

Rotor: SNARKs for Power-of-Two Rings

Chongrong Li ^{*} Yun Li^(✉) [†] Pengfei Zhu [‡] Zhechen Li [§] Michael Dong [¶]
Alan Li ^{||} Yuncong Hu^(✉) ^{**}

Abstract

Succinct proofs for arithmetic modulo powers of two are crucial for verifying machine-word computations in practice. However, designing SNARKs over such rings is challenging, since they lack the algebraic properties that common SNARKs over fields rely on to achieve negligible soundness error. Existing approaches either work over Galois rings, whose arithmetic is expensive, or lift the statement to the integers and prove it over a large random prime field, which is also several times slower than the structured primes used in practice.

We present Rotor, an efficient SNARK for circuits over $\mathbb{Z}_{2^k}$. Unlike prior work, Rotor operates natively over power-of-two rings and derives its soundness from three levels of ring precision. To prove statements over $\mathbb{Z}_{2^k}$, we first construct a sumcheck-style interactive oracle proof with tensor queries over a larger ring $\mathbb{Z}_{2^m}$. We then establish a weakened proximity gap for linear codes, in which proximity testing over a still larger ring $\mathbb{Z}_{2^\ell}$ ensures that the underlying vectors are close to the code after reduction to $\mathbb{Z}_{2^m}$. This yields a Brakedown-style commitment with tensor evaluation proofs over $\mathbb{Z}_{2^m}$. In both cases, a sufficiently large gap between the precisions guarantees negligible soundness error. Finally, we introduce compatible commitments, which link ring witnesses to their binary-field representations and enable permutation checks over fields to verify permutations over $\mathbb{Z}_{2^k}$, completing the construction of Rotor.

1 Introduction

Succinct non-interactive arguments of knowledge (SNARKs) allow a prover to certify a large computation with a short proof that is far cheaper to verify than rerunning the computation. Most efficient SNARKs express computations over finite fields, whose algebraic properties make them a convenient setting for proof systems.

Machine-word programs, however, naturally compute modulo a power of two, that is, over $\mathbb{Z}_{2^k} = \mathbb{Z}/2^k\mathbb{Z}$ with $k = 32$ or 64 on common architectures. Expressing these ring operations over a field requires additional constraints for ranges, lookups, carries, wraparound, and reduction modulo 2^k , so the circuit to be proved can be substantially larger than the computation itself. A proof system designed for power-of-two arithmetic would instead follow machine semantics directly, which makes it attractive for verifying real-world low-level

^{*}Shanghai Jiao Tong University, chongrongli@sjtu.edu.cn

[†]Ant Group, liyun24@antgroup.com

[‡]Tsinghua University, zhupf321@gmail.com

[§]Ant Group, lizhechen.lzc@antgroup.com

[¶]Brevis Network, mdong@brevis.network

^{||}Brevis Network, alan@brevis.network

^{**}Shanghai Jiao Tong University, huyuncong@sjtu.edu.cn

programs specified modulo 2^k , such as hash functions [DP25; BRW26; Hu+25; Set26] and machine execution traces [a1626; Suc26; RIS26; Ope26; Bre26; LZZHZ25]. This motivation has led to a growing body of work on proofs over rings [BBMRS21; BSGJOST23; BBMHS22; LXY24; GNSV23; JLXYY25; WZD25; Gar+25; Abd+26].

Why designing SNARKs over rings is challenging. The soundness of many succinct proof systems relies on random challenges, as in the Schwartz–Zippel lemma [Sch80] and proximity testing [DP23]. Over a field, these techniques work because the challenges are drawn from a large *exceptional set*, a set whose pairwise differences are all invertible, and the failure probability is inversely proportional to its size, the whole field being such a set. Over $\mathbb{Z}_{2^k}$, by contrast, every exceptional set has at most two elements, since among any three elements two have the same parity and their difference is then a zero divisor. Consequently, a nonzero low-degree polynomial can vanish on a large fraction of the ring, while a random linear combination may fail to reveal an error when its random coefficient is divisible by a large power of two. These techniques therefore do not apply directly, and achieving negligible soundness error over $\mathbb{Z}_{2^k}$ is considerably harder than over a field.

Limitations of existing approaches. Existing approaches rely on richer algebraic structures that retain enough field-like properties while still supporting native ring arithmetic, and they pay for these properties with heavier arithmetic and worse concrete performance. Most prior work uses a Galois ring $\text{GR}(2^k, d) = \mathbb{Z}_{2^k}[X]/(f(X))$, where f is a monic polynomial of degree d over $\mathbb{Z}_{2^k}$ that is irreducible modulo two. A Galois ring contains $\mathbb{Z}_{2^k}$ and, more importantly, has residue field $\mathbb{F}_{2^d}$, which allows most field-based techniques to be adapted to rings [GNSV23; WZD25; JLXYY25; LXY24]. Galois ring arithmetic, however, is expensive. Reverse multiplication-friendly embeddings (RMFEs) [EHLXY23] amortize the cost through packing but do not remove it, and as reported in [JLXYY25], SNARKs over Galois rings remain *several orders of magnitude slower* than SNARKs over fields even with an RMFE.

Another route builds SNARKs for the integers or the rationals, such as $\mathbb{Z}$ aratan [CHA26], Zinc [Gar+25] and Zinc+ [Abd+26], which in particular prove statements over any ring $\mathbb{Z}/n\mathbb{Z}$. Their core technique is fingerprinting: the prover sends oracles of integer or rational polynomials, the verifier samples a large random prime p , and every check is projected to $\mathbb{F}_p$ and handled with field techniques. This route supports general modular arithmetic and achieves succinctness without Galois rings. However, $\mathbb{Z}$ aratan relies on heavy hidden-order groups for its polynomial commitment, and all of them pay for arithmetic modulo a random prime, which is several times slower than the structured primes used in practice.

In summary, although these SNARKs support statements over $\mathbb{Z}_{2^k}$, their proof backends still work over Galois-ring extensions or large random prime fields, whose arithmetic incurs substantial overhead. Supporting ring statements thus does not by itself preserve the computational advantage of native power-of-two arithmetic.

This leads to a natural question:

Can we construct a SNARK for $\mathbb{Z}_{2^k}$ that operates natively over power-of-two rings, without resorting to Galois rings or large random prime fields?

1.1 Our Contributions

We answer this question affirmatively by making the following contributions.

An IOP for arithmetic relations over rings. We construct an IOP for arithmetic relations over $\mathbb{Z}_{2^k}$ with linear proving time and logarithmic verifier complexity. To achieve that, we develop a new sumcheck-style protocol for a lifted relation over a larger ring $\mathbb{Z}_{2^m}$, and establish a soundness analysis showing negligible error when the gap $m - k$ is sufficiently large.

Analysis of proximity gaps for linear codes over rings. We formalize and prove a weakened notion of proximity gap for linear codes over power-of-two rings. Intuitively, it states that either all given vectors over $\mathbb{Z}_{2^\ell}$ are close to the code after reduction to a smaller ring $\mathbb{Z}_{2^m}$, or their random tensor combination is far from the code over $\mathbb{Z}_{2^\ell}$, except with negligible probability. Of independent interest, our analysis offers a new perspective on code properties by studying them across a hierarchy of rings, which may be useful more broadly for code-based protocols over rings.

Ring commitments from proximity gaps. We formalize the notion of a ring commitment, which supports tensor evaluation proofs over $\mathbb{Z}_{2^m}$. We then construct a Brakedown-style ring commitment from the proximity gap above. We further show that our construction is low-precision knowledge sound, in the sense that the witness can be extracted over the smaller ring $\mathbb{Z}_{2^k}$.

Permutation checks over rings via compatible commitments. We formalize the notion of a compatible oracle, which answers both ring and field queries on the same underlying vector, and instantiate it with a compatible commitment built from our ring commitment. This compatibility allows any permutation check over the binary field $\mathbb{F}_{2^k}$ to be applied to vectors over $\mathbb{Z}_{2^k}$, which gives the wiring check of our SNARK. Of independent interest, the same mechanism lets Rotor integrate directly with binary-field SNARKs [DP25; Hu+25; BRW26; Set26], so that relations over $\mathbb{F}_{2^k}$, such as XOR and AND, can also be proved natively.

Rotor, a native SNARK over power-of-two rings. Building on our ring IOP and commitment schemes, we obtain Rotor, a native SNARK for Plonkish relations over $\mathbb{Z}_{2^k}$ with linear prover complexity and sublinear verification.

1.2 Related Works

SNARKs through Galois rings. Rinocchio [GNSV23] initiated the Galois-ring approach to proving power-of-two arithmetic. It adapts the Groth16 [Gro16] proof system to Galois rings and achieves constant proof size, but it is designated-verifier and inherits the circuit-specific trusted setup and long common reference strings. More recent work [WZD25; JLXY25] follows the modern SNARK paradigm, which combines a polynomial interactive oracle proof (PIOP) with a polynomial commitment scheme (PCS), and instantiates both over Galois rings. Wei et al. [WZD25] extended the Brakedown [GLSTW23] PCS to Galois rings and built transparent SNARKs on the Libra [XZZPS19] and HyperPlonk [CBBZ23] PIOPs. Jia et al. [JLXY25] studied FRI-based and Liger [GSV21]-based PCSs over Galois rings and obtained a transparent SNARK for rank-one constraint systems (R1CS) over $\mathbb{Z}_{2^k}$.

Galois rings offer a general route to SNARKs for ring arithmetic, but at the price of a more expensive algebraic structure. An element of $\text{GR}(2^k, d)$ is a polynomial with d coefficients in $\mathbb{Z}_{2^k}$, where d has to be large enough for the target security level (e.g., $d = 162$ for 128-bit security in [JLXY25]), so a multiplication in $\text{GR}(2^k, d)$ costs far more than native power-of-two arithmetic. RMFEs [EHLXY23; JLXY25] partially mitigate this by packing several $\mathbb{Z}_{2^k}$ values into one Galois ring element, so that a single Galois ring multiplication accounts for several coordinate-wise multiplications over the base ring. This lowers the amortized cost per multiplication, but the proof system still operates over $\text{GR}(2^k, d)$ and needs additional machinery to enforce the consistency conditions that RMFEs impose. As a result, this line of work still incurs substantial concrete overhead.

SNARKs for integer arithmetic. Campanelli et al. [CHA26] proposed Zaratan, a succinct proof system for R1CS over the integers, together with the fingerprinting technique: the prover commits to integer polynomials, the verifier samples a random prime p , and all identities are checked over $\mathbb{F}_p$ with field techniques. Its PCS follows the DARK [BFS20] compiler based on groups of unknown order, which supports evaluating a

committed integer polynomial over an arbitrary prime field while still allowing the extraction of an integer polynomial. Relations over $\mathbb{Z}_{2^k}$ can be lifted to integer relations by adding quotient witnesses¹ and then proved with Zaratan. Zinc [Gar+25] follows the fingerprinting technique but generalizes it to rational numbers: witnesses are polynomials over $\mathbb{Q}$ of bounded bit-width, projected to a random prime field through rational localization, and committed with a Brakedown [GLSTW23]-style PCS for polynomials with bounded rational coefficients. Zinc+ [Abd+26] extends Zinc to richer algebraic structures such as polynomial rings, and introduces a new family of codes that improves the PCS, with much smaller proofs and faster verification. These constructions are more general than a SNARK tailored to power-of-two rings and avoid Galois rings altogether. However, Zaratan relies on heavy hidden-order groups, and all of them suffer from the computational overhead of a random prime, as arithmetic modulo a random prime is several times slower than arithmetic modulo the structured primes used in practice, such as Mersenne primes.

Native (non-succinct) proofs over power-of-two rings. Inspired by MPC protocols over $\mathbb{Z}_{2^k}$ such as SPD $\mathbb{Z}_{2^k}$ [CDESX18], Appenzeller to Brie [BBMRS21] and Moz $\mathbb{Z}_{2^k}$ arella [BBMHS22] were the first to bring the precision-gap technique to proof systems over $\mathbb{Z}_{2^k}$, building on vector oblivious linear evaluation (VOLE). To authenticate witnesses in $\mathbb{Z}_{2^k}$, Moz $\mathbb{Z}_{2^k}$ arella samples a secret key from $\mathbb{Z}_{2^s}$ and runs the VOLE-based multiplication check over the larger ring $\mathbb{Z}_{2^{k+2s}}$, where $s \approx \lambda + \log \lambda$ for the statistical parameter λ . The extra $2s$ bits make the quadratic identity in the secret key sound, since even if a cheating error is a zero divisor over $\mathbb{Z}_{2^k}$, the identity holds over $\mathbb{Z}_{2^{k+2s}}$ with probability at most $2^{-\lambda}$. Since these protocols prove circuits gate by gate over VOLE correlations, they are designated-verifier and interactive, with linear-time prover and verifier and linear communication. MPC-in-the-head techniques have also been applied to power-of-two arithmetic [BSGJOST23]. All of these works operate natively over power-of-two rings and are concretely efficient, but none of them is succinct.

Field SNARKs for power-of-two arithmetic. Most SNARKs in practice work over prime fields [AST24; STW24; Set26] or binary fields [DP25], and they support power-of-two arithmetic only through emulation. Over a prime field, an operation modulo 2^k is emulated by computing the integer result in the field, whose modulus is large enough to avoid wraparound, and then truncating it to the low k bits with a range proof [BBBPWM18; GWHW24; YQGZ26] or a lookup argument [GW20; Hab22; STW24], which requires the values to be decomposed into bits or limbs. Jolt [AST24] follows this route and combines field constraints with the lookup argument Lasso [STW24] into structured instruction tables. Binius [DP25; DP26] constructs a SNARK that is highly optimized for Boolean operations such as XOR and AND over a binary field. Addition over $\mathbb{Z}_{2^k}$ can be expressed directly with these gates through its carry chain, whereas multiplication has no native counterpart. Binius proposes an optimized GKR-based protocol for it [Irr25], but its cost still grows with the bit-width, so multiplication remains a major overhead. Compared with these approaches, Rotor’s arithmetic IOP checks relations directly over power-of-two rings, avoiding the translation of ring arithmetic into field constraints.

2 Technical Overview

Throughout the paper, we work with three ring precisions $k < m < \ell$: the statement and the extracted witness lie in $\mathbb{Z}_{2^k}$, the IOP and its evaluation claims lie in $\mathbb{Z}_{2^m}$, and encoding and proximity testing lie in $\mathbb{Z}_{2^\ell}$ ². The gaps between these precisions provide the source of soundness.

¹For example, an RICS row over $\mathbb{Z}_{2^k}$ of the form $(Az)_i(Bz)_i = (Cz)_i \pmod{2^k}$ is lifted to an integer equality $(Az)_i(Bz)_i = (Cz)_i + 2^k u_i$ over $\mathbb{Z}$, where u_i is supplied by the prover as an auxiliary witness.

²“ m ” stands for “medium” and “ ℓ ” for “large”.

