsf{degp}(Q) = \sum_{i\in[n]} \deg_{Y_i}(Q)$ is the sum of partial degrees of Q. On the other hand, by our previous arguments, the bit-size of v_0 is at most B, and so

$$\Pr[\mathcal{E} \mid \mathcal{E}_{\mathsf{wf}}] \le \frac{\mathsf{maxdegp}(\mathcal{Q}) \cdot B + (B + \log(n_{\max}(\mathcal{Q})))}{\lambda \cdot (\eta - \xi)},$$

where $\mathsf{maxdegp}(\mathcal{Q}) := \max_{Q\in\mathcal{Q}}(\mathsf{degp}(Q))$ and $\log(n_{\max}(\mathcal{Q})) = \max_{Q\in\mathcal{Q}} \log(n_m(Q))$. This completes the proof that Protocol 7 has soundness error bounded as in Eq. (9). $\square$

4.3 Instantiation over $\mathbb{Q}$ and finite fields

We next instantiate Protocol 7 with the specific setting where $\mathbb{K} = \mathbb{Q}$, $\mathcal{D} = \mathbb{Z}$, $\mathcal{R}_i = \mathbb{Z}_{(q_i)}$ for a prime q_i, and ϕ_i is the canonical projection $\phi_{q_i} : \mathbb{Z}_{(q_i)} \to \mathbb{F}_{q_i}$. To do so, it suffices to show that these subrings and homomorphisms satisfy the requirements in Theorem 4.5.

Proposition 4.6. *Consider the ring of integers $\mathbb{Z}$, which is an integral domain, and its field of fractions $\mathbb{Q}$, i.e. the field of rational numbers. Let $k \ge 1$ and let $\mathcal{P} = \{q_1, \ldots, q_m\}$ be a set of m different primes, each of bit-size at least k. Let $\phi_{q_i} : \mathbb{Z}_{(q_i)} \to \mathbb{F}_{q_i}$ be the canonical projections of the local ring $\mathbb{Z}_{(q_i)} \subseteq \mathbb{Q}$ onto the finite field $\mathbb{F}_{q_i}$ (cf. Section 3.2). Then the pair $(\{\mathbb{Z}_{(q_i)} \mid i \in [m]\}, \{\phi_{q_i} \mid i \in [m]\})$ is k-expanding (Definition 4.7).*

Proof. The first condition in Definition 4.7 is satisfied because each prime q_i has bit-size larger than k, and

$$\mathbb{Q} \setminus \bigcup_{i\in I} \mathbb{Z}_{(q_i)} = \left\{ \tfrac{a}{b} \in \mathbb{Q} \setminus \{0\}, \;\middle|\; b \equiv 0 \mod \prod_{i\in I} q_i \right\},$$

for any $I \subseteq [m]$. Hence, any element from $\mathbb{Q} \setminus \cup_{i\in I} \mathbb{Z}_{(q_i)}$ has a representation a/b in lowest form (cf. Section 3.2) with $b > \prod_{i\in I} q_i \ge 2^{(k-1)\cdot|I|}$. In particular, a/b has bit-size at least $k \cdot |I|$. This proves that Item 1 of Definition 4.7 holds. We now prove that Item 2 does as well.

Let $B, n \ge 1$ and $P \in \mathbb{Z}_B[\mathbf{Y}]$, where $\mathbf{Y} = (Y_1, \ldots, Y_n)$. Assume first that P is multilinear. Let $\mathbf{y} \in \mathbb{Q}^n$ be such that $P(\mathbf{y}) \in \cap_{i\in I} \ker \phi_{q_i} \setminus \{0\}$ for some nonempty $I \subseteq [m]$. Write $\mathbf{y} = \left(\frac{y_1}{y_1'}, \ldots, \frac{y_n}{y_n'}\right)$ where for all $i \in [m]$, (y_i, y_i') is in lowest form, i.e. $y_i, y_i' \in \mathbb{Z}$, $y_i' \ge 1$, and $\gcd(y_i, y_i') = 1$. For any subset $S \subseteq [n]$, we define integers

$$y_S = \prod_{i\in S} y_i, \quad y_S' = \prod_{i\in S} y_i', \quad y_0' = \prod_{i\in[n]} y_i', \quad \overline{y_S'} = \prod_{i\in[n]\setminus S} y_i' = \frac{y_0'}{y_S'}.$$

Further, let $\{c_S \mid S \subseteq [n]\}$ be the coefficients of P, so that $P(\mathbf{y}) = \sum_{S\subseteq[n]} c_S \cdot y_S / y_S'$. Now, since $P(\mathbf{y}) \in \bigcap_{i\in I} \ker \phi_{q_i} \setminus \{0\}$, we have

$$P(\mathbf{y}) = \frac{a}{b} \prod_{i\in I} q_i \tag{11}$$

for some $a/b \in \mathbb{Q}$ in lowest form, where in particular $a, b \in \mathbb{Z}$, $b \geq 1$. Moreover

$$\gcd\left(a \prod_{i \in I} q_i, b\right) = 1, \tag{12}$$

because $\gcd(a, b) = 1$ by definition, and if $\gcd(b, \prod_i q_i) \neq 1$, then because all primes q_i are pairwise different, $\prod_{i \in I} q_i/b$ is not divisible by at least one of the prime $q_{i_0}, i_0 \in I$. Further, q_{i_0} cannot divide a because if it did, $\gcd(a, b)$ would not be 1. Hence, we would have that $P(\mathbf{y})$ is not in $\cap_{i \in I} \ker \phi_{q_i} \setminus \{0\}$, a contradiction.

Further, $a \neq 0$ because $P(\mathbf{y}) \neq 0$. Note that since $P \in \mathbb{Z}_B[\mathbf{Y}]$, all coefficients of P are integers of bit-size at most B. Multiplying (11) by y_0' on both sides we obtain

$$y_0' \cdot P(\mathbf{y}) = \sum_{S \subseteq [n]} c_S \cdot y_S \cdot \overline{y_S'} = y_0' \cdot \frac{a}{b} \cdot \prod_{i \in [I]} q_i \in \mathbb{Z},$$

where we the above expression belongs to $\mathbb{Z}$ because the elements c_S, y_S, y_S' are all integers. This implies that b divides y_0', because of Eq. (12). Moreover $y_0' \cdot (a/b) \neq 0$ because $a \neq 0$ as we argued, and $y_0' \neq 0$ because it is the product of nonzero integers. Hence $y_0' \cdot a/b$ is an integer, and it follows then that $\prod_{i \in I} q_i$ divides the integer $\sum_{S \subseteq [n]} c_S \cdot y_S \cdot \overline{y_S'}$. Then ,

$$\prod_{i \in I} q_i \leq \left| \sum_{S \subseteq [n]} c_S \cdot y_S \cdot \overline{y_S'} \right| \leq \sum_{S \subseteq [n]} |c_S| \cdot |y_S| \cdot |\overline{y_S'}| \leq \max_{S \subseteq [n]}(|c_S|) \cdot n_m \cdot \max_{S \subseteq [n]}(|y_S| \cdot |\overline{y_S'}|),$$

where n_m denotes the number of nonzero elements in the vector $(c_S)_{S \subseteq [n]}$. By assumption, $\max_{S \subseteq [n]}(|c_S|) < 2^B$. Hence, setting $S_{\max}$ to be the subset achieving the maximum value among $|y_S| \cdot |\overline{y_S'}|$, we have

$$|y_{S_{\max}}| \cdot |\overline{y'_{S_{\max}}}| \geq \frac{\prod_{i \in I} q_i}{(n_m \cdot 2^B)} > 2^{k \cdot |I| - B - \log(n_m)}.$$

Then

$$\log(|y_{S_{max}}| \cdot |\overline{y'_{S_{\max}}}|) = \sum_{i \in S_{\max}} \log(|y_i|) + \sum_{i \in [n] \setminus S_{\max}} \log(|y_i'|) > k \cdot |I| - (B + \log(n_m)).$$

Hence, there must exist an index $i \in S_{max}$ or $j \in [n] \setminus S_{max}$ such that either $\log(|y_i|)$ or $\log(|y_j'|)$ is strictly greater than $(k \cdot |I| - (B + \log(n_m)))/n$. This proves that $(\{\mathbb{Z}_{(q_i)} \mid i \in [m]\}, \phi_{q_i} \mid i \in [m])$ satisfies Property 2 of Definition 4.7 when the polynomial P is multilinear.

The case when $P \in \mathbb{Z}_B[\mathbf{Y}]$ is not multilinear can be treated by reducing it to the previous case. Indeed, there exists a multilinear polynomial P' with $n' = \sum_{i \in [n]} \deg_{Y_i}(P)$ variables such that, for all $\mathbf{y} \in \mathbb{Q}^n$, we have $P(\mathbf{y}) = P'(\mathbf{y}')$, where $\mathbf{y}' \in \mathbb{Q}^{n'}$ is the vector $\mathbf{y}$ after creating $\deg_{Y_i}(P)$ copies of y_i, for each $i \in [n]$. Then the result follows.

More precisely, to create P', we introduce, for each $i \in [n]$, $\deg_{Y_i}(P)$ fresh variables $Y_{i,1}, \ldots, \cdots Y_{i,\deg_{Y_i}(P)}$, and then replace, for each i, each occurrence of a factor Y_i^d (with d maximal) in a monomial of P by the factor $Y_{i,1} \cdots Y_{i,d}$. $\square$

Theorem 4.7 (PIOP over $\mathbb{Q}$ from PIOP over finite fields)**.** *Consider the integral domain $\mathbb{Z}$ and its field of fractions $\mathbb{Q}$. Fix global parameters $\mathsf{gp} = (k, m, n, \mu, B)$. let $\eta \geq 1$ and $\mathcal{P}_\lambda = \{q_i \mid i \in [\eta]\}$ be a set of η distinct primes, each of bit-size at least λ. Let $\phi_{q_i} : \mathbb{Z}_{(q_i)} \to \mathbb{F}_{q_i}$ be the canonical projection of $\mathbb{Z}_{(q_i)}$ onto $\mathbb{F}_{q_i}$.*

Let $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}}$ be an algebraic indexed relation. Assume that $\mathcal{Q}$ consists of polynomials with coefficients in $\mathbb{Z}_B$, and that $B > \max_{i\in[\eta]}\{\log(q_i)\}$. Let Π_i be PIOPs for the relations $\mathsf{REL}_{\mathsf{gp},\mathbb{F}_{q_i},\phi_{q_i}(\mathcal{Q})}$, $i \in [\eta]$. Then Protocol 7 instantiated by taking $\mathcal{R} = \mathbb{Q}$, $\mathcal{R}_i = \mathbb{Z}_{(q_i)}$, $\phi_i = \phi_{q_i}$ is a PIOP for the relation $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}}$, with soundness and completeness errors prescribed by Theorem 4.5.

Proof. Proposition 4.6 yields that the pair $(\{\mathbb{Z}_{(q_i)} \mid q_i \in \mathcal{P}_\lambda\}, \{\phi_{q_i} \mid q_i \in \mathcal{P}_\lambda\})$ is λ-expanding. Since $Q \in \mathcal{Q}$ has coefficients in $\mathbb{Z}_B$ and $\mathbb{Z}_B \subseteq \mathbb{Z}_{(q_i)}$ for all i, $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}}$ is compatible with $\{\mathbb{Z}_{(q_i)} \mid q_i \in \mathcal{P}_\lambda\}$. Finally, we argue that the morphisms ϕ_{q_i} are efficient for any $B > \max_{i\in[\eta]}\{\log(q_i)\}$. Indeed, Items 2 and 3 in Definition 4.4 hold trivially, and Item 1 holds because since B is large enough, we have $\phi_{q_i}((\mathbb{Z}_{(q_i)})_B) = \mathbb{F}_{q_i} = (\mathbb{F}_{q_i})_B$ for all q_i, where the first equality is due to the fact that $(\mathbb{Z}_{(q_i)})_B$ contains the integrak interval $[0, q_i - 1]$. The theorem then follows from Theorem 4.5. $\square$

4.4 Application to CCS and lookup relations over $\mathbb{Q}$

In this section, we reformulate the CCS and lookup relations from Section 3.5 as algebraic indexed relations. Then we use Theorem 4.7 to construct a PIOP over $\mathbb{Q}$ for such relations from known PIOPs for the finite-field version of these relations. The latter can be, for example, the PIOP versions of SuperSpartan [STW23a] for the CCS case, and of Lasso or LogUp [STW23b, PH23] in the lookup case. We refer to Section 2.2 for further details.

Informally speaking, notice that having a PIOP $\Pi_{\mathsf{lookup},\mathbb{Q}}$ over $\mathbb{Q}$ for the lookup relation effectively allows us to build PIOPs *over* $\mathbb{Q}$ for relations expressed over the ring of integers $\mathbb{Z}$. This is because one can always use $\Pi_{\mathsf{lookup},\mathbb{Q}}$ to enforce witnesses to have all their entries in a bounded integer set. We emphasize the fact that the latter results in a PIOP *over* $\mathbb{Q}$, i.e. a PIOP whose security holds against provers that are guaranteed to send oracles to polynomials with coefficients in $\mathbb{Q}$. In particular, such PIOP can be compiled with a PCS for rational polynomials. This allows us to avoid making the limiting restriction of working with PIOP's over $\mathbb{Z}$, which later need to be compiled with a PCS for integral polynomials, which is a primitive that is difficult to construct. We refer to our Technical Overview for a more thorough explanation of this matter.

Lookup relations We describe an algebraic indexed relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{\mathsf{lookup}}}$ which, in a sense that we will make precise later, is equivalent to the lookup relation defined in Section 3.5.

Let $\mathsf{gp} = n_a, n_t, B$ be global parameters with $n_a, n_t, B \geq 1$ and both n_a, n_t powers of two. Let $\mathbf{W}$ denote $n_a + n_t$ variables indexed as follows:

$$\mathbf{W} = ((W_{\mathbf{x}})_{\mathbf{x}\in\{0,1\}^{\log(n_a)}}, (W_{\mathbf{y}})_{\mathbf{y}\in\{0,1\}^{\log(n_t)}}).$$

We define the following collection of polynomials:

$$\mathcal{Q}_{\mathsf{Look}} = \left\{ Q_{\mathbf{x}}(\mathbf{W}) = \prod_{\mathbf{y}\in\{0,1\}^{\log(n_t)}} (W_{\mathbf{x}} - W_{\mathbf{y}}) \;\middle|\; \mathbf{x} \in \{0,1\}^{\log(n_a)} \right\}.$$

Assume $\mathcal{D}$ is an integral domain, and let $a(\mathbf{X})$ and $t(\mathbf{Y})$ be two multilinear polynomials on $\log(n_a)$ and $\log(n_t)$ variables, respectively, with coefficients in $\mathcal{D}$. We have that $Q_{\mathbf{x}}(\mathbf{a}, \mathbf{t}) = 0$ if and only if some entry in $\mathbf{t}$ is equal to $a(\mathbf{x})$, where $\mathbf{a} = (a(\mathbf{x}))_{\mathbf{x}\in\{0,1\}^{\log(n_a)}}$ and $\mathbf{t} = (t(\mathbf{y}))_{\mathbf{y}\in\{0,1\}^{\log(n_t)}}$. Hence, if $Q_{\mathbf{x}}(\mathbf{a}, \mathbf{t}) = 0$ for all $\mathbf{x} \in \{0,1\}^{\log(n_a)}$, then all the values in $\mathbf{a}$ appear in $\mathbf{t}$.

With this in mind, we define the following algebraic indexed relation:

$$\mathsf{REL}_{\mathsf{gp},\mathcal{D},\mathcal{Q}_{\mathsf{Look}}} = \left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} = (k = 2, m = 0, n = 1, \mu = \log(n_a) + \log(n_t), B), \\ \mathbb{i} = ([[t]]), \\ \mathbb{x} = ([[a]]), \\ a(\mathbf{X}) \in \mathcal{D}_B^{\mathsf{multilin}}[\mathbf{X}],\ \mathbf{X} = (X_1, \ldots, X_{\log(n_a)}), \\ t(\mathbf{Y}) \in \mathcal{D}_B^{\mathsf{multilin}}[\mathbf{Y}],\ \mathbf{Y} = (Y_1, \ldots, Y_{\log(n_t)}), \\ \mathbb{w} = (a(\mathbf{X})), \\ Q_{\mathbf{x}}(\mathbf{a}, \mathbf{t}) = 0 \text{ for all } \mathbf{x} \in \{0,1\}^{\log(n_a)} \end{array} \right\},$$

where $\mathbf{a}, \mathbf{t}$ have the same meaning as above.

Then we have that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{\mathsf{Look}}}$ if and only if $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{Look}_{\mathsf{gp}'}$ where $\mathsf{Look}_{\mathsf{gp}'}$ is defined as in Section 3.5, and $\mathsf{gp}' = (\mathcal{R}, n_a, n_t, B)$.

Regarding the maximum of the sum of partial degrees, degp, we have

$$\mathsf{maxdegp}(\mathcal{Q}) = \max_{Q_{\mathbf{x}}\in\mathcal{Q}_{\mathsf{Look}}} \mathsf{degp}(Q_{\mathbf{x}}) = \max_{Q_{\mathbf{x}}\in\mathcal{Q}_{\mathsf{Look}}} \sum_{W\in\mathbf{W}} \deg_W(Q_{\mathbf{x}}) = n_a + n_t.$$

Remark 4.8. Notice that $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{\mathsf{Look}}}$ and $\mathcal{Q}_{\mathsf{Look}}$ do not fully fit into the definition of algebraic indexed relation, because both $a(\mathbf{X})$ and $t(\mathbf{Y})$ should be polynomials on the same variables, and the order of the entries in $Q_{\mathbf{x}}$ is not consistent with the order used in Definition 4.1. We opted for this for ease of presentation and readability. To fully adhere to Definition 4.1, it suffices to formally consider $a(\mathbf{X})$ and $t(\mathbf{Y})$ as polynomials on the variables $(\mathbf{X}, \mathbf{Y})$ (even though the variables $\mathbf{Y}$ do not appear in $a(\mathbf{X})$ and $\mathbf{X}$ do not appear in $t(\mathbf{Y})$), and reorder the way the variables in $\mathbf{W}$ are indexed.

Now assume $\mathcal{R} = \mathbb{Q}$. Let $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$, for a prime q. Then $\mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\phi_q(\mathcal{Q}_{\mathsf{Look}})} = \mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\mathcal{Q}_{\mathsf{Look}}}$ is a lookup relation expressed over the finite field $\mathbb{F}_q$. Using any suitable family of PIOPs for the lookup relations over finite fields (eg. [STW23b, PH23]), we obtain a PIOP for $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}_{\mathsf{Look}}}$ from Theorem 4.7 with knowledge soundness and completeness errors as prescribed in Theorem 4.5. This PIOP is the result of instantiating Protocol 7 accordingly.

Remark 4.9 (Using Protocol 7 to prove range checks into big integer intervals)**.** As we discussed in Section 2.2, we are especially interested in using $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}_{\mathsf{Look}}}$ with $\mathbf{t}(\mathbf{Y})$ having as coefficients the integers in an integer interval of the form $[-2^B+1, 2^B-1]$, where $B = \mathsf{poly}(\lambda)$.

In this case, we have that $\mathsf{maxdegp}(\mathcal{Q}) \leq 2^{B+3+\log(n_a)}$. Say we want to prove statements about such a relation with Protocol 7. Then the soundness and knowledge soundness error of the resulting PIOP has a dominant factor of the form $\mathsf{maxdegp}(\mathcal{Q})/\eta$ in one of its term (cf. Theorem 4.5), where, recall, η is the number of subrings and ring homomorphisms our PIOP is built from, i.e. it is the number of morphisms ϕ_i that can be sampled by V in Step 1 of Protocol 7. Hence, to make sure the error stays negligible, one must choose η as, roughly, $2^{B+\log(n_a)+\lambda}$.

Alternatively, one can write a different algebraic indexed relation for membership to the range $[-2^B+1, 2^B-1]$ that leverages Lasso's notion of SOS decomposability [STW23b]. With such technique one can bring down the value of $\mathsf{maxdegp}(\mathcal{Q})$ as low as necessary, at the expense of slightly increasing the number of witness vectors, and adding some extra small linear constraints that will need to be proved by the PIOP.

Remark 4.10. Even though $\mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\phi_q(\mathcal{Q}_{\mathsf{Look}})} = \mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\mathcal{Q}_{\mathsf{Look}}}$ has relatively complicated polynomial constraints in its definition, the relation is still equivalent to a lookup relation, in the sense that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\phi_q(\mathcal{Q}_{\mathsf{Look}})}$ if and only if $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{Look}_{\mathsf{gp}'}$, where gp' is a suitable choice of global parameters. Hence, one does not need to prove those polynomial constraints directly. One can indeed use known lookup arguments such as Lasso and logUp [STW23b, PH23].

Customizable Constraint Systems Next, we describe an algebraic indexed relation $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{\mathsf{CCS}}}$ which is equivalent to the CCS relation defined in Section 3.5, in the same sense as seen above.

Fix global parameters $\mathsf{gp} = (k, m, n, \mu, B)$ with $k = 1$. Let $\mathbf{c} = (c_1, \ldots, c_q) \in \mathcal{R}_B^m$, and let $\mathbf{S} = (S_1, \ldots, S_q)$ be q multisets whose elements belong to $[n]$, each with size at most d, for d an auxiliary parameter. Let $\mu_1, \mu_2 \geq 1$ be numbers of variables, and write $\mu = \mu_1 + \mu_2$. Let $1 \leq \ell \leq \mu_2$, and fix $n \cdot 2^\mu + 2^{\mu_2}$ variables

$$\mathbf{M} = (M_j(\mathbf{x}, \mathbf{y}))_{j\in[n],\,(\mathbf{x},\mathbf{y})\in\{0,1\}^\mu}, \quad \mathbf{Z} = (Z(\mathbf{y}))_{\mathbf{y}\in\{0,1\}^{\mu_2}}.$$

Then we define the collection of polynomials

$$\mathcal{Q}_{CCS}^{\mathbf{S},\mathbf{c},\mu_1,\mu_2,\ell} = \left\{ \begin{array}{l} Q_{\mathbf{x}}(\mathbf{M}, \mathbf{Z}) = \sum_{i\in[m]} c_i \cdot \left(\prod_{j\in S_i} \left(\sum_{\mathbf{y}\in\{0,1\}^{\mu_2}} M_j(\mathbf{x}, \mathbf{y}) \cdot Z(\mathbf{y}) \right) \right) \\ Q'(\mathbf{M}, \mathbf{Z}) = Z_{(1,\overset{\mu_2}{\ldots},1)} - 1 \end{array} \;\middle|\; \mathbf{x} \in \{0,1\}^{\mu_1} \right\}.$$

For ease of notation, we denote the above collection of polynomials simply as $\mathcal{Q}_{CCS}$. Then we consider the following algebraic indexed relation

$$\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{CCS}} =$$

$$\left\{ (\mathbb{i}, \mathbb{x}; \mathbb{w}) \;\middle|\; \begin{array}{l} \mathsf{gp} = (k = 1, m = 2^{\ell}, n, \mu, B), \\ \mathbb{i} = ([[M_1]], \ldots, [[M_n]]), \\ M_1, \ldots, M_n \in \mathcal{R}_B^{\mathsf{multilin}}[\mathbf{X}],\ \mathbf{X} = (X_1, \ldots, X_\mu), \\ \mathbb{x} = (\mathbf{y}, [[f_1]]) \text{ for } \mathbf{y} \in \mathcal{R}_B^{2^\ell}, \\ \mathbb{w} = f_1(X) \in \mathcal{R}_B^{\mathsf{multilin}}[X_{\mu_1+1,\ldots,\mu_2-\ell}], \\ \mathbf{X}_{\mu_2} = (X_{\mu_1+1}, \ldots, X_\mu) \subseteq \mathbf{X}, \\ Q((M_1(\mathbf{x}), \ldots, M_n(\mathbf{x}))_{\mathbf{x}\in\{0,1\}^\mu}, (f_1(\mathbf{y}))_{\mathbf{y}\in\{0,1\}^{\mu_2-\ell}}, \mathbf{y}) = 0 \text{ for all } Q \in \mathcal{Q}_{CCS} \end{array} \right\}.$$

Then we have that $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{CCS}}$ if and only if $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{CCS}_{\mathsf{gp}'}$, where $\mathsf{CCS}_{\mathsf{gp}}$ is defined as in Section 3.5, and $\mathsf{gp}' = (\mathcal{R}, 2^{\mu_1}, 2^{\mu_2}, \ell, n, q, d, \mathbf{S}, \mathbf{c})$.

Remark 4.11. Notice that, similarly to the case of $\mathcal{Q}_{\mathsf{Look}}$ and $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{\mathsf{Look}}}$ discussed in Remark 4.8, when we defined $\mathcal{Q}_{CCS}$ and $\mathsf{REL}_{\mathsf{gp},\mathcal{R},\mathcal{Q}_{CCS}}$, we slightly deviated from Definition 4.1 in that we ordered the variable entries of the polynomials in $\mathcal{Q}$ differently. Again, this was done for ease of presentation, but it should be clear how to reorder the variables to fit the original Definition 4.1.

Regarding the sum of partial degrees, degp, we have $\mathsf{degp}(Q') = 1$ and

$$\mathsf{degp}(Q_{\mathbf{x}}) \leq (m+1) \cdot 2^{\mu_2} \cdot d,$$

where we recall that $m = 2^{\ell}$.

Now assume $\mathcal{R} = \mathbb{Q}$ and $\mathbf{c} \in \mathbb{Z}_B^m$. Let $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$, for a prime q. Then $\mathsf{REL}_{\mathsf{gp},\mathbb{F}_q,\phi_q(\mathcal{Q}_{CCS})}$ is a CCS relation expressed over the finite field $\mathbb{F}_q$. Using any suitable family of PIOPs for CCS relations over finite fields (eg. [STW23a]), we obtain a PIOP for $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}_{CCS}}$ from Theorem 4.7 with knowledge soundness and completeness errors as prescribed in Theorem 4.5. The PIOP is the result of instantiating Protocol 7 accordingly.

5 Zip: An efficient PCS from an IOP of proximity to the integers

Zip is a Brakedown-like polynomial commitment scheme [GLS+23] for multilinear polynomials with (bounded) rational coefficients. Zip is based on what we call an *IOP of proximity (IOPP) to the integers*. In a nutshell, Zip is meant to be used to commit to polynomials with integer coefficients of bit-size less than B', but it only guarantees that the coefficients are rational numbers of bit-size less than B, for certain $B > B'$. So, intuitively, Zip allows to prove proximity to integers of B' bits. This scenario is analogous to what one has with IOPPs to a linear code (see e.g. [BSBHR18]). Such a primitive guarantees that a prover knows a word that is close to being a codeword, but it is meant to be used by provers that actually know the codeword.

We refer to Section 2.3 for an intuitive but detailed explanation of the ideas used in this section.