We build **Rotor** from four components. An IOP (Section 4) proves arithmetic relations over $\mathbb{Z}_{2^k}$ by lifting them to $\mathbb{Z}_{2^m}$, where the verifier has tensor-query access to the witness. Instantiating these queries requires a commitment scheme with tensor evaluations over $\mathbb{Z}_{2^m}$. To this end, we first prove a weakened proximity gap (Section 5), showing that checks over $\mathbb{Z}_{2^\ell}$ ensure proximity after reduction to $\mathbb{Z}_{2^m}$. From it we construct a Brakedown-style commitment scheme (Section 6) with evaluations over $\mathbb{Z}_{2^m}$. Finally, we define a compatible commitment (Section 7), which ties a ring witness to its binary-field representation, so that a permutation IOP over fields can enforce the wiring constraints over $\mathbb{Z}_{2^k}$. Combining the arithmetic IOP with this permutation check yields **Rotor**, a SNARK for general circuits over $\mathbb{Z}_{2^k}$ (Section 8).

2.1 An IOP for Arithmetic Relations over Rings

Our starting point is an IOP for arithmetic relations over $\mathbb{Z}_{2^k}$. Roughly speaking, such a relation is specified by a public constraint polynomial g , and requires that g vanishes on every coordinate of the witness vectors.

To avoid Galois rings, we instead lift the relation to a larger ring $\mathbb{Z}_{2^m}$ with $m > k$. A constraint that vanishes over $\mathbb{Z}_{2^k}$ may no longer vanish after its inputs are lifted to $\mathbb{Z}_{2^m}$, but its lifted value is still a multiple of 2^k . To account for this overflow, the prover additionally provides an auxiliary vector h such that the lifted constraint equals $2^k h$. One might then try to prove this identity exactly over $\mathbb{Z}_{2^m}$, but this brings us back to the original problem, only over a larger ring. We resolve this by relaxing the problem. Specifically, if the lifted constraint differs from $2^k h$ only by an error divisible by 2^k , this error disappears after reduction to $\mathbb{Z}_{2^k}$, and the original relation still holds. Therefore, instead of rejecting every instance for which the lifted identity fails over $\mathbb{Z}_{2^m}$, it suffices to only reject those whose error is not divisible by 2^k .

Sumcheck-style construction. We construct an IOP for this relaxed problem, inspired by the recursive multiplication check in the MPC protocol [Li+24]. At a high level, the protocol follows an idea similar to the sumcheck protocol by iteratively folding the vectors and deriving a new claim for the same constraint over the folded vectors. One major difference is that, instead of folding all vectors using the same random challenge, we fold each vector using individual random challenges for soundness. After logarithmic rounds of folding, the vectors are folded as scalars, and the verifier checks the final claim by querying the witness oracles. By construction, these scalars are tensor evaluations of the original witness vectors, so the queries are exactly the tensor queries supported by the oracles. Consequently, for a constraint polynomial of degree d and witness vectors of length T , the prover time is $O(d2^d T)$ ring operations, while the verification cost and the proof size are $O(2^d \log T)$.

Establishing soundness. For a cheating instance to pass the check, the malicious prover must eliminate the error in the initial claim. Suppose the initial error has fewer than k factors of two. Then it must gain at least $m - k + 1$ additional factors of two to vanish over $\mathbb{Z}_{2^m}$. As the verifier will check the consistency of the prover's messages with the current claim in each folding round, any gain of extra factors of two can only come from the verifier's random folding challenges. The probability that the error gains at least s additional factors of two in each folding step is at most 2^{-s} . Equivalently, the gain from each folding step can be bounded by a geometric random variable Γ satisfying $\Pr[\Gamma \geq s] = 2^{-s}$. Accumulating this bound over all folding steps, the overall soundness error is bounded by the tail probability of a sum of independent geometric random variables. For a precision gap $m - k = \lambda + O(\log \lambda)$, this probability is at most $2^{-\lambda}$.

Removing interaction. To make the IOP non-interactive, we apply the Fiat–Shamir transform by deriving the verifier's challenges from a random oracle. However, this transform leads to a great loss in soundness using the same precision gap as the interactive protocol. Intuitively, in the interactive protocol, the verifier samples the challenges after the prover has fixed its messages, so the malicious prover only receives a single set of random challenges in each round and passes only if these challenges supply all the missing factors of

two. With a random oracle, the prover can repeatedly obtain challenges with different candidate messages, keep the message whose challenges supply the most factors of two, and then move on to the next round with this gain fixed. The factors of two thus accumulate across rounds, at a cost that is the sum rather than the product of the costs of the individual rounds. To compensate, every round has to resist such an attack on its own, which requires a precision gap of about $(t + 1)\lambda$ for t folding rounds instead of λ , and running the whole protocol over such a large ring incurs a large computational overhead. To avoid this overhead, we observe that, instead of carrying out all folding rounds directly over $\mathbb{Z}_{2^m}$, we can gradually lift the size of the ring from one round to the next. Concretely, we fix precision levels $k < m_0 < m_1 < \dots < m_t = m$, and the reduction of round i lifts the claim from $\mathbb{Z}_{2^{m_i}}$ to $\mathbb{Z}_{2^{m_{i+1}}}$, so that the computation of round i takes place over $\mathbb{Z}_{2^{m_{i+1}}}$ and $\mathbb{Z}_{2^m}$ is only reached in the last round. Intuitively, soundness is preserved because raising the precision does not add factors of two to an error that is nonzero at the current precision. A malicious prover therefore has to gain the $m_{i+1} - m_i + 1$ missing factors of two within round i alone, which is infeasible for a proper choice of the precision levels. Since the most computationally intensive part of the protocol lies in the first several rounds of reduction, this ensures that the bulk of the prover's work is performed over small rings without sacrificing soundness.

2.2 Proximity Gaps for Linear Codes over Rings

To turn the oracles of our IOP into commitments, we follow Brakedown [GLSTW23] and encode the witness row by row with a linear code. The verifier then has to test that all rows are close to the codeword, and the tool that makes this possible is the *proximity gap* [AHIV17; GLSTW23; DP23; AER24]. However, the soundness of existing analyses relies on a large *exceptional set* in the underlying ring, that is, a set whose pairwise differences are all invertible. Every exceptional set has at most two elements over $\mathbb{Z}_{2^m}$, and the traditional proximity gap therefore fails.

Visible distance. As in the IOP construction, we instead resort to a sufficiently larger ring $\mathbb{Z}_{2^\ell}$ with $\ell > m$. This move alone does not restore the proximity gap. However, it allows us to replace the Hamming distance in the proximity gap by a weaker notion, which we call the *q-visible distance*. The Hamming distance counts every entry at which two words differ, whereas the *q-visible distance* counts only the differences that still exist after being reduced to $\mathbb{Z}_{2^q}$.

Weakened proximity gap. With this notion in hand, we establish a weakened proximity gap for binary linear codes, which we state informally. Let r be a power of two, let $u_0, \dots, u_{r-1}$ be vectors of length r over $\mathbb{Z}_{2^\ell}$, and let $\alpha_0, \beta_0, \dots, \alpha_{\log r - 1}, \beta_{\log r - 1}$ be random scalars from $\mathbb{Z}_{2^\ell}$. Then either $u_0, \dots, u_{r-1}$ are all close to the code once reduced to $\mathbb{Z}_{2^m}$, or, for most choices of the scalars, their combination with the coefficients $\bigotimes_{i=0}^{\log r - 1} (\alpha_i, \beta_i)$ is far from the code over $\mathbb{Z}_{2^\ell}$. To see why the weakened statement holds, suppose that the vectors are far from the code after reduction to $\mathbb{Z}_{2^m}$. Then the errors that a random combination would have to cancel are nonzero modulo 2^m and therefore they have fewer than m factors of two. For the combination to become a codeword over $\mathbb{Z}_{2^\ell}$, the random coefficients have to supply the $\ell - m$ or more missing factors of two by themselves, which happens only with negligible probability once the gap between the two rings is large enough.

2.3 Ring Commitments from Proximity Gaps

We next instantiate the oracles of our IOP with a ring commitment scheme based on the above weakened proximity gap. Recall that the IOP queries a witness oracle at tensor points, so the commitment scheme has

to support evaluation claims of the form $\bigotimes_{i=0}^{\log T-1} (\alpha_i, \beta_i) \cdot v = s$ for a committed vector $v \in \mathbb{Z}_{2^m}^T$. We obtain such a scheme by following Brakedown over a sufficiently larger ring $\mathbb{Z}_{2^\ell}$, where the proximity test is sound.

Brakedown-style commitment. Since the weakened gap certifies closeness only after reduction to a smaller ring, we commit and test over $\mathbb{Z}_{2^\ell}$ in order to obtain a guarantee over $\mathbb{Z}_{2^m}$. To commit to $v \in \mathbb{Z}_{2^m}^T$, the prover lifts it canonically to $\mathbb{Z}_{2^\ell}$, arranges it as a matrix, encodes every row, and commits to the columns of the encoded matrix with a Merkle tree. To prove an evaluation at a tensor point, the prover sends the tensor-weighted combination of the rows and opens randomly sampled columns. The verifier checks over $\mathbb{Z}_{2^\ell}$ that the encoding of the row combination agrees with the same combination of the opened columns, then reduces the row combination to $\mathbb{Z}_{2^m}$ and applies the remaining tensor factor to recover the claimed evaluation. By the weakened proximity gap, a committed matrix that is far from the code after reduction to $\mathbb{Z}_{2^m}$ yields a row combination that is far from the code over $\mathbb{Z}_{2^\ell}$ except with negligible probability, and the column checks then fail.

Security definition. We define the security of a ring commitment through three properties: low-precision binding, evaluation binding, and low-precision knowledge soundness. The first two are the ring analogues of the usual binding notions. The main difference from commitment schemes over fields lies in knowledge extraction: rather than extracting the entire witness over $\mathbb{Z}_{2^m}$, we only require extracting its reduction modulo 2^k .

This relaxation is motivated by an inherent obstacle to full extraction over $\mathbb{Z}_{2^m}$. Roughly speaking, the emulator repeatedly rewinds the prover, collects accepted evaluation proofs at freshly sampled challenges, and reconstructs the witness by solving the resulting linear system. However, a malicious prover may answer only those challenges that vanish on some fixed nonzero vector, so that the collected system stays underdetermined. If this vector is divisible by a large power of two, many random challenges vanish on it, and the prover retains a nonnegligible success probability while ruling out full extraction. We show that for a sufficient precision gap $m - k$, the collected equations nevertheless determine the witness modulo 2^k with high probability. Although this notion is weaker than full extraction, we prove that it suffices to establish knowledge soundness for relations over $\mathbb{Z}_{2^k}$.

2.4 Permutation Check over Rings and a Full SNARK

The IOP and the commitment scheme together prove the arithmetic relations, which capture the gate evaluation of a circuit. To construct a SNARK for general circuits, we still need a permutation check that enforces the wiring constraints, that is, that the values on wires connecting different gates agree.

Transferring to $\mathbb{F}_{2^k}$. Existing permutation checks over fields [GWC19; Hab22; STW24; STZ26; BCD25] require constraints of degree at least logarithmic in the instance size. Since the prover cost of our IOP grows exponentially with the constraint degree, they incur a polynomial overhead for the prover, which makes them impractical in our setting. We instead transfer the permutation check from $\mathbb{Z}_{2^k}$ to the binary field $\mathbb{F}_{2^k}$ through the bijection φ that identifies elements with the same bit representation. The key point is that, since φ is bijective, two vectors are permutations of each other if and only if their images are. Thus, the permutation relation over $\mathbb{Z}_{2^k}$ is equivalent to that over $\mathbb{F}_{2^k}$. To exploit this equivalence, we introduce a new oracle model, which we call a *compatible oracle*. Intuitively, a compatible oracle supports both ring and field queries to a consistent vector, where ring queries access the vector itself and field queries access its image under φ . Consequently, using compatible oracles, any IOP for the permutation relation over $\mathbb{F}_{2^k}$ can establish the corresponding permutation relation over $\mathbb{Z}_{2^k}$.

Compatible commitment. We next define a *compatible commitment* to instantiate the compatible oracle. At

a high level, it combines a ring commitment with a field polynomial commitment [DP25] under a shared commitment procedure. We show that it suffices to ensure that any valid ring opening can also be interpreted as a valid field opening under φ . The binding property of the field commitment then guarantees that the witness extracted on the field side is exactly the image under φ of the witness extracted on the ring side.

We build the compatible commitment on our ring commitment scheme. Since this scheme encodes the witness vector with the lift of a binary linear code, reducing the encoding modulo 2 yields a Brakedown-style commitment over $\mathbb{F}_2$, which naturally serves as the field commitment. However, this reduction only keeps the least significant bit of each entry. We therefore let the prover commit to the bit decomposition of the vector, so that the whole vector over $\mathbb{F}_{2^k}$ can be recovered from the committed bits. To ensure that the committed components form a valid bit decomposition, we use our IOP to verify that they are binary, which guarantees that the witnesses extracted on the two sides are consistent.

Combining the arithmetic IOP for the gate constraints, the permutation check for the wiring constraints, and the compatible commitment that instantiates their oracles yields Rotor, a SNARK for general circuits over $\mathbb{Z}_{2^k}$.

3 Preliminaries

Rings and precisions. Throughout, $1 \leq k < m < \ell$ are three precisions of rings. For $q \leq q'$, we write $\pi_q : \mathbb{Z}_{2^{q'}} \rightarrow \mathbb{Z}_{2^q}$ for reduction modulo 2^q , applied entrywise to vectors and matrices. A *lift* of $a \in \mathbb{Z}_{2^q}$ to $\mathbb{Z}_{2^{q'}}$ is any $\hat{a} \in \mathbb{Z}_{2^{q'}}$ with $\pi_q(\hat{a}) = a$, and its *canonical lift* is the representative in $\{0, \dots, 2^q - 1\}$. For $x \in \mathbb{Z}_{2^q}$, we define $\text{Po}2_q(x) := \max\{a \in \{0, \dots, q\} : x \in 2^a \mathbb{Z}_{2^q}\}$ to be the number of factors of two of x , so that $\text{Po}2_q(x) = q$ exactly when $x = 0$. For a nonempty vector or matrix, $\text{Po}2_q$ is the minimum over its entries.

Vectors. We write $[T] := \{0, \dots, T-1\}$, $u \circ v$ for the Hadamard (coordinate-wise) product and $u \cdot v$ or $\langle u, v \rangle$ for the inner product of two vectors of equal length, $u \parallel v$ for their concatenation, and 1^T for the all-one vector of length T . For d vectors $v_1, \dots, v_d$ of length T , we define their multi-inner product as $\langle v_1, \dots, v_d \rangle := \sum_{i=0}^{T-1} \prod_{j=1}^d v_{j,i}$.

Tensor products. Let R be a commutative ring; in this paper, R is a ring $\mathbb{Z}_{2^q}$ or a binary field. For pairs $(\alpha_0, \beta_0), \dots, (\alpha_{n-1}, \beta_{n-1}) \in R^2$, the tensor product $\otimes_{i=0}^{n-1} (\alpha_i, \beta_i) \in R^{2^n}$ is defined inductively by $\otimes_{i=0}^{n-1} (\alpha_i, \beta_i) := (\alpha_0 \cdot s \parallel \beta_0 \cdot s)$, with $s := \otimes_{i=1}^{n-1} (\alpha_i, \beta_i)$. For $\eta = (\alpha_0, \beta_0, \dots, \alpha_{n-1}, \beta_{n-1})$, we write $\text{Tensor}(\eta) := \otimes_{i=0}^{n-1} (\alpha_i, \beta_i)$.

3.1 Algebraic Facts

Below we give some important algebraic facts used in our soundness analysis. Our soundness arguments bound the probability that random ring elements supply the factors of two that a nonzero value is missing. Each random combination supplies at most a geometric number of factors of two, and over several rounds these contributions add up. We therefore record the exact tail of a sum of independent geometric variables, which gives the concrete error bounds in our theorems and in the choice of parameters.