5.1 Projectable codes

Before describing Zip we discuss linear codes over $\mathbb{Q}$ with integral generator matrices, and introduce the notion of a family of projectable linear codes. In short, this is a collection of codes $\{\mathcal{C}_\lambda\}$ with integral generator matrices M_λ, parameterized by the security parameter λ, such that, for each λ, there is a large number of primes q such that $\phi_q(M_\lambda)$ has full rank and generates a code with minimal distance similar to the minimal distance of $\mathcal{C}$. Recall that $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$ is the canonical projection of $\mathbb{Z}_{(q)}$ onto $\mathbb{F}_q$, and $\phi_q(M_\lambda)$ is the result of applying ϕ_q to all components of M_λ. Zip requires working with such family of codes, because part of its evaluation procedure is run modulo a random prime.

Let $\mathcal{C}$ be a linear code over $\mathbb{Q}$ of length n and dimension dim. Let $M_\mathcal{C}$ be a generator matrix of $\mathcal{C}$. By $\mathsf{Enc}_\mathcal{C} : \mathbb{Q}^{\mathsf{dim}} \to \mathbb{Q}^{\mathsf{n}}$ we denote the linear map assigning to each vector $\mathbf{v} \in \mathbb{Q}^{\mathsf{dim}}$ an encoding in $\mathcal{C}$, which is obtained by multiplying $\mathbf{v}$ with $M_\mathcal{C}$. If $M_\mathcal{C}$ contains only integer entries, i.e. if $M_\mathcal{C} \in \mathbb{Z}^{\mathsf{dim}\times\mathsf{n}}$, we say that $\mathcal{C}$ is an *integral linear code over* $\mathbb{Q}$.

If $\mathcal{C}$ is such a code, given any prime q, we define a new linear code $\mathcal{C}_q$ over $\mathbb{F}_q$ of length n and of dimension $\mathsf{dim}_q \leq \mathsf{dim}$. Concretely, $\mathcal{C}_q$ is the $\mathbb{F}_q$-vector space spanned by the rows of $\phi_q(M_\mathcal{C})$, where $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$ is the canonical projection of $\mathbb{Z}_{(q)}$ onto $\mathbb{F}_q$.

As we mentioned, $\mathsf{dim}_q \leq \mathsf{dim}$. Moreover, $\phi_q(M_\mathcal{C})$ is a generator matrix of $\mathcal{C}_q$ if and only if $\mathsf{dim}_q = \mathsf{dim}$. Additionally, the relative distance of $\mathcal{C}_q$, which we denote dist_q, is at most the relative distance dist of $\mathcal{C}$. We call $\mathcal{C}_q$ the *q-projection* of $\mathcal{C}$.

Sometimes we denote the dimension and relative distance of a code $\mathcal{C}$ by $\mathsf{dist}(\mathcal{C})$ and $\mathsf{dim}(\mathcal{C})$.

Definition 5.1 (Projectable family of linear codes)**.** *Let* $0 < \mathsf{dist}_0 < 1$*. For each security parameter* λ*, let* $\mathcal{C}_\lambda$ *be an integral linear code*[7] *over* $\mathbb{Q}$ *and let* $\mathcal{P}_\lambda$ *be a finite set of primes. Write* $\mathcal{C}_{\lambda q}$ *for the q-projection of* $\mathcal{C}_\lambda$ *with respect to* $q \in \mathcal{P}_\lambda$*. We say that* $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ *is a* $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable family of linear codes over $\mathbb{Q}$ with error $\varepsilon_{\mathsf{proj}}(\lambda)$ *if, for all* $\lambda \geq 1$*, for at least* $(1 - \varepsilon_{\mathsf{proj}}(\lambda)) \cdot |\mathcal{P}_\lambda|$ *primes in* $\mathcal{P}_\lambda$ *we have that* $\mathsf{dim}_q = \mathsf{dim}$ *and* $\mathsf{dist}_q \geq \mathsf{dist}_0$*, where* $\mathsf{dim}_q = \mathsf{dim}(\mathcal{C}_{\lambda q})$ *and* $\mathsf{dist}_q = \mathsf{dist}(\mathcal{C}_{\lambda q})$*.*

We say that a prime $q \in \mathcal{P}_\lambda$ *such that* $\mathsf{dim}_q = \mathsf{dim}$ *and* $\mathsf{dist}_q \geq \mathsf{dist}_0$ *is* good *with respect to* $\mathcal{C}_\lambda, \mathsf{dist}_0$*, and denote the set of such primes by* $\mathcal{P}_{\lambda\mathsf{good}}$*.*

In Section 6 we describe a projectable family of codes with small projection error, large distance parameter dist_0, and with encoding matrices whose entries have bit-size of at most 2λ.

Lemma 5.1. *Let* $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ *be a* $(\mathcal{P}_\lambda, \mathsf{dist}_0)$*-projectable family of linear codes over* $\mathbb{Q}$*. Let* $\mathcal{C}_{\lambda q}$ *denote the q-projection of* $\mathcal{C}_\lambda$*. Then for* $q \in \mathcal{P}_{\lambda\mathsf{good}}$*,* $\phi_q(M_{\mathcal{C}_\lambda})$ *is a generator matrix of* $\mathcal{C}_{\lambda q}$*. Denote by* $\mathsf{Enc}_{\mathcal{C}_\lambda}$ *and* $\mathsf{Enc}_{\mathcal{C}_{\lambda q}}$ *the encoding maps for* $\mathcal{C}_\lambda$ *and* $\mathcal{C}_{\lambda q}$ *consisting of multiplying a vector in* $\mathbb{Q}$ *or in* $\mathbb{F}_q$ *by the matrix* $M_{\mathcal{C}_\lambda}$ *or* $\phi_q(M_{\mathcal{C}_\lambda})$*, respectively. Then, for all* $\mathbf{v} \in \mathbb{Z}_{(q)}^{\mathsf{dim}} \subseteq \mathbb{Q}^{\mathsf{dim}}$*,*

$$\mathsf{Enc}_{\mathcal{C}_{\lambda q}}(\phi_q(\mathbf{v})) = \phi_q(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v})).$$

Proof. Let dim denote the dimension of $\mathcal{C}_\lambda$ and dim_q the dimension of $\mathcal{C}_{\lambda q}$. Since $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ is a $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable family of linear codes over $\mathbb{Q}$, $\mathsf{dim}_q = \mathsf{dim}$ for all $\mathcal{C}_\lambda$ and $q \in \mathcal{P}_{\lambda\mathsf{good}}$.

[7] The length and dimension of the code, its minimal distance, etc., depend on λ, but we omit referring to λ when speaking of these.

Thus, the rank of $\phi_q(M_{\mathcal{C}_\lambda})$ is $\mathsf{dim}_q = \mathsf{dim}$, making it a generator matrix for $\mathcal{C}_{\lambda q}$. Hence, for $\mathbf{v} \in \mathbb{Z}_{(q)}^{\mathsf{dim}}$, since ϕ_q is a ring homomorphism, we have

$$\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\mathbf{v})) = \phi_q(\mathbf{v}) \cdot \phi_q(M_{\mathcal{C}_\lambda}) = \phi_q(\mathbf{v} \cdot M_{\mathcal{C}_\lambda}) = \phi_q(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v})).$$

□

5.2 A Brakedown-type PCS from an IOP of Proximity to the Integers

In this section we describe Zip. We start by fixing some notation and terminology.

Preliminary terminology and notation Throughout the rest of Section 5.2 we let $\mathcal{P}_\lambda$ be a set of primes each of bit-size at least λ, we fix a distance parameter $0 < \mathsf{dist}_0 < 1$, and let $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ be a $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable family of linear integer codes over $\mathbb{Q}$ with error $\varepsilon_{\mathsf{proj}}(\lambda)$, length n, and dimension dim (which may depend on λ), such as the family described in Section 6. By Proposition 4.6, the family of rings and morphisms $\{\mathbb{Z}_{(q)} \mid q \in \mathcal{P}_\lambda\}$, $\{\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q \mid q \in \mathcal{P}_\lambda\}$ is λ-expanding (Definition 4.7).

We will use symbols like $\mathbf{v}$ or $\mathbf{u}$ to denote matrices. In this case, $\mathbf{v}_i$ denotes the i-th row of $\mathbf{v}$, and $\mathbf{v}_{i,j}$ denotes the (i, j)-th entry of $\mathbf{v}$. Recall that, for any rational number $v \in \mathbb{Q}$, we denote its absolute value by $|v|$, and, given $\mathbf{v}$ a vector, we denote by $\|\mathbf{v}\|_\infty = \max_i\{|\mathbf{v}_i|\}$. Similarly, if $\mathbf{v}$ is a matrix then $\|\mathbf{v}\|_\infty$ is the largest absolute value of one of the entries in $\mathbf{v}$.

Recall that by bit-length of a rational number $a/b \in \mathbb{Q}$ we refer to the quantity $\log(|a|) + \log(b)$, where (a, b) are in lowest form (cf. Section 3.2). We also recall (Remark 3.1) that we fix an encoding of $\mathbb{Q}$ as strings of bits such that any a/b in lowest form has an encoding with at most $\log(|a|) + \log(b) + 2$ bits. For simplicity we assume this encoding actually has at most $\log(|a|) + \log(b)$ bits.

Let μ be an even number of variables , $\mathbf{X} = (X_1, \ldots, X_\mu)$, and $f \in \mathbb{Q}^{\mathsf{multilin}}[\mathbf{X}]$ be a multilinear polynomial on μ variables. Let $\mathbf{v}^f = (f(\mathbf{x}))_{\mathbf{x}\in\{0,1\}^\mu}$ be the vector of evaluations of f on the hypercube $\{0, 1\}^\mu$. Denote $\mathsf{dim} = \sqrt{2^\mu}$. Following, e.g. [GLS+23], we see $\mathbf{v}^f$ as a matrix from $\mathbb{Q}^{\mathsf{dim}\times\mathsf{dim}}$, which we call the *coefficient matrix of f*. We use the well-known fact [Tha22] that, for any $\mathbf{q} \in \mathbb{Q}^\mu$, there exists $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{Q}^{\mathsf{dim}}$ such that

$$f(\mathbf{q}) = \mathbf{q}_1 \cdot \mathbf{v}^f \cdot \mathbf{q}_2^\mathsf{T}.$$

Precisely, $\mathbf{q}_1 = (\widetilde{\mathsf{eq}}(\mathbf{x}; \mathbf{q}^{(1)}))_{\mathbf{x}\in\{0,1\}^{\mu/2}}$ and $\mathbf{q}_1 = (\widetilde{\mathsf{eq}}(\mathbf{x}; \mathbf{q}^{(2)}))_{\mathbf{x}\in\{0,1\}^{\mu/2}}$, where $\widetilde{\mathsf{eq}}$ is the equality multilinear polynomial (cf. Section 3.2) on $\mu/2$ variables, $\mathbf{q}^{(1)} = (\mathbf{q}_1, \ldots, \mathbf{q}_{\mu/2})$, and $\mathbf{q}^{(2)} = (\mathbf{q}_{\mu/2+1}, \ldots, \mathbf{q}_\mu)$.

Remark 5.2 (If $\mathbf{q}$ is integral, then so are $\mathbf{q}_1$ and $\mathbf{q}_2$)**.** It follows that if $\mathbf{q} \in \mathbb{Z}^\mu$, then $\mathbf{q}_1 \in \mathbb{Z}^{\mathsf{dim}}$ and $\mathbf{q}_2 \in \mathbb{Z}^{\mathsf{dim}}$.

Description of Zip Next we describe Zip, our version of the Brakedown PCS [GLS+23] for multilinear polynomials with bounded rational coefficients. Following [GLS+23], we present the scheme as a PCS with oracles, in the sense of Definition 3.5. The oracles can then be replaced by Merkle tree commitments using standard compilation methods, as outlined in Section 5.5.

The global parameters of Zip are of the form $\mathsf{gp} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0)$ where μ is an even number of variables, $B_{\mathbf{v}}, B_{\mathsf{pt}} \geq 1$ are bit-size bounds, δ is a distance parameter (for linear codes), and q_0 is a prime of $\Omega(\lambda)$ bits. We further assume $\delta < \mathsf{dist}_0/3$. If all these conditions are met, we say that the global parameters gp are *well-formed*. We also implicitly include $\lambda, \mathbb{Q}$, $\mathcal{P}_\lambda, \mathcal{C}_\lambda$ in gp, but omit writing or referring to them explicitly. Recall that by $\mathbb{Q}_B$ we denote the set of rational numbers whose bit-size is less than B. We also let $\mathbf{X} = (X_1, \ldots, X_\mu)$.

Given such parameters, Zip works as follows:

$\mathsf{Commit}(\mathsf{gp}, f, \mathsf{aux})$**:** Takes global parameters gp, a multilinear polynomial

$$f \in \mathbb{Q}^{\mathsf{multilin}}_{B_{\mathbf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}[\mathbf{X}],$$

with coefficient matrix $\mathbf{v}^f$, and a $\mathsf{dim} \times \mathsf{n}$ matrix $\mathsf{aux} = \hat{\mathbf{u}}$ so that each row $\hat{\mathbf{u}}_i$ of $\hat{\mathbf{u}}$ is δ-close to the codeword $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i)$, for all $i \in [\mathsf{dim}]$.

We place no constraint on the form of the entries of $\hat{\mathbf{u}}_i$ that do not agree with $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i)$. We do not even constraint those to be rational numbers. This reflects the fact that, after replacing oracles by Merkle tree commitments, a prover can potentially get away with making a few (but not many) entries in $[[\hat{\mathbf{u}}_i]]$ be any piece of data of its choice.

Then the commit procedure outputs as commitment the vector of oracles $\mathsf{cm} = ([[\hat{\mathbf{u}}_i]])_{i\in[\mathsf{dim}]}$, and the hint $\mathsf{hint} = \hat{\mathbf{u}}$.

$\mathsf{Open}(\mathsf{gp}, \mathsf{cm}, f, \mathsf{hint})$**:** Check that gp is well-formed and that $f \in \mathbb{Q}^{\mathsf{multilin}}_{B_{\mathbf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}[\mathbf{X}]$. Let $\mathbf{v}^f$ be the coefficient matrix of f. Parse $\mathsf{hint} = \{\hat{\mathbf{u}}_i\}_{i\in\mathsf{dim}}$. The opening procedure returns 1 if and only if $\hat{\mathbf{u}}_i$ is δ-close to $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i)$ for all $i \in [\mathsf{dim}]$, and the oracles in cm are oracles to the vectors $(\hat{\mathbf{u}}_i)_{i\in[\mathsf{dim}]}$ (this can be checked by reading $\hat{\mathbf{u}}$ and cm entirely). We have the procedure stop and return 0 if at any point while reading the oracles in cm or while reading the vectors in hint, some entry is seen to not be a rational number or to have bit-size larger than $\log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_{\mathbf{v}} + (2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})$.

$\mathsf{Eval} = (\mathsf{Indexer}, \mathsf{P}, \mathsf{V})$**.** This is an IOP for the relation $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$. The relation and the IOP are described and discussed below.

Recall that $\mathbf{v}^f$ denotes the coefficient matrix of f. As usual, the expression $\Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i), \hat{\mathbf{u}}_i)$ refers to the relative number of positions where $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i)$ and $\hat{\mathbf{u}}_i$ agree.

$$\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}} = \left\{ \begin{pmatrix} \mathbb{i}, \\ \mathbb{x}; \\ \mathbb{w} \end{pmatrix} = \begin{pmatrix} ([[\widehat{\mathbf{u}}]], \hat{\mathbf{u}}), \\ (\mathbf{q}, y), \\ f \end{pmatrix} \;\middle|\; \begin{array}{l} \mathsf{gp} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0), \\ \widehat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i\in[\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim}\times\mathsf{n}}, \\ [[\widehat{\mathbf{u}}]] = \{[[\hat{\mathbf{u}}_i]]\}_{i\in[\mathsf{dim}]}, \\ \mathbf{q} \in \mathbb{Z}^\mu, \|\mathbf{q}\|_\infty < 2^{B_{\mathsf{pt}}}, \\ y \in \mathbb{Z}, \ |y| < 2^{\mu+B_{\mathbf{v}}+\mu\cdot B_{\mathsf{pt}}}, \\ f \in \mathbb{Q}^{\mathsf{multilin}}_{B_{\mathbf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}[\mathbf{X}], \\ \mathbf{v}^f = \{\mathbf{v}^f_i\}_{i\in[\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim}\times\mathsf{dim}}_{B_{\mathbf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}, \\ \Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}^f_i), \hat{\mathbf{u}}_i) < \delta \text{ for all } i \in [\mathsf{dim}], \\ f(\mathbf{q}) = y \end{array} \right\}$$

Before describing the Zip evaluation IOP, we introduce some terminology. We say $(\mathbb{i}, \mathbb{x})$ are *well-formed* if they satisfy all but the last three constraints from $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$.

Definition 5.2 (Strong witnesses)**.** *Let* $(\mathbb{i}, \mathbb{x})$ *be a well-formed index-instance pair for* $\mathsf{REL}_{\mathsf{Eval}}$*, and write* $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$*. We say that* $\mathbb{w} = f$ *is a* strong witness for $(\mathbb{i}, \mathbb{x})$ *if the following two conditions hold:*

- *The witness* $\mathbb{w} = f$ *is such that* $f \in \mathbb{Z}_{B_{\mathsf{v}}}[\mathbf{X}]$ *(i.e. all coefficients of* f *are integer numbers –as opposed to rational– within the range* $[-2^{B_{\mathsf{v}}} + 1, 2^{B_{\mathsf{v}}} - 1]$*).*
- *We have* $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ *and* $\mathsf{Enc}(\mathbf{v}_i^f) = \hat{\mathbf{u}}_i$ *for all* $i \in [\mathsf{dim}]$*.*

In our schemes, we expect honest provers to only use strong witnesses. However, we cannot fully enforce a dishonest prover to use such witnesses, and hence, in the definition of $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, we allow for a broader class of witnesses. This scenario is analogous to when, in a finite field setting, one uses an IOPP (IOP of proximity) to construct a PCS (see e.g. [GLS+23]): in most cases, the honest prover is expected to commit to one (or more) codewords, but it is only possible to guarantee that the prover has committed to a word that is only close to being a codeword. Additionally, full completeness is only guaranteed in case the prover has committed to a codeword. We refer to Section 2.3 for further intuitive explanation of this phenomenon.

We next describe the evaluation IOP of Zip. The indexer $\mathsf{Indexer}$ receives $(\mathsf{gp}, \mathbb{i})$ and outputs

$$(\mathsf{pp}, \mathsf{vp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i}), \quad \mathsf{pp} = (\mathsf{gp}, \widehat{\mathbf{u}}), \quad \mathsf{vp} = (\mathsf{gp}, [[\widehat{\mathbf{u}}]]).$$

The interactive phase between $\mathsf{P}(\mathsf{pp}, \mathbb{x}, \mathbb{w})$ and verifier $\mathsf{V}(\mathsf{vp}, \mathbb{x})$ is described in Protocol 8.

Protocol 8 Zip's evaluation IOPP for $\mathsf{REL}_{\mathsf{Eval}}$.

Input: Let $\mathsf{gp} = (\mu, B_{\mathsf{v}}, B_{\mathsf{pt}}, \delta, q_0)$ and $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$ with $\hat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i\in[\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim}\times \mathsf{n}}$ be well-formed. Let $\mathbb{x} = (\mathbf{q}, y) \in \mathbb{Z}^{\mu+1}$ with $\|\mathbf{q}\|_\infty < 2^{B_{\mathsf{pt}}}$, $|y| < 2^{\mu+B_{\mathsf{v}}+\mu\cdot B_{\mathsf{pt}}}$. Let $\mathsf{pp} = (\mathsf{gp}, \hat{\mathbf{u}})$, $\mathsf{vp} = (\mathsf{gp}, [[\hat{\mathbf{u}}]])$ be the output of $\mathsf{Indexer}(\mathsf{gp}, \mathbb{i})$. V receives $(\mathsf{vp}, \mathbb{x})$ as input. P receives $(\mathsf{vp}, \mathbb{x}, \mathbb{w})$ with $\mathbb{w} = f$ where $f \in \mathbb{Q}^{\mathsf{multilin}}_{B_{\mathsf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}[\mathbf{X}]$ and $\Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i^f), \hat{\mathbf{u}}_i) < \delta$ for all $i \in [\mathsf{dim}]$, where $\mathbf{v}^f$ is the coefficient matrix of f.

Testing phase:

1: V and P execute the testing procedure Protocol 9 with inputs (vp) and (pp, f), respectively. V rejects if the verifier's checks in Protocol 9 fail.
Otherwise, let $\bar{\mathbf{v}} \in [-\mathsf{dim}\cdot q_0 \cdot 2^{B_{\mathsf{v}}}, \mathsf{dim}\cdot q_0\cdot 2^{B_{\mathsf{v}}}]^{\mathsf{dim}}$ be the vector sent by P at Step 2 of Protocol 9 and proceed to the evaluation phase.

Evaluation phase:

1: V samples a random prime $q \in \mathcal{P}_\lambda$, and sends q to P.

2: If $\mathbf{v}_i^f \notin \mathbb{Z}_{(q)}^{\mathsf{dim}}$ for some $i \in [\mathsf{dim}]$, P aborts. Otherwise, let $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$ be the canonical projection of $\mathbb{Z}_{(q)}$ onto $\mathbb{F}_q$. Let $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{Z}^{\mathsf{dim}}$ be such that $f(\mathbf{q}) = \mathbf{q}_1 \cdot \mathbf{v}^f \cdot \mathbf{q}_2^{\mathsf{T}}$. P sends V the vector
$$\bar{\mathbf{v}}_q = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i})\cdot\phi_q(\mathbf{v}_i^f) \in \mathbb{F}_q^{\mathsf{dim}}.$$
// P is expected to use a strong witness, in which case $\mathbf{v}_i^f \in \mathbb{Z}^{\mathsf{dim}} \subseteq \mathbb{Z}_{(q)}^{\mathsf{dim}}$ for all i. In this case, $\phi_q(\mathbf{v}_i^f)$ is well-defined.

3: V randomly chooses a subset $J \subseteq [\mathsf{dim}]$ of size $|J| = \Theta(\lambda)$. For each $j \in J$:

- V queries the oracles $[[\hat{\mathbf{u}}_i]]$ at position j, for each $i \in [\mathsf{dim}]$. Let $\hat{\mathbf{u}}_{1,j}, \ldots, \hat{\mathbf{u}}_{\mathsf{dim},j}$ be the received values. If $\hat{\mathbf{u}}_{i,j}$ is not an integer from $\mathbb{Z}_{\log(\|M_{\mathcal{C}_\lambda}\|_\infty\cdot\mathsf{dim})+B_{\mathsf{v}}+(2\mathsf{dim}+1)\cdot\log(q_0\cdot\mathsf{dim})}$ for some $i \in [\mathsf{dim}], j \in J$, then V rejects. // P is expected to use a strong witnesses, in which case $\hat{\mathbf{u}}$ contains only integers with the required bit-size.

- V checks whether
$$\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\bar{\mathbf{v}}_q)_j = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i})\cdot\phi_q(\hat{\mathbf{u}}_{i,j}), \qquad \phi_q(y) = \sum_{i\in[\mathsf{dim}]} \bar{\mathbf{v}}_{q,i}\cdot\phi_q(\mathbf{q}_{2,i}).$$

Protocol 9 Zip's testing protocol.

Input: Let $\mathsf{gp}, \mathbb{i}, \mathsf{pp}, \mathsf{vp}, \mathbf{v}^f, f$ be as in Protocol 8. The prover receives pp, f as input, and the verifier receives vp.

1: V sends P uniformly sampled elements $r_1, \ldots, r_{\mathsf{dim}} \in [0, q_0 - 1]$, where q_0 is a prime.

2: P sends V the vector $\bar{\mathbf{v}} = \sum_{i \in [\mathsf{dim}]} r_i \cdot \mathbf{v}_i^f \in \mathbb{Q}^{\mathsf{dim}}$.

3: If for some $j \in [\mathsf{dim}]$, $|\bar{\mathbf{v}}_j| > \mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathbf{v}}}$ or if $\bar{\mathbf{v}}_j$ is not an integer, V rejects. // $\mathbf{v}^f$ is expected to be the coefficient matrix of a strong witness f. In that case this check passes.

4: V randomly chooses a subset $J \subseteq [\mathsf{n}]$ with $|J| = \Theta(\lambda)$. For each $j \in J$:

- V queries the oracles $[[\hat{\mathbf{u}}_i]]$ at position j, for each $i \in [\mathsf{dim}]$. Let $\hat{\mathbf{u}}_{1,j}, \ldots, \hat{\mathbf{u}}_{\mathsf{dim},j}$ be the received values. If $\hat{\mathbf{u}}_{i,j}$ is not an integer from $\mathbb{Z}_{\log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_{\mathbf{v}} + (2\mathsf{dim}+1)\cdot\log(q_0 \cdot \mathsf{dim})}$ some $i \in [\mathsf{dim}], j \in J$, V rejects. // Again, P is expected to use a strong witnesses, in which case $\hat{\mathbf{u}}$ contains only integers with the appropriate bit-size.

- V checks whether $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}})_j = \sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_{i,j}$.

Remark 5.3 (Polynomial verifier)**.** The verifier in Protocol 8 can be made to run in polynomial time by making sure that, at Steps 3 and 4 of Protocol 9, V stops reading a value as soon as the value is seen to have more bits than it is supposed to. For example, at Step 4, when trying to read a value $\hat{\mathbf{u}}_{i,j}$, V reads the first $\log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_{\mathbf{v}} + (2\mathsf{dim} + 1) \cdot \log(q_0 \cdot \mathsf{dim})$ bits of the value. If the value still contains further bits, then V stops and rejects the proof.

5.3 Completeness

Following the notation and conventions of Section 5.2, recall that a strong witness $\mathbb{w}$ for a well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ satisfies the properties $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, and $\mathbb{w} = f$ is such that $f \in \mathbb{Z}_{B_{\mathbf{v}}}^{\mathsf{multilin}}[\mathbf{X}]$, $\mathbf{X} = (X_1, \ldots, X_\mu)$, and $\mathsf{Enc}(\mathbf{v}_i^f) = \hat{\mathbf{u}}_i$ for all $i \in [\mathsf{dim}]$, where $\mathbf{v}^f$ is the coefficient matrix of f.