Let Γ be the geometric random variable on $\{0, 1, \dots\}$ with

$$\Pr[\Gamma = g] = 2^{-(g+1)}, \quad \Pr[\Gamma \geq g] = 2^{-g} \quad (g \geq 0),$$

let $\Gamma_1, \Gamma_2, \dots$ be independent copies, and put $S_t := \sum_{j=1}^t \Gamma_j$, with $S_0 := 0$.

Lemma 1. For positive integers S, t , define $\Phi(S, t) := \Pr[S_t \geq S]$. Then

$$\Phi(S, t) = 2^{-(S+t-1)} \sum_{j=0}^{t-1} \binom{S+t-1}{j} = \sum_{j=0}^{t-1} \binom{S+j-1}{j} 2^{-(S+j)}.$$

In particular, $\Phi(S, t)$ is nonincreasing in S and nondecreasing in t .

Proof. Interpret Γ_j as the number of failures before the next success in a sequence of independent fair coin tosses. Then $S_t \geq S$ exactly when the S -th failure occurs before the t -th success, that is, when the first $S+t-1$ tosses contain at most $t-1$ successes, which gives the first expression. For the second, partition the event according to the number $j < t$ of successes preceding the S -th failure: the toss at which it occurs is a failure, and the $S+j-1$ preceding tosses contain exactly j successes, which has probability $\binom{S+j-1}{j} 2^{-(S+j)}$. Monotonicity is clear from the definition as a tail probability. $\square$

The basic source of factors of two is a single random combination of two vectors. The following lemma is the ring analogue of the Schwartz–Zippel lemma [Sch80; Zip79] for this case: the combination vanishes only if the random coefficients supply all the factors of two that the vectors lack.

Lemma 2. Let $v_0, v_1 \in \mathbb{Z}_{2^q}^r$ and define $a := \min\{\text{Po2}_q(v_0), \text{Po2}_q(v_1)\}$. If $a < q$, then for independent uniform θ_0, θ_1 over $\mathbb{Z}_{2^q}$, $\Pr[\theta_0 v_0 + \theta_1 v_1 = 0] \leq 2^{-(q-a)}$.

Proof. Assume without loss of generality that $\text{Po2}_q(v_0) = a$. Then choose a coordinate i with $\text{Po2}_q(v_{0,i}) = a$. The i -th coordinate of the equation implies $\theta_0 v_{0,i} = -\theta_1 v_{1,i}$. Since $v_{0,i}$ has exactly a factors of two, multiplication by $v_{0,i}$ over $\mathbb{Z}_{2^q}$ has a kernel of size 2^a . Hence, for any θ_1 , the above equation has either no solution or exactly 2^a solutions in θ_0 . As θ_0 is uniform over $\mathbb{Z}_{2^q}$, for any fixed θ_1 , the coordinate equation holds with probability at most $2^{-(q-a)}$. Averaging over θ_1 proves the claim. $\square$

Applying this bound once per tensor challenge pair gives a contraction bound that remains valid after all previous prover messages are fixed, so that one statement covers the initial batching and each adaptive folding round of our protocols.

Lemma 3. Let $T = 2^t$. For integers $q, t \geq 1$ and $u \in \mathbb{Z}_{2^q}^T$, let $a := \text{Po2}_q(u)$. Let $(\theta_0^{(i)}, \theta_1^{(i)})$ be independent and uniform over $\mathbb{Z}_{2^q}^2$ for every $i \in [t]$, and define $U := u \cdot \otimes_{i=0}^{t-1} (\theta_0^{(i)}, \theta_1^{(i)})$. Then, for every integer b , $\Pr[\text{Po2}_q(U) \geq b] \leq \Pr[a + S_t \geq b]$. In particular, if $a < q$, then $\Pr[U = 0] \leq \Phi(q - a, t)$.

Proof. Set $u^{(0)} := u$ and, for $i \in [t]$, define $u^{(i+1)} := \theta_0^{(i)} u^{(i,0)} + \theta_1^{(i)} u^{(i,1)}$, where $u^{(i,0)}$ and $u^{(i,1)}$ are the low and high halves of $u^{(i)}$, so that $u^{(t)} = U$. Let $a_i := \text{Po2}_q(u^{(i)})$, and therefore $a_0 = a$ and $a_t = \text{Po2}_q(U)$. Consider every integer b with $a_i < b \leq q$. The event $a_{i+1} \geq b$ is equivalent to $\theta_0^{(i)} u^{(i,0)} + \theta_1^{(i)} u^{(i,1)} = 0 \pmod{2^b}$. Since $a_i < b$, reducing modulo 2^b gives $a_i = \min\{\text{Po2}_b(u^{(i,0)}), \text{Po2}_b(u^{(i,1)})\}$. Moreover, the reductions of $\theta_0^{(i)}$ and $\theta_1^{(i)}$ modulo 2^b remain independent and uniform over $\mathbb{Z}_{2^b}$. Applying Lemma 2 therefore gives $\Pr[a_{i+1} \geq b \mid u^{(i)}] \leq 2^{-(b-a_i)} = \Pr[a_i + \Gamma_{i+1} \geq b \mid a_i]$. The same inequality holds trivially for $b \leq a_i$ and $b > q$.

We now prove by induction that, for every integer b , $\Pr[a_i \geq b] \leq \Pr[a + S_i \geq b]$. The claim is immediate for $i = 0$. Assuming it holds for i , the above result implies

$$\begin{aligned} \Pr[a_{i+1} \geq b] &\leq \sum_{g \geq 0} 2^{-(g+1)} \Pr[a_i \geq b - g] \\ &\leq \sum_{g \geq 0} 2^{-(g+1)} \Pr[a + S_i \geq b - g] = \Pr[a + S_{i+1} \geq b]. \end{aligned}$$

Thus, taking $i = t$ gives $\Pr[\text{Po2}_q(U) \geq b] \leq \Pr[a + S_t \geq b]$. Finally, if $a < q$, taking $b = q$ gives $\Pr[U = 0] = \Pr[\text{Po2}_q(U) \geq q] \leq \Pr[a + S_t \geq q] = \Phi(q - a, t)$, which proves the second claim. $\square$

To choose parameters for soundness error $2^{-\lambda}$, we use the following estimate from [Li+24, Appendix G.6].

Lemma 4. *Let S and t be positive integers and let $\lambda > 0$. If*

$$S - 1 \geq \max \left\{ 3t, \lambda + t \left(\frac{1}{2} + \log \left(\frac{5}{2} + \frac{3\lambda}{t} \right) \right) \right\},$$

then $\Phi(S, t) \leq 2^{-\lambda}$.

3.2 Interactive Oracle Proofs

Indexed relations. An indexed relation $\mathcal{R}$ is a set of tuples $(\mathfrak{i}, \mathfrak{x}; \mathfrak{w})$, where $\mathfrak{i}$ is the index, $\mathfrak{x}$ is the instance, and $\mathfrak{w}$ is the witness. We allow both the index and the instance to contain explicit components and oracle components. The language associated with $\mathcal{R}$ is $\mathcal{L}(\mathcal{R}) := \{(\mathfrak{i}, \mathfrak{x}) : \exists \mathfrak{w}, \text{ s.t. } (\mathfrak{i}, \mathfrak{x}; \mathfrak{w}) \in \mathcal{R}\}$.

Interactive oracle proofs. Let $\mathcal{R}$ be an indexed relation. An interactive oracle proof (IOP) for $\mathcal{R}$ is a public-coin interactive protocol $\Pi = (\mathcal{P}, \mathcal{V})$ with the following syntax. The verifier $\mathcal{V}$ receives the explicit components of the index and the instance and query access to their oracle components. The prover $\mathcal{P}$ receives the same information, the contents of these oracles, and a witness $\mathfrak{w}$. The parties interact for a number of rounds. In each round, $\mathcal{P}$ may send explicit messages and oracles, and $\mathcal{V}$ replies with independent uniformly random challenges. After the interaction, $\mathcal{V}$ queries the oracle components of the index and the instance and the oracles sent by $\mathcal{P}$. The query points are determined by the verifier's randomness, its explicit input, and the explicit messages received during the interaction. The verifier then outputs a bit as a deterministic function of this information and the query answers.

What an oracle holds, which queries it answers, and from which structure the explicit messages and the challenges are drawn are not fixed here. They are fixed by an *oracle model*, which each section specifies for the protocols it considers: Section 4 uses oracles that hold vectors over $\mathbb{Z}_{2^m}$ and answer tensor queries, and Section 7 lets the same oracles answer a second kind of query over a binary field. A model may allow an oracle component of the index or the instance to hold any one of several contents for the same component, for instance any lift of a vector over $\mathbb{Z}_{2^k}$; completeness and soundness are then required for every such choice.

The protocol is perfectly complete if, for every $(\mathfrak{i}, \mathfrak{x}; \mathfrak{w}) \in \mathcal{R}$ and every choice of the contents of the oracle components of $\mathfrak{i}$ and $\mathfrak{x}$ allowed by the model, the honest prover is accepted with probability 1. The protocol has soundness error ϵ if, for every $(\mathfrak{i}, \mathfrak{x}) \notin \mathcal{L}(\mathcal{R})$, every such choice of contents, and every prover strategy $\mathcal{P}^*$, $\Pr[\langle \mathcal{P}^*, \mathcal{V}(\mathfrak{i}, \mathfrak{x}) \rangle = 1] \leq \epsilon$.

Promise problems. We use the same protocol syntax for an indexed promise problem (YES, NO), where YES and NO are disjoint sets of pairs $(\mathfrak{i}, \mathfrak{x})$, and the index and the instance may again contain both explicit and oracle components. The protocol is perfectly complete if every $(\mathfrak{i}, \mathfrak{x}) \in \text{YES}$, with any contents of its oracle components allowed by the model, is accepted by the honest prover with probability 1, and it has soundness error ϵ if, for every $(\mathfrak{i}, \mathfrak{x}) \in \text{NO}$, every such choice of contents, and every prover strategy $\mathcal{P}^*$, $\Pr[\langle \mathcal{P}^*, \mathcal{V}(\mathfrak{i}, \mathfrak{x}) \rangle = 1] \leq \epsilon$. No condition is imposed on pairs outside $\text{YES} \cup \text{NO}$.

4 An IOP for Arithmetic Relations over Rings

In this section, we construct an IOP for arithmetic relations over $\mathbb{Z}_{2^k}$.

4.1 Arithmetic Relations over $\mathbb{Z}_{2^k}$ and Their Lifts

We first define the ring oracle used in this section.

Definition 1. A ring oracle over $\mathbb{Z}_{2^m}$ holds a vector $\hat{u} \in \mathbb{Z}_{2^m}^T$ and answers a tensor query at a point $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$ with $\text{Tensor}(\eta) \cdot \hat{u} \in \mathbb{Z}_{2^m}$. For $u \in \mathbb{Z}_{2^m}^T$, the oracle $\llbracket u \rrbracket$ holds u itself. For $u \in \mathbb{Z}_{2^k}^T$, the oracle $\llbracket u \rrbracket$ holds a lift $\hat{u} \in \mathbb{Z}_{2^m}^T$ of u , such that $\pi_k(\hat{u}) = u$. For a list of vectors $u = (u_0, \dots, u_{r-1})$, we write $\llbracket u \rrbracket := (\llbracket u_0 \rrbracket, \dots, \llbracket u_{r-1} \rrbracket)$.

We next define the arithmetic relations over $\mathbb{Z}_{2^k}$ together with their lifted counterparts over $\mathbb{Z}_{2^m}$. We consider coordinate-wise constraints specified by a polynomial defined over $\mathbb{Z}_{2^k}$. Specifically, given an r -variable polynomial $g(X_0, \dots, X_{r-1})$ of total degree d over $\mathbb{Z}_{2^k}$ and r vectors $v := (v_0, \dots, v_{r-1}) \in (\mathbb{Z}_{2^k}^T)^r$ each of length T , the arithmetic relation requires the constraint defined by g to hold on every coordinate $i \in [T]$ of the vectors, i.e., $g(v_{0,i}, \dots, v_{r-1,i}) = 0$ for all $i \in [T]$. We use the abbreviation $g(v) \in \mathbb{Z}_{2^k}^T$ for the vector whose i -th entry is $g(v_{0,i}, \dots, v_{r-1,i})$ for all $i \in [T]$, and give the following definition.

Definition 2. The arithmetic relation $\mathcal{R}_{\text{arith}}^k$ is the set of all tuples $(\mathfrak{i}, \mathfrak{x}; \mathfrak{w}) = (g, \llbracket v \rrbracket; v)$, where $v \in (\mathbb{Z}_{2^k}^T)^r$ and $g(v) = 0^T$.

We now lift the arithmetic constraint from $\mathbb{Z}_{2^k}$ to $\mathbb{Z}_{2^m}$. Let $\hat{g}$ be the canonical lift of g to $\mathbb{Z}_{2^m}$. For lifts $\hat{v} := (\hat{v}_0, \dots, \hat{v}_{r-1}) \in (\mathbb{Z}_{2^m}^T)^r$, a constraint that vanishes after reduction to $\mathbb{Z}_{2^k}$ need not vanish exactly over $\mathbb{Z}_{2^m}$. Instead, its residual can be a multiple of 2^k . We therefore introduce an additional vector h .

Definition 3. The lifted arithmetic relation $\hat{\mathcal{R}}_{\text{arith}}^{k,m}$ is the set of all tuples $(\mathfrak{i}, \mathfrak{x}; \mathfrak{w}) = (\hat{g}, (\llbracket \hat{v} \rrbracket, \llbracket h \rrbracket); (\hat{v}, h))$, where $(\hat{v}, h) \in (\mathbb{Z}_{2^m}^T)^{r+1}$ and $\hat{g}(\hat{v}) - 2^k \cdot h = 0^T$.

For a fixed pair $(\hat{v}, h)$, the exact equality required by $\hat{\mathcal{R}}_{\text{arith}}^{k,m}$ is stronger than merely requiring the constraint to hold after reduction to $\mathbb{Z}_{2^k}$, as the residual $\hat{g}(\hat{v}) - 2^k h$ may be nonzero in $\mathbb{Z}_{2^m}^T$ while still vanishing modulo 2^k . Such an input induces a valid arithmetic constraint over $\mathbb{Z}_{2^k}$, even though it does not belong to the lifted relation. Therefore, we need only consider to reject lifted inputs whose residual remains nonzero after reduction to $\mathbb{Z}_{2^k}$, and no soundness guarantee is needed for the intermediate case above as they similarly implies a valid tuple for $\mathcal{R}_{\text{arith}}^k$. This naturally leads to the following promise problem.

Definition 4. The lifted arithmetic promise problem $\hat{\text{AP}}_{\text{arith}}^{k,m} := (\text{YES}, \text{NO})$ is defined by $\text{YES} := \mathcal{L}(\hat{\mathcal{R}}_{\text{arith}}^{k,m})$ and $\text{NO} := \{(\hat{g}, (\hat{v}, h)) : \text{Po2}_m(\hat{g}(\hat{v}) - 2^k h) < k\}$.

The following theorem states that an IOP for the lifted arithmetic promise problem can be used to construct an IOP for the original arithmetic relation.

Theorem 1. Let $\hat{\Pi}$ be an IOP for the lifted arithmetic promise problem $\hat{\text{AP}}_{\text{arith}}^{k,m}$ with soundness error ϵ . Then there exists an IOP Π for the arithmetic relation $\mathcal{R}_{\text{arith}}^k$ with the same soundness error ϵ . Additionally, if $\hat{\Pi}$ is perfectly complete, then so is Π .

Proof. We construct Π from $\hat{\Pi}$ as follows. Let $\hat{v} = (\hat{v}_0, \dots, \hat{v}_{r-1})$ denote the lifted vectors submitted by the prover to the oracle. The prover evaluates $\hat{g}(\hat{v})$ over $\mathbb{Z}_{2^m}$ and computes the corresponding $h \in \mathbb{Z}_{2^m}^T$. Then the prover submits it to the oracle and the verifier obtains $\llbracket h \rrbracket$. Finally, the parties run $\hat{\Pi}$ on the lifted instance $(\hat{v}, h)$.

For completeness, suppose $g(v_0, \dots, v_{r-1}) = 0$ in $\mathbb{Z}_{2^k}^T$. We have $\pi_k(\hat{g}(\hat{v})) = g(v) = 0$. Hence every coordinate of $\hat{g}(\hat{v})$ is divisible by 2^k and therefore there exists $h \in \mathbb{Z}_{2^m}^T$ such that $\hat{g}(\hat{v}) = 2^k h$ in $\mathbb{Z}_{2^m}^T$. The

honest prover then computes such an h , and thus $(\hat{v}, h) \in \text{YES}$. Consequently, if $\hat{\Pi}$ is perfectly complete, it accepts with probability 1, and so does Π .

For soundness, suppose that $(g, \llbracket v \rrbracket) \notin \mathcal{L}(\mathcal{R}_{\text{arith}}^k)$. Then there is at least one coordinate $i \in [T]$ such that $g(v)_i \neq 0$. Therefore, this implies $\pi_k(\hat{g}(\hat{v})_i) \neq 0$. Equivalently, $\text{Po2}_m(\hat{g}(\hat{v})_i) < k$. For every $h \in \mathbb{Z}_{2^m}^T$, the term $2^k h_i$ is divisible by 2^k , and therefore subtracting it cannot change the result reduced to $\mathbb{Z}_{2^k}$. Hence $\pi_k(\hat{g}(\hat{v})_i - 2^k h_i) \neq 0$, and thus $\text{Po2}_m(\hat{g}(\hat{v}) - 2^k h) < k$. Therefore $(\hat{v}, h) \in \text{NO}$ for every possible choice of h .

Consequently, regardless of the oracle $\llbracket h \rrbracket$ sent by a cheating prover and of its subsequent strategy, the execution of $\hat{\Pi}$ is on a NO instance. By the soundness of $\hat{\Pi}$, the verifier accepts with probability at most ϵ . Hence Π has soundness error ϵ . $\square$

4.2 Our Construction of IOP

We now give the IOP for the lifted arithmetic promise problem $\widehat{\text{AP}}_{\text{arith}}^{k,m}$.

Example: Hadamard products of two vectors. We begin with an illustrative example to explain our idea. We consider the tuple $(\hat{g}, (\hat{v}_0, \hat{v}_1, 0^T)) \in \widehat{\mathcal{R}}_{\text{arith}}^{k,m}$, with $\hat{g}(X_0, X_1) = X_0 \cdot X_1$. Hence, the claim amounts to $\hat{v}_0 \circ \hat{v}_1 = 0^T$.

The T individual claims are first batched into a single inner-product claim. Let $T = 2^t$. The verifier samples $(\theta_i^{(0)}, \theta_i^{(1)})$ from $\mathbb{Z}_{2^m}^2$ for every $i \in [t]$ and sends them to the prover. Both parties define a challenge vector $\theta := \otimes_{i=0}^{t-1} (\theta_i^{(0)}, \theta_i^{(1)}) \in \mathbb{Z}_{2^m}^T$, and then batch the claims by defining $z_0 := \langle \theta \circ \hat{v}_0, \hat{v}_1 \rangle = 0$. For convenience, we further let $w_0 := \theta \circ \hat{v}_0$ and $w_1 := \hat{v}_1$, and thus $z_0 = \langle w_0, w_1 \rangle$.

The protocol then proceeds to the folding step. Starting from $z_0 = \langle w_0, w_1 \rangle$, in each round $i \in [t]$ the two parties have an inner-product claim $z_i = \langle w_0, w_1 \rangle$, where w_0, w_1 have length 2^{t-i} , and the goal of each round is to halve the vector lengths and then derive the new claim z_{i+1} . This is achieved by random folding. Unlike the standard sumcheck protocol over fields, where both vectors are folded using the same random challenge, we use two random challenges for each vector. Concretely, the verifier samples four random challenges $\gamma_i^{(0,0)}, \gamma_i^{(0,1)}, \gamma_i^{(1,0)}, \gamma_i^{(1,1)}$. For every $j \in \{0, 1\}$, let $w_j^{(0)}$ and $w_j^{(1)}$ denote the low and high halves of w_j , that is, the subvectors formed by its first and last 2^{t-i-1} entries, respectively. The vectors are then folded as

$$w_0 \leftarrow \gamma_i^{(0,0)} w_0^{(0)} + \gamma_i^{(0,1)} w_0^{(1)}, \quad w_1 \leftarrow \gamma_i^{(1,0)} w_1^{(0)} + \gamma_i^{(1,1)} w_1^{(1)} \in \mathbb{Z}_{2^m}^{2^{t-i-1}}.$$

Accordingly, the claim for the folded vectors becomes

$$\begin{aligned} z_{i+1} &= \langle \gamma_i^{(0,0)} w_0^{(0)} + \gamma_i^{(0,1)} w_0^{(1)}, \gamma_i^{(1,0)} w_1^{(0)} + \gamma_i^{(1,1)} w_1^{(1)} \rangle \\ &= \sum_{b_0, b_1 \in \{0,1\}} \gamma_i^{(0,b_0)} \gamma_i^{(1,b_1)} \langle w_0^{(b_0)}, w_1^{(b_1)} \rangle. \end{aligned}$$

To enable the verifier to obtain the updated claim z_{i+1} , the prover sends the four partial inner products $z_i^{(b)} := \langle w_0^{(b_0)}, w_1^{(b_1)} \rangle$ for every $b := (b_0, b_1) \in \{0, 1\}^2$, before the challenges are sampled. To ensure the consistency of the prover's messages with the previous claim, the verifier checks that $z_i = z_i^{(0,0)} + z_i^{(1,1)}$. After this, the verifier samples the challenges and computes the weighted sum above to obtain z_{i+1} , which yields a new claim over vectors of half the length.

After t rounds, the claim concerns the product of two scalars. By construction, these scalars can be viewed as tensor evaluations of the corresponding original vectors, with the batching tensor θ absorbed into

PROTOCOL 1. *IOP for the Lifted Arithmetic Promise Problem*

Let $T = 2^t$ and let $\widehat{G}(X_0, \dots, X_{r+1}) = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} X_{\mu_j}$ be a $(r+2)$ -variable homogeneous polynomial of degree $d \geq 2$. $\mathcal{P}$ holds $(\widehat{v}, h) \in (\mathbb{Z}_{2^m}^T)^{r+1}$, where $\widehat{v} = (\widehat{v}_0, \dots, \widehat{v}_{r-1})$. $\mathcal{V}$ has tensor-query access to $\llbracket \widehat{v} \rrbracket$ and $\llbracket h \rrbracket$. By a slight abuse of notation, we denote $\widehat{v}_r := h$ and $\widehat{v}_{r+1} := 1^T$. For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, set $w_j^{(\mu)} := \widehat{v}_{\mu_j}$.

1. $\mathcal{V}$ samples $(\theta_i^{(0)}, \theta_i^{(1)})$ from $\mathbb{Z}_{2^m}^2$ for $i \in [t]$ and sends them to $\mathcal{P}$. Both parties set $z_0 := 0$. $\mathcal{P}$ computes $\theta := \otimes_{i=0}^{t-1} (\theta_i^{(0)}, \theta_i^{(1)})$ and sets $w_0^{(\mu)} \leftarrow \theta \circ w_0^{(\mu)}$ for every $\mu \in \widehat{\mathcal{I}}$.

2. For $i = 0, \dots, t-1$:

(a) For every $\mu \in \widehat{\mathcal{I}}$, $j \in [d]$, and $b \in \{0, 1\}$, let $w_j^{(\mu,0)}$ and $w_j^{(\mu,1)}$ be the low and high halves of $w_j^{(\mu)}$. For every $b = (b_0, \dots, b_{d-1}) \in \{0, 1\}^d$, $\mathcal{P}$ computes

$$z_i^{(b)} := \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \cdot \left\langle w_0^{(\mu, b_0)}, \dots, w_{d-1}^{(\mu, b_{d-1})} \right\rangle,$$

and sends all $\{z_i^{(b)}\}_{b \in \{0,1\}^d}$ to $\mathcal{V}$.

(b) $\mathcal{V}$ checks $z_i = z_i^{(0^d)} + z_i^{(1^d)}$, and rejects if the check fails. Otherwise, for every $j \in [d]$, it samples $(\gamma_i^{(j,0)}, \gamma_i^{(j,1)})$ from $\mathbb{Z}_{2^m}^2$ and sends the pairs to $\mathcal{P}$. Both parties then set

$$z_{i+1} := \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j, b_j)} \right) z_i^{(b)}.$$

(c) For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, the prover updates the vectors as

$$w_j^{(\mu)} \leftarrow \gamma_i^{(j,0)} w_j^{(\mu,0)} + \gamma_i^{(j,1)} w_j^{(\mu,1)}.$$

3. $\mathcal{V}$ computes the first query point $\eta_0 := ((\theta_i^{(0)} \gamma_i^{(0,0)}, \theta_i^{(1)} \gamma_i^{(0,1)}))_{i \in [t]}$, and the other query points $\eta_j := ((\gamma_i^{(j,0)}, \gamma_i^{(j,1)}))_{i \in [t]}$ for every $j \in \{1, \dots, d-1\}$. For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, if $\widehat{v}_{\mu_j} = 1^T$, $\mathcal{V}$ locally computes its evaluation as $q_{\mu,j}$. Otherwise, $\mathcal{V}$ queries $\llbracket \widehat{v}_{\mu_j} \rrbracket$ at η_j and obtains $q_{\mu,j}$. Finally, $\mathcal{V}$ accepts if $z_t = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} q_{\mu,j}$.

the first evaluation. More precisely, they are given by

$$q^{(0)} := \left\langle \bigotimes_{i=0}^{t-1} (\theta_i^{(0)} \gamma_i^{(0,0)}, \theta_i^{(1)} \gamma_i^{(0,1)}), \widehat{v}_0 \right\rangle, \quad q^{(1)} := \left\langle \bigotimes_{i=0}^{t-1} (\gamma_i^{(1,0)}, \gamma_i^{(1,1)}), \widehat{v}_1 \right\rangle.$$

Thus, the verifier obtains the two final scalars with one tensor query to each oracle and checks that their product equals the final claimed value z_t .

Formal Construction. We now present the IOP for the lifted arithmetic promise problem $\widehat{\text{AP}}_{\text{arith}}^{k,m}$. Let the lifted polynomial have degree at most d and write

$$\widehat{g}(X_0, \dots, X_{r-1}) = \sum_{\mu \in \mathcal{I}} \widehat{c}_\mu \prod_{j=0}^{|\mu|-1} X_{\mu_j},$$

where $\mathcal{I}$ indexes the monomials of g , and $\widehat{c}_\mu \in \mathbb{Z}_{2^m}$ denotes the canonical lift of each coefficient $c_\mu \in \mathbb{Z}_{2^k}$ in g . Recall that the lifted constraint is $\widehat{g}(\widehat{v}) - 2^k h = 0^T$. For clarity of presentation, we homogenize this constraint by introducing two additional variables and defining

$$\widehat{G}(X_0, \dots, X_{r+1}) := \sum_{\mu \in \mathcal{I}} \widehat{c}_\mu \left(\prod_{j=0}^{|\mu|-1} X_{\mu_j} \right) X_{r+1}^{d-|\mu|} - 2^k X_r \cdot X_{r+1}^{d-1}.$$

Equivalently, we write the resulting degree- d homogeneous polynomial as

$$\widehat{G}(X_0, \dots, X_{r+1}) = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} X_{\mu_j}.$$

By construction, $\widehat{G}$ is homogeneous of degree d , and in particular,

$$\widehat{G}(\widehat{v}_0, \dots, \widehat{v}_{r-1}, h, 1^T) = \widehat{g}(\widehat{v}_0, \dots, \widehat{v}_{r-1}) - 2^k h.$$

With this notation in place, we give our construction in Protocol 1. It follows the high-level structure of the illustrative example. Differently, in each folding round, the prover now sends 2^d partial sums corresponding to all choices of low or high halves of the vectors, and the verifier samples d independent challenge pairs that are reused across all monomials, to derive the new folded claim.

Theorem 2. *Protocol 1 is an IOP with tensor queries for the lifted arithmetic promise problem $\widehat{\text{AP}}_{\text{arith}}^{k,m}$. It is perfectly complete and has soundness error $\Phi(m - k + 1, (d + 1)t)$.*

Proof. Completeness. Suppose that the input $(\widehat{g}, (\widehat{v}, h)) \in \text{YES}$. By definition $\widehat{G}(\widehat{v}_0, \dots, \widehat{v}_{r+1}) = 0^T$. Hence, the honest prover's initial claim is $z_0 = \langle \theta, 0^T \rangle = 0$ as required.

We show inductively that every round preserves the claimed value. At the beginning of round i , suppose that $z_i = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \langle w_0^{(\mu)}, \dots, w_{d-1}^{(\mu)} \rangle$. For each $b \in \{0, 1\}^d$, the honest prover sends

$$z_i^{(b)} = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \left\langle w_0^{(\mu, b_0)}, \dots, w_{d-1}^{(\mu, b_{d-1})} \right\rangle, \forall b \in \{0, 1\}^d.$$

By definition, $z_i = z_i^{(0^d)} + z_i^{(1^d)}$ and thus the consistency check succeeds.

After sampling the challenges, the verifier updates its claim to

$$\begin{aligned} z_{i+1} &= \sum_{b \in \{0, 1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j, b_j)} \right) z_i^{(b)} \\ &= \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \left\langle \gamma_i^{(0,0)} w_0^{(\mu,0)} + \gamma_i^{(0,1)} w_0^{(\mu,1)}, \dots, \gamma_i^{(d-1,0)} w_{d-1}^{(\mu,0)} + \gamma_i^{(d-1,1)} w_{d-1}^{(\mu,1)} \right\rangle. \end{aligned}$$

As specified in Protocol 1, the last expression is exactly the new claim associated with the folded vectors. Thus the invariant is preserved.

After all t rounds, each vector $w_j^{(\mu)}$ has been reduced to a scalar $q_{\mu,j}$. The folding procedure implies that for every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, this scalar is exactly the tensor evaluation of the corresponding vector $\widehat{v}_{\mu_j}$ at the point η_j , since the fold of round i contracts the index bit $t - 1 - i$, which is the bit of the i -th pair of η_j . For $j = 0$, the batching tensor θ is absorbed into the query point η_0 . Consequently, $z_t = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} q_{\mu,j}$, which is precisely the verifier's final check. Hence, Protocol 1 is perfectly complete.