Theorem 5.4 (Completeness of Protocols 8 and 9 for strong witnesses)**.** *Let $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ be a $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable family of integer linear codes over $\mathbb{Q}$ with error $\varepsilon_{\mathsf{proj}}(\lambda)$. Let $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ be such that $\mathbb{w}$ is a strong witness for $(\mathbb{i}, \mathbb{x})$. Then $\mathsf{V}(\mathsf{vp}, \mathbb{i}, \mathbb{x})$ accepts after interacting with the honest prover $\mathsf{P}(\mathsf{pp}, \mathbb{i}, \mathbb{x}; \mathbb{w})$ in Protocol 8 except with probability $\varepsilon_{\mathsf{comp}}(\lambda) = \varepsilon_{\mathsf{proj}}(\lambda)$.*

Proof. We follow the notation from Protocols 8 and 9, and assume the protocols are executed honestly. Since $\mathbb{w}$ is a strong witness, for any choice of $r_1, \ldots, r_{\mathsf{dim}} \in [0, q_0 - 1]$ and all $j \in [\mathsf{dim}]$, $\bar{\mathbf{v}}_j$ is an integer and $|\bar{\mathbf{v}}_j| < \mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathbf{v}}}$. So V accepts Step 3 of Protocol 9 with probability 1. Next, by linearity of the encoding map $\mathsf{Enc}_{\mathcal{C}_\lambda}$ and since $\hat{\mathbf{u}}_i = \mathsf{Enc}_{\mathcal{C}}(\mathbf{v}_i)$ for all $i \in [\mathsf{dim}]$ due to $\mathbb{w}$ being a strong witness, we have

$$\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}) = \mathsf{Enc}_{\mathcal{C}_\lambda}\left(\sum_{i \in [\mathsf{dim}]} r_i \cdot \mathbf{v}_i^f\right) = \sum_{i \in [\mathsf{dim}]} r_i \cdot \mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i^f) = \sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i.$$

Hence, for all $J \subseteq [\mathsf{n}]$, the probability that V accepts at the end of Step 4 of Protocol 9 is 1.

We proceed to analyze Protocol 8. Again, by the fact that $\mathbb{w}$ is a strong witness, we know that $\mathbf{v}_i^f \in \mathbb{Z}^{\mathsf{dim}} \subset \mathbb{Z}_{(q)}^{\mathsf{dim}}$ for all $i \in [\mathsf{dim}]$ and $q \in \mathcal{P}_\lambda$. Therefore, P will always proceed to Step 3 of the evaluation phase of Protocol 8. By linearity of $\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}$ and ϕ_q,

$$\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\bar{\mathbf{v}}_q) = \sum_{i \in [\mathsf{dim}]} \mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\mathbf{v}_i^f)) = \sum_{i \in [\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\mathbf{v}_i^f)).$$

If $\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\mathbf{v}_i^f)) = \phi_q(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i^f)) = \phi_q(\hat{\mathbf{u}}_i)$, then the verifier's first check in Step 3 of Protocol 8 passes for any choice of $J \subseteq [\mathsf{n}]$. By Lemma 5.1 this happens if the prime q belongs to $\mathcal{P}_{\lambda\mathsf{good}}$. Hence, nu definition of projectable family of codes, V's first check at Step 3 of Protocol 8 will pass with probability $1 - \varepsilon_{\mathsf{proj}}(\lambda)$. Finally, since $y = \mathbf{q}_1 \cdot \mathbf{v}^f \cdot \mathbf{q}_2^\mathsf{T}$. we have

$$\phi_q(y) = \phi_q(\mathbf{q}_1) \cdot \phi_q(\mathbf{v}^f) \cdot \phi_q(\mathbf{q}_2)^\mathsf{T} = \bar{\mathbf{v}}_q \cdot \phi_q(\mathbf{q}_2)^\mathsf{T} = \sum_{i \in [\mathsf{dim}]} \bar{\mathbf{v}}_{q,i} \cdot \phi_q(\mathbf{q}_{2,i}),$$

finishing the proof. $\square$

5.4 Knowledge soundness

Theorem 5.5 (Knowledge soundness of Protocols 8 and 9)**.** *Protocol 8 is an IOP for $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ with knowledge soundness error*

$$\begin{aligned} &\varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})) \\ \leq &\frac{\mathsf{n}+1}{q_0} + 2 \cdot (1-\delta)^{|J|} + \left(1 - \frac{2\mathsf{dist}_0}{3}\right)^{|J|} + \frac{2 \cdot (\mathsf{dim}-1)}{\varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) \cdot q_0} + \theta + \varepsilon_{\mathsf{proj}}(\lambda), \end{aligned} \tag{13}$$

where $\varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \Pr[\langle \mathsf{P}^*(\mathsf{gp}, \mathbb{i}, \mathbb{x}), \mathsf{V}(\mathsf{gp}, \mathbb{i}, \mathbb{x})\rangle = 1]$, $\varepsilon_{\mathsf{proj}}(\lambda)$ is the error of the projectable code $\mathcal{C}_\lambda$, and

$$\theta = \frac{2 \cdot \mathsf{dim}^2 \cdot (B_\mathbf{v} + \log(\mathsf{dim}) + 2 \cdot \log(q_0)) + B_\mathbf{v} + \mu \cdot B_{\mathsf{pt}} + 3\mu + 2}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

Next, we explain how one can instantiate Protocol 8 in a way that makes $\varepsilon_{\mathsf{ks}}$ as small as practically needed.

Remark 5.6 (Obtaining a small knowledge soundness error)**.** To make sure $\varepsilon_{\mathsf{ks}}$ is small enough for practical purposes, when $\varepsilon_{\mathsf{P}^*}$ is not negligible, it suffices to:

- Take $q_0 = O(2^\lambda)$, $\mathsf{n} = \mathsf{poly}(\lambda)$,$\mathsf{dim} = \mathsf{poly}(\lambda)$, $|J|$ such that $(1 - 2\mathsf{dist}_0/3)^{|J|} = \mathsf{negl}(\lambda)$ and $(1-\delta)^{|J|} = \mathsf{negl}(\lambda)$. This ensures that the first four terms in the right-hand side of (13) are negligible in λ, whenever $\varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})^{-1}$ is not negligible in λ.
- Take $\mathcal{P}_\lambda$ such that $|\mathcal{P}_\lambda| = O(2^\lambda)$, and take $B_\mathbf{v}, B_{\mathsf{pt}}, \mu$ to be $\mathsf{poly}(\lambda)$, ensuring that θ is negligible.

- Regarding the last term $\varepsilon_{\mathsf{proj}}(\lambda)$, ideally, one would use a family of projectable codes such that $\varepsilon_{\mathsf{proj}}(\lambda) = \mathsf{negl}(\lambda)$ make the last terms as small as possible. In our instantiation (Section 6) we do not achieve an asymptotic negligible such error, but we achieve concrete projection errors that are sufficiently small for practical scenarios. Namely, in Remark 6.4 we explain how to attain $\varepsilon_\lambda \approx 2^{-100}$ under standard parameter choices. Referring to [GLS+23, BFK+24], this suffices for practical purposes.

The rest of this section is devoted to proving Theorem 5.5. We begin by stating a series of auxiliary lemmas. Then, we will describe an extractor and use the lemmas to prove the extractor is PPT and satisfies the probability bound claimed in Theorem 5.5.

5.4.1 Random linear combinations of large rational numbers

We start by stating two key auxiliary lemmas regarding the behavior of random linear combinations of rational numbers when the combination coefficients are integers. Overall, the two results amount to say that, if such a random linear combination results in a small integer with high probability, then the rational numbers have bit-size not much larger than the size of the combination coefficients.

Lemma 5.7. *Let $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Q}^n$ be a vector of n rational numbers, not all of them zero. Let $N > 0$ be a positive rational number and let $M > 1$ be a positive integer. Then*

$$P = \Pr\left[|\sum_{i\in[n]} r_i \cdot v_i| < N \;\middle|\; r_i \leftarrow [0, M-1] \textit{ for all } i \in [n] \right] \leq \min\left\{1, \frac{N}{\|\mathbf{v}\|_\infty \cdot M}\right\}.$$

Proof. Assume without loss of generality that $|v_n| = \|\mathbf{v}\|_\infty = \max_{i\in[n]}\{|v_i|\}$. We have $v_n \neq 0$, because not all the v_i's are zero. Let $\mathbf{r} = (r_1, \ldots, r_{n-1}) \in [0, M-1]^{n-1}$ be $(n-1)$ fixed integers from $[0, M-1]$. Consider first the probability

$$P_{\mathbf{r}} = \Pr\left[|r_n \cdot v_n + \sum_{i\in[n-1]} r_i \cdot v_i| < N \;\middle|\; r_n \leftarrow [0, M-1] \right].$$

Denote $\mathbf{r} \cdot \mathbf{v} = \sum_{i\in[n-1]} r_i \cdot v_i$. Assume first that $\mathbf{r} \cdot \mathbf{v} \geq 0$ and $v_n < 0$. Write $v_n = -v'_n$ with $v'_n > 0$. Let $\mathcal{E}_1$ be the event that $\mathbf{r} \cdot \mathbf{v} \geq r_n \cdot v'_n$.

If $\mathcal{E}_1$ holds and, simultaneously $|\mathbf{r} \cdot \mathbf{v} - r_n \cdot v'_n| < N$, then we have $|\mathbf{r} \cdot \mathbf{v} - r_n \cdot v'_n| = \mathbf{r} \cdot \mathbf{v} - r_n \cdot v'_n < N$, and so, since $v'_n > 0$,

$$\frac{\mathbf{r} \cdot \mathbf{v}}{v'_n} - \frac{N}{v'_n} < r_n \leq \frac{\mathbf{r} \cdot \mathbf{v}}{v'_n}.$$

Hence, r_n, which is an integer, belongs to a rational interval of length N/v'_n. In particular, there are at most N/v'_n values for r_n such that $\mathcal{E}_1$ holds. Hence

$$\Pr\left[|r_n \cdot v_n + \sum_{i\in[n-1]} r_i \cdot v_i| < N \;\middle|\; \begin{array}{l} r_n \leftarrow [0, M-1], \\ \mathcal{E}_1 \text{ holds} \end{array} \right] \leq \frac{N}{|v_n| \cdot M}.$$

Now assume $\mathcal{E}_1$ does not hold. Then $|\mathbf{r} \cdot \mathbf{v} - r_n \cdot v'_n| = r_n \cdot v'_n - \mathbf{r} \cdot \mathbf{v} < N$, and so

$$\frac{\mathbf{r} \cdot \mathbf{v}}{v'_n} < r_n < \frac{N}{v'_n} + \frac{\mathbf{r} \cdot \mathbf{v}}{v'_n}.$$

Again, the same reasoning gives

$$\Pr \left[|r_n \cdot v_n + \sum_{i \in [n-1]} r_i \cdot v_i| < N \;\middle|\; \begin{array}{l} r_n \leftarrow [0, M-1], \\ \neg\mathcal{E}_1 \text{ holds} \end{array} \right] \leq \frac{N}{|v_n| \cdot M}.$$

Since $\mathcal{E}_1$ and $\mathcal{E}_2$ are complementary, the law of total probability now yields $P_{\mathbf{r}} \leq N/(|v_n| \cdot M)$.

Now assume $\mathbf{r} \cdot \mathbf{v} \geq 0$ and $v_n > 0$. In that case $|\mathbf{r} \cdot \mathbf{v} + r_n v_n| \geq |\mathbf{r} \cdot \mathbf{v} - r_n v_n|$, and then

$$P_{\mathbf{r}} \leq \Pr \left[|r_n \cdot (-v_n) + \sum_{i \in [n-1]} r_i \cdot v_i| < N \;\middle|\; r_n \leftarrow [0, M-1] \right].$$

We know by our previous analysis that the right-hand side of the inequality above is at most $N/(|v_n| \cdot M)$ as needed. The remaining cases, i.e. $\mathbf{r} \cdot \mathbf{v} \leq 0, v_n > 0$, and $\mathbf{r} \cdot \mathbf{v} \leq 0, v_n < 0$ all reduce to the previous two cases since the absolute value is invariant under multiplication by -1.

Now, using again the law of total probability,

$$P = \sum_{\mathbf{r} \in [0, M-1]^{n-1}} P_{\mathbf{r}} \cdot \Pr[\mathbf{r}] \leq \frac{N}{|v_n| \cdot M} \cdot \left(\sum_{\mathbf{r} \in [0, M-1]^{n-1}} \Pr[\mathbf{r}] \right) = \frac{N}{|v_n| \cdot M},$$

where P is the probability from the statement of the lemma that we want to bound, and $\Pr[\mathbf{r}]$ denotes the probability that sampling a tuple from $[0, M-1]^{n-1}$ uniformly at random results in the tuple $\mathbf{r}$. □

Recall that, by gcd and lcm we denote the greatest common divisor and the lowest common multiple of a tuple of integers, respectively.

Lemma 5.8. *Let $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Q}^n$ be a vector of $n \geq 1$ rational numbers, not all of them zero. For each $i \in [n]$, write $v_i = a_i/b_i$ with $a_i, b_i \in \mathbb{Z}$ and* $\gcd(a_i, b_i) = 1$*. Let $M \geq 1$ be a positive integer, and assume there exists $i \in [n]$ such that $|b_i| > M$ and $v_i \neq 0$. Then the probability of uniformly sampling n integers $r_1, \ldots, r_n$ in the interval $[0, M-1]$ such that $\sum_{i \in [n]} r_i v_i$ is an integer is at most $1/M$. More formally,*

$$P = \Pr \left[\left(\sum_{i \in [n]} r_i \cdot v_i \right) \in \mathbb{Z} \;\middle|\; r_i \leftarrow [0, M-1] \textit{ for all } i \in [n] \right] \leq \frac{1}{M}.$$

Proof. Assume without loss of generality that $b_n > M$ and $v_n \neq 0$. Let $\mathbf{r} = (r_1, \ldots, r_{n-1}) \in [0, M-1]^{n-1}$ be $(n-1)$ fixed integers from $[0, M-1]$. Consider first the probability

$$P_{\mathbf{r}} = \Pr \left[\left(r_n \cdot v_n + \sum_{i \in [n-1]} r_i \cdot v_i \right) \in \mathbb{Z} \;\middle|\; r_n \leftarrow [0, M-1] \right].$$

Denote $\mathbf{r} \cdot \mathbf{v} = \sum_{i \in [n-1]} r_i \cdot v_i$. Observe that if $\mathbf{r} \cdot \mathbf{v} = 0$, then $\sum_{i \in [n]} r_i \cdot v_i = r_n \cdot v_n$. But $r_n \cdot v_n$ is an integer if and only if b_n divides r_n. However we assumed $r_n \in [0, M-1]$, while $b_n > M$. Hence, in this case, $P_{\mathbf{r}} = 0$.

Assume from now on that $\mathbf{r} \cdot \mathbf{v} \neq 0$, and write $\mathbf{r} \cdot \mathbf{v} = a/b$ with (a, b) in lowest form, i.e. $a, b \in \mathbb{Z}$, $b \geq 1$, $\gcd(a, b) = 1$. We have $a \neq 0$. Let $L = \operatorname{lcm}(b, b_n)$. Note that $L \neq 0$ because $b, b_n > 0$. Let $r_n \in [0, M-1]$, and assume $r_n \cdot u_n + \mathbf{r} \cdot \mathbf{v} \in \mathbb{Z}$. Multiplying and dividing by L (which, as we argued, is not zero) we obtain that

$$r_n \cdot u_n + \mathbf{r} \cdot \mathbf{v} \in \mathbb{Z} \iff r_n \cdot a_n \cdot \frac{L}{b_n} + a \cdot \frac{L}{b} \equiv 0 \pmod{L},$$

where, note L/b and L/b_n are integers. Denote $A = a \cdot L/b$ and $A_n = a_n \cdot L/b_n$. We claim that the congruence

$$r_n \cdot A_n + A \equiv 0 \pmod{L} \tag{14}$$

has at most one solution on r_n satisfying $r_n \in [0, M-1]$. Indeed, let $D = \gcd(A_n, L)$. Then (14) has a solution $r_n \in \mathbb{Z}$ if and only if D divides A, and in that case the set of integer solutions is

$$S = \left\{ r_n^0 + \mu \cdot \frac{L}{D} \mid \mu \in \mathbb{Z} \right\}$$

where r_n^0 is any one solution. Denote $r_n^0 + \mu \cdot L/D$ by $S(\mu)$. Later we will prove that $L/D = b_n$. In that case, we argue that S contains at most one element from the interval $[0, M-1]$. Indeed, assume S contains one element from $[0, M-1]$, say $S(\mu_0)$. Let $k \in \mathbb{Z}$ be a nonzero integer and assume that $S(\mu_0 + k) \in [0, M-1]$. Then, if k is positive,

$$M > S(\mu_0 + k) = r_n^0 + (\mu_0 + k) \cdot b_n = r_n^0 + \mu_0 \cdot b_n + k \cdot b_n = S(\mu_0) + k \cdot b_n \geq k \cdot b_n > M,$$

where we have used that $S(\mu_0 + k) < M$ because $S(\mu_0 + k) \in [0, M-1]$, that $S(\mu_0) \geq 0$ because $S(\mu_0) \in [0, M-1]$, and that $k \cdot b_n > M$ because we assumed that $b_n > M$, and that $k \geq 1$. The above inequalities constitute a contradiction. Similarly, if k is negative, writing $k = -k'$ for $k' \geq 1$,

$$0 < S(\mu_0 - k') = r_n^0 + (\mu_0 - k') \cdot b_n = r_n^0 + \mu_0 \cdot b_n - k' \cdot b_n = S(\mu_0) - k' \cdot b_n < M - k' \cdot b_n.$$

This yields that $k' \cdot b_n < M$, but then $M < k' \cdot b_n < M$, again a contradiction. Hence, there cannot be more than one element in S that belongs to $[0, M-1]$.

We now show that $L/D = b_n$. Indeed, write $L = \operatorname{lcm}(b, b_n) = b \cdot b_n/d$, with $d = \gcd(b, b_n)$. Then

$$\begin{aligned} D = \gcd(A_n, L) &= \gcd\left(a_n \cdot \frac{L}{b_n}, L\right) = \gcd\left(a_n \cdot \frac{b \cdot b_n}{d \cdot b_n}, \frac{b \cdot b_n}{d}\right) \\ &= \frac{b}{d} \cdot \gcd\left(a_n \cdot \frac{b_n}{b_n}, b_n\right) = \frac{b}{d} \cdot \gcd(a_n, b_n) = \frac{b}{d}, \end{aligned}$$

and so

$$\frac{L}{D} = \frac{b \cdot b_n}{d \cdot \frac{b}{d}} = b_n.$$

Hence, we have shown that the congruence $r_n \cdot A_n + A \equiv 0 \pmod{L}$ has at most one solution with $r_n \in [0, M-1]$ and with A_n and A fixed constant. This implies that $P_{\mathbf{r}} \leq 1/M$. Then, using the law of total probability as in Lemma 5.7,

$$P = \sum_{\mathbf{r} \in [0,M-1]^{n-1}} P_{\mathbf{r}} \cdot \Pr[\mathbf{r}] \leq \frac{1}{M},$$

where P is the probability from the statement of the lemma that we want to bound, and $\Pr[\mathbf{r}]$ denotes the probability that when uniformly sampling $n-1$ elements in $[0, M-1]$, one samples the vector $\mathbf{r}$. $\square$

5.4.2 Auxiliary lemmas

Next we state and prove a series of lemmas regarding the probability of several kinds of events occurring during the execution of Zip.

Definition 5.3 (Correlated codewords for an index-instance pair)**.** *Let* $\mathsf{gp} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0)$, $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$ *and* $\mathbb{x} = (\mathbf{q}, y)$ *be well-formed for* $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$*. Write* $\hat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$*. Assume each* $\hat{\mathbf{u}}_i$ *is* δ*-close to* $\mathcal{C}_\lambda$*. Then, since* δ *is within the unique decoding radius (cf. Section 3.3), there exist unique vectors* $\mathbf{v}_i \in \mathbb{Q}^{\mathsf{dim}}$ *such that* $\Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i), \hat{\mathbf{u}}_i) \leq \delta$ *for all* $i \in [\mathsf{dim}]$*. In this event we define*

$$\mathsf{Words}(\hat{\mathbf{u}}) = \mathbf{v}.$$

The value $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = \max_{i,j \in [\mathsf{dim}] \times [\mathsf{n}]} \{\mathbf{v}_{i,j}\}$ *will play an important role in the security analysis of Protocol 8.*

If $\hat{\mathbf{u}}_i$ *is not* δ*-close to* $\mathcal{C}_\lambda$ *for some* $i \in [\mathsf{dim}]$*, then we define* $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty$ *to be an empty set, and we convene that* $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = 1$*.*

Let P^* be a malicious prover for Protocol 8. Let $\mathsf{gp} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0)$, $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$, and $\mathbb{x} = (\mathbf{q}, y)$ be well formed, so that $\hat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$, $\mathbf{q} \in [-2^{B_{\mathsf{pt}}}, 2^{B_{\mathsf{pt}}}]^\mu$, $y \in \mathbb{Z}$, $|y| < 2^{\mu + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}}}$, and $[[\hat{\mathbf{u}}]] = \{[[\hat{\mathbf{u}}_i]]\}_{i \in [\mathsf{dim}]}$. Let $(\mathsf{pp}, \mathsf{vp}) \leftarrow \mathsf{Indexer}(\mathsf{gp}, \mathbb{i})$, so that $\mathsf{pp} = (\mathsf{gp}, \hat{\mathbf{u}})$ and $\mathsf{vp} = (\mathsf{gp}, [[\hat{\mathbf{u}}]])$.

We define $\mathcal{E}_{\mathsf{test}}$ to be the event that $\mathsf{V}(\mathsf{vp}, \mathbb{x})$ accepts at the end of the testing phase of Protocol 8 during the execution of $\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{i}, \mathbb{x}) \rangle$.

Lemma 5.9 (The words $\hat{\mathbf{u}}_i$ should have δ-correlated agreement)**.** *Assume that the words* $(\hat{\mathbf{u}}_i)_{i \in [\mathsf{dim}]}$ *do not have* δ*-correlated agreement in* $\mathcal{C}_\lambda$*. Then*

$$\Pr[\mathcal{E}_{\mathsf{test}}] \leq \frac{\mathsf{n}}{q_0} + (1-\delta)^{|J|}. \tag{15}$$

Proof. By Lemma 3.2, $\mathcal{C}_\lambda$ has $(\delta, \alpha, [0, q_0 - 1])$-correlated agreement with $\delta < \mathsf{dist}/3$ and $\alpha = \mathsf{n}/q_0$. Hence,

$$\Pr\left[\sum_{i \in [\mathsf{dim}]} \Delta(r_i \cdot \hat{\mathbf{u}}_i, \mathcal{C}_\lambda) \leq \delta \;\middle|\; r_1, \ldots, r_{\mathsf{dim}} \leftarrow [0, q_0 - 1]\right] < \alpha = \frac{\mathsf{n}}{q_0}.$$

Assume V samples elements $r_1, \ldots, r_{\mathsf{dim}}$ such that $\Delta(\sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i, \mathcal{C}_\lambda) > \delta$. Then $\sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i$ agrees with a codeword on less than $(1-\delta) \cdot \mathsf{n}$ positions. Since at the end of the testing phase, V checks whether $\sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i$ agrees with the codeword $\mathsf{Enc}(\bar{\mathbf{v}})$ on $|J|$ random positions, we have that V accepts with probability less than $(1-\delta)^{|J|}$, and so Eq. (15) follows. $\square$

From now on until the end of this subsection we assume the words $(\hat{\mathbf{u}}_i)_{i\in[\mathsf{dim}]}$ have δ-correlated agreement in $\mathcal{C}_\lambda$. Let $E \subseteq [\mathsf{n}]$ be a maximal subset of $[\mathsf{n}]$ (with respect to inclusion) such that each $\hat{\mathbf{u}}_i$ agrees with a codeword $\hat{\mathbf{v}}_i$ from $\mathcal{C}_\lambda$ on the positions E, for all $i \in [\mathsf{dim}]$. Then, since we assumed the words $(\hat{\mathbf{u}}_i)_{i\in[\mathsf{dim}]}$ have δ-correlated agreement and δ is within the unique decoding radius, we have $|E| \geq (1-\delta)\cdot \mathsf{n}$. Let $\mathbf{v}_i \in \mathbb{Q}^{\mathsf{dim}}$ be such that $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i) = \hat{\mathbf{v}}_i$ for all $i \in [\mathsf{dim}]$, and denote the rows of $\mathbf{v}$ and $\hat{\mathbf{v}}$ as $\mathbf{v} = \{\mathbf{v}_i\}_{i\in[\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim}\times\mathsf{dim}}$, $\hat{\mathbf{v}} = \{\hat{\mathbf{v}}_i\}_{i\in[\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim}\times\mathsf{n}}$. Note that, by unicity, $\mathsf{Words}(\hat{\mathbf{u}}) = \mathbf{v}$.

Lemma 5.10 ($\bar{\mathbf{v}}$ should indeed be $\sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i$)**.** *Let $\mathcal{E}_0$ be the event that $\mathsf{P}^*(\mathsf{pp}, \mathbb{x})$ sends $\bar{\mathbf{v}} = \sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i$ at Step 2 of Protocol 9, where $r_1, \ldots, r_{\mathsf{dim}}$ are $\mathsf{V}(\mathsf{vp}, \mathbb{x})$'s challenges sent at Step 1 of Protocol 9. Then*

$$\Pr[\mathcal{E}_{\mathsf{test}} \mid \neg\mathcal{E}_0] < (1-\delta)^{|J|}.$$

Proof. Since δ is within the unique decoding radius, there is a unique codeword whose relative Hamming distance from $\sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i$ is at most δ. Since each $\hat{\mathbf{u}}_i$ agrees with $\hat{\mathbf{v}}_i$ on E and $|E| \geq (1-\delta)\cdot\mathsf{n}$, we have that this codeword is $\bar{\mathbf{v}}$. Now, assume P sends a vector $\bar{\mathbf{v}}'$ with $\bar{\mathbf{v}}' \neq \bar{\mathbf{v}}$ at Step 2 of Protocol 9. Then, by uniqueness, the codeword $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}')$ is such that $\Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}'), \sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i) > \delta$. In particular, $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}')$ and $\sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i$ agree on less than $(1-\delta)\cdot\mathsf{n}$ positions. Hence, the probability that $\mathcal{E}_{\mathsf{test}}$ occurs while $\mathcal{E}_0$ does not is less than $(1-\delta)^{|J|}$. $\square$

We refer to Section 2.3 for an intuitive explanation of what is the role of the following lemma regarding the knowledge soundness of Zip.

Lemma 5.11 ($\hat{\mathbf{u}}$ should contain polynomially sized entries)**.** *Let $\mathcal{E}_1$ be the event that $\mathcal{E}_0$ occurs and V does not reject at Step 3 of Protocol 9. That is, $\mathcal{E}_1$ is the event where $\mathsf{P}^*(\mathsf{pp}, \mathbb{x})$ sends the vector $\bar{\mathbf{v}} = \sum_{i\in[\mathsf{dim}]} r_i \cdot \mathbf{v}_i$ after receiving the challenges $r_1, \ldots, r_{\mathsf{dim}}$, and all entries of $\bar{\mathbf{v}}$ are integers with absolute value at most $\mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathsf{v}}}$. Then*

$$\Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \leq \min\left\{1, \frac{\mathsf{dim}\cdot 2^{B_{\mathsf{v}}}+1}{\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty}\right\},$$

where, recall, $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = \|\mathbf{v}\|_\infty = \max_{i,j\in[\mathsf{dim}]}\{|\mathbf{v}_{i,j}|\}$. If $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = 0$ then we convene that the right-hand side of the inequality above means 1.

Proof. Assume $\mathcal{E}_0$ occurs. If $\mathbf{v}$ consists entirely of zeros, then $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = 0$ and so the inequality holds trivially. Otherwise, assume $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty > 0$, and consider Lemma 5.7 when taking $M = q_0$, $N = \mathsf{dim}\cdot q_0 \cdot 2^{B_{\mathsf{v}}} + 1$, $n = \mathsf{dim}$, and the rational numbers $v_1, \ldots, v_n = \mathbf{v}_{1,j_0}, \ldots, \mathbf{v}_{\mathsf{dim},j_0}$, where $j_0 \in [\mathsf{dim}]$ is a column of $\mathbf{v}$ that contains a largest entry (in absolute value) of $\mathbf{v}$. Note that not all the values $v_1, \ldots, v_n$ are zero, because $\mathsf{Words}(\hat{\mathbf{u}}) > 0$. Then the lemma yields that the probability that $\bar{\mathbf{v}}_{j_0} < \mathsf{dim}\cdot q_0 \cdot 2^{B_{\mathsf{v}}}$ is at most the probability claimed in this lemma. $\square$

Lemma 5.12. *Suppose that* $\mathsf{Words}\{\hat{\mathbf{u}}\}$ *has a nonzero entry* $\mathbf{v}_{i,j}$ *with* $\mathbf{v}_{i,j} = c/d$ *with* $c, d \in \mathbb{Z}$, $c \neq 0$, $\gcd(c, d) = 1$, *and* $d > q_0$. *Then*

$$\Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \leq \frac{1}{q_0}.$$

Proof. For $\mathcal{E}_1$ to occur, it must be the case that $\bar{\mathbf{v}}$ is a vector filled with integers. In particular, it must be the case that $\bar{\mathbf{v}}_j$ is an integer. Now the lemma follows by applying Lemma 5.8 to the rational numbers $\mathbf{v}_{1,j}, \ldots, \mathbf{v}_{\mathsf{dim},j}$, taking $n = \mathsf{dim}$, $M = q_0$. □

Because of Lemma 5.12, from now on we assume that all denominators in the nonzero entries of $\mathsf{Words}(\hat{\mathbf{u}})$ (in lowest form) are positive integers in the range $[1, q_0]$.

Lemma 5.13. *Under the assumption above, if* $\mathcal{E}_1$ *holds, then every entry in* $\mathsf{Words}(\hat{\mathbf{u}}) = \mathbf{v}$ *has bit-size at most*

$$\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0).$$

Proof. Since $\mathcal{E}_1$ holds, V does not reject at Step 3 of Protocol 9. By our assumption above, each entry $\mathbf{v}_{i,j}$ of $\mathbf{v}$ is a rational number of the form $\mathbf{v}_{i,j} = a_{i,j}/b_{i,j}$ with $a_{i,j}, b_{i,j} \in \mathbb{Z}$, $b_{i,j} \geq 1$, $\gcd(a_{i,j}, b_{i,j}) = 1$, and $b_{i,j} \leq q_0$. Then, for all $i, j \in [\mathsf{dim}]$,

$$|a_{i,j}| \leq \|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty \cdot b_{i,j} \leq \|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty \cdot q_0,$$

and so $\mathbf{v}_{i,j}$ has bit-size at most $\log(|a_{i,j}|) + \log(b_{i,j}) \leq \log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0)$. □

Let $\mathcal{E}_{\mathsf{proj}}$ be the event that, at Step 1 of the evaluation phase in Protocol 8, V samples a prime $q \leftarrow \mathcal{P}_\lambda$ that is good with respect to $\mathcal{C}_\lambda$ (see Definition 5.1). By definition, $\Pr[\neg\mathcal{E}_{\mathsf{proj}}] \leq \varepsilon_{\mathsf{proj}}(\lambda)$.

Let $\mathcal{E}_{\mathsf{local}}$ be the event that, in Step 1 of the evaluation phase (Protocol 8), V samples a prime $q \in \mathcal{P}_\lambda$ such that $\mathbf{v} \in \mathbb{Z}_{(q)}^{\mathsf{dim}\times\mathsf{dim}}$.

Lemma 5.14. *We have*

$$\Pr[\neg\mathcal{E}_{\mathsf{local}}] \leq \frac{\mathsf{dim}^2 \cdot (\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0))}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

Proof. Assume $\mathcal{E}_1$ occurs. Let q be the prime sampled by V. For $\mathcal{E}_{\mathsf{local}}$ to not hold, there must be $i, j \in [\mathsf{dim}]$ with $\mathbf{v}_{i,j} \in \mathbb{Q} \setminus \mathbb{Z}_{(q)}$. Hence

$$\Pr[\neg\mathcal{E}_{\mathsf{local}}] \leq \sum_{i,j \in [\mathsf{dim}]} \Pr[\mathbf{v}_{i,j} \notin \mathbb{Z}_{(q)} \mid q \leftarrow \mathcal{P}_\lambda].$$

Fix $i, j \in [\mathsf{dim}]$, and let $\mathcal{P}_\lambda(i, j)$ be such that $\mathbf{v}_{i,j} \in \mathbb{Q} \setminus \mathbb{Z}_{(q)}$ for all $q \in \mathcal{P}_\lambda(i, j)$. Since, by Proposition 4.6, the families of subrings and homomorphisms $\{\mathbb{Z}_{(q)}, \phi_q \mid q \in \mathcal{P}_\lambda\}$ are λ-expanding (recall we assumed that all primes in $\mathcal{P}_\lambda$ have bit-size at least λ), we have by Item 2 of Definition 4.7 that the bit-size of $\mathbf{v}_{i,j}$ is at least $\lambda \cdot |\mathcal{P}_\lambda(i, j)|$. Hence, $\lambda \cdot |\mathcal{P}_\lambda(i, j)| \leq L$, where L is the largest bit-size of an entry in $\mathbf{v}$. Then

$$\Pr[\mathbf{v}_{i,j} \notin \mathbb{Z}_{(q)} \mid q \leftarrow \mathcal{P}_\lambda] = \frac{|\mathcal{P}_\lambda(i, j)|}{|\mathcal{P}_\lambda|} \leq \frac{L}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

Finally, Lemma 5.13 yields that $L \leq \log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0)$, and the lemma follows. □

Let $\bar{\mathbf{v}}' = \sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathbf{v}_i$. Recall that $0 < \mathsf{dist}_0 < 1$ is a parameter such that the family of linear codes $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ is $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable.

Lemma 5.15 ($\bar{\mathbf{v}}_q$ should indeed be $\sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\mathbf{v}_i)$)**.** *Let $\mathcal{E}_2$ be the event that $\mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}} \wedge \mathcal{E}_1$ holds, and the vector $\bar{\mathbf{v}}_q$ sent by P^* at Step 2 of the evaluation phase (Protocol 8) satisfies $\bar{\mathbf{v}}_q = \phi_q(\bar{\mathbf{v}}')$. Then*

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}} \wedge \mathcal{E}_1] \leq \left(1 - \frac{2\mathsf{dist}_0}{3}\right)^{|J|}.$$

Proof. Assume $\mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}} \wedge \mathcal{E}_1$ occurs but $\mathcal{E}_2$ does not hold. Then $\bar{\mathbf{v}}_q \neq \phi_q(\bar{\mathbf{v}}')$, and so, we have $\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\bar{\mathbf{v}}_q) \neq \mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\bar{\mathbf{v}}'))$ (for any encoding map for $\mathcal{C}_{\lambda q}$, which since we assume q is good with respect to $\mathcal{C}_\lambda$, in this case consists of multiplication by the matrix $\phi_q(M_{\mathcal{C}_\lambda})$). Let $A_q \subseteq [\mathsf{n}]$ be the set of positions where $\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\bar{\mathbf{v}}_q)$ and $\mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\bar{\mathbf{v}}'))$ agree. Recall that $\hat{\mathbf{v}}_i$ denotes the vector $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i)$, and that $\hat{\mathbf{v}}_i$ agrees with $\hat{\mathbf{u}}_i$ on all positions of the set $E \subseteq [\mathsf{n}]$. Note we have $|A_q| < \mathsf{n} - \mathsf{dist}_q \cdot \mathsf{n}$, where dist_q is the relative distance of $\mathcal{C}_{\lambda q}$.

Before we continue, we argue that $\mathbf{q}_{1,i}, \mathbf{q}_{2,i}, \hat{\mathbf{v}}_{i,j}, \hat{\mathbf{u}}_{i,j} \in \mathbb{Z}_{(q)}$ for all $i \in [\mathsf{dim}]$ and $j \in E$. Indeed, $\mathbf{q}_{1,i}, \mathbf{q}_{2,i} \in \mathbb{Z}_{(q)}$ because $\mathbf{q}_1, \mathbf{q}_2$ are vectors filled with integer entries, by Remark 5.2. Further, since $\mathcal{E}_{\mathsf{local}}$ is assumed to hold, $\mathbf{v}_{i,j} \in \mathbb{Z}_{(q)}$. Then, since $\hat{\mathbf{v}}_{i,j}$ is the j-th component of $\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i)$ and the generator matrix of $\mathcal{C}_\lambda$ is integer, we have that $\hat{\mathbf{v}}_{i,j} \in \mathbb{Z}_{(q)}$. Finally, if $j \in E$ then $\hat{\mathbf{u}}_{i,j} = \hat{\mathbf{v}}_{i,j}$ and so $\hat{\mathbf{u}}_{i,j} \in \mathbb{Z}_{(q)}$ as well.

Then, using that these elements belong to $\mathbb{Z}_{(q)}$ and thus their ϕ_q-image is well-defined, we have for all $j \in E \setminus A_q$:

$$\sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\hat{\mathbf{u}}_{i,j}) = \sum_{i\in[\mathsf{dim}]} \phi_q(\mathbf{q}_{1,i}) \cdot \phi_q(\hat{\mathbf{v}}_{i,j}) = \qquad \text{Since } \hat{\mathbf{u}}_i \text{ agrees with } \hat{\mathbf{v}}_i \text{ on } E, \forall i$$

$$\phi_q\left(\sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \hat{\mathbf{v}}_{i,j}\right) = \phi_q\left(\sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot (\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i))_j\right) = \qquad \text{Since taking the } j\text{-th coordinate and } \phi \text{ are homomorphisms, and by definition of } \hat{\mathbf{v}}_i$$

$$\phi_q\left(\sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i)\right)_j = \phi_q\left(\mathsf{Enc}_{\mathcal{C}_\lambda}\left(\sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathbf{v}_i\right)\right)_j \qquad \text{By linearity of the code}$$

$$= \phi_q(\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}'))_j \qquad \text{By definition of } \bar{\mathbf{v}}'$$

$$= \mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\phi_q(\bar{\mathbf{v}}'))_j \qquad \text{By Lemma 5.1, because } \mathcal{E}_{\mathsf{proj}} \text{ holds}$$

$$\neq \mathsf{Enc}_{\mathcal{C}_{\lambda_q}}(\bar{\mathbf{v}}_q)_j \qquad \text{Because } j \notin A_q.$$

Hence, if $(\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}} \wedge \mathcal{E}_1$ holds, V rejects if it samples $j \in E \setminus A_q$. For each sampled $j \in [\mathsf{n}]$, we have that $j \in E \setminus A_q$ with probability

$$\frac{|E \setminus A_q|}{\mathsf{n}} \geq \frac{|E| - |A_q|}{\mathsf{n}} > \frac{(1-\delta)\cdot\mathsf{n} - (\mathsf{n} - \mathsf{dist}_q \cdot \mathsf{n})}{\mathsf{n}} > \frac{3\mathsf{dist}_0 - \mathsf{dist}}{3} \geq \frac{2\mathsf{dist}_0}{3},$$

where in the last two inequalities we have used that $\mathsf{dist}_q \geq \mathsf{dist}_0$ (because q is good with respect to $\mathcal{C}_\lambda$ since $\mathcal{E}_{\mathsf{proj}}$ holds); $\delta < \mathsf{dist}/3$; and $\mathsf{dist} \geq \mathsf{dist}_q \geq \mathsf{dist}_0$. Hence, if $\mathcal{E}_2$ does not occur and $\mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}} \wedge \mathcal{E}_1$ does, then V accepts with probability at most $(1 - (2\mathsf{dist}_0/3))^{|J|}$. $\square$

Lemma 5.16 (y should be the correct evaluation of f at $\mathbf{q}$)**.** *Let $\mathcal{E}_3$ be the event that $\mathcal{E}_2$ occurs and $y = \mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^\top$. Let $\mathcal{E}_{\mathsf{accept}}$ be the event that the verifier accepts at the end of Protocol 8. Then*

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \leq \frac{\mathsf{dim}^2 \cdot (\log(\|\mathsf{Words}(\hat{\mathbf{u}}\|_\infty)) + 2 \cdot \log(q_0)) + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}} + 3\mu + 2}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

Proof. Assume $(\neg\mathcal{E}_3) \wedge \mathcal{E}_2$ holds. Then, if $\mathcal{E}_{\mathsf{accept}}$ holds we have

$$\begin{aligned}
\phi_q(y) &= \sum_{j\in[\mathsf{dim}]} \bar{\mathbf{v}}_{q,j} \cdot \phi_q(\mathbf{q}_{2,j}) && \text{Because } \mathcal{E}_{\mathsf{accept}} \text{ occurs}\\
&= \sum_{j\in[\mathsf{dim}]} (\phi_q(\bar{\mathbf{v}}'))_j \cdot \phi_q(\mathbf{q}_{2,j}) && \text{Because } \mathcal{E}_2 \text{ occurs}\\
&= \phi_q\left(\sum_{j\in[\mathsf{dim}]} \bar{\mathbf{v}}'_j \cdot \mathbf{q}_{2,j}\right) && \text{Because } \phi_q \text{ is a morphism and } (\phi_q(\mathbf{a}))_j = \phi_q(\mathbf{a}_j) \text{ for any vector } \mathbf{a}\\
&= \phi_q\left(\sum_{j\in[\mathsf{dim}]} \sum_{i\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathbf{v}_{i,j} \cdot \mathbf{q}_{2,j}\right) && \text{By definition of } \bar{\mathbf{v}}'\\
&= \phi_q(\mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^\top)
\end{aligned}$$

Hence,

$$y - \mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^\top \in \ker\phi_q \setminus \{0\}, \tag{16}$$

where we have $y - \mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^\top \neq 0$ because we assumed $\neg\mathcal{E}_3$ holds. Note that $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{Z}_{(q)}$ by Remark 5.2. Because of this, and since $\mathcal{E}_{\mathsf{local}}$ is assumed to hold (and since $\mathbb{Z}_{(q)}$ is a field), all the applications of the map ϕ_q above are well-defined.

Define a polynomial P on variables $(V_{i,j})_{i,j\in[\mathsf{dim}]}$ as follows:

$$P(V_{i,j})_{i,j\in[\mathsf{dim}]} = y - \sum_{i,j\in[\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathbf{q}_{2,j} \cdot V_{i,j}.$$

Note that P is a linear polynomial on dim^2 variables, with integer coefficients since $y \in \mathbb{Z}$ and all components of $\mathbf{q}_1, \mathbf{q}_2$ are integers, by Remark 5.2). The polynomial P has at most $1 + \mathsf{dim}^2$ nonzero coefficients, and the absolute value of its coefficients is at most

$$\max\{|y|, |\mathbf{q}_{1,i} \cdot \mathbf{q}_{2,j}|\}_{i,j\in[\mathsf{dim}]} \leq \max\{|y|, 2^{B_{\mathsf{pt}}\cdot\mu}\} \leq 2^{\mu+B_{\mathbf{v}}+\mu\cdot B_{\mathsf{pt}}},$$

where we used $|y| < 2^{\mu+B_{\mathbf{v}}+\mu\cdot B_{\mathsf{pt}}}$ because $(\mathbb{i}, \mathbb{x})$ is assumed to be well-formed. Note that $P(\mathbf{v}) \in \ker\phi_q \setminus \{0\}$.

We have argued that (16) if $\mathcal{E}_{\mathsf{accept}}$ occurs, conditioned on $(\neg\mathcal{E}_3) \wedge \mathcal{E}_2$ happening. The randomness is taken over the sampling of q in $\mathcal{P}_\lambda$. Hence, there are at least

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \cdot |\mathcal{P}_\lambda|$$

primes $q \in \mathcal{P}_\lambda$ such that (16) holds conditioned on $\neg\mathcal{E}_3 \wedge \mathcal{E}_2$ occurring, i.e. such that $P(\mathbf{v}) \in \ker\phi_q \setminus \{0\}$. Then, since we chose $\{\mathbb{Z}_{(q)} \mid q \in \mathcal{P}_\lambda\}$, $\{\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q \mid q \in \mathcal{P}_\lambda\}$ to be

λ-expanding (Definition 4.7), we have by definition that some entry in $\mathbf{v}$ has bit-size at least

$$\frac{\lambda \cdot \Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \cdot |\mathcal{P}_\lambda| - (\mu + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}} - \log(\mathsf{dim}^2 + 1))}{\mathsf{dim}^2}.$$

On the other hand, by Lemma 5.13, all entries in $\mathsf{Words}(\hat{\mathbf{u}}) = \mathbf{v}$ have bit-size at most $\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2\log(q_0)$. Hence

$$\frac{\lambda \cdot \Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \cdot |\mathcal{P}_\lambda| - (\mu + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}} - \log(\mathsf{dim}^2 + 1))}{\mathsf{dim}^2} \leq \log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0).$$

It follows that

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \leq \frac{\mathsf{dim}^2 \cdot (\log(\|\mathsf{Words}(\hat{\mathbf{u}}\|_\infty)) + 2 \cdot \log(q_0)) + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}} + 3\mu + 2}{\lambda \cdot |\mathcal{P}_\lambda|},$$

where we have used that $\log(\mathsf{dim}^2 + 1) \leq \log(4\mathsf{dim}^2) = 2\log(2\mathsf{dim}) = 2 + \mu$. $\square$

5.4.3 Description and analysis of the extractor

In Protocol 10 we describe a PPT extractor Ext that, for all malicious prover P^* for Protocol 8, and for all well-formed index-instance pair $(\mathbb{i}, \mathbb{x})$ for $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, the probability that $\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathsf{vp}, \mathbb{x})\rangle = 1$ and $(\mathbb{i}, \mathbb{x}; \mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})) \notin \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ is bounded by the right-hand side of (13).

Protocol 10 Extractor Ext for Protocol 8.

Input: Let $\mathsf{gp} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0)$ be well-formed global parameters for $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, and let $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$, $\mathbb{x} = (\mathbf{q}, y)$ with $\hat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$, $\mathbf{q} \in \mathbb{Z}^\mu$, $\|\mathbf{q}\|_\infty < 2^{B_{\mathsf{pt}}}$, $y \in \mathbb{Z}$, $|y| < 2^{\mu + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}}}$. Let $\mathsf{pp} = (\mathsf{gp}, [[\hat{\mathbf{u}}]])$ and $\mathsf{vp} = (\mathsf{gp}, \hat{\mathbf{u}})$. Let P^* be a malicious prover for Protocol 8. Ext receives $(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ as input, and black-box access to P^*.

1: Initialize an empty list S.
2: Pick a nonzero vector $\mathbf{r} = (r_1, \ldots, r_{\mathsf{dim}}) \in [0, q_0 - 1]^{\mathsf{dim}}$ uniformly at random. Give $\mathbf{r}$ to $\mathsf{P}^*(\mathsf{pp}, \mathbb{x})$ as if $\mathbf{r}$ was the first message sent by $\mathsf{V}(\mathsf{vp}, \mathbb{x})$ in the testing phase of Protocol 8.
3: Let $\bar{\mathbf{v}}_{\mathbf{r}}$ be the vector output by $\mathsf{P}^*(\mathsf{pp}, \mathbb{x})$ in the testing phase (Protocol 9) after receiving $\mathbf{r}$.
4: Check whether all entries of $\bar{\mathbf{v}}_{\mathbf{r}}$ are integers with absolute value at most $\mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathbf{v}}}$. If they are not, go back to Step 2.
5: Run Protocol 11 with input $([[\hat{\mathbf{u}}]], \mathbf{r}, \bar{\mathbf{v}}_{\mathbf{r}})$. Let b be the output of Protocol 11.
6: If $b = 0$, go back to Step 2. If $b = 1$, add $\mathbf{r}$ to the list S. If $|S| < \mathsf{dim}$, go back to Step 2. Otherwise, if $|S| = \mathsf{dim}$, proceed to the next step.
7: Check whether the vectors contained in $S = (\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(\mathsf{dim})})$ are linearly independent over $\mathbb{Q}$. If they are not, output $\perp$. Otherwise, if they are, compute and output $\mathbf{r}^{-1} \cdot \bar{\mathbf{v}}$, where $\mathbf{r}$ is the $\mathsf{dim} \times \mathsf{dim}$ matrix whose i-th row is the vector $\mathbf{r}^{(i)}$, and $\bar{\mathbf{v}}$ is the $\mathsf{dim} \times \mathsf{dim}$ matrix whose i-th row is the vector $\bar{\mathbf{v}}_{\mathbf{r}^{(i)}}$, $i \in [\mathsf{dim}]$.

Protocol 11 Testing subprocedure for the extractor Protocol 10.

Input: $([[\hat{\mathbf{u}}]], \mathbf{r}, \bar{\mathbf{v}}_{\mathbf{r}})$, where $\hat{\mathbf{u}}' = \{\hat{\mathbf{u}}'_i\}_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$, $\mathbf{r} = (r_1, \ldots, r_{\mathsf{dim}}) \in [0, q_0 - 1]^{\mathsf{dim}}$, and $\bar{\mathbf{v}}_{\mathbf{r}} \in \mathbb{Z}^{\mathsf{dim}}_{\mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathbf{v}}}}$.

1: Let $\perp$ be a special fresh symbol. For each $i \in [\mathsf{dim}]$, compute a string $\hat{\mathbf{u}}'_i \in (\mathbb{Q} \cup \{\perp\})^{\mathsf{n}}$ as follows. In short, $\hat{\mathbf{u}}'_i$ is the string obtained from $\hat{\mathbf{u}}_i$ by replacing each of its entries with bit-size larger than B_{BoundEnc} with the symbol $\perp$. This can be done efficiently as follows:

- For each $j \in [\mathsf{n}]$, Ext queries the oracle $[[\widehat{\mathbf{u}}_i]]$ at position j and starts reading the received value $\widehat{\mathbf{u}}_{i,j}$.
- Ext either reads $\widehat{\mathbf{u}}_{i,j}$ in its entirety, or stops reading it after having read more bits than
$$\log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_{\mathbf{v}} + (2\mathsf{dim} + 1) \cdot \log(q_0 \cdot \mathsf{dim}).$$
- In the first case, Ext assigns the j-th entry $\hat{\mathbf{u}}'_{i,j}$ of $\hat{\mathbf{u}}'_i$ to be $\hat{\mathbf{u}}_{i,j}$. Otheriwse $\hat{\mathbf{u}}'_{i,j}$ is set as the symbol $\perp$.

2: Compute the vector $\widehat{\mathbf{u}}'_{\mathbf{r}} = \sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}'_i$, where we interpret $r_i \cdot \perp = \perp$ and $\perp + a = a + \perp = \perp$ for any integer a.

3: Count the number k of positions from $[\mathsf{n}]$ where $\widehat{\mathbf{u}}'_{\mathbf{r}}$ and $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}_{\mathbf{r}})$ agree. Every position of $\widehat{\mathbf{u}}'_{\mathbf{r}}$ occupied by the symbol $\perp$ is counted as a position of disagreement with $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}_{\mathbf{r}})$.

4: Output 1 if $(k/\mathsf{n})^{|J|} > \frac{\varepsilon_{\mathsf{test}}}{2}$, output $b = 1$. Otherwise, output $b = 0$. Here $\varepsilon_{\mathsf{test}}$ is the probability that $\mathsf{V}(\mathsf{vp}, \mathbb{x})$ accepts at the end of Protocol 9 when interacting with $\mathsf{P}^*(\mathsf{pp}, \mathbb{x})$.

Denote by $\mathcal{E}_{\mathsf{accept}}$ the event $\langle \mathsf{P}^*(\mathsf{pp}, \mathbb{x}), \mathsf{V}(\mathbf{v}, \mathbb{x}) \rangle = 1$, and let $\varepsilon_{\mathsf{P}^*} = \varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) = \Pr[\mathcal{E}_{\mathsf{accept}}]$. Denote

$$\varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}) = \Pr\left[\mathcal{E}_{\mathsf{accept}} \wedge ((\mathbb{i}, \mathbb{x}; \mathbb{w}) \notin \mathsf{REL}_{\mathsf{gp}, \mathsf{Eval}}) \quad \middle| \; \mathbb{w} \leftarrow \mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})\right]$$

Our goal is to prove that Protocol 10 runs in expected polynomial time on $\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}^{-1}$ and that $\varepsilon_{\mathsf{ks}}$ satisfies the bound (13).

By Lemma 5.9, if the words $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{\mathsf{dim}}$ do not have δ-correlated agreement in $\mathcal{C}_\lambda$, then

$$\varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}) \leq \Pr[\mathcal{E}_{\mathsf{test}}] \leq \frac{\mathsf{n}}{q_0} + (1 - \delta)^{|J|},$$

and then the bound (13) is satisfied. Hence, we assume from now on that these words do have δ-correlated agreement in $\mathcal{C}_\lambda$. Then the matrix $\mathbf{v} = \mathsf{Words}(\hat{\mathbf{u}}) = (\mathbf{v}_i)_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{dim}}$ is well defined (cf. Definition 5.3). Let $E \subseteq [\mathsf{n}]$ be a maximal correlated agreement (cf. Definition 3.1) subset of $[\mathsf{n}]$ for the words $\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{\mathsf{dim}}$, i.e. E is a maximal (with respect to inclusion) subset of $[\mathsf{n}]$ such that the codewords $\hat{\mathbf{v}}_i = \mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i) \in \mathcal{C}_\lambda$ agree with $\hat{\mathbf{u}}_i$ on E, for all $i \in [\mathsf{dim}]$. We have $|E| \geq (1 - \delta) \cdot \mathsf{n}$.

By Lemma 5.12, if some denominator in lowest form of a nonzero entry in $\mathbf{v}$ is larger than q_0, then $\varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}) \leq \Pr[\mathcal{E}_{\mathsf{test}}] \leq 1/q_0$, and then (13) is satisfied. Hence, we also

[7] Note that Ext doesn't know the value $\varepsilon_{\mathsf{test}}$. However, it can estime such value by running and rewinding P^* several times. See Remark 5.26 for further details.

assume from now on that all denominators of the nonzero entries of $\mathbf{v}$ (in lowest form) lie in the interval $[1, q_0]$. Then, by Lemma 5.13, we have that every entry in $\mathbf{v}$ has bit-length at most $\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0)$.