Soundness. For every instance $(\widehat{g}, (\widehat{v}, h)) \in \text{NO}$, we write $\Xi = \widehat{G}(\widehat{v}_0, \dots, \widehat{v}_{r+1})$. By assumption, we have $\text{Po2}_m(\Xi) < k$. Let $f_0 = \langle \theta, \Xi \rangle$ denote the honest batched value. Since the verifier initializes $z_0 = 0$, the initial error is $\epsilon_0 := f_0 - z_0 = \langle \theta, \Xi \rangle$. Because θ is the tensor product of t independent challenge pairs, Lemma 3 implies that, for every integer q ,

$$\Pr[\text{Po2}_m(\epsilon_0) \geq q] \leq \Pr[k - 1 + S_t \geq q]. \quad (1)$$

We next analyze each round $i \in [t]$. Let f_i denote the honest value determined by the current vectors, let z_i denote the claimed value maintained by the verifier, and set $\epsilon_i := f_i - z_i$. For every $b \in \{0, 1\}^d$, let $f_i^{(b)}$ be the honest partial message determined by the current vectors, let $z_i^{(b)}$ denote the corresponding message sent by the prover, and set $\epsilon_i^{(b)} := f_i^{(b)} - z_i^{(b)}$. By construction, $f_i = f_i^{(0^d)} + f_i^{(1^d)}$. If the consistency check fails, the verifier rejects immediately. Hence, it suffices to consider the case in which $z_i = z_i^{(0^d)} + z_i^{(1^d)}$. Therefore,

$$\epsilon_i = \epsilon_i^{(0^d)} + \epsilon_i^{(1^d)}, \quad \text{Po2}_m((\epsilon_i^{(b)})_{b \in \{0,1\}^d}) \leq \text{Po2}_m(\epsilon_i).$$

After the challenges are sampled, define the honest value for the next round as

$$f_{i+1} := \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j,b_j)} \right) f_i^{(b)}.$$

By the definition of z_{i+1} in Protocol 1, the discrepancy after this update is

$$\begin{aligned} \epsilon_{i+1} &:= f_{i+1} - z_{i+1} \\ &= \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j,b_j)} \right) \epsilon_i^{(b)} \\ &= \left\langle \bigotimes_{j=0}^{d-1} (\gamma_i^{(j,0)}, \gamma_i^{(j,1)}), (\epsilon_i^{(b)})_{b \in \{0,1\}^d} \right\rangle. \end{aligned}$$

Conditioning on the transcript before the fresh challenges of round i are sampled, applying Lemma 3, and then averaging over the transcript, we obtain

$$\Pr[\text{Po2}_m(\epsilon_{i+1}) \geq q] \leq \Pr[\text{Po2}_m(\epsilon_i) + S_d \geq q], \quad (2)$$

where S_d is the sum of d fresh independent copies of Γ , independent of the transcript before round i .

We now prove by induction on $i \in \{0, \dots, t\}$ that, for every integer q ,

$$\Pr[\text{Po2}_m(\epsilon_i) \geq q] \leq \Pr[k - 1 + S_{t+di} \geq q]. \quad (3)$$

The base case $i = 0$ follows from (1).

Suppose (3) holds for i , where $i \in [t]$. Then, (2) gives

$$\begin{aligned} \Pr[\text{Po2}_m(\epsilon_{i+1}) \geq q] &\leq \sum_{u \geq 0} \Pr[S_d = u] \Pr[\text{Po2}_m(\epsilon_i) \geq q - u] \\ &\leq \sum_{u \geq 0} \Pr[S_d = u] \Pr[k - 1 + S_{t+di} \geq q - u] \\ &= \Pr[k - 1 + S_{t+d(i+1)} \geq q], \end{aligned}$$

where the last equality follows from the independence of the d fresh geometric variables.

Finally, if the verifier accepts, the final check implies $z_t = f_t$ and therefore $\epsilon_t = 0$. Thus, we have

$$\begin{aligned} \Pr[\text{accept}] &\leq \Pr[\text{Po2}_m(\epsilon_t) \geq m] \\ &\leq \Pr[k - 1 + S_{(d+1)t} \geq m] \\ &= \Phi(m - k + 1, (d + 1)t). \end{aligned}$$

□

By Lemma 4, if $m - k = \lambda + O(dt \log \lambda)$, then the soundness error is at most $2^{-\lambda}$.

Complexity. Let $M := |\widehat{\mathcal{I}}|$. The prover and verifier require $O(d2^d MT)$ and $O(2^d \log T + Md)$ $\mathbb{Z}_{2^m}$ operations, respectively. The non-oracle communication consists of $2^d \log T$ $\mathbb{Z}_{2^m}$ elements, with $\log T$ rounds and at most $d(r + 1)$ queries.

Remark 1 (Higher-arity folding). *Protocol 1 naturally extends to arbitrary folding arity $B \geq 2$. In each round, the vectors are partitioned into B blocks, and the prover sends all B^d partial sums. Then, the verifier samples B challenges for each position, checks the consistency with the current claim using B diagonal sums and updates the claim accordingly. The initial batching step extends to the same arity similarly. One benefit of using a larger folding arity is that it leads to a decreased soundness error of $\Phi(m - k + 1, (d + 1) \log_B T)$, which allows using a smaller m for the same target security level.*

4.3 IOP in the Random Oracle Model

The construction above relies on an interactive public-coin verifier. To obtain a non-interactive protocol, it seems natural to apply the Fiat–Shamir transform by deriving the challenges from a random oracle H . However, as discussed in [Li+24, Appendix D.2], this leads to a significant loss in soundness for the resulting protocol, which we briefly recall in the following.

Attack on the Fiat–Shamir transform. Recall from the proof of Theorem 2 that, for a NO instance to pass the check, the verifier’s challenges must supply at least $m - k + 1$ factors of two. Since the prover fixes its messages before the verifier samples the challenges, it receives a single sample per contraction and cannot obtain another one by changing its messages. Under this condition, each contraction supplies s additional factors with probability about 2^{-s} , and the prover passes only if the $(d + 1)t$ challenge pairs happen to supply all the missing factors, which happens with probability $\Phi(m - k + 1, (d + 1)t)$.

After the Fiat–Shamir transform, the challenges of round i are computed by querying the random oracle H on the transcript tr_i , which contains the prover’s messages of round i . The prover is no longer bound to a single sample, since it can evaluate H on many candidate messages and keep the candidate whose challenges introduce the largest number of additional factors of two. Since a single sample supplies s additional factors

with probability about 2^{-s} , about 2^s queries suffice to gain s factors of two in one round. More importantly, these gains accumulate across rounds. Once a favorable challenge has been fixed for round i , the prover repeats this for round $i + 1$ independently, so gaining $s_0, \dots, s_t$ factors of two in the $t + 1$ stages of the protocol, namely the batching step and the t folding rounds, costs about $2^{s_0} + \dots + 2^{s_t}$ queries, whereas in the interactive protocol all these gains have to occur in a single execution, which happens with probability about $2^{-(s_0 + \dots + s_t)}$.

To restore the soundness error, every stage has to resist the attack on its own. Divide the gap $m - k + 1$ into shares $s_0, \dots, s_t$ of the stages. If the prover passes, some stage has supplied at least its share of factors of two, and a single attempt does so with probability at most $\Phi(s_0, t)$ for the batching step and at most $\Phi(s_j, d)$ for the j -th folding round. To ensure soundness against 2^λ queries, it thus suffices to bound these probabilities by $2^{-\lambda}$, for which $s_0 = \lambda + O(t \log \lambda)$ and $s_j = \lambda + O(d \log \lambda)$ suffice by Lemma 4. Summing over the stages, the total gap has to be about $(t + 1)\lambda + O(dt \log \lambda)$, which is significantly larger than the gap $\lambda + O(dt \log \lambda)$ of the interactive protocol.

Gradually lifting the ring. Naively running the whole protocol over the enlarged ring incurs a large computational overhead. In the following, we discuss how to reduce this overhead. Recall that every folding round reduces a claim to another claim of smaller size. The key insight is that, while the ring $\mathbb{Z}_{2^m}$ has to be large, we do not have to carry out all these reductions directly over it. Instead, we can gradually lift the size of the ring for each reduction step. By doing this, the first several rounds of reduction, which involve most of the computation, are performed over smaller rings, and $\mathbb{Z}_{2^m}$ is only reached in the last rounds, where little computation remains.

Specifically, we fix precision levels $k < m_0 < m_1 < \dots < m_t = m$ and maintain the following invariant. The claim z_i at the beginning of round i is defined over $\mathbb{Z}_{2^{m_i}}$, and round i turns it into a claim z_{i+1} over $\mathbb{Z}_{2^{m_{i+1}}}$. To facilitate the computation at every precision level, the vectors remain over $\mathbb{Z}_{2^m}$ as in Protocol 1. A folding round proceeds as follows. The prover reduces its vectors modulo $2^{m_{i+1}}$ and computes the partial sums $z_i^{(b)}$ over $\mathbb{Z}_{2^{m_{i+1}}}$, since they have to produce the claim of the next precision level. As z_i is only known modulo 2^{m_i} , the verifier checks the consistency of the partial sums with z_i at this precision, that is, $z_i = \pi_{m_i}(z_i^{(0d)} + z_i^{(1d)})$. The sum $z_i^{(0d)} + z_i^{(1d)} \in \mathbb{Z}_{2^{m_{i+1}}}$ thereby becomes the lift of z_i to the next precision level, whose higher digits are chosen by the prover. The new claim z_{i+1} is then computed over $\mathbb{Z}_{2^{m_{i+1}}}$ from the partial sums and the challenges reduced modulo $2^{m_{i+1}}$. The prover then folds its vectors over $\mathbb{Z}_{2^m}$ as before. After t rounds, the claim is over $\mathbb{Z}_{2^{m_t}} = \mathbb{Z}_{2^m}$ and concerns the tensor evaluations of the original vectors, so the final check is the same as in Protocol 1.

The formal construction is presented in Protocol 2.

Theorem 3. *Protocol 2 is an IOP with tensor queries for the lifted arithmetic promise problem $\widehat{\text{AP}}_{\text{arith}}^{k,m}$ in the random oracle model. It is perfectly complete and, against any prover making at most Q queries to $\mathcal{H}$, has soundness error*

$$Q \cdot \delta_{\max} + \sum_{j=0}^t \delta_j \leq (Q + t + 1) \cdot \delta_{\max},$$

where $\delta_0 := \Phi(m_0 - k + 1, t)$, $\delta_{i+1} := \Phi(m_{i+1} - m_i + 1, d)$ for $i \in [t]$, and $\delta_{\max} := \max_{0 \leq j \leq t} \delta_j$.

Proof. Completeness. Suppose that the input $(\widehat{g}, (\widehat{v}, h)) \in \text{YES}$. By definition, $\widehat{G}(\widehat{v}_0, \dots, \widehat{v}_{r+1}) = 0^T$, and hence the honest prover's initial claim is $z_0 = \pi_{m_0}((\widehat{v}, 0^T)) = 0$ as required.

PROTOCOL 2. *Non-Interactive IOP for the Lifted Arithmetic Promise Problem in the ROM*

Let $T = 2^t$ and let $\widehat{G}(X_0, \dots, X_{r+1}) = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} X_{\mu_j}$ be an $(r+2)$ -variable homogeneous polynomial of degree $d \geq 2$, as in Protocol 1. Fix precision levels $k < m_0 < m_1 < \dots < m_t = m$. $\mathcal{P}$ holds $(\widehat{v}, h) \in (\mathbb{Z}_{2^m}^T)^{r+1}$, where $\widehat{v} = (\widehat{v}_0, \dots, \widehat{v}_{r-1})$. $\mathcal{V}$ has tensor-query access to $\llbracket \widehat{v} \rrbracket$ and $\llbracket h \rrbracket$ over $\mathbb{Z}_{2^m}$. Both parties have access to a random oracle H , whose output is parsed as a tuple of uniform elements of $\mathbb{Z}_{2^m}$ of the required length, and hold a string x that identifies the index and the instance, including the contents of the oracles. By a slight abuse of notation, we denote $\widehat{v}_r := h$ and $\widehat{v}_{r+1} := 1^T$. For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, set $\overline{w}_j^{(\mu)} := \widehat{v}_{\mu_j} \in \mathbb{Z}_{2^m}^T$.

- Both parties derive the batching challenges $(\bar{\theta}_i^{(0)}, \bar{\theta}_i^{(1)})_{i \in [t]} \in (\mathbb{Z}_{2^m}^2)^t$ from $H(x \parallel \text{batch})$ and set $z_0 := 0 \in \mathbb{Z}_{2^{m_0}}$. $\mathcal{P}$ computes $\bar{\theta} := \otimes_{i=0}^{t-1} (\bar{\theta}_i^{(0)}, \bar{\theta}_i^{(1)}) \in \mathbb{Z}_{2^m}^T$ and sets $\overline{w}_0^{(\mu)} \leftarrow \bar{\theta} \circ \overline{w}_0^{(\mu)}$ for every $\mu \in \widehat{\mathcal{I}}$.

- For $i = 0, \dots, t-1$:

- For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, $\mathcal{P}$ computes $w_j^{(\mu)} := \pi_{m_{i+1}}(\overline{w}_j^{(\mu)}) \in \mathbb{Z}_{2^{m_{i+1}}}^{2^{t-i}}$ and lets $w_j^{(\mu,0)}$ and $w_j^{(\mu,1)}$ be the low and high halves of $w_j^{(\mu)}$. For every $b = (b_0, \dots, b_{d-1}) \in \{0, 1\}^d$, $\mathcal{P}$ computes

$$z_i^{(b)} := \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \cdot \left\langle w_0^{(\mu, b_0)}, \dots, w_{d-1}^{(\mu, b_{d-1})} \right\rangle \in \mathbb{Z}_{2^{m_{i+1}}},$$

and sends all $\{z_i^{(b)}\}_{b \in \{0,1\}^d}$ to $\mathcal{V}$.

- $\mathcal{V}$ checks $z_i = \pi_{m_i}(z_i^{(0^d)} + z_i^{(1^d)})$, and rejects if the check fails.
- Both parties derive the folding challenges $(\bar{\gamma}_i^{(j,0)}, \bar{\gamma}_i^{(j,1)})_{j \in [d]} \in (\mathbb{Z}_{2^m}^2)^d$ from $H(x \parallel \text{fold} \parallel i \parallel \text{tr}_i)$, where $\text{tr}_i := (\{z_{i'}^{(b)}\}_{b \in \{0,1\}^d})_{i'=0}^i$ consists of the messages of rounds $0, \dots, i$, and set $\gamma_i^{(j,b)} := \pi_{m_{i+1}}(\bar{\gamma}_i^{(j,b)})$ for every $j \in [d]$ and $b \in \{0, 1\}$. Both parties then set

$$z_{i+1} := \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j, b_j)} \right) z_i^{(b)} \in \mathbb{Z}_{2^{m_{i+1}}}.$$

- For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, let $\overline{w}_j^{(\mu,0)}$ and $\overline{w}_j^{(\mu,1)}$ be the low and high halves of $\overline{w}_j^{(\mu)}$, and $\mathcal{P}$ updates the vectors as

$$\overline{w}_j^{(\mu)} \leftarrow \bar{\gamma}_i^{(j,0)} \overline{w}_j^{(\mu,0)} + \bar{\gamma}_i^{(j,1)} \overline{w}_j^{(\mu,1)} \in \mathbb{Z}_{2^m}^{2^{t-i-1}}.$$

- $\mathcal{V}$ computes the first query point $\eta_0 := ((\bar{\theta}_i^{(0)} \bar{\gamma}_i^{(0,0)}, \bar{\theta}_i^{(1)} \bar{\gamma}_i^{(0,1)}))_{i \in [t]}$, and the other query points $\eta_j := ((\bar{\gamma}_i^{(j,0)}, \bar{\gamma}_i^{(j,1)}))_{i \in [t]}$ for every $j \in \{1, \dots, d-1\}$. For every $\mu \in \widehat{\mathcal{I}}$ and $j \in [d]$, if $\widehat{v}_{\mu_j} = 1^T$, $\mathcal{V}$ locally computes its evaluation as $q_{\mu,j}$. Otherwise, $\mathcal{V}$ queries $\llbracket \widehat{v}_{\mu_j} \rrbracket$ at η_j and obtains $q_{\mu,j}$. Finally, $\mathcal{V}$ accepts if $z_t = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} q_{\mu,j}$ in $\mathbb{Z}_{2^m}$.