Write $\mathsf{gp} = (\lambda, \mu, \mathcal{C}_\lambda, \mathcal{P}_\lambda, B_\mathbf{v}, B_{\mathsf{pt}}, \delta)$ and $\mathbb{i} = ([[\hat{\mathbf{u}}]], \hat{\mathbf{u}})$, $\mathbb{x} = (\mathbf{q}, y)$ with $\hat{\mathbf{u}} = \{\hat{\mathbf{u}}_i\}_{i \in [\mathsf{dim}]} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$, $\mathbf{q} \in [0, q_0 - 1]^\mu$, $y \in \mathbb{Z}$, $|y| < 2^{\mu + B_{\mathsf{pt}} + \mu \cdot B_\mathbf{v}}$, and $[[\hat{\mathbf{u}}]] = \{[[\hat{\mathbf{u}}_i]]\}_{i \in [\mathsf{dim}]}$. Assume $\varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x}) > 0$, otherwise there is nothing to prove since $\varepsilon_{\mathsf{ks}}(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}) \leq \varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$.

As in Lemma 5.9, let $\mathcal{E}_{\mathsf{test}}$ be the event that all of the checks made by V at Steps 3 and 4 of the testing phase of Protocol 9 are correct, and let $\varepsilon_{\mathsf{test}} = \Pr[\mathcal{E}_{\mathsf{test}}]$. Assume $\varepsilon_{\mathsf{P}^*} \geq 2 \cdot (1 - \delta)^{|J|}$. Otherwise, $\varepsilon_{\mathsf{ks}} \leq \varepsilon_{\mathsf{P}^*}$ is bounded by the right-hand side of Eq. (13), regardless of whether $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \in \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ or not, where $\mathbb{w}$ is the extractor's output.

We will use the lemmas from the previous two sections to prove that this is indeed the case. We set the following notation

$$B_{\mathsf{BoundEnc}} = \log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_\mathbf{v} + (2\mathsf{dim} + 1) \cdot \log(q_0 \cdot \mathsf{dim}).$$

Given $\hat{\mathbf{u}} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{n}}$ we define

$$\mathsf{Big} = \{j \in [\mathsf{n}] \mid \hat{\mathbf{u}}_{i,j} \notin \mathbb{Z}_{B_{\mathsf{BoundEnc}}} \text{ for some } i \in [\mathsf{dim}]\}. \tag{17}$$

Given $\mathbf{r} \in [0, q_0 - 1]^{\mathsf{dim}}$, let $\mathcal{E}_{\mathsf{Big}}$ be the event that V samples $J \subseteq [\mathsf{n}]$ such that $J \cap \mathsf{Big} = \emptyset$. Let $\mathcal{E}_\mathbf{r}$ be the event that V sends the challenge vector $\mathbf{r}$ at Step 1 of Protocol 9. Let T be the set of nonzero vectors from $[0, q_0 - 1]^{\mathsf{dim}}$ such that, if $\mathcal{E}_\mathbf{r}$ holds, then all of the verifier's checks in Steps 3 and 4 of Protocol 9 pass with probability strictly larger than $\varepsilon_{\mathsf{test}}/2$. In other words,

$$T = \left\{ \mathbf{r} \in [0, q_0 - 1]^{\mathsf{dim}} \;\middle|\; \Pr[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_\mathbf{r}] > \varepsilon_{\mathsf{test}}/2 \right\} \setminus \{\mathbf{0}\},$$

where $\mathbf{0}$ denotes the vector consisting entirely of zeros.

As in Lemma 5.10, let $\mathcal{E}_0$ be the event that, given $\mathbf{r} = (r_1, \ldots, r_{\mathsf{dim}})$ at Step 1 of Zip's testing phase (Protocol 9), P^* sends the vector $\sum_{i \in [\mathsf{dim}]} r_i \cdot \mathbf{v}_i$.

Lemma 5.17. *Let $\mathcal{E}_T$ be the event that the verifier samples a vector $\mathbf{r}$ in T at Step 1 of Protocol 9. Then, if $\mathcal{E}_T$ occurs, also $\mathcal{E}_0$ occurs.*

Proof. Due to Lemma 5.10 and our previous assumptions, $\Pr[\mathcal{E}_{\mathsf{test}} \mid \neg\mathcal{E}_0] \leq (1 - \delta)^{|J|} \leq \varepsilon_{\mathsf{P}^*}/2 \leq \varepsilon_{\mathsf{test}}/2$. Hence, if V samples $\mathbf{r} \in T$, then $\mathcal{E}_0$ must necessarily hold, because we have $\Pr[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_\mathbf{r}] > \varepsilon_{\mathsf{test}}/2 \geq \Pr[\mathcal{E}_{\mathsf{test}} \mid \neg\mathcal{E}_0]$, where $\mathbf{r} \in T$ denotes the event that the sampled vector $\mathbf{r}$ belongs to T. $\square$

Let T' be the set of nonzero challenge vectors $\mathbf{r} \in [0, q_0 - 1]^{\mathsf{dim}}$ such that Protocol 11 outputs 1 when given $([[\widehat{\mathbf{u}}]], \mathbf{r}, \bar{\mathbf{v}}_\mathbf{r})$ as input, where $\bar{\mathbf{v}}_\mathbf{r}$ denotes the vector sent by P^* at Step 2 of Protocol 9 after receiving the challenge vector $\mathbf{r}$. In what follows, we let $\widehat{\mathbf{u}}'_\mathbf{r}$ denote the vector computed at Step 2 of Protocol 11 with input $([[\widehat{\mathbf{u}}]], \mathbf{r}, \bar{\mathbf{v}}_\mathbf{r})$.

Lemma 5.18. *We have that $T' = T$.*

Proof. Let $\mathsf{Big} \subseteq [\mathsf{n}]$ be defined as in (17). The event $[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_\mathbf{r}]$ occurs if and only if $\mathcal{E}_{\mathsf{Big}}$ does not occur, and if J is contained in the set of positions where $\sum_{i \in [\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}_i$ agrees with $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}_\mathbf{r})$. In that case, since $J \cap \mathsf{Big} = \emptyset$, such a set J is contained in the set of positions where $\hat{\mathbf{u}}'_{\mathbf{r}}$ (as defined in Protocol 11) does not have the symbol $\perp$, which is precisely the set $[\mathsf{n}] \setminus \mathsf{Big}$. Hence $[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_{\mathbf{r}}]$ occurs if and only if J is contained in the set of positions $S_{\mathbf{r}} \subseteq [\mathsf{n}]$ where $\sum_{i\in[\mathsf{dim}]} r_i \cdot \hat{\mathbf{u}}'_i$ agrees with $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}_{\mathbf{r}})$, where $\hat{\mathbf{u}}'_i$ has the meaning given in Protocol 11. Formally,

$$\Pr[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_{\mathbf{r}}] = \Pr[J \subseteq S_{\mathbf{r}} \mid \mathcal{E}_{\mathbf{r}}] = \left(\frac{|S_{\mathbf{r}}|}{\mathsf{n}}\right)^{|J|}.$$

Now assume $\mathbf{r} \in T'$. Then $(|S_{\mathbf{r}}|/\mathsf{n})^{|J|} > \varepsilon_{\mathsf{test}}/2$, and so, for such $\mathbf{r}$,

$$\Pr[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_{\mathbf{r}}] = \left(\frac{|S_{\mathbf{r}}|}{\mathsf{n}}\right)^{|J|} > \frac{\varepsilon_{\mathsf{test}}}{2}.$$

Since, additionally, $\mathbf{r}$ is nonzero by definition of T', this means that $\mathbf{r} \in T$. Conversely, if $\mathbf{r} \in T$, then

$$\Pr[\mathcal{E}_{\mathsf{test}} \mid \mathcal{E}_{\mathbf{r}}] = \left(\frac{|S_{\mathbf{r}}|}{\mathsf{n}}\right)^{|J|} > \frac{\varepsilon_{\mathsf{test}}}{2},$$

and so $\mathbf{r} \in T'$, where, again, we used the fact that $\mathbf{r}$ is nonzero. We conclude that $T' = T$. ☐

Lemma 5.19. *The following inequality holds:*

$$|T| \geq {q_0}^{\mathsf{dim}} \cdot \frac{\varepsilon_{\mathsf{test}}}{2}.$$

Proof. Recall that T is the set of *nonzero* vectors $\mathbf{r} \in [0, q_0 - 1]^{\mathsf{dim}}$ such that $\mathcal{E}_{\mathsf{test}}$ holds with probability larger than $\varepsilon_{\mathsf{test}}/2$ if $\mathbf{r}$ is the challenge vector sent by V. Let T_0 be the same set, but without the restriction that the set does not contain the zero vector. Clearly $|T_0| \leq |T| \leq |T_0| + 1$. We have

$$\begin{aligned}\varepsilon_{\mathsf{test}} &= \Pr[\mathcal{E}_{\mathsf{test}} \mid \mathbf{r} \in T_0] \cdot \Pr[\mathbf{r} \in T_0] + \Pr[\mathcal{E}_{\mathsf{test}} \mid \mathbf{r} \notin T_0] \cdot \Pr[\mathbf{r} \notin T_0] \\ &\leq \Pr[\mathbf{r} \in T_0] + (\varepsilon_{\mathsf{test}}/2) \cdot (1 - \Pr[\mathbf{r} \in T_0]) = \Pr[\mathbf{r} \in T_0] \cdot (1 - \varepsilon_{\mathsf{test}}/2) + \varepsilon_{\mathsf{test}}/2 \\ &\leq \Pr[\mathbf{r} \in T_0] + \varepsilon_{\mathsf{test}}/2,\end{aligned}$$

and so $|T_0| \geq (\varepsilon_{\mathsf{test}}/2) \cdot {q_0}^{\mathsf{dim}}$, where by $\Pr[\mathbf{r} \in T]$ we denote the probability that the vector $\mathbf{r}$ of challenges sent by V at Step 1 of Protocol 9 belongs to T. It follows that also $|T| \geq (\varepsilon_{\mathsf{test}}/2) \cdot {q_0}^{\mathsf{dim}}$. ☐

We now state and prove an auxiliary lemma relating linear independence over a finite field and over $\mathbb{Q}$.

Lemma 5.20 (Linear independence in $\mathbb{F}_q$ implies linear independence in $\mathbb{Q}$)**.** *Let $n, k \geq 1$ and let $\mathbf{w}_1, \ldots, \mathbf{w}_k$ be k n-dimensional vectors with integer entries. Let q be a prime and let $\phi_q : \mathbb{Z} \to \mathbb{F}_q$ be the natural projection of $\mathbb{Z}$ onto $\mathbb{F}_q$. Assume the vectors $\phi_q(\mathbf{w}_1), \ldots, \phi_q(\mathbf{w}_k)$ are linearly independent over $\mathbb{F}_q$ and are not all zero. Then the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are linearly independent over $\mathbb{Q}$.*

Proof. Assume towards contradiction that the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ are linearly dependent over $\mathbb{Q}$. Let $c_1, \ldots, c_k$ be k rational numbers, not all zero, such that $\sum_{i\in[k]} c_i \cdot \mathbf{w}_i = 0$. By multiplying each c_i by the lowest common multiple of $c_1, \ldots, c_k$, we can assume without loss of generality that c_i is an integer, for all $i \in [k]$. Further, we can assume without loss of generality that there exists $j \in [k]$ such that c_j is not divisible by q, i.e. $\phi_q(c_j) \neq 0$, since otherwise we have $\sum_{i\in[k]}(c_i/q^t) \cdot \mathbf{w}_i = 0$ where $t > 0$ is the largest positive integer such that q^t divides each c_i for all $i \in [k]$. Note that each coefficient c_i/q^t is an integer.

Now, we have $\sum_{i\in[k]} \phi_q(c_i) \cdot \phi_q(\mathbf{w}_i) = 0$. Note that not all the elements $\phi_q(c_1), \ldots, \phi_q(c_k)$ are zero, because we assumed q does not divide c_j. Since we assumed that the vectors $\phi_q(\mathbf{w}_1), \ldots, \phi_q(\mathbf{w}_k)$ are linearly independent over $\mathbb{F}_q$, the only possibility left is that $\phi_q(\mathbf{w}_i) = 0$ for all $i \in [k]$, which contradicts the hypothesis of the lemma. Hence, the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$ had to be linearly independent over $\mathbb{Q}$ to begin with. $\square$

Lemma 5.21. *Except with probability at most*

$$\frac{2 \cdot (\mathsf{dim} - 1)}{\varepsilon_{\mathsf{test}} \cdot q_0} \tag{18}$$

the dim *vectors in* S *at the end of the execution of* $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ *are linearly independent over* $\mathbb{Q}$*. In particular,* $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ *outputs a matrix* $\mathbf{v} \in \mathbb{Q}^{\mathsf{dim}\times\mathsf{dim}}$ *(as opposed to the symbol* $\perp$*) with probability at least* $1 - (2 \cdot (\mathsf{dim} - 1))/(\varepsilon_{\mathsf{test}} \cdot q_0)$*.*

Proof. Let $S = (\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(\mathsf{dim})})$ be the list S at the end of executing $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$. By definition of Ext, all of the vectors $\mathbf{r}^{(i)}$ belong to T'. In particular, all these vectors are nonzero. For each $i \in [\mathsf{dim}]$, let $\mathcal{E}_{\mathsf{l.d.}\mathbb{Q},i}$ be the event that the vectors $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(i)}$ are linearly dependent over $\mathbb{Q}$. As usual, let ϕ_{q_0} be the natural projection of $\mathbb{Z}$ onto $\mathbb{F}_{q_0}$. Let $\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}$ be the event that the vectors $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i)})$ are linearly dependent over $\mathbb{F}_q$. Next, we prove that

$$\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{Q},i}] \leq \Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}].$$

Indeed, assume $\mathcal{E}_{\mathsf{l.d.}\mathbb{Q},i}$ occurs, so that the nonzero vectors $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(i)}$ are linearly dependent over $\mathbb{Q}$. We claim that $\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}$ must hold as well. Indeed, if it does not, then by definition the vectors $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i)})$ are linearly independent over $\mathbb{F}_{q_0}$. But then Lemma 5.20 implies that $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(i)}$ are linearly independent over $\mathbb{Q}$, a contradiction (note that none of the vectors $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i)})$ is the zero vector because $\mathbf{r}^{(j)}$ is non-zero fo all $j \in [\mathsf{dim}]$, and $\phi_{q_0}(\mathbf{r}^{(j)}) = \mathbf{r}^{(j)}$, because all entries of $\mathbf{r}^{(j)}$ belong to $[0, q_0 - 1]$). Hence, $\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{Q},i}] \leq \Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}]$, as needed.

We now prove that

$$\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}] \leq \frac{2 \cdot (i-1)}{\varepsilon_{\mathsf{test}} \cdot q_0 - 2}$$

for all $i \in [\mathsf{dim}]$. We proceed by induction. If $i = 1$, then $\phi_{q_0}(\mathbf{r}^{(1)})$ is linearly dependent over $\mathbb{F}_{q_0}$ if and only if $\phi_{q_0}(\mathbf{r}^{(1)}) = \mathbf{r}^{(1)}$ is the zero vector. However, as already argued, by definition of T, $\mathbf{r}^{(1)}$ is not the zero vector. Hence $\phi_{q_0}(\mathbf{r}^{(1)})$ is linearly dependent over $\mathbb{F}_{q_0}$ with probability 0, as needed.

Now assume that $i \geq 2$. We have

$$\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}] \leq \Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i} \mid \neg\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_q,i-1}] + \Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i-1}].$$

By induction hypothesis, $\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_q,i-1}] \leq 2 \cdot (i-2)/(\varepsilon_{\mathsf{test}} \cdot q_0)$. We proceed to bound $\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i} \mid \neg\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_q,i-1}]$. Assume the vectors $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i-1)})$ are linearly independent over $\mathbb{F}_q$. Then $\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}$ holds if and only if $\phi_{q_0}(\mathbf{r}^{(i)})$ is in the span of $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i-1)})$

over the field $\mathbb{F}_{q_0}$. There are at most ${q_0}^{i-1}$ vectors in such span. Now, $\phi_{q_0}(\mathbf{r}^{(i)}) = \mathbf{r}^{(i)}$ is sampled uniformly at random in $[0, q_0 - 1]^{\mathsf{dim}}$, and it is included in S if and only if it belongs to T'. By Lemmas 5.18 and 5.19, $|\phi_{q_0}(T')| = |T'| = |T| \geq (\varepsilon_{\mathsf{test}}/2) \cdot {q_0}^{\mathsf{dim}}$. Hence, letting $\mathcal{E}_{\mathsf{span},i}$ be the event that $\phi_{q_0}(\mathbf{r}^{(i)})$ is in the span of $\phi_{q_0}(\mathbf{r}^{(1)}), \ldots, \phi_{q_0}(\mathbf{r}^{(i-1)})$ we have

$$\Pr[\mathcal{E}_{\mathsf{span},i} \mid \mathbf{r}^{(i)} \in T'] = \frac{\Pr[\mathcal{E}_{\mathsf{span,i}} \wedge \mathbf{r}^{(i)} \in T']}{\Pr[\mathbf{r}^{(i)} \in T']} \leq \frac{{q_0}^{i-1}}{\left(\frac{\varepsilon_{\mathsf{test}}}{2}\right) \cdot {q_0}^{\mathsf{dim}}} \leq \frac{2}{\varepsilon_{\mathsf{test}} \cdot q_0}.$$

(Note that the probability that $\mathbf{r}^{(i)} \in T'$, i.e. $\Pr[\mathbf{r}^{(i)} \in T']$, is nonzero because $T' = T$ is nonempty due to Lemma 5.19 and our assumption that $\varepsilon_{\mathsf{test}} \neq 0$). Hence, we conclude that

$$\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},i}] \leq \frac{2}{\varepsilon_{\mathsf{test}} \cdot q_0} + \frac{2 \cdot (i-2)}{\varepsilon_{\mathsf{test}} \cdot q_0} \leq \frac{2 \cdot (i-1)}{\varepsilon_{\mathsf{test}} \cdot q_0}.$$

In particular $\Pr[\mathcal{E}_{\mathsf{l.d.}\mathbb{F}_{q_0},\mathsf{dim}}] \leq 2 \cdot (\mathsf{dim} - 1)/(\varepsilon_{\mathsf{test}} \cdot q_0)$, as claimed. ☐

Lemma 5.22. *Suppose that* $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ *outputs a matrix* $\mathbf{v} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{dim}}$. *Then* $\mathbf{v} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{dim}}_{B_{\mathsf{v}} + (2\mathsf{dim}+1) \cdot \log(q_0 \cdot \mathsf{dim})}$.

Proof. By definition, $\mathbf{v} = \mathbf{r}^{-1} \cdot \bar{\mathbf{v}}$, where $\mathbf{r} = (\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(\mathsf{dim})}) \in [0, q_0 - 1]^{\mathsf{dim} \times \mathsf{dim}}$ with $\mathbf{r}^{(i)} \in T'$ for all $i \in [\mathsf{dim}]$, and $\bar{\mathbf{v}}_{\mathbf{r}} = (\bar{\mathbf{v}}_{\mathbf{r}^{(1)}}, \ldots, \bar{\mathbf{v}}_{\mathbf{r}^{(\mathsf{dim})}})$. We claim that all entries in $\bar{\mathbf{v}}$ are nonnegative integers of size at most $q_0 \cdot \mathsf{dim} \cdot 2^{B_{\mathsf{v}}}$. Indeed, since $T' = T$ by Lemma 5.18, for any $i \in [\mathsf{dim}]$ we have that if V samples $\mathbf{r}^{(i)}$, then P^* passes the testing phase of Protocol 9 with probability at least $\varepsilon_{\mathsf{test}}/2$, and so in particular, the vector $\bar{\mathbf{v}}_{\mathbf{r}^{(i)}}$ passes the checks at Step 3 of Protocol 9. This proves the claim.

Hence, all entries in $\mathbf{v}$ have bit-size at most $\mathsf{len}(\mathbf{r}^{-1}) + \log(\mathsf{dim}) + \log(q_0 \cdot \mathsf{dim} \cdot 2^{B_{\mathsf{v}}})$, where $\mathsf{len}(\mathbf{r}^{-1})$ denotes the largest bit-size of the entries in $\mathbf{r}^{-1}$. Since all entries in $\mathbf{r}$ have bit-size at most $\log(q_0)$, it follows using standard expressions for $\mathbf{r}^{-1}$ and simple crude bounding arguments, that $\mathsf{len}(\mathbf{r}^{-1}) \leq 2 \cdot \mathsf{dim} \cdot \log(q_0 \cdot \mathsf{dim})$, and the lemma follows. ☐

Let $\mathcal{E}_{\mathsf{extraction}}$ be the event that $\mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ outputs $\mathbf{v} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{dim}}$, where $\mathbf{v} = \mathbf{r}^{-1} \cdot \bar{\mathbf{v}}$ as defined in Protocol 10. Further, let $\mathcal{E}_{\mathsf{total}}$ be the event $\mathcal{E}_{\mathsf{accept}} \wedge (\mathbb{i}, \mathbb{x}, \mathbb{w}) \notin \mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, where $\mathbb{w} \leftarrow \mathsf{Ext}^{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})$. Then

$$\Pr[\mathcal{E}_{\mathsf{total}}] \leq \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] + \Pr[\neg\mathcal{E}_{\mathsf{extraction}}] \leq \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] + \frac{2 \cdot (\mathsf{dim} - 1)}{\varepsilon_{\mathsf{test}} \cdot q_0},$$

where the last inequality holds due to Lemma 5.21. Further, since $\varepsilon_{\mathsf{test}} \geq \varepsilon_{\mathsf{P}^*}$, we have

$$\frac{2 \cdot (\mathsf{dim} - 1)}{\varepsilon_{\mathsf{test}} \cdot q_0} \leq \frac{2 \cdot (\mathsf{dim} - 1)}{\varepsilon_{\mathsf{P}^*} \cdot q_0}$$

We next bound $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}]$.

Lemma 5.23. *The following holds:*

$$\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] \leq \theta + \varepsilon_{\mathsf{proj}}(\lambda) + \left(1 - \frac{2\mathsf{dist}_0}{3\mathsf{n}}\right)^{|J|} + (1 - \delta)^{|J|},$$

where

$$\theta = \frac{2 \cdot \mathsf{dim}^2 \cdot (B_{\mathsf{v}} + \log(\mathsf{dim}) + \log(q_0)) + B_{\mathsf{pt}} \cdot \mathsf{dim}^3 + \mathsf{dim}^2 + B_{\mathsf{v}} + \mu \cdot B_{\mathsf{pt}} + 2 \cdot \mu + 2}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

Proof. Assume $\mathcal{E}_{\mathsf{extraction}}$ holds. Let $S = (\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(\mathsf{dim})})$ be the list after executing $\mathsf{Ext}^{\mathsf{P}^*}(\mathbb{i}, \mathbb{x})$. For each $i \in [\mathsf{dim}]$, let $\bar{\mathbf{v}}_{\mathbf{r}^{(i)}} \in \mathbb{Z}^{\mathsf{dim}}$ be the vector sent by P^* in Step 2 of Protocol 9 after receiving $\mathbf{r}^{(i)}$ in Step 1 of Protocol 9. Since $\mathbf{r}^{(i)} \in T'$ by definition of Ext, we have by Lemmas 5.17 and 5.18 that $\bar{\mathbf{v}}_{\mathbf{r}^{(i)}} = \sum_{j \in [\mathsf{dim}]} \mathbf{r}_j^{(i)} \cdot \mathbf{v}_j$, for all $i \in [\mathsf{dim}]$. Hence, if $\mathcal{E}_{\mathsf{extraction}}$ occurs, we have that the output matrix $\mathbf{v} = \mathbf{r}^{-1} \cdot \bar{\mathbf{v}}_{\mathbf{r}}$ is precisely the $\mathsf{dim} \times \mathsf{dim}$ matrix whose i-th row is the vector $\mathbf{v}_i$, for all $i \in [\mathsf{dim}]$.

Let $f_{\mathbf{v}} \in \mathbb{Q}^{\mathsf{multilin}}[\mathbf{X}]$ be the multilinear polynomial whose coefficient matrix is $\mathbf{v}$. We claim that for $(\mathbb{i}, \mathbb{x}; f_{\mathbf{v}})$ to not belong to $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, it must be the case that $f_{\mathbf{v}}(\mathbf{q}) \neq y$. Indeed, all but the last three conditions in the definition of $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$ are met because $(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ are well-formed for $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$. On the other hand, since $\mathcal{E}_{\mathsf{extraction}}$ holds, Lemma 5.22 guarantees that

$$\mathbf{v} \in \mathbb{Q}^{\mathsf{dim} \times \mathsf{dim}}_{B_{\mathbf{v}} + (2\mathsf{dim}+1) \cdot \log(q_0 \cdot \mathsf{dim})}.$$

Further, for each $i \in [\mathsf{dim}]$, each $\mathbf{v}_i$ is the unique vector from $\mathbb{Q}^{\mathsf{dim}}$ such that $\Delta(\mathsf{Enc}_{\mathcal{C}_\lambda}(\mathbf{v}_i), \hat{\mathbf{u}}_i) < \delta$, which we assumed exists (since, as we saw, otherwise $\Pr[\mathcal{E}_{\mathsf{accept}}] < \mathsf{dim}/q_0$). Hence, $(\mathsf{gp}, \mathbb{i}, \mathbb{x}, \mathbb{w})$ satisfies all but the last condition in the definition of $\mathsf{REL}_{\mathsf{gp},\mathsf{Eval}}$, and so it must be the case that $f_{\mathbf{v}}(\mathbf{q}) = \mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^{\mathsf{T}} \neq y$.

Next, we follow the notation used in Section 5.4.2. Namely, we define:

- $\mathcal{E}_0$ is the event that $\mathsf{P}^*(\mathsf{gp}, \mathbb{i}, \mathbb{x})$ sends $\bar{\mathbf{v}} = \sum_{i \in [\mathsf{dim}]} r_i \cdot \mathbf{v}_i$ at Step 2 of Protocol 9
- $\mathcal{E}_1$ is the event that $\mathcal{E}_0$ occurs and V does not reject at Step 3 of Protocol 9.
- $\mathcal{E}_{\mathsf{local}}$ is the event that, in Step 1 of the evaluation phase (Protocol 8), V samples a prime $q \in \mathcal{P}_\lambda$ such that $\bar{\mathbf{v}}' \in \mathbb{Z}^{\mathsf{dim}}_{(q)}$, where $\bar{\mathbf{v}}' = \sum_{i \in [\mathsf{dim}]} \mathbf{q}_{1,i} \cdot \mathbf{v}_i$.
- $\mathcal{E}_{\mathsf{proj}}$ is the event that, at Step 1 of the evaluation phase in Protocol 8, V samples a prime $q \in \mathcal{P}_\lambda$ that is good with respect to $\mathcal{C}_\lambda$ (see Definition 5.1).
- $\mathcal{E}_2$ is the event that $\mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{local}} \wedge \mathcal{E}_{\mathsf{proj}}$ occurs and the vector $\bar{\mathbf{v}}_q$ sent by P in the evaluation phase satisfies $\bar{\mathbf{v}}_q = \phi_q(\bar{\mathbf{v}}')$ (note that $\phi_q(\bar{\mathbf{v}}')$ is well-defined because $\mathcal{E}_{\mathsf{local}}$ is assumed to hold).
- $\mathcal{E}_3$ is the event that $\mathcal{E}_2$ holds and $y = \mathbf{q}_1 \cdot \mathbf{v} \cdot \mathbf{q}_2^{\mathsf{T}}$ as rational numbers.

Then

$$\begin{aligned} &\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] \\ \leq &\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0] + \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \neg\mathcal{E}_0]. \end{aligned}$$

We have that $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \neg\mathcal{E}_0] \leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \neg\mathcal{E}_0]$ since, if $\mathcal{E}_{\mathsf{total}}$ occurs, then so does $\mathcal{E}_{\mathsf{accept}}$. Now, $\mathcal{E}_{\mathsf{extraction}}$ and $\mathcal{E}_{\mathsf{accept}}$ are independent events, and so $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \neg\mathcal{E}_0] = \Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_0]$. By Lemma 5.10, $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_0] \leq (1-\delta)^{|J|}$. Next we bound $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0]$. We have

$$\begin{aligned} &\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0] \\ \leq &\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0 \wedge \mathcal{E}_1] \cdot \Pr[\mathcal{E}_1 \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0] + \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0 \wedge \neg\mathcal{E}_1]. \end{aligned}$$

Note that $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0 \wedge \neg\mathcal{E}_1] = 0$, since, if $\neg\mathcal{E}_1$ holds, then the verifier rejects at Step 3 of the testing phase (Protocol 9), and so $\mathcal{E}_{\mathsf{accept}}$ cannot hold. In particular, neither does $\mathcal{E}_{\mathsf{total}}$. Also observe that $\Pr[\mathcal{E}_1 \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0] = \Pr[\mathcal{E}_1 \mid \mathcal{E}_0]$, because $\mathcal{E}_1$ and $\mathcal{E}_{\mathsf{extraction}}$ are independent events. By Lemma 5.11,

$$\Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \leq \min\left\{1, \frac{\mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathbf{v}}}}{\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty \cdot q_0}\right\}.$$

Note that $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_0 \wedge \mathcal{E}_1] = \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1]$, because if $\mathcal{E}_1$ holds, then by definition so does $\mathcal{E}_0$. Next we have

$$\begin{aligned}\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1] &\leq \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1 \wedge \neg\mathcal{E}_{\mathsf{local}}] \cdot \Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1] \\ &\quad + \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{local}}] \\ &\leq \Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1] + \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{local}}].\end{aligned}$$

Similarly as before, since the event $\mathcal{E}_{\mathsf{extraction}}$ and the event $\mathcal{E}_{\mathsf{local}}$ are independent, $\Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1] = \Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_1]$. By Lemma 5.14,

$$\Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_1] \leq \frac{\mathsf{dim}^2 \cdot (\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) + 2 \cdot \log(q_0))}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

So far we have seen that

$$\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] \leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_0] + \Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \cdot (\Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_1] + \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{local}}]).$$

We proceed to bound $\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{local}}]$. To avoid cluttering the text, we denote $\mathcal{E}_{\mathsf{total}}, \mathcal{E}_{\mathsf{extraction}}, \mathcal{E}_{\mathsf{local}}$ simply by $\mathcal{E}_{\mathsf{to}}, \mathcal{E}_{\mathsf{ex}}, \mathcal{E}_{\mathsf{lo}}$ for the duration of this proof, respectively. We now have

$$\Pr[\mathcal{E}_{\mathsf{to}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}}] \leq \Pr[\mathcal{E}_{\mathsf{to}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge \mathcal{E}_3] + \Pr[\mathcal{E}_{\mathsf{to}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)].$$

We have $\Pr[\mathcal{E}_{\mathsf{to}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_{\mathsf{to}} \wedge \mathcal{E}_{\mathsf{lo}} \wedge \mathcal{E}_3] = 0$ because, as we argued, if $(\mathbb{i}, \mathbb{x}; \mathbb{w}) \notin \mathsf{REL}_{\mathsf{Eval}}$ and $\mathcal{E}_{\mathsf{ex}} = \mathcal{E}_{\mathsf{extraction}}$ happens, then $f_{\mathbf{v}}(\mathbf{q}) \neq y$ and so $\mathcal{E}_3$ cannot hold, where $\mathbb{w} = (f_{\mathbf{v}}) \leftarrow \mathsf{Ext}^{\mathsf{P}^*}(\mathbb{i}, \mathbb{x})$. Further, since $\mathcal{E}_{\mathsf{to}}$ implies $\mathcal{E}_{\mathsf{accept}}$,

$$\Pr[\mathcal{E}_{\mathsf{to}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)] \leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)]$$

Moreover, $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_{\mathsf{ex}} \wedge \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)] = \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)]$ because $\mathcal{E}_{\mathsf{ex}}$ and $\mathcal{E}_{\mathsf{accept}}$ are independent events. Hence, we proceed to bound $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)]$. We have

$$\begin{aligned}&\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3)] \\ &\leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] + \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3) \wedge (\neg\mathcal{E}_2)].\end{aligned}$$

By definition of $\mathcal{E}_2$, if $\mathcal{E}_2$ occurs, then so do $\mathcal{E}_1$ and $\mathcal{E}_{\mathsf{lo}}$. Hence $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] = \Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2]$. Now, by Lemma 5.16,

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid (\neg\mathcal{E}_3) \wedge \mathcal{E}_2] \leq \frac{\mathsf{dim}^2 \cdot (\log(\|\mathsf{Words}(\hat{\mathbf{u}}\|_\infty)) + 2 \cdot \log(q_0)) + B_{\mathbf{v}} + \mu \cdot B_{\mathsf{pt}} + 3\mu + 2}{\lambda \cdot |\mathcal{P}_\lambda|}.$$

On the other hand,

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_3) \wedge (\neg\mathcal{E}_2)] = \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2)]$$

because, by definition of $\mathcal{E}_3$, if $\mathcal{E}_2$ does not happen, then $\mathcal{E}_3$ cannot happen either. Now,

$$\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2)] \leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{proj}}] + \Pr[\neg\mathcal{E}_{\mathsf{proj}}].$$

By definition, $\Pr[\neg\mathcal{E}_{\mathsf{proj}}] = \varepsilon_{\mathsf{proj}}(\lambda)$, and by Lemma 5.15, $\Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{proj}}] \leq \left(1 - \frac{2\mathsf{dist}_0}{3}\right)^{|J|}$. Overall, we have shown that

$$\begin{aligned}
&\Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] \\
&\leq \Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_0] + \Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \cdot \begin{pmatrix} \Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_1] + \Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_3 \wedge \mathcal{E}_2] \\ \quad + \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{proj}}] + \Pr[\neg\mathcal{E}_{\mathsf{proj}}] \end{pmatrix}.
\end{aligned}$$

We next show that that the right-hand side of the above equality satisfies the bound stated in the lemma. Indeed, first note that the right-hand side above is at most

$$\begin{aligned}
&\Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_0] + M \cdot (\Pr[\neg\mathcal{E}_{\mathsf{local}} \mid \mathcal{E}_1] + \Pr[\mathcal{E}_{\mathsf{accept}} \mid \neg\mathcal{E}_3 \wedge \mathcal{E}_2]) \\
&\qquad + \Pr[\mathcal{E}_{\mathsf{accept}} \mid \mathcal{E}_1 \wedge \mathcal{E}_{\mathsf{lo}} \wedge (\neg\mathcal{E}_2) \wedge \mathcal{E}_{\mathsf{proj}}] + \Pr[\neg\mathcal{E}_{\mathsf{proj}}].
\end{aligned} \tag{19}$$

where

$$M = \min\left\{1, \frac{\mathsf{dim} \cdot q_0 \cdot 2^{B_{\mathsf{v}}}}{\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty \cdot q_0}\right\},$$

and where we have used Lemma 5.11, which states that $\Pr[\mathcal{E}_1 \mid \mathcal{E}_0] \leq M$.

We next show that

$$M \cdot \log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) \leq B_{\mathsf{v}} + \log(\mathsf{dim}). \tag{20}$$

The reason this holds is that if $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty < \mathsf{dim} \cdot 2^{B_{\mathsf{v}}}$, then $M = 1$, and $\log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) \leq B_{\mathsf{v}} + \log(\mathsf{dim})$. On the other hand, if $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty \geq \mathsf{dim} \cdot 2^{B_{\mathsf{v}}}$, write $\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty = k \cdot \mathsf{dim} \cdot 2^{B_{\mathsf{v}}}$ for some $k \geq 1$. Then $M = \mathsf{dim} \cdot 2^{B_{\mathsf{v}}} / \|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty$, and

$$\frac{\mathsf{dim} \cdot 2^{B_{\mathsf{v}}}}{\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty} \cdot \log(\|\mathsf{Words}(\hat{\mathbf{u}})\|_\infty) = \frac{\log(k) + B_{\mathsf{v}} + \log(\mathsf{dim})}{k} \leq B_{\mathsf{v}} + \log(\mathsf{dim}),$$

where, in the last inequality, we used that $k \geq 1$. This proves the inequality (20).

The bound in the lemma now follows by applying the bounds for the different terms in (19) we have given during the course of the proof. □

We have proved that, if the words $\{\hat{\mathbf{u}}_i\}_{i \in [\mathsf{dim}]}$ have δ-correlated agreement in $\mathcal{C}_\lambda$, then

$$\varepsilon_{\mathsf{ks}} = \Pr[\mathcal{E}_{\mathsf{total}}] \leq \Pr[\mathcal{E}_{\mathsf{total}} \mid \mathcal{E}_{\mathsf{extraction}}] + \frac{2 \cdot (\mathsf{dim} - 1)}{\varepsilon_{\mathsf{test}} \cdot q_0}.$$

Otherwise, we saw that $\varepsilon_{\mathsf{ks}} \leq \mathsf{n}/q_0 + (1-\delta)^{|J|}$. We conclude that $\varepsilon_{\mathsf{ks}}(\mathbb{i}, \mathbb{x}, \varepsilon_{\mathsf{P}^*}(\mathbb{i}, \mathbb{x}))$ satisfies the bound stated in Theorem 5.5. Next, we prove that Ext runs in expected polynomial time.

Lemma 5.24. *The extractor's testing subprocedure Protocol 11 runs in time*

$$\mathsf{poly}(B_{\mathbf{v}}, \log(\|M_{\mathcal{C}}\|_\infty), \log(q_0), \mathsf{n}, \mathsf{dim}).$$

Proof. Recall that $B_{\mathsf{BoundEnc}} = \log(\|M_{\mathcal{C}_\lambda}\|_\infty \cdot \mathsf{dim}) + B_{\mathbf{v}} + (2\mathsf{dim}+1) \cdot \log(q_0 \cdot \mathsf{dim})$. Step 1 of Protocol 11 can be performed in time at most $\mathsf{dim} \cdot \mathsf{n} \cdot B_{\mathsf{BoundEnc}}$. Step 2 requires computing at most $\mathsf{dim} \cdot \mathsf{n}$ multiplications and additions of rational numbers whose bit-size is at most B_{BoundEnc}, and so it can be performed in the claimed polynomial time. Step 3 can also be performed in polynomial time since it requires comparing the entries of the n-dimensional vectors $\hat{\mathbf{u}}'_{\mathbf{r}}$ and $\mathsf{Enc}_{\mathcal{C}_\lambda}(\bar{\mathbf{v}}_{\mathbf{r}})$, both of which have entries whose bit-size have length polynomial on B_{BoundEnc}. □

Lemma 5.25. *Protocol 10 runs in expected time* $\mathsf{poly}(\varepsilon_{\mathsf{P}^*}(\mathsf{gp}, \mathbb{i}, \mathbb{x})^{-1}, B_{\mathbf{v}}, \log(\|M_{\mathcal{C}}\|_\infty),$ $\log(q_0), \mathsf{n}, \mathsf{dim})$.

Proof. By Lemmas 5.18 and 5.19, the probability that a randomly sampled $\mathbf{r} \in [0, q_0 - 1]^{\mathsf{dim}}$ belongs to T' is at least $\varepsilon_{\mathsf{test}}/2$. By definition of T', a vector $\mathbf{r}$ is added to S if and only if $\mathbf{r} \in T'$. Hence, the expected number of vectors $\mathbf{r}$ tried before S has size at least dim is $\mathsf{dim} \cdot 2 \cdot \varepsilon_{\mathsf{test}}^{-1}$, and so, in expectation, Protocol 10 passes

$$\mathsf{dim} \cdot 2 \cdot \varepsilon_{\mathsf{test}}^{-1} \geq \mathsf{dim} \cdot 2 \cdot \varepsilon_{\mathsf{P}^*}^{-1}$$

times through Step 2 before reaching Step 7, where we have used that $\varepsilon_{\mathsf{P}^*} \leq \varepsilon_{\mathsf{test}}$.

The lemma now follows, since each individual step in Protocol 10 can be performed in polynomial time due to Lemma 5.24. In particular, in the last step, Ext can compute the matrix $\mathbf{v}$ as $\mathbf{v} = \mathbf{r}^{-1} \cdot \bar{\mathbf{v}}$. Since each entry in $\mathbf{r}$ has bit size at most $\log(q_0)$, the $\mathsf{dim} \times \mathsf{dim}$ matrix $\mathbf{r}^{-1}$ can be computed in polynomial time, and in particular each entry in $\mathbf{r}^{-1}$ has bit-size of polynomial size (cf. the proof of Lemma 5.22 for more details). Further, as argued in the proof of Lemma 5.22, all entries in $\bar{\mathbf{v}}$ are integers with bit-size at most $\log(\mathsf{dim} \cdot q_0 \cdot B_{\mathbf{v}})$, and so Ext can compute $\mathbf{v}$ in the claimed polynomial time. □

This completes the proof of Theorem 5.5.

Remark 5.26 (Removing the dependence on the aborting probability)**.** In the above proof (concretely, in Protocol 11), it is assumed that the probability $\varepsilon_{\mathsf{test}}$ with which the verifier passes the testing phase is known to the extractor and that the running time of the extractor depends in this error. To remove this dependency and construct an extractor that runs in expected time independent of $\varepsilon_{\mathsf{test}}$, we can rely on a standard argument of Goldreich and Kahan [GK96] where we estimate the aborting probability of the verifier in the testing phase using rewinds, i.e. by repeatedly simulating executions of the evaluation IOP (Protocol 8) between P^* and V. Note that V runs in polynomial time due to Remark 5.3. The application of this technique to the Brakedown PCS has been described in the proof of [GLS+23, Lemma 3]. The adaptation of this lemma to our setting is straightforward by taking into account the sizes of the challenges, its response, as well as the success probability and the running time of our extractor.

5.5 Compiling Zinc-PIOP into a succinct argument using Zip and the COS transformation

In this section, we provide an outline of how to use the Zinc PIOP from Section 4 and Zip in order to obtain a succinct argument. In short, the Zinc PIOP is turned into an IOP by replacing Zinc's polynomial oracles with Zip's oracle commitments. Then, the resulting IOP is turned into a succinct argument by replacing all oracles by Merkle tree commitments MT. The approach relies on standard compilation arguments, which can be found in [BFS19, CHM+19, COS20, CY24]. However, in our context, a point deserving special attention is how the global parameters of Zinc and Zip are set up, so that a secure compiled protocol is ensured. Precisely, one has to make sure the bit-bounds B and $B_{\mathbf{v}}$ in Zinc and Zip are chosen appropriately. An intuitive explanation of how this is done can be found in Section 2.3.

Let $\mathsf{REL}_{\mathsf{gp},\mathbb{Q},\mathcal{Q}}$ be an algebraic indexed relation over $\mathbb{Q}$, with $\mathsf{gp} = (k, m, n, \mu, B_{\mathbf{v}}, \mathbb{Q}, \mathcal{Q})$. Here, $B_{\mathbf{v}}$ is the bit-size of the integer witness coefficients we expect honest provers to use. In other words, we expect honest provers to use witnesses $\mathbb{w}$ consisting of polynomials with coefficients in $\mathbb{Z}_{B_{\mathbf{v}}}$ (cf. the beginning of Section 2 for more information).

Let $\mathcal{P}_\lambda$ be a set of primes, and let $\{\mathcal{C}_\lambda \mid \lambda \geq 1\}$ be a $(\mathcal{C}_\lambda, \mathsf{dist}_0)$-projectable family of linear integer codes over $\mathbb{Q}$, following the same setting as in Section 6, of code-length n and dimension dim (depending on λ).

Now let q_0 be a prime and let $B = B_{\mathbf{v}} + (2\mathsf{dim} + 1) \cdot \log(q_0 \cdot \mathsf{dim})$, and take parameters for our Zip PCS in IOP form $\mathsf{gp}_{\mathsf{Zip}} = (\mu, B_{\mathbf{v}}, B_{\mathsf{pt}}, \delta, q_0)$ (cf. Section 5.2). Let gp' be a new tuple of global parameters for $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$ defined as $\mathsf{gp}' = (k, m, n, \mu, B, \mathbb{Q}, \mathcal{Q})$. Precisely, gp' is gp after replacing the bit-size bound $B_{\mathbf{v}}$ by the larger bound B. We refer to Section 2.3 for an explanation of the role and the need for the two bounds $B_{\mathbf{v}}$ and B.

Let $\Pi_{\mathsf{PIOP}} = (\mathsf{Indexer}_{\mathsf{PIOP}}, \mathsf{P}_{\mathsf{PIOP}}, \mathsf{V}_{\mathsf{PIOP}})$ be a PIOP over $\mathbb{Q}_B$ for the relation $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$. We construct an IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}} = (\mathsf{Indexer}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{P}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{V}_{\mathsf{IOP}}^{\mathsf{compiled}})$ for the relation $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$ by replacing each polynomial oracle $[[h]]$ in $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$ with a (non-polynomial) oracle to the encoding of the coefficient matrix $\mathbf{v}^h$ of the polynomial h, as defined in the Zip commitment procedure, and by making the same replacement with the polynomial oracles sent by P during the execution of Π_{PIOP}. We adjust the behavior of the indexer accordingly, i.e., if $\mathsf{Indexer}_{\mathsf{PIOP}}$ outpus $(\mathsf{vp}_{\mathsf{PIOP}}, \mathsf{pp}_{\mathsf{PIOP}})$, then $\mathsf{Indexer}_{\mathsf{IOP}}^{\mathsf{compiled}}$ outputs verifier and prover parameters $(\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{pp}_{\mathsf{IOP}}^{\mathsf{compiled}})$ where $\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}$ is $\mathsf{vp}_{\mathsf{PIOP}}$ after replacing all oracles to a polynomial h by the oracle to the encodings of the coefficient matrix $\mathbf{v}^h$; and $\mathsf{pp}_{\mathsf{IOP}}^{\mathsf{compiled}}$ is the same as $\mathsf{pp}_{\mathsf{PIOP}}$, where the polynomial h is replaced with its coefficient matrix $\mathbf{v}^h$. Now, every time $\mathsf{V}_{\mathsf{PIOP}}$ would query a polynomial oracle during the execution of Π_{PIOP}, we have $\mathsf{P}_{\mathsf{IOP}^{\mathsf{compiled}}}$ instead report the corresponding evaluation, and then $\mathsf{P}_{\mathsf{IOP}^{\mathsf{compiled}}}$ and $\mathsf{V}_{\mathsf{IOP}^{\mathsf{compiled}}}$ execute Zip's evaluation IOP Protocol 8 to certify that the reported evaluation is indeed correct.

It remains to argue that the resulting IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ achieves knowledge soundness. To justify this, we construct an extractor $\mathsf{Ext}_{\mathsf{IOP}^{\mathsf{compiled}}}$ that, when interacting with a malicious prover $\mathsf{P}^*_{\mathsf{IOP}^{\mathsf{compiled}}}$ for $\Pi_{\mathsf{IOP}^{\mathsf{compiled}}}$, is able to extract the witness while relying on the extractor $\mathsf{Ext}_{\mathsf{PIOP}}$ of the underlying Π_{PIOP} and the extractor $\mathsf{Ext}_{\mathsf{Zip}}$ of the Zip evaluation IOP. Our extractor $\mathsf{Ext}_{\mathsf{IOP}^{\mathsf{compiled}}}$ works as follows: it receives the global paramaters gp' and the index-instance pair $(\mathbb{i}, \mathbb{x})$ from $\mathsf{P}^*_{\mathsf{IOP}^{\mathsf{compiled}}}$, executes the indexer $(\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{pp}_{\mathsf{IOP}}^{\mathsf{compiled}}) \leftarrow \mathsf{Indexer}_{\mathsf{IOP}}^{\mathsf{compiled}}(\mathsf{gp}', \mathbb{i}, \mathbb{x})$

and starts (simulating) an execution of $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ between $\mathsf{P}^*_{\mathsf{IOP}^{\mathsf{compiled}}}$ and the verifier of $\Pi_{\mathsf{IOP}^{\mathsf{compiled}}}$, where the verifier obtains the calculated $\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}$, until the first oracle of the index, as part of $\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}$, is queried.[8] Once this query is being asked, the extractor $\mathsf{Ext}_{\mathsf{Zip}}$ of the Zip IOP is executed to obtain the polynomial underlying the oracle. After the execution of $\mathsf{Ext}_{\mathsf{Zip}}$ has finished, $\mathsf{P}^*_{\mathsf{IOP}^{\mathsf{compiled}}}$ is rewound until the beginning of the execution of $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$. Next, another execution is simulated until the second oracle of the index, as part of $\mathsf{vp}_{\mathsf{IOP}}^{\mathsf{compiled}}$, is being queried, and the extraction proceeds as described before. This is done until the coefficient matrices $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}$ underlying all oracles of the index-instance pair are extracted, and continues for all the coefficient matrix oracles $\mathbf{v}'^{(1)}, \dots, \mathbf{v}'^{(m)}$ that are exchanged during the protocol. We highlight that, when the oracles that are exchanged during the execution of $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ are extracted, the protocol is only rewound until after the extracted oracle has been output by $\mathsf{P}^*_{\mathsf{IOP}^{\mathsf{compiled}}}$. The execution then continues from this point on. In the final step, the extractor $\mathsf{Ext}_{\mathsf{PIOP}}$ of the underlying PIOP Π_{PIOP} is executed interpreting the extracted matrices $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}, \mathbf{v}'^{(1)}, \dots, \mathbf{v}'^{(m)}$ as polynomials, and its output is used as the final output for the witness $\mathbb{w}$. An avid reader might have observed that the standard notion of knowledge soundness does not capture the above described rewinding techniques. Fortunately, in a work by Lindell [Lin01], the notion of witness-extended emulation has been introduced that provides an additional procedure that allows to simulate the mentioned transcripts. Lindell further proved that the notions of knowledge soundness and witness-extended emulation are equivalent, which allows us to directly rely on this notion and concludes the argument.

The above described technique is an informal overview of the proof of [BFS20, Theorem 4]. We refer to this work for the formal details.

From Π_{IOP} to a succinct interactive argument for $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$. Next, we compile the IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}} = (\mathsf{Indexer}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{P}_{\mathsf{IOP}}^{\mathsf{compiled}}, \mathsf{V}_{\mathsf{IOP}}^{\mathsf{compiled}})$ for the relation $\mathsf{REL}_{\mathsf{gp}',\mathbb{Q},\mathcal{Q}}$ obtained in the previous section into a succinct argument $\Pi_{\mathsf{ARG}}^{\mathsf{succ}} = (\mathsf{Indexer}_{\mathsf{ARG}}^{\mathsf{succ}}, \mathsf{P}_{\mathsf{ARG}}^{\mathsf{succ}}, \mathsf{V}_{\mathsf{ARG}}^{\mathsf{succ}})$ for the same relation by relying on the so-called COS transformation [COS20]. In more detail, the COS transformation allows us to compile an IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$, with oracles contained in the index-instance pair $(\mathbb{i}, \mathbb{x})$, into a SNARK. To do this, we extend the underlying indexer $\mathsf{Indexer}_{\mathsf{IOP}}^{\mathsf{compiled}}$ by committing to the oracle using Merkle trees MT into the indexer $\mathsf{Indexer}_{\mathsf{ARG}}^{\mathsf{succ}}$. In more detail, the verifier parameters $\mathsf{vp}_{\mathsf{ARG}}^{\mathsf{succ}}$ output by $\mathsf{Indexer}_{\mathsf{ARG}}^{\mathsf{succ}}$ contain Merkle commitments to the oracles that are part of the index-instance pair, i.e., $\mathsf{rt} := \mathsf{MT.Com}^{\mathcal{H}}(\mathbf{v}^h)$ where $\mathcal{H}$ is a random oracle, and the output prover parameters $\mathsf{pp}_{\mathsf{ARG}}^{\mathsf{succ}}$ contain the corresponding openings which, in our case, is simply the coefficient matrix $\mathbf{v}^h$ of the polynomial h. The remainder of the transformation proceeds as the iBCS transformation [CDGS23], i.e., instead of sending oracles $[[\hat{\mathbf{u}}]]$ to the verifier, the prover sends Merkle commitments to these oracles $\mathsf{rt}' := \mathsf{MT.Com}^{\mathcal{H}}(\hat{\mathbf{u}})$ and the final oracle queries are answered by providing the corresponding openings to the Merkle commitments. To answer the queries made by the verifier $\mathsf{V}_{\mathsf{ARG}}^{\mathsf{succ}}$ to the Merkle commitments rt of the index, the prover $\mathsf{P}_{\mathsf{ARG}}^{\mathsf{succ}}$ relies on the corresponding openings $\hat{\mathbf{u}}$, as part of $\mathsf{pp}_{\mathsf{ARG}}^{\mathsf{succ}}$. For the queries to the Merkle commitments exchanged during the protocol execution rt', the prover $\mathsf{P}_{\mathsf{ARG}}^{\mathsf{succ}}$ relies on its knowledge of $\hat{\mathbf{u}}$. In the last step, the whole

[8] Here, we are assuming without loss of generality that all the oracles in the index-instance pair as well as the ones that are exchanged during the protocol are queried. Oracles that are not being queried can simply be omitted.

protocol can be made non-interactive by applying the Fiat-Shamir transformation [FS86]. Therefore, obtaining a SNARK with preprocessing which follows from the succinctness of the parameters $\mathsf{vp}_{\mathsf{ARG}}^{\mathsf{succ}}$ and $\mathsf{pp}_{\mathsf{ARG}}^{\mathsf{succ}}$ and the succinctness of the iBCS transformation.