We show inductively that every round preserves the claimed value. At the beginning of round i , let $\overline{w}_j^{(\mu)} \in \mathbb{Z}_{2^m}^{2^{t-i}}$ be the vectors held by the honest prover, and suppose that $z_i = \pi_{m_i}(\sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \langle \overline{w}_0^{(\mu)}, \dots, \overline{w}_{d-1}^{(\mu)} \rangle)$. For

each $b \in \{0, 1\}^d$, the honest prover sends

$$z_i^{(b)} = \sum_{\mu \in \hat{\mathcal{I}}} \hat{c}_\mu \left\langle w_0^{(\mu, b_0)}, \dots, w_{d-1}^{(\mu, b_{d-1})} \right\rangle \in \mathbb{Z}_{2^{m_{i+1}}}, \quad \text{where } w_j^{(\mu)} = \pi_{m_{i+1}}(\bar{w}_j^{(\mu)}).$$

Since reduction is a ring homomorphism, $z_i^{(0^d)} + z_i^{(1^d)} = \pi_{m_{i+1}}(\sum_{\mu \in \hat{\mathcal{I}}} \hat{c}_\mu \langle \bar{w}_0^{(\mu)}, \dots, \bar{w}_{d-1}^{(\mu)} \rangle)$, whose reduction modulo 2^{m_i} is z_i , and thus the consistency check succeeds. After the challenges are derived, the verifier updates its claim to

$$\begin{aligned} z_{i+1} &= \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j, b_j)} \right) z_i^{(b)} \\ &= \pi_{m_{i+1}} \left(\sum_{\mu \in \hat{\mathcal{I}}} \hat{c}_\mu \left\langle \bar{\gamma}_i^{(0,0)} \bar{w}_0^{(\mu,0)} + \bar{\gamma}_i^{(0,1)} \bar{w}_0^{(\mu,1)}, \dots, \bar{\gamma}_i^{(d-1,0)} \bar{w}_{d-1}^{(\mu,0)} + \bar{\gamma}_i^{(d-1,1)} \bar{w}_{d-1}^{(\mu,1)} \right\rangle \right), \end{aligned}$$

where we used that $\gamma_i^{(j,b)} = \pi_{m_{i+1}}(\bar{\gamma}_i^{(j,b)})$ and again that reduction is a ring homomorphism. As specified in Protocol 2, the vectors in the last expression are exactly the folded vectors of the honest prover, so the invariant is preserved with m_{i+1} in place of m_i .

After all t rounds, each vector $\bar{w}_j^{(\mu)}$ has been reduced to a scalar, which is the tensor evaluation $q_{\mu,j}$ of $\hat{v}_{\mu_j}$ at η_j exactly as in the proof of Theorem 2, since the folds use the maximum-precision challenges. Consequently, as $m_t = m$, we have $z_t = \sum_{\mu \in \hat{\mathcal{I}}} \hat{c}_\mu \prod_{j=0}^{d-1} q_{\mu,j}$, which is precisely the verifier's final check. Hence, Protocol 2 is perfectly complete.

Soundness. For every instance $(\hat{g}, (\hat{v}, h)) \in \text{NO}$, we write $\Xi = \hat{G}(\hat{v}_0, \dots, \hat{v}_{r+1})$. By assumption, we have $\text{Po2}_m(\Xi) < k$, and hence $\text{Po2}_{m_0}(\pi_{m_0}(\Xi)) < k$ as $k \leq m_0$. Let $f_0 = \pi_{m_0}(\langle \bar{\theta}, \Xi \rangle) = \langle \theta, \pi_{m_0}(\Xi) \rangle$ denote the honest batched value, where $\theta := \otimes_{i=0}^{t-1} (\theta_i^{(0)}, \theta_i^{(1)})$. Since the verifier initializes $z_0 = 0$, the initial error is $\epsilon_0 := f_0 - z_0 = \langle \theta, \pi_{m_0}(\Xi) \rangle$. Because θ is the tensor product of t independent uniform pairs over $\mathbb{Z}_{2^{m_0}}$, Lemma 3 and the monotonicity of Φ in its first argument imply

$$\Pr[\epsilon_0 = 0] \leq \Phi(m_0 - \text{Po2}_{m_0}(\pi_{m_0}(\Xi)), t) \leq \Phi(m_0 - k + 1, t) = \delta_0.$$

We next analyze each round $i \in [t]$. Let $\bar{w}_j^{(\mu)} \in \mathbb{Z}_{2^{m_i}}^{2^{t-i}}$ denote the vectors that the honest prover holds at the beginning of round i , which are determined by the challenges so far, let $f_i \in \mathbb{Z}_{2^{m_i}}$ denote the honest value determined by these vectors, namely the reduction modulo 2^{m_i} of $\sum_{\mu \in \hat{\mathcal{I}}} \hat{c}_\mu \langle \bar{w}_0^{(\mu)}, \dots, \bar{w}_{d-1}^{(\mu)} \rangle$, let z_i denote the claimed value maintained by the verifier, and set $\epsilon_i := f_i - z_i \in \mathbb{Z}_{2^{m_i}}$. For every $b \in \{0, 1\}^d$, let $f_i^{(b)} \in \mathbb{Z}_{2^{m_{i+1}}}$ be the honest partial message determined by the current vectors, as in the completeness proof, let $z_i^{(b)}$ denote the corresponding message sent by the prover, and set $\epsilon_i^{(b)} := f_i^{(b)} - z_i^{(b)} \in \mathbb{Z}_{2^{m_{i+1}}}$. By construction, $f_i = \pi_{m_i}(f_i^{(0^d)} + f_i^{(1^d)})$. If the consistency check fails, the verifier rejects immediately. Hence, it suffices to consider the case in which $z_i = \pi_{m_i}(z_i^{(0^d)} + z_i^{(1^d)})$. Therefore, $\epsilon_i^{(0^d)} + \epsilon_i^{(1^d)}$ is a lift of ϵ_i to $\mathbb{Z}_{2^{m_{i+1}}}$. We now show that raising the precision does not create factors of two: if $u \in \mathbb{Z}_{2^q}$ satisfies $a := \text{Po2}_q(u) < q$ and $q < q'$, then every lift $\hat{u} \in \mathbb{Z}_{2^{q'}}$ of u satisfies $\text{Po2}_{q'}(\hat{u}) = a$, because writing $u = 2^a v$ with v odd gives $\hat{u} = u + 2^q c = 2^a(v + 2^{q-a} c)$ for some integer c , and $v + 2^{q-a} c$ is odd as $a < q$. Hence, whenever $\epsilon_i \neq 0$,

$$\text{Po2}_{m_{i+1}}((\epsilon_i^{(b)})_{b \in \{0,1\}^d}) \leq \text{Po2}_{m_{i+1}}(\epsilon_i^{(0^d)} + \epsilon_i^{(1^d)}) = \text{Po2}_{m_i}(\epsilon_i) \leq m_i - 1.$$

After the challenges are derived, define the honest value for the next round as

$$f_{i+1} := \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j,b_j)} \right) f_i^{(b)},$$

which is the honest value determined by the folded vectors, as shown in the completeness proof. By the definition of z_{i+1} in Protocol 2, the discrepancy after this update is

$$\epsilon_{i+1} := f_{i+1} - z_{i+1} = \sum_{b \in \{0,1\}^d} \left(\prod_{j=0}^{d-1} \gamma_i^{(j,b_j)} \right) \epsilon_i^{(b)} = \left\langle \bigotimes_{j=0}^{d-1} (\gamma_i^{(j,0)}, \gamma_i^{(j,1)}), (\epsilon_i^{(b)})_{b \in \{0,1\}^d} \right\rangle.$$

The pairs $(\gamma_i^{(j,0)}, \gamma_i^{(j,1)})$ are the reductions of fresh independent uniform pairs over $\mathbb{Z}_{2^m}^2$, so they are independent and uniform over $\mathbb{Z}_{2^{m_{i+1}}}^2$. Conditioning on the transcript before the challenges of round i are derived and applying Lemma 3 with $q = m_{i+1}$, we obtain that, whenever $\epsilon_i \neq 0$,

$$\Pr[\epsilon_{i+1} = 0] \leq \Phi(m_{i+1} - m_i + 1, d) = \delta_{i+1}.$$

This is where the gradual lifting of the ring enters, since even if ϵ_i has as many as $m_i - 1$ factors of two, it still lacks $m_{i+1} - m_i + 1$ factors of two at the new precision, all of which have to be supplied by the fresh challenges of round i .

Finally, if the verifier accepts, the final check implies $z_t = f_t$ and therefore $\epsilon_t = 0$, since $f_t = \sum_{\mu \in \widehat{\mathcal{I}}} \widehat{c}_\mu \prod_{j=0}^{d-1} q_{\mu,j}$ by the completeness proof. It remains to account for the queries of $\mathcal{A}$ to the random oracle, which we expose by lazy sampling: a query on a new input is answered with a fresh uniform value, and a repeated query receives the previous answer. We call an input of the form $(x \parallel \text{batch})$ a batching query, and an input of the form $(x \parallel \text{fold} \parallel i \parallel \text{tr}_i)$ a round- i query. Together with the oracle values on the shorter inputs determined by tr_i , which are independent of its own answer whether they are sampled before or after it, a round- i query determines the transcript up to and including the messages of round i , and hence $z_0, \dots, z_i$ and $\epsilon_0, \dots, \epsilon_i$. We call a fresh batching query *bad* if its answer gives $\epsilon_0 = 0$, and a fresh round- i query *bad* if the consistency check of round i passes, $\epsilon_i \neq 0$, and its answer gives $\epsilon_{i+1} = 0$. The answer to a fresh query is uniform and independent of everything else, so the bounds above show that a fresh batching query is bad with probability at most δ_0 and a fresh round- i query is bad with probability at most δ_{i+1} , even though $\mathcal{A}$ chooses its queries adaptively.

Now suppose that the verifier accepts the proof output by $\mathcal{A}$. Its $t + 1$ challenges are determined by one batching query and one round- i query for each $i \in [t]$, and those not made by $\mathcal{A}$ are fresh queries made during verification. Acceptance means that every consistency check passes and, as shown above, $\epsilon_t = 0$. Let i^* be the smallest index with $\epsilon_{i^*} = 0$. If $i^* = 0$, the batching query of the proof is bad; otherwise, $\epsilon_{i^*-1} \neq 0$ and the round- $(i^* - 1)$ query of the proof is bad. Hence acceptance implies a bad query among the at most Q fresh queries of $\mathcal{A}$, each bad with probability at most $\delta_{\max}$, and the at most one fresh query per stage $j \in \{0, \dots, t\}$ made during verification, bad with probability at most δ_j . A union bound gives the first inequality, and the second is immediate. $\square$

Complexity. Let $M := |\widehat{\mathcal{I}}|$. In round i , the prover computes the 2^d partial sums from vectors of length 2^{t-i} , which takes $O(d2^d M 2^{t-i})$ operations over $\mathbb{Z}_{2^{m_{i+1}}}$, and folds its $O(r)$ distinct vectors, which takes $O(r 2^{t-i})$ multiplications over $\mathbb{Z}_{2^m}$. In total, the prover performs $O(d2^d M T)$ ring operations as in Protocol 1, of which $O(d2^d M 2^{t-i})$ are over $\mathbb{Z}_{2^{m_{i+1}}}$ for each $i \in [t]$. Hence the number of operations halves with every increase of the precision, and only $O(d2^d M + rT)$ operations are over $\mathbb{Z}_{2^m}$. The verifier performs $O(2^d)$ operations

over $\mathbb{Z}_{2^{m_{i+1}}}$ in round i and $O(Md)$ operations over $\mathbb{Z}_{2^m}$ for the final check, that is, $O(2^d \log T + Md)$ ring operations in total. The proof consists of 2^d elements of $\mathbb{Z}_{2^{m_{i+1}}}$ for each $i \in [t]$, that is, $2^d \log T$ ring elements, the verifier makes at most $d(r+1)$ tensor queries over $\mathbb{Z}_{2^m}$, and the challenges are derived with $\log T + 1$ queries to H .

Remark 2 (Higher-arity folding). *Protocol 2 naturally extends to arbitrary folding arity $B \geq 2$ as in Remark 1, with $\log_B T$ folding rounds and $\log_B T + 1$ precision levels. Since the maximum precision grows with the number of precision levels, a larger folding arity reduces the required gap $m - k$ to about $(\log_B T + 1)\lambda$, which allows using a smaller m for the same target security level.*

5 Proximity Gap of Linear Codes over Rings

In this section, we study the proximity gap of linear codes over the ring $\mathbb{Z}_{2^m}$.

5.1 Lifted Codes and Visible Distances

We start by defining the notation used in the analysis.

Linear codes. Let $\mathcal{C} \subseteq \mathbb{F}_2^N$ be a binary linear $[N, n, d]$ code, that is, a code of length N , dimension n and minimum distance d , with generator matrix $G \in \mathbb{F}_2^{n \times N}$. All codes in this paper are of this form.

Lifted codes. By identifying the entries of G with $\{0, 1\} \subseteq \mathbb{Z}_{2^q}$, we regard G as a matrix over $\mathbb{Z}_{2^q}$ for every $q \geq 1$. For $x \in \mathbb{Z}_{2^q}^n$, let $\text{Enc}_q(x) := xG$, and define the *lifted code* of $\mathcal{C}$ over $\mathbb{Z}_{2^q}$ as $\mathcal{C}_q := \{\text{Enc}_q(x) : x \in \mathbb{Z}_{2^q}^n\} \subseteq \mathbb{Z}_{2^q}^N$. Since G is binary, $\pi_q \circ \text{Enc}_{q'} = \text{Enc}_q \circ \pi_q$ for all $q \leq q'$. We first show that lifting preserves the minimum distance.