A drawback of the above transformation is that it requires the underlying IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ to satisfy state-restoration knowledge soundness for the resulting *non-interactive* argument to be extractable. Since the $\mathsf{Zip\text{-}IOP}$ $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ presented in this work only achieves standard knowledge soundness and not state-restoration knowledge soundness, we conclude by realizing a succinct interactive argument instead of a non-interactive one. We do this by executing the compilation described above without applying the Fiat-Shamir transformation at the end, i.e, we execute the iBCS transformation together with the modifed indexer $\mathsf{Indexer}_{\mathsf{ARG}}^{\mathsf{succ}}$. We refer to this transformation as the iCOS transformation. It remains to argue that the resulting interactive protocol achieves knowledge soundness, since the succinctness directly follows from the succinctness of the applied Merkle commitments MT. Fortunately, in [CY24, Section 32.8.1] it has been shown how to reduce the security of the COS transformation to the security of the BCS transformation (the non-interactive version of the iBCS transformation by applying Fiat-Shamir [FS86]). The authors further extend this to the iBCS transformation [CY24, Remark 32.8.7], i.e., the iCOS transformation is knowledge sound if the underlying iBCS transformation is knowledge sound. It remains to argue the knowledge soundness of the iBCS transformation which has been proven in [CDGS23] for general vector commitments VC and, therefore, also holds for Merkle commitments MT in the random oracle model. This results in our final succinct argument for the relation $\mathsf{REL}_{\mathsf{gp}',\mathcal{R},\mathcal{Q}}$, relying on the IOP $\Pi_{\mathsf{IOP}}^{\mathsf{compiled}}$ for the same relation and the random oracle model.

6 Integral Juxtaposed Expand-Accumulate (JEA) codes over $\mathbb{Q}$

Let $n, d \geq 1$, and let $A \in \mathbb{Z}^{n\times n}$ be an upper-triangular matrix where all entries on and above the main diagonal are equal to 1. We call A an *accumulator matrix*. Let $S = [0, s-1]$, for some integer $s > 1$. Define sparse matrices $E_1, E_2 \in \mathbb{Z}^{d\times n}$ as follows:

- Let $\gamma \geq 1$ be a constant and let $t = \gamma \log(\mathsf{n})$. Define the set $S_{(n,t)} := \{\mathbf{x} \in S^n \mid \mathsf{wt}(\mathbf{x}) = t\}$, where $\mathsf{wt}(\mathbf{x})$ denotes the number of nonzero entries of the vector $\mathbf{x}$. Denote sampling uniformly at random from $S_{(n,t)}$ by $\mathsf{BP}(d, n, t, S)$. E_1 is the matrix constructed by independently sampling each row from $\mathsf{BP}(d, n, t, S)$.

- Let $0 < p < 1$ be a parameter such that $p = t/\mathsf{n}$, with t as above. Denote by $\mathsf{Ber}_p(S)$ the generalized Bernoulli distribution with parameter p and sampling set S, more precisely, $\mathsf{Ber}_p(S)$ outputs a nonzero element $s \in S$ with probability p and zero with probability $1-p$. Let E_2 be the matrix constructed by sampling each entry independently from $\mathsf{Ber}_p(S)$.

We call E_1 and E_2 *expander matrices*. Note that all entries of E_1 and E_2 come from a finite integral interval. Moreover, the choice of p and t guarantees that E_1 and E_2 have the same sparsity.

Remark 6.1 (On the rank of E_1 and E_2)**.** For E_2 to have rank d, there must exist at least one non-singular $d \times d$ submatrix. By [VJS21, Corollary 1.3] the probability that a $d \times d$ matrix sampled from $\mathsf{Ber}_p(S)$ is singular is $2d \cdot (1-p)^n + (1 + O(e^{-c_p d})) \cdot d \cdot (d-1)(p^2 + (1-p)^2)^d$, for some constant $c_p > 0$. We conjecture that the probability E_1 is singular is negligible. We believe this follows from [COEG+20, Theorem 1.1] but we do not have a concrete proof at the time of writing this work. In any case, one can always append the identity matrix to E_1. This ensures we obtain the desired full row rank. The downside, however, is that it increases the length of $\mathcal{C}_{\mathsf{Ber}}$.

Let $M_{\mathsf{BP}} = E_1 \cdot A$ and $M_{\mathsf{Ber}} = E_2 \cdot A$, where "$\cdot$" denotes matrix multiplication. By construction, these are matrices in $\mathbb{Z}^{d \times n}$. Define integral linear codes $\mathcal{C}_{\mathsf{BP}}$ and $\mathcal{C}_{\mathsf{Ber}}$ over $\mathbb{Q}$ as the $\mathbb{Q}$-vector spaces spanned by the rows of M_1 and M_2 respectively. When $\mathsf{rk}(E_1) = \mathsf{rk}(E_2) = d$, the integral linear codes $\mathcal{C}_{\mathsf{BP}}$ and $\mathcal{C}_{\mathsf{Ber}}$ have dimensions $\mathsf{dim}(\mathcal{C}_{\mathsf{BP}}) = \mathsf{dim}(\mathcal{C}_{\mathsf{Ber}}) = d$ and code length $\mathsf{n}(\mathcal{C}_{\mathsf{BP}}) = \mathsf{n}(\mathcal{C}_{\mathsf{Ber}}) = n$.

Definition 6.1 (Integral juxtaposed expand-accumulate code over $\mathbb{Q}$ – cf. [BFK+24])**.** *Let $n, d, \gamma \geq 1$ and let $t = \gamma \log(n)$ and $p = t/n$. The* integral juxtaposed expand-accumulate (JEA) code over $\mathbb{Q}$*, denoted $\mathcal{C}_{\mathsf{JEA}}$, is the integral linear code over $\mathbb{Q}$ whose codewords are those in the image of the map*

$$\begin{aligned} \mathsf{Enc}_{\mathcal{C}_{1,2}} : \mathbb{Q}^{\mathsf{dim}} &\to \mathbb{Q}^{\mathsf{n}}, \\ \mathbf{v} &\mapsto (\mathbf{v} \cdot M_{\mathsf{BP}}) || (\mathbf{v} \cdot M_{\mathsf{Ber}}). \end{aligned}$$

In other words, the codewords of $\mathcal{C}_{\mathsf{JEA}}$ are concatenations of codewords of the integral linear codes $\mathcal{C}_{\mathsf{BP}}$ and $\mathcal{C}_{\mathsf{Ber}}$ defined above. Assuming $\mathsf{rk}(E_1) = \mathsf{rk}(E_2) = d$, the dimension of $\mathcal{C}_{\mathsf{JEA}}$ is $\mathsf{dim} = d$, and its code length is $\mathsf{n} = 2n$.

We fix $\mathcal{P}_\lambda$ a finite set of consecutive primes. The prime in $\mathcal{P}_\lambda$ with the smallest bit-size is denoted by $q_{\min}$ and its respective bit-size $\log(q_{\min})$. Using the Prime Number Theorem, we can infer that the largest prime $q_{\max}$ in $\mathcal{P}_\lambda$ is at most $q_{\min} + |\mathcal{P}_\lambda| \cdot \log_e(q_{\min})$ and has respective bit-size $\log(q_{\max}) \leq \log(q_{\min} + |\mathcal{P}_\lambda| \cdot \log(q_{\min}) \cdot \log_e(2))$. Here $\log_e$ denotes the logarithm in base e.

In this section, we prove that the family $\{\mathcal{C}_{\lambda 1,2} \mid \lambda \geq 1\}$ of integral JEA codes over $\mathbb{Q}$ are $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable, where $0 < \mathsf{dist}_0 \leq \frac{1}{2(5 \cdot 2^5 \cdot e)^2}$.

Lemma 6.2 (Rank is preserved modulo good primes)**.** *Let $S = [0, s-1]$, for some positive integer $s > 1$, and let $\mathcal{P}_\lambda$ and $q_{\min}$ be as above. For $n > d$, let M be a $d \times n$ matrix with entries in S and rank $\mathsf{rk}(M) = d$. Let $\phi_q : \mathbb{Z}_{(q)} \to \mathbb{F}_q$, for $q \in \mathcal{P}_\lambda$, be the canonical projection. Then*

$$\mathsf{rk}(M) = \mathsf{rk}(\phi_q(M)),$$

except for at most $\frac{d^2 \log(|S|) + 2\log(d) + 1}{\log(q_{\min})}$ primes in $\mathcal{P}_\lambda$.

Proof. Denote by $\mathcal{M}_{\mathsf{ns},d}$ the set of non-singular $d \times d$- submatrices of M, i.e. matrices obtained from M by deleting $n-d$ columns at a time, whose determinant is nonzero. $\mathcal{M}_{\mathsf{ns},d}$ is non-empty since $\mathsf{rk}(M) = d$. Observe that $\mathsf{rk}(\phi_q(M)) \leq \mathsf{rk}(M)$ always holds. The inequality becomes strict if all $d \times d$-submatrices of $\phi_q(M)$ have determinant zero. Index the primes in

$\mathcal{P}_\lambda$ by $[m]$. Let $\mathcal{P}_{\lambda\mathsf{bad}} = \{q \in \mathcal{P}_\lambda \mid \det(\phi_q(M_d)) = 0 \text{ for all } M_d \in \mathcal{M}_{\mathsf{ns},d}\}$. Index $\mathcal{P}_{\lambda\mathsf{bad}}$ by a set $I \subset [m]$.

Write $M = (m_{i,j})_{i\in[d],j\in[n]}$. Then any $M_d \in \mathcal{M}_{\mathsf{ns},d}$ can be written as $M_d = (m_{i,j})_{i\in[d],j\in D}$, for $D \subseteq [n]$, with $|D| = d$. Let $D = \{j_1, \ldots, j_d\}$. Then $P = \sum_{\sigma\in S_d} \mathsf{sgn}(\sigma) \prod_{i=1}^{d} m_{\sigma(i),j_i}$ is the determinant of M_d, where S_d is the symmetric group on a set of d elements. P can be viewed as a multilinear integral polynomial on d^2 variables $m_{\sigma(i),j_i}$ with at most d^2 nonzero coefficients, lying in the set $\{-1, 1\}$. Suppose that for $\mathbf{m} = (m_{\sigma(1),j_1}, \ldots, m_{\sigma(d),j_d})$, we have

$$P(\mathbf{m}) \in \bigcap_{i\in I} \ker(\phi_{q_i}) \setminus \{0\}.$$

Since all $M_d \in \mathcal{M}_{\mathsf{ns},d}$ are non-singular, $P(\mathbf{m}) \neq 0$. Notice that the above statement is equivalent to saying that the determinant of a $d \times d$-submatrix of M vanishes after projecting to $\mathbb{F}_{q_i}$ by ϕ_{q_i}, for all $q_i \in \mathcal{P}_{\lambda\mathsf{bad}}$. Since all submatrices of $\phi_{q_i}(M)$, $q_i \in \mathcal{P}_\lambda$, result from projecting M through ϕ_{q_i}, and both P and ϕ_{q_i} are linear, finding a bound on $|I|$ proves our statement.

Let $\mathcal{R} = \mathbb{Q}, \mathfrak{R} = \{\mathbb{Z}_{(q_i)} \mid i \in [m]\}$ and $\Phi = \{\phi_{q_i} : \mathbb{Z}_{(q_i)} \to \mathbb{F}_{q_i} \mid i \in [m]\}$, and $[m]$ as above. By Proposition 4.6 $(\mathfrak{R}, \Phi)$ is a k-expanding family of homomorphisms, with $k = \log(q_{\min})$. Hence, by Definition 4.7 there is a $m_{\sigma(i),j_i}$, $i \in [d]$, with bit-size larger than

$$\frac{\log(q_{\min}) \cdot |I| - \log(d^2) - 1}{d^2}$$

bits. But since $m_{\sigma(i),j_i} \in S$, has bit-size at most $\log(|S|)$, leading to

$$|I| < \frac{d^2 \log(|S|) + 2\log(d) + 1}{\log(q_{\min})},$$

finishing the proof. □

Theorem 6.3 (Integral JEA codes over $\mathbb{Q}$ are $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable)**.** *Let $S = [0, s-1]$, for some positive integer $s > 1$, and let $\mathcal{P}_\lambda, q_{\min}$ and $q_{\max}$ be as above. Let $\{\mathcal{C}_{\lambda,\mathsf{JEA}} \mid \lambda \geq 1\}$ be a family of linear integral codes over $\mathbb{Q}$, where each $\mathcal{C}_{\lambda,\mathsf{JEA}}$ is an integral JEA code over $\mathbb{Q}$, with corresponding generator matrices M_{BP} and M_{Ber} as in Definition 6.1. Then, for $0 < \mathsf{dist}_0 < \frac{1}{2(5\cdot 2^5\cdot e)^2}$, $\gamma \geq 1, c^* \geq 5$ and $t = \gamma\log(\mathsf{n})$ and $p = t/\mathsf{n}$ the family $\{\mathcal{C}_{\lambda,\mathsf{JEA}} \mid \lambda \geq 1\}$ is a $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable family of linear codes over $\mathbb{Q}$ with error*

$$\varepsilon_{\mathsf{proj}} \leq 2\gamma\cdot\log(\mathsf{n})\cdot\mathsf{dim}\cdot\left(\frac{\log(|S|)}{\log(q_{\min})\cdot|\mathcal{P}_\lambda|} + \frac{q_{\max}}{|S|\cdot|\mathcal{P}_\lambda|}\right) + 2\cdot\frac{d^2\log(|S|) + 2\log(d) + 1}{\log(q_{\min})\cdot|\mathcal{P}_\lambda|} + O(\mathsf{n}^{5-c^*}).$$

Using that $q_{\max} \leq q_{\min} + |\mathcal{P}_\lambda| \cdot \log_e(q_{\min})$, it is possible to express the above bound in terms of $q_{\min}$ and $|\mathcal{P}_\lambda|$.

Proof. Keeping notation and convention as in Definition 6.1, to analyze the q-projection $\mathcal{C}_{\lambda q}$ of $\mathcal{C}_{\lambda,\mathsf{JEA}}$ for any prime $q \in \mathcal{P}_\lambda$, we need to understand the matrices $\phi_q(M_{\mathsf{BP}})$ and $\phi_q(M_{\mathsf{Ber}})$. Clearly, the accumulator matrix A satisfies $\phi_q(A) = A$, for all $q \in \mathcal{P}_\lambda$. Since ϕ_q is a homomorphism, the above reduces to understanding $\phi_q(E_1)$ and $\phi_q(E_2)$.

For each $q \in \mathcal{P}_\lambda$, we call a subset $S_q \subset S$ a q-block in S if $|S_q| = q$ and S_q contains a full set of residue classes mod q, i.e. if $S_q = [k \cdot q, \ldots, (k+1) \cdot q - 1]$ for some $k < q$. Let $k_q = \lfloor \frac{s-1}{q} \rfloor$. Since S is an integral interval, it could be split into k_q full q-blocks and a partial block L_q. More precisely, $L_q = [k_q \cdot q, s-1]$. Therefore, by definition $|L_q| < q$.

Define the following events:

- $\mathcal{E}_{\mathsf{mul},x}$: a fixed $x \in S$ is divisible by a sampled prime from $\mathcal{P}_\lambda$,
- $\mathcal{E}_{\mathsf{mul},E_1}$: a row of E_1 contains an entry divisible by a sampled prime from $\mathcal{P}_\lambda$,
- $\mathcal{E}_{\mathsf{mul},E_2}$: a nonzero entry of E_2 is divisible by a sampled prime from $\mathcal{P}_\lambda$,
- $\mathcal{E}_{L_q,x}$: a fixed $x \in S$ belongs to L_q, after $q \in \mathcal{P}_\lambda$ is sampled,
- $\mathcal{E}_{\mathsf{size},E_1}$: a row of E_1 contains an entry from L_q, after $q \in \mathcal{P}_\lambda$ is sampled
- $\mathcal{E}_{\mathsf{size},E_2}$: a nonzero entry of E_2 is in L_q, after $q \in \mathcal{P}_\lambda$ is sampled,
- $\mathcal{E}_{\mathsf{BP}}$: none of the entries of E_1 belong to L_q and when projecting by ϕ_q, the row weights of E_1 do not change, for a sampled q from $\mathcal{P}_\lambda$,
- $\mathcal{E}_{\mathsf{Ber}}$: none of the entries of E_2 are divisible by q and none of the entries belong to L_q, for a sampled q from $\mathcal{P}_\lambda$.

We proceed by analyzing the probabilities that the above events occur. All probabilities are calculated with respect to sampling a prime q in $\mathcal{P}_\lambda$. We start with $\Pr[\mathcal{E}_{\mathsf{mul},x}]$. Since any element of S has bit-size at most $\log(|S|)$, there are at most $d = \log(|S|)/\log(q_{\min})$ primes that can divide x. Then

$$\Pr[\mathcal{E}_{\mathsf{mul},x}] \leq \frac{d}{|\mathcal{P}_\lambda|} = \frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|}.$$

Next, consider $\mathcal{E}_{L_q,x}$. For all $q \in \mathcal{P}_\lambda$, we have $|L_q| < q$ by construction and $q \leq q_{\max}$ by definition of $q_{\max}$. Hence:

$$\Pr[\mathcal{E}_{L_q,x}] = \frac{|L_q|}{|S|} \cdot \frac{1}{|\mathcal{P}_\lambda|} \leq \frac{q}{|S| \cdot |\mathcal{P}_\lambda|} \leq \frac{q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|},$$

Let $E_{(1,i)}$ denote the i-th row of E_1 and $E_{(1,i,j)}$ its (i,j)-th entry. Recall that E_1 is constructed by sampling each row $E_{(1,i)}$ independently and uniformly at random from $S_{(n,t)} := \{\mathbf{x} \in S^n \mid \mathsf{wt}(\mathbf{x}) = t\}$. In particular, an element $\mathbf{x} \in S_{(n,t)}$ has the form $\mathbf{x} = (x_1, \ldots, x_n)$, where all $x_i \in S$, $x_i \neq 0$ for $i \in [t]$, with $[t] \subset [n]$, and $x_i = 0$ otherwise. Note that technically, the nonzero indices belong to a subset of $[n]$ of size t rather than necessarily $[t]$, but for our purposes, without loss of generality we can assume this subset is $[t]$. Let $\mathcal{E}_{\mathsf{mul},S_{(n,t)},\mathbf{x}}$ be the event that a prime q divides an entry x_i of $\mathbf{x} \in S_{(n,t)}$ for some $i \in [t]$. Since $x_i \in S$, we have

$$\Pr[\mathcal{E}_{\mathsf{mul},S_{(n,t)},\mathbf{x}}] = \Pr[\mathcal{E}_{\mathsf{mul},x_i} \text{ for some } i] \leq \sum_{i \in [t]} \Pr[\mathcal{E}_{\mathsf{mul},x_i}] \leq t \cdot \frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|}$$

Similarly, we define $\mathcal{E}_{L_q,\mathbf{x}}$ to be the event that $\mathbf{x} \in S_{(n,t)}$ defined as before, contains an entry from L_q. More precisely, $x_i \in L_q$, for some $i \in [t]$. Since $x_i \in S$, we get

$$\Pr[\mathcal{E}_{L_q,\mathbf{x}}] = \Pr[\mathcal{E}_{L_q,x_i} \text{ for some } i] \leq \sum_{i\in[t]} \Pr[\mathcal{E}_{L_q,x_i}] \leq t \cdot \frac{q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|}$$

By definition, rows $E_{(1,i)}$, $i \in [\mathsf{dim}]$ of E_1 are elements of $S_{(n,t)}$. Using $\Pr[\mathcal{E}_{\mathsf{mul},S_{(n,t)},\mathbf{x}}]$ we calculated above and $t = \gamma \log(\mathsf{n})$, we calculate:

$$\begin{aligned}\Pr[\mathcal{E}_{\mathsf{mul},E_1}] &= \Pr[\mathcal{E}_{\mathsf{mul},S_{(n,t)},E_{(1,i)}} \text{ for some } i] \leq \sum_{i\in[\mathsf{dim}]} \Pr[\mathcal{E}_{\mathsf{mul},S_{(n,t)},E_{(1,i)}}] \\ &\leq t \cdot \mathsf{dim} \cdot \frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|} = \frac{\gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot \log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|}.\end{aligned}$$

Similarly, using $\Pr[\mathcal{E}_{L_q,\mathbf{x}}]$ this time, we obtain:

$$\begin{aligned}\Pr[\mathcal{E}_{\mathsf{size},E_1}] &= \Pr[\mathcal{E}_{L_q,E_{(1,i)}} \text{ for some } i] \leq \sum_{i\in[\mathsf{dim}]} \Pr[\mathcal{E}_{L_q,E_{(1,i)}}] \\ &\leq t \cdot \mathsf{dim} \cdot \frac{q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|} = \frac{\gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|}\end{aligned}$$

By definition, $\mathcal{E}_{\mathsf{BP}}$ is the event that neither $\mathcal{E}_{\mathsf{size},E_1}$ nor $\mathcal{E}_{\mathsf{mul},E_1}$ occur. Then,

$$\begin{aligned}\Pr[\neg\mathcal{E}_{\mathsf{BP}}] &= \Pr[\mathcal{E}_{\mathsf{mul},E_1} \vee \mathcal{E}_{\mathsf{size},E_1}] \leq \Pr[\mathcal{E}_{\mathsf{mul},E_1}] + \Pr[\mathcal{E}_{\mathsf{size},E_1}] \\ &\leq \gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot \left(\frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|} + \frac{q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|} \right)\end{aligned}$$

We now analyze the expander matrix E_2. Recall that it is sampled by the generalized Bernoulli distribution: i.e. entries are sampled independently at random from S, with the probability of an entry being nonzero equal to $p = t/\mathsf{n}$. Let $E_{(2,i,j)}$ denote the (i,j)-the entry of E_2. We next calculate the probabilities of the events $\mathcal{E}_{\mathsf{size},E_2}$ and $\mathcal{E}_{\mathsf{mul},E_2}$. For both calculations we use $p = t/\mathsf{n}$ and $t = \gamma \log(\mathsf{n})$:

$$\begin{aligned}\Pr[\mathcal{E}_{\mathsf{size},E_2}] &= \Pr[E_{(2,i,j)} \neq 0 \wedge E_{(2,i,j)} \in L_q \text{ for some } i \text{ and } j] \\ &\leq \sum_{\substack{i\in[\mathsf{dim}] \\ j\in[\mathsf{n}]}} \Pr[E_{(2,i,j)} \neq 0 \wedge \mathcal{E}_{L_q,E_{(2,i,j)}}] \leq \mathsf{n} \cdot \mathsf{dim} \cdot p \cdot \frac{q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|} \\ &= \frac{\gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot q_{\max}}{|S| \cdot |\mathcal{P}_\lambda|}\end{aligned}$$

$$\begin{aligned}\Pr[\mathcal{E}_{\mathsf{mul},E_2}] &\leq \sum_{i\in[\mathsf{dim}],j\in[\mathsf{n}]} \Pr[E_{(2,i,j)} \neq 0 \wedge \mathcal{E}_{\mathsf{mul},E_{(2,i,j)}} \text{ for some } i \text{ and } j] \\ &\leq \mathsf{n} \cdot \mathsf{dim} \cdot p \cdot \frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|} = \frac{\gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot \log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|}\end{aligned}$$

Finally, we look at the probability of $\mathcal{E}_{\mathsf{Ber}}$ not occurring:

$$\begin{aligned}\Pr[\neg\mathcal{E}_{\mathsf{Ber}}] &= \Pr[\mathcal{E}_{\mathsf{size},E_2} \vee \mathcal{E}_{\mathsf{mul},E_2}] \leq \Pr[\mathcal{E}_{\mathsf{size},E_2}] + \Pr[\mathcal{E}_{\mathsf{mul},E_2}] \\ &\leq \gamma \log(\mathsf{n}) \cdot \mathsf{dim} \cdot \left(\frac{\log(|S|)}{\log(q_{\min}) \cdot |\mathcal{P}_\lambda|} + \frac{q_{\max}}{|S|} \right)\end{aligned}$$

Notice that $\Pr[\neg\mathcal{E}_{\mathsf{Ber}}] = \Pr[\neg\mathcal{E}_{\mathsf{BP}}]$. Call this error $\varepsilon_{\mathsf{sample}}$.

For any $q \in \mathcal{P}_\lambda$, by construction $S \setminus L_q$ maps to k_q copies of the full set of residue classes modulo q after projecting by ϕ_q. Therefore, each $y \in \mathbb{F}_q$ has k_q distinct ϕ_q-preimages in $S \setminus L_q$. Call these $x_{y,1}, \ldots, x_{y,k_q}$. Since S is an integral interval and residue classes of a prime q are uniformly distributed over $\mathbb{Z}$, the probability of picking y uniformly at random in $\mathbb{F}_q$ is the same as the probability of picking a preimage $x_{y,i}$ in $S \setminus L_q$. Therefore, when events $\mathcal{E}_{L_q}$ occur, sampling uniformly at random in $\mathbb{F}_q$ is the same as sampling uniformly at random from the set $\phi_q(S \setminus L_q)$.

Let $\mathcal{E}_{\mathsf{BP},\phi_q}$ be the event that for any sampled prime $q \in \mathcal{P}_\lambda$ and a matrix E_1 sampled by $\mathsf{BP}(\mathsf{dim}, \mathsf{n}, t, S)$, there is a matrix M_1 with entries in $\mathbb{F}_q$, sampled by $\mathsf{BP}(\mathsf{dim}, \mathsf{n}, t, \mathbb{F}_q)$ such that $\phi_q(E_1) = M_1$ ($\mathsf{BP}(\mathsf{dim}, \mathsf{n}, t, \mathbb{F}_q)$ is defined in the same way as $\mathsf{BP}(\mathsf{dim}, \mathsf{n}, t, S)$ with sampling set $\mathbb{F}_q$ instead of S). Similarly, let $\mathcal{E}_{\mathsf{Ber},\phi_q}$ be the event that for any sampled prime $q \in \mathcal{P}_\lambda$ and a matrix E_2 sampled by $\mathsf{Ber}_q(S)$, there is a matrix M_2 with entries in $\mathbb{F}_q$, sampled by $\mathsf{Ber}_q(\mathbb{F}_q)$ such that $\phi_q(E_2) = M_2$ ($\mathsf{Ber}_p(\mathbb{F}_q)$ is defined in the same way as $\mathsf{Ber}_p(S)$ with sampling set $\mathbb{F}_q$ instead of S). We claim that

$$\Pr[\mathcal{E}_{\mathsf{BP},\phi_q} \mid \mathcal{E}_{\mathsf{BP}}] = 1 \quad \text{and} \quad \Pr[\mathcal{E}_{\mathsf{Ber},\phi_q} \mid \mathcal{E}_{\mathsf{Ber}}] = 1$$

Since $\mathcal{E}_{\mathsf{size},E_1}$ and $\mathcal{E}_{\mathsf{size},E_2}$ do not occur, no entries of E_1 and E_2 respectively come from L_q. Therefore, uniform sampling in S and projecting by ϕ_q is the same as uniform sampling from $\phi_q(S)$ which in turn corresponds to uniform sampling in $\mathbb{F}_q$. On the other hand, the fact that $\mathcal{E}_{\mathsf{mul},E_1}$ and $\mathcal{E}_{\mathsf{mul},E_2}$ do not occur guarantees that our sampled q does not divide entries of E_1 and E_2 respectively. This ensures that row weights of E_1 remain unchanged, hence $\phi_q(E_{(1,i)}) \in \mathbb{F}_{q,(n,t)}$ for all $i \in [\mathsf{dim}]$. On E_2's side, $\neg\mathcal{E}_{\mathsf{mul},E_2}$ guarantees that the sparsity of the matrix remains unchanged. This means that if events $\mathcal{E}_{\mathsf{BP}}$ and $\mathcal{E}_{\mathsf{Ber}}$ occur, $\phi_q(E_1)$ and $\phi_q(E_2)$ correspond to the expander matrices E_1 and E_2 over $\mathbb{F}_q$ as defined in [BFK+24].

Finally, by Lemma 6.2, $\mathsf{rk}(\phi_q(M_{\mathsf{BP}})) = \mathsf{rk}(M_{\mathsf{BP}})$ and $\mathsf{rk}(\phi_q(M_{\mathsf{Ber}})) = \mathsf{rk}(M_{\mathsf{Ber}})$, i.e. $\mathsf{dim}_q = \mathsf{dim}$, for all but at most

$$2 \cdot \frac{d^2 \log(|S|) + 2\log(d) + 1}{\log(q_{\min})}$$

primes in $\mathcal{P}_\lambda$. Then the probability of sampling such a prime is

$$\varepsilon_{\mathsf{rk}} = 2 \cdot \frac{d^2 \log(|S|) + 2\log(d) + 1}{\log(q_{\min})|\mathcal{P}_\lambda|}.$$

Hence, by [BFK+24, Theorem 4.1], there exist constants $\mathsf{dist}_0 \leq \frac{1}{2(5 \cdot 2^5 \cdot e)^2}$ and $c^* \geq 5$ such that

$$\Pr_{\substack{E_1 \leftarrow \mathsf{BP}(\mathsf{dim},\mathsf{n},t,\mathbb{F}_q) \\ E_2 \leftarrow \mathsf{Ber}_p(\mathbb{F}_q)}}[\mathsf{dist}_q \geq \mathsf{dist}_0] \geq 1 - O(\mathsf{n}^{5-c^*}).$$

This gives our final error $\varepsilon_{\mathsf{dist}_0} = O(\mathsf{n}^{5-c^*})$. In other words, the integral JEA $\mathcal{C}_{\lambda,\mathsf{JEA}}$ codes over $\mathbb{Q}$ are $(\mathcal{P}_\lambda, \mathsf{dist}_0)$-projectable with error $\varepsilon_{\mathsf{proj}} \leq 2 \cdot \varepsilon_{\mathsf{sample}} + \varepsilon_{\mathsf{rk}} + \varepsilon_{\mathsf{dist}_0}$. $\square$

Remark 6.4 (On the error $O(\mathsf{n}^{5-c^*})$)**.** To be more concrete in the probability bound above, we take a closer look at the error $O(\mathsf{n}^{5-c^*})$. Here, we can directly rely on [BFK+24, Section 6]. In this section, the authors argue how to achieve a provable distance of $\mathsf{dist}_0 = 0.1$ with a failure probability of 2^{-100}. The parameters for this probability are obtained by numerically analyzing the low- and high-weight message cases, which are used to derive the final error of $O(\mathsf{n}^{5-c^*})$.

For the low-weight message case, they obtain the bound

$$\Pr[\exists \mathbf{x} \in \mathbb{F}_{q^{\mathsf{n}}} \setminus \{0^{\mathsf{n}}\} : \mathsf{wt}(\mathbf{x}E_1A) \leq \mathsf{dist}_0 N \wedge \mathsf{wt}(\mathbf{x}) \leq \beta\mathsf{n}/t] \\ \leq \sum_{r=1}^{\beta\mathsf{n}/t} \binom{n}{r} \sum_{w=rct}^{2\mathsf{dist}_0 N} \sum_{h=\lceil w/2 \rceil}^{\mathsf{dist}_0 N} \sum_{i=0}^{w-1} \frac{\binom{N-h}{\lfloor \frac{w-i}{2} \rfloor}\binom{h-1}{\frac{w-i}{2}-1}\binom{h-\lceil \frac{w-i}{2} \rceil}{i}}{\binom{N}{w}} ,$$

and for the high-weight message case the bound

$$\Pr[\exists \mathbf{x} \in \mathbb{F}_{q^{\mathsf{n}}} \setminus \{0^{\mathsf{n}}\} : \mathsf{wt}(\mathbf{x}E_2A) \leq \mathsf{dist}_0 N \wedge \mathsf{wt}(\mathbf{x}) \geq \beta\mathsf{n}/t] \\ \leq \mathsf{negl}(N) + \sum_{r=\beta\mathsf{n}/t}^{\mathsf{n}} \binom{\mathsf{n}}{r} \sum_{w=rct}^{2\mathsf{dist}_0 N} \sum_{h=\lceil w/2 \rceil}^{\mathsf{dist}_0 N} \sum_{i=0}^{w-1} \frac{\binom{N-h}{\lfloor \frac{w-i}{2} \rfloor}\binom{h-1}{\lceil \frac{w-i}{2} \rceil-1}\binom{h-\lceil \frac{w-i}{i} \rceil}{i}}{\binom{N}{w}} .$$

The authors of [BFK+24], estimate these parameters to be $\mathsf{n} = 1024, N = 2048, \beta = 2, \varepsilon = 0.1$ and $\gamma \geq 18$, where ε is a parameter of the expander E_1 for the low-weight case and $c = (1-2\varepsilon)$. We can rely on the same parameters in our work.

7 References

[ACC+22] Thomas Attema, Ignacio Cascudo, Ronald Cramer, Ivan Damgard, and Daniel Escudero. Vector commitments over rings and compressed σ-protocols. In *Theory of Cryptography Conference*, pages 173–202. Springer, 2022.

[AHIV22] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. Cryptology ePrint Archive, Paper 2022/1608, 2022. URL: https://eprint.iacr.org/2022/1608, doi:10.1145/3133956.

[BBMH+21] Carsten Baum, Lennart Braun, Alexander Munch-Hansen, Benoit Razet, and Peter Scholl. Appenzeller to brie: efficient zero-knowledge proofs for mixed-mode arithmetic and z2k. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 192–211, 2021.

[BBMHS22] Carsten Baum, Lennart Braun, Alexander Munch-Hansen, and Peter Scholl. Moz z 2 k arella: efficient vector-ole and zero-knowledge proofs over z 2 k. In *Annual International Cryptology Conference*, pages 329–358. Springer, 2022.

[BCG+17] Jonathan Bootle, Andrea Cerulli, Essam Ghadafi, Jens Groth, Mohammad Hajiabadi, and Sune K. Jakobsen. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 336–365. Springer, Cham, December 2017. doi:10.1007/978-3-319-70700-6_12.

[BCG20] Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part II*, volume 12551 of *LNCS*, pages 19–46. Springer, Cham, November 2020. doi:10.1007/978-3-030-64378-2_2.

[BCS21] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. Sumcheck arguments and their applications. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part I 41*, pages 742–773. Springer, 2021.

[BFK+24] Alexander R. Block, Zhiyong Fang, Jonathan Katz, Justin Thaler, Hendrik Waldner, and Yupeng Zhang. Field-agnostic snarks from expand-accumulate codes. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*, volume 14929 of *Lecture Notes in Computer Science*, pages 276–307. Springer, 2024. doi:10.1007/978-3-031-68403-6_9.

[BFS19] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from dark compilers. Cryptology ePrint Archive, Paper 2019/1229, 2019. URL: https://eprint.iacr.org/2019/1229.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent SNARKs from DARK compilers. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 677–706. Springer, Cham, May 2020. doi:10.1007/978-3-030-45721-1_24.

[BG23] Jeremy Bruestle and Paul Gafni. Risc zero zkvm: scalable, transparent arguments of risc-v integrity. *Draft. July*, 29, 2023.

[BHR+21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. Time- and space-efficient arguments from groups of unknown order. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 123–152, Virtual Event, August 2021. Springer, Cham. doi:10.1007/978-3-030-84259-8_5.

[BS23] Ward Beullens and Gregor Seiler. Labrador: compact proofs for r1cs from module-sis. In *Annual International Cryptology Conference*, pages 518–548. Springer, 2023.

[BSBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon Interactive Oracle Proofs of Proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:17, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2018.14, doi:10.4230/LIPIcs.ICALP.2018.14.

[BSCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed-solomon codes. 70(5), 2023. doi:10.1145/3614423.

[CBBZ22] Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. Hyperplonk: Plonk with linear-time prover and high-degree custom gates. Cryptology ePrint Archive, Paper 2022/1355, 2022. URL: https://eprint.iacr.org/2022/1355.

[CCKP19] Shuo Chen, Jung Hee Cheon, Dongwoo Kim, and Daejun Park. Verifiable computing for approximate computation. *Cryptology ePrint Archive*, 2019.

[CDGS23] Alessandro Chiesa, Marcel Dall'Agnol, Ziyi Guan, and Nicholas Spooner. On the security of succinct interactive arguments from vector commitments. Cryptology ePrint Archive, Report 2023/1737, 2023. URL: https://eprint.iacr.org/2023/1737.

[CHA24] Matteo Campanelli and Mathias Hall-Andersen. Fully-succinct arguments over the integers from first principles. Cryptology ePrint Archive, Paper 2024/1548, 2024. URL: https://eprint.iacr.org/2024/1548.

[CHM+19] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. Cryptology ePrint Archive, Paper 2019/1047, 2019. https://eprint.iacr.org/2019/1047. URL: https://eprint.iacr.org/2019/1047.

[COEG+20] Amin Coja-Oghlan, Alperen A Ergür, Pu Gao, Samuel Hetterich, and Maurice Rolvien. The rank of sparse random matrices. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 579–591. SIAM, 2020.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part I 39*, pages 769–793. Springer, 2020.

[CY24] Alessandro Chiesa and Eylon Yogev. *Building Cryptographic Proofs from Hash Functions*. 2024. URL: https://github.com/hash-based-snargs-book.

[DP23] 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.

[DP24] Benjamin E Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. *Cryptology ePrint Archive*, 2024.

[Eli] Ben-Sasson Eli. State of stwo. URL: https://www.youtube.com/watch?v=Gs2HwJIgdxc.

[Fou] Ethereum Foundation. Zk grants round announcement. URL: https://blog.ethereum.org/2024/06/25/zk-grants-round-announce.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986. doi:10.1007/3-540-47721-7_12.

[GK96] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(3):167–190, June 1996. doi:10.1007/s001459900010.

[GLS$^+$23] Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*, volume 14082 of *Lecture Notes in Computer Science*, pages 193–226. Springer, 2023. doi:10.1007/978-3-031-38545-2_7.

[GNSV23] Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: Snarks for ring arithmetic. *Journal of Cryptology*, 36(4):41, 2023.

[HLP24] Ulrich Haböck, David Levit, and Shahar Papini. Circle STARKs. Cryptology ePrint Archive, Paper 2024/278, 2024. URL: https://eprint.iacr.org/2024/278.

[HMZ25] Mi-Ying Miryam Huang, Xinyu Mao, and Jiapeng Zhang. Sublinear proofs over polynomial rings. Cryptology ePrint Archive, Paper 2025/199, 2025. URL: https://eprint.iacr.org/2025/199.

[KS23] Abhiram Kothapalli and Srinath Setty. Hypernova: Recursive arguments for customizable constraint systems. Cryptology ePrint Archive, Paper 2023/573, 2023. https://eprint.iacr.org/2023/573. URL: https://eprint.iacr.org/2023/573.

[Lin01] Yehuda Lindell. Parallel coin-tossing and constant-round secure two-party computation. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 171–189. Springer, Berlin, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_10.

[LXY24] Fuchun Lin, Chaoping Xing, and Yizhou Yao. More efficient zero-knowledge protocols over $\mathbb{Z}_{2^k}$ via galois rings. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part IX*, volume 14928 of *LNCS*, pages 424–457. Springer, Cham, August 2024. doi:10.1007/978-3-031-68400-5_13.

[OKMZ24] Michele Orrù, George Kadianakis, Mary Maller, and Greg Zaverucha. Beyond the circuit: How to minimize foreign arithmetic in zkp circuits. *Cryptology ePrint Archive*, 2024.

[PH23] Shahar Papini and Ulrich Haböck. Improving logarithmic derivative lookups using gkr. Cryptology ePrint Archive, Paper 2023/1284, 2023. https://eprint.iacr.org/2023/1284. URL: https://eprint.iacr.org/2023/1284.

[Plo] Plonky3. Plonky3. URL: https://github.com/Plonky3/Plonky3.

[Set19] Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. Cryptology ePrint Archive, Paper 2019/550, 2019. https://eprint.iacr.org/2019/550. URL: https://eprint.iacr.org/2019/550.

[STW23a] Srinath Setty, Justin Thaler, and Riad Wahby. Customizable constraint systems for succinct arguments. 2023. URL: https://eprint.iacr.org/2023/552.

[STW23b] Srinath Setty, Justin Thaler, and Riad Wahby. Unlocking the lookup singularity with lasso. Cryptology ePrint Archive, Paper 2023/1216, 2023. https://eprint.iacr.org/2023/1216. URL: https://eprint.iacr.org/2023/1216.

[SV22] Eduardo Soria-Vazquez. Doubly efficient interactive proofs over infinite and non-commutative rings. In *Theory of Cryptography Conference*, pages 497–525. Springer, 2022.

[Tha22] Justin Thaler. Proofs, arguments, and zero-knowledge. *Foundations and Trends® in Privacy and Security*, 4(2–4):117–660, 2022.

[VJS21] Ashwin Sah Vishesh Jain and Mehtaab Sawhney. Singularity of discrete random matrices. *Geometric and Functional Analysis*, 31:1160–1218, 2021. doi:10.1007/s00039-021-00580-6.

[WZD25] Yuanju Wei, Xinxuan Zhang, and Yi Deng. Transparent SNARKs over galois rings. Cryptology ePrint Archive, Paper 2025/263, 2025. URL: https://eprint.iacr.org/2025/263.

[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.

A Correlated agreement for linear codes over infinite fields

In this section we prove the correlated agreement result from the Ligero paper [AHIV22] in the case of $\delta < \mathsf{dist}/3$ but for infinite fields. The proof is essentially the same with minor straightforward modifications. Recall that we write Δ for the relative Hamming distance in $\mathcal{C}$, $\Delta(\mathbf{v}, \mathcal{C})$ for the relative Hamming distance between $\mathbf{v} \in \mathbb{K}^{\mathsf{n}}$ and the closest codeword in $\mathcal{C}$, and dist for the relative distance of $\mathcal{C}$. Additionally, for the purpose of this section, we define $E(\mathbf{v}, \mathcal{C})$ to be the set of positions where $\mathbf{v}$ differs from the closest codeword in $\mathcal{C}$. The goal is to prove the following statement:

Lemma A.1 (Correlated agreement for linear codes over infinite fields, c.f. [AHIV22, Lemma 4.5])**.** *Let $\mathcal{C}$ be a linear code over a field $\mathbb{K}$ with dimension dim, length n and relative distance dist. Let $K \subseteq \mathbb{K}$ be a finite nonempty subset of $\mathbb{K}$. Then $\mathcal{C}$ has (δ, α, K)-correlated agreement as defined in 3.1 with $\delta < \mathsf{dist}/3$, $\alpha = \mathsf{n}/|K|$.*

Following [AHIV22], we first need two preliminary results for linear codes over infinite fields.

Lemma A.2 ([AHIV22, Lemma 4.3, Claim A.1])**.** *Let $\mathcal{C}$ be a linear code over $\mathbb{K}$ of length n and minimum distance dist, and let $\delta < \mathsf{dist}/3$. Let K be a finite subset of $\mathbb{K}$ with $|K| > \delta \cdot \mathsf{n}$. Let $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{K}^n$ and codewords $\mathsf{c}_1, \dots, \mathsf{c}_k$ be such that each $\mathbf{v}_i$ disagrees with c_i on at least $\delta \cdot \mathsf{n}$ positions. Then there exists a vector $\mathbf{v}^*$ of the form $\mathbf{v}^* = \sum_{i \in [k]} r_i \cdot \mathbf{v}_i$, for $r_i \in K$, such that $\Delta(\mathbf{v}^*, \mathcal{C}) > \delta$.*

Proof. The proof follows the proof of [AHIV22, Lemma 4.3, Claim A.1]. Let V_K^k be the K-span of the vectors $\{\mathbf{v}_1, \dots, \mathbf{v}_k\}$, i.e. $V_K^k = \{\sum_{i \in [k]} r_i \cdot \mathbf{v}_i \mid r_i \in K, i \in [k]\}$. Suppose $\Delta(\mathbf{v}^*, \mathcal{C}) \le \delta$ for all $\mathbf{v}^* \in V_K^k$. Suppose $\mathbf{v}_0^* \in V_K^k$ maximizes the distance from $\mathcal{C}$. Since for all $i \in [k]$, $\mathbf{v}_i$ disagree with codewords c_i on at least $\delta \cdot \mathsf{n}$ positions, $E(\mathbf{v}_i, \mathcal{C}) \setminus E(\mathbf{v}_0^*, \mathcal{C}) \neq \emptyset$. Let $\mathbf{v}_0^* = \mathbf{u}_0 + \mathbf{x}_0$ and $\mathbf{v}_i = \mathbf{u}_i + \mathbf{x}_i$, for $\mathbf{u}_0, \mathbf{u}_i \in \mathcal{C}$ and $\mathbf{x}_0, \mathbf{x}_i \in V_{\mathbb{K}}^k$ such that $\mathsf{wt}(\mathbf{x}_0), \mathsf{wt}(\mathbf{x}_i) \le \delta \cdot \mathsf{n}$. We claim there is $a \in K$, such that for $\hat{\mathbf{v}} = \mathbf{v}_0^* + a\mathbf{v}_i$, we have $\Delta(\hat{\mathbf{v}}, \mathcal{C}) > \Delta(\mathbf{v}_0^*, \mathcal{C})$, contradicting the choice of $\mathbf{v}_0^*$. This follows by a union bound, noting that for any $j \in E(\mathbf{v}_0^*, \mathcal{C}) \cup E(\mathbf{v}_i, \mathcal{C})$ there is at most one choice of $a \in K$, such that the j-th coordinate of $\hat{\mathbf{v}}$ is zero. □

Lemma A.3 ([AHIV22, Claim A.2])**.** *Let $\mathcal{C}$ be an arbitrary linear code over $\mathbb{K}$ of length n and minimum distance dist. Let K be a finite subset of $\mathbb{K}$, such that $|K| > \mathsf{n}$. Let $\delta < \mathsf{dist}/3$. Then, for every $\mathbf{u}, \mathbf{v} \in \mathbb{K}^{\mathsf{n}}$ defining an affine line $l_{\mathbf{u},\mathbf{v}} = \{\mathbf{u} + a\mathbf{v} : a \in K\}$, either:*

1. *For every $\mathbf{x} \in l_{\mathbf{u},\mathbf{v}}$, we have $\Delta(\mathbf{x}, \mathcal{C}) \le \delta$, or*
2. *For at most n points $\mathbf{x} \in l_{\mathbf{u},\mathbf{v}}$, we have $\Delta(\mathbf{x}, \mathcal{C}) \le \delta$.*

Proof. For any two length n vectors $\mathbf{u}$ and $\mathbf{v}$ of weight at most $\delta \cdot \mathsf{n}$, the affine line $l_{\mathbf{u},\mathbf{v}}$ contains N points of distance at most $\delta \cdot \mathsf{n}$ from $\mathcal{C}$ if and only if $l_{\mathbf{u},\mathbf{v}+\mathsf{c}}$ contains N points of distance at most $\delta \cdot \mathsf{n}$ from $\mathcal{C}$ for any codeword $\mathsf{c} \in \mathcal{C}$. Hence, it suffices to prove the claim for vectors $\mathbf{u}$ and $\mathbf{v}$ of weight at most $\delta \cdot \mathsf{n}$. Let $\sup(\mathbf{u})$ and $\sup(\mathbf{v})$ denote the indexing sets of non-zero positions of $\mathbf{u}$ and $\mathbf{v}$ respectively. We consider two cases:

Case 1: $|\sup(\mathbf{u}) \cup \sup(\mathbf{v})| < \delta \cdot \mathsf{n}$. Hence, $l_{\mathbf{u},\mathbf{v}}$ is entirely contained in the ball $B_{\delta \cdot \mathsf{n}}(0)$ of radius $\delta \cdot \mathsf{n}$ around 0, i.e. $B_{\delta \cdot \mathsf{n}}(0) = \{\mathbf{x} \in \mathbb{K}^{\mathsf{n}} \mid \Delta(\mathbf{x}, 0) \le \delta \cdot \mathsf{n}\}$, where 0 is the all 0s vector in $\mathbb{K}^{\mathsf{n}}$. This proves part 1 of the statement.

Case 2: $|\sup(\mathbf{u}) \cup \sup(\mathbf{v})| \ge \delta \cdot \mathsf{n}$. Since $\mathsf{wt}(\mathbf{u}), \mathsf{wt}(\mathbf{v}) \le \delta \cdot \mathsf{n}$, then $|\sup(\mathbf{u}) \cap \sup(\mathbf{v})| \le \delta \cdot \mathsf{n} - 1$. For each of the coordinates in the intersection of the supports, there can be at most one vector in $l_{\mathbf{u},\mathbf{v}}$ such that the entry in that coordinate is 0. Therefore, there are at most

$$\delta \cdot \mathsf{n} - 1 < \frac{\mathsf{dist} \cdot \mathsf{n}}{3} - 1 < \frac{\mathsf{n} - \mathsf{dim}}{3} - 1 < \mathsf{n}$$

vectors in $l_{\mathbf{u},\mathbf{v}}$ contained in the ball $B_{\delta \cdot \mathsf{n}}(0)$, where 0 is the all 0s vector in $\mathbb{K}^{\mathsf{n}}$. Suppose there exists a nonzero codeword $\mathsf{c} \in \mathcal{C}$ such that $l_{\mathbf{u},\mathbf{v}}$ intersects the ball $B_{\delta \cdot \mathsf{n}}(\mathsf{c})$. Then, there is a vector $\mathbf{w} \in \mathbb{K}^{\mathsf{n}}$ such that $\mathsf{wt}(\mathbf{w}) \leq \delta \cdot \mathsf{n}$ and $\mathsf{c} + \mathbf{w} = \mathbf{u} + a\mathbf{v}$, for some $a \in K$. Hence, c is equal to the sum of three vectors each of weight at most $\delta \cdot \mathsf{n}$ which is strictly less than the minimum distance in $\mathcal{C}$, leading to a contradiction since $\mathsf{c} \in \mathcal{C}$. $\square$

Proof. (of Lemma A.1) To establish the (δ, α, K)-correlated agreement, we will prove the counterpositive: Let $\mathbf{v}_1, \ldots, \mathbf{v}_k$ be any words that agree on at most $\delta \cdot \mathsf{n}$ positions. Then

$$\Pr\left[\Delta\left(\sum_{i \in [k]} r_i \cdot \mathbf{v}_i, \mathcal{C}\right) \leq \delta \;\middle|\; r_i \leftarrow K, i \in [k]\right] < \alpha.$$

Let U be the $k \times \mathsf{dim}$-matrix with i-th row $\mathbf{v}_i, i \in [k]$ and $\delta < \mathsf{dist}/3$ and $\alpha = \mathsf{n}/|K|$. The agreement of $\mathbf{v}_1, \ldots, \mathbf{v}_k$ on at most $\delta \cdot \mathsf{n}$ positions is equivalent to $\Delta(U, \mathcal{C}^k) > \delta$, where $\mathcal{C}^k$ is the k-interleaved code of $\mathcal{C}$.

Let $V^k_{\mathbb{K}}$ and V^k_K denote the $\mathbb{K}$-span and K-span of $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ respectively (see the beginning of this proof for an explanation of this terminology). Note that since K is not closed under addition or scalar multiplication V^k_K is not a vector subspace of $\mathbb{K}^{\mathsf{n}}$. The proof closely follows [AHIV22, Lemma 4.5]. We consider two cases.

Case 1: (cf. [AHIV22, Case 1, Lemma 4.2]) Suppose there exists a vector $\mathbf{v}^* \in V^k_K$ such that $\Delta(\mathbf{v}^*, \mathcal{C}) > 2\delta$. We can write any $\mathbf{w}^* \in V^k_K$ as $\mathbf{w}^* = a\mathbf{v}^* + \mathbf{x}$, where $a \in K, \mathbf{x} \in V^k_{\mathbb{K}}$. We argue that conditioned on any choice of $\mathbf{x}$, there can be at most one choice of a such that $\Delta(a\mathbf{v}^* + x, \mathcal{C}) \leq \delta$. Suppose there exist $a_1, a_2 \in K$ with $a_1 \neq a_2$, such that $\Delta(a_1\mathbf{v}^* + x, \mathcal{C}) \leq \delta$ and $\Delta(a_2\mathbf{v}^* + x, \mathcal{C}) \leq \delta$. By the triangle inequality $\Delta((a_2 - a_1)\mathbf{v}^*, \mathcal{C}) \leq 2\delta$. If $a_2 - a_1 \notin K$, our claim follows. If $a_2 - a_1 \in K$, since $a_2 - a_1 \neq 0$, $\Delta((a_2 - a_1)\mathbf{v}^*, \mathcal{C}) = \Delta(\mathbf{v}^*, \mathcal{C}) \leq 2\delta$ contradicting our assumption $\Delta(\mathbf{v}^*, \mathcal{C}) > 2\delta$.

Note that the choice of $\mathbf{x}$ is conditioned on a. Since $a \in K$, there are exactly $|K|$ options for choosing it. Therefore,

$$\Pr[\Delta(\mathbf{w}^*, \mathcal{C}) < \delta \mid \mathbf{w} \leftarrow V^k_K] \leq \frac{1}{|K|}.$$

Case 2: For every $\mathbf{v}^* \in V^k_K$, we have $\Delta(\mathbf{v}^*, \mathcal{C}) \leq 2\delta$. Since $\Delta(U, \mathcal{C}^k) > \delta$, by Lemma A.2 there exists $\mathbf{v}^* \in V^k_K$ such that $\Delta(\mathbf{v}^*, \mathcal{C}) > \delta$. As in Case 1, we write points in V^k_K as $\mathbf{x} + a\mathbf{v}^*$, where $a \in K$ and $\mathbf{x} \in V^k_K$. For any fixed $\mathbf{x}$, we there exists an a such that $\Delta(\mathbf{x} + a\mathbf{v}^*, \mathcal{C}) > \delta$. By Lemma A.3 there are at most n values for a such that $\Delta(\mathbf{x} + a\mathbf{v}^*, \mathcal{C}) \leq \delta$. Since this is true for each $\mathbf{x}$, it is true for the entire V^k_K. Therefore,

$$\Pr\left[\Delta\left(\sum_{i \in [k]} r_i \cdot \mathbf{v}_i, \mathcal{C}\right) \leq \delta \;\middle|\; r_i \leftarrow K, i \in [k]\right] \leq \frac{\mathsf{n}}{|K|},$$

finishing the proof. $\square$