Lemma 5. *For every $q \geq 1$, the lifted code $\mathcal{C}_q$ has minimum distance d .*

Proof. Let $c = \text{Enc}_q(x) \neq 0$ for some $x \in \mathbb{Z}_{2^q}^n$, and let $u := \text{Po}2_q(x)$. Since $x \neq 0$, we have $u < q$. Set $x = 2^u x'$, where x' has at least one odd coordinate and hence $\pi_1(x')$ is nonzero. By the identity $\pi_1 \circ \text{Enc}_q = \text{Enc}_1 \circ \pi_1$, we have $\pi_1(\text{Enc}_q(x')) = \text{Enc}_1(\pi_1(x'))$ is a nonzero codeword of $\mathcal{C}$ and therefore has Hamming weight at least d . For every coordinate j in the support of $\pi_1(\text{Enc}_q(x'))$, the entry $\text{Enc}_q(x')_j$ is odd. Thus, $c_j = \text{Enc}_q(x)_j = 2^u \text{Enc}_q(x')_j \neq 0$. Hence $\text{wt}(c) \geq d$.

Conversely, choose $\bar{x} \in \mathbb{F}_2^n$ such that $\text{wt}(\text{Enc}_1(\bar{x})) = d$ and regard $\bar{x}$ as an element of $\mathbb{Z}_{2^q}^n$. Then $\text{Enc}_q(2^{q-1}\bar{x}) = 2^{q-1}\text{Enc}_q(\bar{x})$. Its j -th coordinate is nonzero in $\mathbb{Z}_{2^q}$ exactly when $\text{Enc}_1(\bar{x})_j = 1$ in $\mathbb{F}_2$, which implies $\text{wt}(\text{Enc}_q(2^{q-1}\bar{x})) = d$. Thus $\mathcal{C}_q$ has minimum distance d . $\square$

Interleaved codes and distances. For an integer $r \geq 1$, we extend Enc_q to matrices by defining $\text{Enc}_q(X) := XG$ for every $X \in \mathbb{Z}_{2^q}^{r \times n}$, and define the r -fold interleaved code as $\mathcal{C}_q^{(r)} := \{\text{Enc}_q(X) : X \in \mathbb{Z}_{2^q}^{r \times n}\} \subseteq \mathbb{Z}_{2^q}^{r \times N}$. We view $\mathcal{C}_q^{(r)}$ as a length- N block code over the alphabet $\mathbb{Z}_{2^q}^r$. In particular, its elements are naturally identified with matrices in $\mathbb{Z}_{2^q}^{r \times N}$. For $X \in \mathbb{Z}_{2^q}^{r \times N}$ we write $X_{:,j}$ for its j -th column, and set $\text{supp}(X) := \{j \in [N] : X_{:,j} \neq 0\}$ and $\text{wt}(X) := |\text{supp}(X)|$. For $X, Y \in \mathbb{Z}_{2^q}^{r \times N}$, we write $\Delta^{(r)}(X, Y) := \text{supp}(X - Y)$ for the set of columns at which X and Y differ, called their *disagreement set*, and $d(X, Y) := |\Delta^{(r)}(X, Y)|$. For a nonempty set $\mathcal{S} \subseteq \mathbb{Z}_{2^q}^{r \times N}$, we further write $d(X, \mathcal{S}) := \min_{Y \in \mathcal{S}} d(X, Y)$.

Visible distance. The standard Hamming distance captures the number of disagreements. We additionally record how visible each disagreement is.

Definition 5. For $X, Y \in \mathbb{Z}_{2^\ell}^{r \times N}$ and $q \in \{0, \dots, \ell\}$, the q -visible disagreement set and the q -visible distance are

$$\Delta_q^{(r)}(X, Y) := \{j \in [N] : \text{Po}2_\ell(X_{:,j} - Y_{:,j}) < q\}, \quad d_q(X, Y) := \left| \Delta_q^{(r)}(X, Y) \right|.$$

For a nonempty set $\mathcal{S} \subseteq \mathbb{Z}_{2^\ell}^{r \times N}$, we write $d_q(X, \mathcal{S}) := \min_{Y \in \mathcal{S}} d_q(X, Y)$, and we say that X is (e, q) -far from $\mathcal{S}$ if $d_q(X, \mathcal{S}) > e$.

Intuitively, the q -visible distance is the Hamming distance observed after reduction to $\mathbb{Z}_{2^q}$. As an immediate consequence of the definition, the visible distance is monotone in the parameter q .

Theorem 4. For every $X \in \mathbb{Z}_{2^\ell}^{r \times N}$, every nonempty set $\mathcal{S} \subseteq \mathbb{Z}_{2^\ell}^{r \times N}$, and all $0 \leq q \leq q' \leq \ell$, we have $d_q(X, \mathcal{S}) \leq d_{q'}(X, \mathcal{S})$.

Error precision and error profile. Fix an error bound $e \geq 0$. For every $j \in \{0, \dots, e\}$, we define the j -error precision of X as the largest q at which X remains within j errors of the code after reduction to $\mathbb{Z}_{2^q}$.

Definition 6. For $X \in \mathbb{Z}_{2^\ell}^{r \times N}$ and $j \in \{0, \dots, e\}$, the j -error precision of X is

$$\rho_j(X) := \max \left\{ q \in \{0, \dots, \ell\} : d_q(X, \mathcal{C}_\ell^{(r)}) \leq j \right\}.$$

By monotonicity of the visible distance, $\rho_j(X) \geq q$ if and only if $d_q(X, \mathcal{C}_\ell^{(r)}) \leq j$. Particularly, this implies X is (j, q) -far from $\mathcal{C}_\ell^{(r)}$ if and only if $\rho_j(X) < q$. Since increasing the error bound can only increase the precision, we have $\rho_0(X) \leq \rho_1(X) \leq \dots \leq \rho_e(X)$. Consequently, if X is (e, q) -far from $\mathcal{C}_\ell^{(r)}$, then $\rho_j(X) < q$ for every $j \leq e$. We call $(\rho_0(X), \rho_1(X), \dots, \rho_e(X))$ the *error profile* of X .

5.2 Main Result

In this part, we prove the following theorem.

Theorem 5. Let $\mathcal{C}_\ell \subseteq \mathbb{Z}_{2^\ell}^N$ be the lift of an $[N, n, d]$ binary code, let $r \geq 2$ be a power of two, and let $e \in \{0, \dots, \lfloor \frac{d-1}{3} \rfloor\}$. Let $U := (u_i)_{i=0}^{r-1} \in \mathbb{Z}_{2^\ell}^{r \times N}$, and suppose that U is (e, m) -far from $\mathcal{C}_\ell^{(r)}$. Then, for independent uniform $\alpha_0, \beta_0, \dots, \alpha_{\log r-1}, \beta_{\log r-1}$ over $\mathbb{Z}_{2^\ell}$,

$$\Pr \left[d \left(\left[\bigotimes_{i=0}^{\log r-1} (\alpha_i, \beta_i) \right] \cdot U, \mathcal{C}_\ell \right) \leq e \right] \leq (e+1) \Phi(\ell - m + 1, \log r).$$

At a high level, we view the random tensor combination as a sequence of $\log r$ binary folds. If the input is far from the code after reduction to $\mathbb{Z}_{2^m}$, then all error precisions should lie below m . On the other hand, if the final folded word is e -close to the code over $\mathbb{Z}_{2^\ell}$, then $\rho_e = \ell$. Thus, it suffices to understand how the error profile can increase over the course of these folds.

We begin with a single fold, which depends on two random challenges. For clarity, we write $X = \begin{pmatrix} X_0 \\ X_1 \end{pmatrix} \in \mathbb{Z}_{2^\ell}^{2t \times N}$ for a matrix split into two blocks, and for a coefficient pair $p = (\alpha, \beta) \in \mathbb{Z}_{2^\ell}^2$, we define the *fold* $F_p(X) := \alpha X_0 + \beta X_1 \in \mathbb{Z}_{2^\ell}^{t \times N}$.

Theorem 6. Let $X \in \mathbb{Z}_{2^\ell}^{2t \times N}$ and $a_j := \rho_j(X)$ for $j \in \{0, \dots, e\}$. For every $q \in \{1, \dots, \ell\}$ and uniform $p \leftarrow \mathbb{Z}_{2^\ell}^2$,

$$\sum_{\substack{0 \leq j \leq e \\ a_j < q}} \Pr_p [\rho_j(F_p(X)) \geq q] \leq \sum_{\substack{0 \leq j \leq e \\ a_j < q}} 2^{-(q-a_j)}.$$

Proof. Fix q . Since $\pi_q(F_p(X)) = F_{\pi_q(p)}(\pi_q(X))$ and the q -visible distance is the ordinary distance after reduction modulo 2^q , the event $\rho_j(F_p(X)) \geq q$ depends only on p and X modulo 2^q . Moreover, a uniform $p \leftarrow \mathbb{Z}_{2^\ell}^2$ reduces to a uniform element of $\mathbb{Z}_{2^q}^2$. Hence we may work over $\mathbb{Z}_{2^q}$ and, with a slight abuse of notation, write X for $\pi_q(X)$.

For $j \in \{0, \dots, e\}$, define the *passing sets* as

$$\mathcal{S}_j := \left\{ p \in \mathbb{Z}_{2^q}^2 : d\left(F_p(X), \mathcal{C}_q^{(t)}\right) \leq j \right\}.$$

Clearly, $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \dots \subseteq \mathcal{S}_e$. For $\mathcal{S} \subseteq \mathbb{Z}_{2^q}^2$, write $\Pr[\mathcal{S}] := |\mathcal{S}|/2^{2q}$. Then by the definition of $\mathcal{S}_j$, we have $\Pr_p[\rho_j(F_p(X)) \geq q] = \Pr[\mathcal{S}_j]$.

We start by analyzing the structure of subsets in $\mathbb{Z}_{2^q}^2$.

Lemma 6. *Let $\mathcal{S} \subseteq \mathbb{Z}_{2^q}^2$ be nonempty and $h := \min_{p,p' \in \mathcal{S}} \text{Po2}_q(\det(p, p'))$. Then $\Pr[\mathcal{S}] \leq 2^{-h}$.*

Proof. If $\mathcal{S} = \{0\}$, the claim is immediate. Otherwise let $s_0 := \min\{\text{Po2}_q(p) : p \in \mathcal{S}\} < q$, and choose $p_0 \in \mathcal{S}$ attaining this minimum. There exists an invertible matrix $Q \in \mathbb{Z}_{2^q}^{2 \times 2}$ such that $p_0 Q = (2^{s_0}, 0)$. For $p \in \mathcal{S}$, let $pQ = (x, y)$. We claim that $\text{Po2}_q(pQ) = \text{Po2}_q(p)$. Indeed, each coordinate of pQ is a linear combination of the coordinates of p , which implies $\text{Po2}_q(pQ) \geq \text{Po2}_q(p)$. The reverse implication follows in the same way from $p = (pQ)Q^{-1}$. Hence, by the definition of s_0 , it follows that $\text{Po2}_q(pQ) \geq s_0$ for every $p \in \mathcal{S}$.

Now set $s_1 := \min\{\text{Po2}_q(y) : pQ = (x, y), p \in \mathcal{S}\}$, and choose $p_1 \in \mathcal{S}$ such that $p_1 Q = (x_1, y_1)$ with $\text{Po2}_q(y_1) = s_1$. Since $\text{Po2}_q(x_1) \geq s_0$, subtracting a suitable multiple of $p_0 Q$ from $p_1 Q$ eliminates its first coordinate. Hence, for $\mathcal{H} := \text{RowSpan}_{\mathbb{Z}_{2^q}}(p_0, p_1)$, we have $\mathcal{H}Q = 2^{s_0}\mathbb{Z}_{2^q} \times 2^{s_1}\mathbb{Z}_{2^q}$. Moreover, by the definitions of s_0 and s_1 , every $pQ = (x, y)$ with $p \in \mathcal{S}$ satisfies $x \in 2^{s_0}\mathbb{Z}_{2^q}$ and $y \in 2^{s_1}\mathbb{Z}_{2^q}$. Thus $\mathcal{S} \subseteq \mathcal{H}$.

Next, take arbitrary $p, p' \in \mathcal{S}$, and write $pQ = (x, y)$ and $p'Q = (x', y')$. Since $x, x' \in 2^{s_0}\mathbb{Z}_{2^q}$ and $y, y' \in 2^{s_1}\mathbb{Z}_{2^q}$, we have $\det(pQ, p'Q) = xy' - x'y \in 2^{s_0+s_1}\mathbb{Z}_{2^q}$. Consequently, $\text{Po2}_q(\det(p, p')) = \text{Po2}_q(\det(pQ, p'Q)) \geq \min\{s_0 + s_1, q\}$. On the other hand, $\det(p_0 Q, p_1 Q) = 2^{s_0}y_1$, so $\text{Po2}_q(\det(p_0, p_1)) = \min\{s_0 + s_1, q\}$. Thus, $h = \min\{s_0 + s_1, q\}$. Finally, since right multiplication by Q is a bijection, $|\mathcal{S}| \leq |\mathcal{H}| = |2^{s_0}\mathbb{Z}_{2^q}| \cdot |2^{s_1}\mathbb{Z}_{2^q}| = 2^{q-s_0}2^{q-s_1} \leq 2^{2q-h}$. $\square$

Let $h := \min_{p,p' \in \mathcal{S}_e} \text{Po2}_q(\det(p, p'))$, then we have $\Pr[\mathcal{S}_e] \leq 2^{-h}$. If $h = q$, then $\Pr[\mathcal{S}_j] \leq \Pr[\mathcal{S}_e] \leq 2^{-q} \leq 2^{-(q-a_j)}$ for every j with $a_j < q$, so the desired bound already follows. We may therefore assume $h < q$ in the following.

Choose $p_0, p_1 \in \mathcal{S}_e$ as in the proof of Lemma 6, so that $\text{Po2}_q(\det(p_0, p_1)) = h$. Then there exist $C_0, C_1 \in \mathcal{C}_q^{(t)}$ and errors $\Xi_0, \Xi_1 \in \mathbb{Z}_{2^q}^{t \times N}$ such that

$$F_{p_0}(X) = C_0 + \Xi_0, \quad F_{p_1}(X) = C_1 + \Xi_1,$$

with $\text{wt}(\Xi_i) \leq e$ for $i \in \{0, 1\}$. Let $E := \text{supp}(\Xi_0) \cup \text{supp}(\Xi_1)$, so $|E| \leq 2e < d$.

Let A be the matrix with rows p_0, p_1 . By the proof of Lemma 6, there exist invertible matrices $P, Q \in \mathbb{Z}_{2^q}^{2 \times 2}$ such that $PAQ = \text{diag}(2^{s_0}, 2^{s_1})$ with $s_0 + s_1 = h$. Moreover, $\mathcal{H} := \text{RowSpan}_{\mathbb{Z}_{2^q}}(p_0, p_1)$ satisfies $\mathcal{H}Q = 2^{s_0}\mathbb{Z}_{2^q} \times 2^{s_1}\mathbb{Z}_{2^q}$, $\mathcal{S}_e \subseteq \mathcal{H}$ and $|\mathcal{H}| = 2^{2q-h}$. Define $Y := \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} := Q^{-1} \begin{pmatrix} X_0 \\ X_1 \end{pmatrix}$, where Q^{-1} acts on the two blocks X_0, X_1 . Then, we have $PAQ \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} = P \begin{pmatrix} C_0 \\ C_1 \end{pmatrix} + P \begin{pmatrix} \Xi_0 \\ \Xi_1 \end{pmatrix}$. This further implies

$$2^{s_0}Y_0 = C'_0 + \Xi'_0, \quad 2^{s_1}Y_1 = C'_1 + \Xi'_1,$$

where $C'_0, C'_1 \in \mathcal{C}_q^{(t)}$ and $\text{supp}(\Xi'_0), \text{supp}(\Xi'_1) \subseteq E$. Additionally, since Q is invertible, it preserves the visible distance to the code at every level. That is, $d_{q'}(Y, \mathcal{C}_q^{(2t)}) = d_{q'}(X, \mathcal{C}_q^{(2t)})$ for every $q' \leq q$.

Let J be the number of indices $j \in \{0, \dots, e\}$ such that $a_j + h < q$. By monotonicity of the error precision, these are precisely $j = 0, \dots, J - 1$. Fix such a j . By the definition of a_j , we have $d_{a_j+1}(X, C_\ell^{(2t)}) > j$ and therefore $d_{a_j+1}(Y, C_q^{(2t)}) > j$.

To exploit this distance lower bound, we would like to compare each Y_i directly with a codeword in $C_q^{(t)}$. However, the argument above only shows that $2^{s_i} Y_i$ is close to a codeword C'_i . Since 2^{s_i} is not invertible in $\mathbb{Z}_{2^q}$, we cannot simply divide this relation by 2^{s_i} . We therefore take a different approach. Specifically, we demonstrate that the codeword C'_i itself contains a factor of 2^{s_i} . This allows us to isolate the common factor in the relation and analyze the remaining part separately.

Lemma 7. *Let $E \subseteq [N]$ satisfy $|E| < d$, let $a \in \{0, \dots, q - 1\}$, and let $0 \leq s < q - a$. Suppose that $W \in \mathbb{Z}_{2^q}^{r \times N}$, $C \in C_q^{(r)}$ and $\Xi \in \mathbb{Z}_{2^q}^{r \times N}$ satisfy $2^s W = C + \Xi$ with $\text{supp}(\Xi) \subseteq E$. Then there exists $V \in C_q^{(r)}$ such that $2^s V = C$ and $\Delta_{a+1}^{(r)}(W, V) \subseteq E$. In particular, $d_{a+1}(W, V) \leq |E|$.*

Proof. We first claim that for every $E \subseteq [N]$ with $|E| < d$ there exists $I \subseteq [N] \setminus E$ with $|I| = n$ such that the submatrix $G_I \in \mathbb{Z}_{2^q}^{n \times n}$ is invertible, where G is the generator matrix of C_q . Indeed, since $|E| < d$, the submatrix $G_{[N] \setminus E}$ has rank n over $\mathbb{F}_2$. Hence there is a set $I \subseteq [N] \setminus E$ of size n for which $\det(G_I)$ is nonzero over $\mathbb{F}_2$. Therefore, the determinant of G_I is odd when considered on $\mathbb{Z}_{2^q}$ and is also invertible. Write $C = MG$. Restricting the assumed equality to I gives $2^s W_I = M G_I$, hence $M = 2^s W_I G_I^{-1}$. Write $M = 2^s M'$ and set $V := M' G$. Then $V \in C_q^{(r)}$ and $2^s V = C$. For every $j \notin E$, we have $2^s(W_{:,j} - V_{:,j}) = 0$. Thus $\text{Po2}_q(W_{:,j} - V_{:,j}) \geq q - s > a$. Hence $\Delta_{a+1}^{(r)}(W, V) \subseteq E$ and $d_{a+1}(W, V) = |\Delta_{a+1}^{(r)}(W, V)| \leq |E|$. $\square$

Since $s_i \leq h < q - a_j$, Lemma 7 implies that there exist $V_j^0, V_j^1 \in C_q^{(t)}$ such that $C'_i = 2^{s_i} V_j^i$ for $i \in \{0, 1\}$ and $\Delta_{a_j+1}^{(t)}(Y_i, V_j^i) \subseteq E$. Let $V_j := \begin{pmatrix} V_j^0 \\ V_j^1 \end{pmatrix}$. Since $d_{a_j+1}(Y, C_q^{(2t)}) > j$ and $V_j \in C_q^{(2t)}$, it must hold that $d_{a_j+1}(Y, V_j) \geq j + 1$.

For $v \in E$, define $\mu_v := \min_{i \in \{0,1\}} \text{Po2}_q(\Xi'_{i,:}, v)$. Since $2^{s_i} V_j^i = C'_i$, we have $\Xi'_i = 2^{s_i}(Y_i - V_j^i)$. As shown above, there are at least $j + 1$ columns $v \in E$ for which $\min_{i \in \{0,1\}} \text{Po2}_q(Y_{i,:}, v - V_{j,:}^i) \leq a_j$. For each such column, choose $i \in \{0, 1\}$ attaining this minimum. Since $a_j + s_i \leq a_j + h < q$, we get $\mu_v \leq \text{Po2}_q(\Xi'_{i,:}, v) = s_i + \text{Po2}_q(Y_{i,:}, v - V_{j,:}^i) \leq a_j + h$ for each of these $j + 1$ columns.

Order the values $\{\mu_v\}_{v \in E}$ increasingly, and let $v_0, v_1, \dots$ denote the corresponding columns. Since, for every $j < J$, at least $j + 1$ of these values are at most $a_j + h$, the $(j + 1)$ -st smallest one satisfies $\mu_{v_j} \leq a_j + h$ for every $0 \leq j < J$.

We now bound the sizes of the passing sets. Sample $\theta = (\theta_0, \theta_1)$ uniformly from $\mathbb{Z}_{2^q}^2$ and set $p := \theta \text{diag}(2^{s_0}, 2^{s_1}) Q^{-1}$. Since $\mathcal{H}Q = 2^{s_0} \mathbb{Z}_{2^q} \times 2^{s_1} \mathbb{Z}_{2^q}$, p is uniformly distributed over $\mathcal{H}$. For such p , we have

$$F_p(X) = \theta_0 (2^{s_0} Y_0) + \theta_1 (2^{s_1} Y_1) = (\theta_0 C'_0 + \theta_1 C'_1) + \Xi_\theta,$$

where $\Xi_\theta := \theta_0 \Xi'_0 + \theta_1 \Xi'_1$. The first term belongs to $C_q^{(t)}$, while $\text{supp}(\Xi_\theta) \subseteq E$. Therefore, $\text{wt}(\Xi_\theta) \leq 2e$.

Suppose that $p \in \mathcal{S}_j$. Then there exist $\hat{C} \in C_q^{(t)}$ and an error term $\hat{\Xi}$ with $\text{wt}(\hat{\Xi}) \leq j$ such that $F_p(X) = \hat{C} + \hat{\Xi}$. This suggests

$$\hat{C} - (\theta_0 C'_0 + \theta_1 C'_1) = \Xi_\theta - \hat{\Xi}.$$

The right-hand side has weight at most $2e + j \leq 3e < d$. Since the left-hand side is a codeword and the minimum distance of the code is d , it must therefore be the zero codeword. Consequently, $\Xi_\theta = \hat{\Xi}$ and hence $\text{wt}(\Xi_\theta) \leq j$.

The above reasoning shows that $p \in \mathcal{S}_j$ only if $\text{wt}(\Xi_\theta) \leq j$. Therefore, there are at most $\max\{J - \text{wt}(\Xi_\theta), 0\}$ indices $j \in [J]$ such that $p \in \mathcal{S}_j$. On the other hand, among the J distinguished columns $v_0, \dots, v_{J-1}$, at most $\text{wt}(\Xi_\theta)$ can contain a nonzero block of Ξ_θ . Hence at least $\max\{J - \text{wt}(\Xi_\theta), 0\}$ of these columns contain a zero block, which implies $|\{j \in [J] : p \in \mathcal{S}_j\}| \leq |\{j \in [J] : (\Xi_\theta)_{:,v_j} = 0\}|$. Taking expectations over θ gives

$$\sum_{j=0}^{J-1} \Pr_{p \leftarrow \mathcal{H}} [p \in \mathcal{S}_j] \leq \sum_{j=0}^{J-1} \Pr_{\theta} [(\Xi_\theta)_{:,v_j} = 0].$$

By Lemma 2 and $\mu_{v_j} \leq a_j + h < q$, each term on the right is at most $2^{-(q-\mu_{v_j})}$. Hence

$$\sum_{j=0}^{J-1} \Pr_{p \leftarrow \mathcal{H}} [p \in \mathcal{S}_j] \leq \sum_{j=0}^{J-1} 2^{-(q-\mu_{v_j})}.$$

Since $\mathcal{S}_j \subseteq \mathcal{S}_e \subseteq \mathcal{H}$ and $|\mathcal{H}| = 2^{2q-h}$, we have $\Pr[\mathcal{S}_j] = 2^{-h} \Pr_{p \leftarrow \mathcal{H}} [p \in \mathcal{S}_j]$, and we obtain

$$\sum_{j=0}^{J-1} \Pr[\mathcal{S}_j] \leq \sum_{j=0}^{J-1} 2^{-h} 2^{-(q-\mu_{v_j})} \leq \sum_{j=0}^{J-1} 2^{-(q-a_j)},$$

where the last inequality follows from $\mu_{v_j} \leq a_j + h$.

It remains to consider the indices $j \geq J$ for which $a_j < q$. By the definition of J , we have $a_j + h \geq q$. Since $\mathcal{S}_j \subseteq \mathcal{S}_e \subseteq \mathcal{H}$, it follows that $\Pr[\mathcal{S}_j] \leq 2^{-h} \leq 2^{-(q-a_j)}$. Combining the two ranges gives the desired bound. $\square$

We next show how the result for single fold can be iteratively applied to obtain the final results. For $X \in \mathbb{Z}_{2^\ell}^{r \times N}$ and integer q , we define the *crossing count* as the number of error precisions that have reached q , namely

$$K_X(q) := |\{j \in \{0, \dots, e\} : \rho_j(X) \geq q\}|. \quad (4)$$

Corollary 1. *Let $X \in \mathbb{Z}_{2^\ell}^{2t \times N}$, let p be uniform over $\mathbb{Z}_{2^\ell}^{2t}$, and let Γ be the geometric random variable of Section 3.1, so that $\Pr[\Gamma \geq s] = 2^{-s}$ for every $s \geq 0$. Then, for every integer $q \leq \ell$,*

$$\mathbb{E}_p [K_{F_p(X)}(q)] \leq \mathbb{E}_\Gamma [K_X(q - \Gamma)]. \quad (5)$$

Proof. If $q \leq 0$, both sides of (5) equal $e + 1$, so we assume $1 \leq q \leq \ell$ in the following. Let $a_j := \rho_j(X)$ for every $j = 0, \dots, e$. By the definition of crossing count, $\mathbb{E}_p [K_{F_p(X)}(q)] = \sum_{j=0}^e \Pr_p [\rho_j(F_p(X)) \geq q]$. We split this sum according to a_j . If $a_j \geq q$, then $d_q(X, \mathcal{C}_\ell^{(2t)}) \leq j$. Choose $V = \begin{pmatrix} V_0 \\ V_1 \end{pmatrix} \in \mathcal{C}_\ell^{(2t)}$ such that $d_q(X, V) \leq j$. Outside $\Delta_q^{(2t)}(X, V)$ we have $\pi_q(X_i) = \pi_q(V_i)$ for $i \in \{0, 1\}$. Hence, for every $p = (\alpha, \beta)$, we have $\pi_q(F_p(X)) = \pi_q(\alpha V_0 + \beta V_1)$ outside the same set of columns. Since $\alpha V_0 + \beta V_1 \in \mathcal{C}_\ell^{(t)}$, it follows that $d_q(F_p(X), \mathcal{C}_\ell^{(t)}) \leq j$ and therefore $\rho_j(F_p(X)) \geq q$ for every p . This leads to $\Pr_p [\rho_j(F_p(X)) \geq q] = 1$. On the other hand, since $a_j \geq q$ and $\Gamma \geq 0$, we also have $\Pr[a_j + \Gamma \geq q] = 1$. Therefore, for every j with $a_j \geq q$, $\Pr_p [\rho_j(F_p(X)) \geq q] = \Pr[a_j + \Gamma \geq q] = 1$.

It remains to consider the indices with $a_j < q$. By Theorem 6, we have $\sum_{j:a_j < q} \Pr_p [\rho_j(F_p(X)) \geq q] \leq \sum_{j:a_j < q} 2^{-(q-a_j)}$. By the distribution of Γ , $2^{-(q-a_j)} = \Pr[\Gamma \geq q - a_j] = \Pr[a_j + \Gamma \geq q]$. Hence

$$\sum_{j:a_j < q} \Pr_p [\rho_j(F_p(X)) \geq q] \leq \sum_{j:a_j < q} \Pr[a_j + \Gamma \geq q].$$

Finally, for every fixed Γ ,

$$K_X(q - \Gamma) = |\{j \in \{0, \dots, e\} : a_j + \Gamma \geq q\}|.$$

Taking the expectation over Γ yields $\mathbb{E}_\Gamma [K_X(q - \Gamma)] = \sum_{j=0}^e \Pr[a_j + \Gamma \geq q]$. Consequently, $\mathbb{E}_p [K_{F_p(X)}(q)] \leq \mathbb{E}_\Gamma [K_X(q - \Gamma)]$. $\square$

We now consider the fold process. Set $U^{(0)} := U$. For every $i \in \{0, \dots, \log r - 1\}$, split $U^{(i)} \in \mathbb{Z}_{2^\ell}^{(r/2^i) \times N}$ as $U^{(i)} = \begin{pmatrix} U_0^{(i)} \\ U_1^{(i)} \end{pmatrix}$ into two blocks of equal size. Given $\alpha_i, \beta_i \in \mathbb{Z}_{2^\ell}$, we define $U^{(i+1)} := \alpha_i U_0^{(i)} + \beta_i U_1^{(i)} = F_{(\alpha_i, \beta_i)}(U^{(i)})$.

Theorem 7. *Let $\Gamma_1, \Gamma_2, \dots$ be independent copies of Γ and put $S_i := \Gamma_1 + \dots + \Gamma_i$, with $S_0 := 0$. For every $i \in \{0, \dots, \log r\}$ and every integer $q \leq \ell$,*

$$\mathbb{E} [K_{U^{(i)}}(q)] \leq \mathbb{E}_{S_i} [K_U(q - S_i)], \quad (6)$$

where the expectation on the left is over the randomness of the first i folds.

Proof. We proceed by induction on i . For $i = 0$, we have $U^{(0)} = U$ and $S_0 = 0$, and hence

$$\mathbb{E} [K_{U^{(0)}}(q)] = K_U(q) = \mathbb{E}_{S_0} [K_U(q - S_0)].$$

Now suppose that the claim holds for some $i < \log r$. Conditioning on $U^{(i)}$ and applying Corollary 1 to the fresh randomness $p_i = (\alpha_i, \beta_i)$ gives

$$\mathbb{E} [K_{U^{(i+1)}}(q) \mid U^{(i)}] \leq \mathbb{E}_{\Gamma_{i+1}} [K_{U^{(i)}}(q - \Gamma_{i+1})].$$

Taking the expectation over the first i folds, we obtain

$$\mathbb{E} [K_{U^{(i+1)}}(q)] \leq \mathbb{E}_{\Gamma_{i+1}} [\mathbb{E} [K_{U^{(i)}}(q - \Gamma_{i+1})]].$$

Since $\Gamma_{i+1} \geq 0$, we have $q - \Gamma_{i+1} \leq \ell$, so the induction hypothesis applies at the shifted target $q - \Gamma_{i+1}$. Therefore,

$$\mathbb{E} [K_{U^{(i+1)}}(q)] \leq \mathbb{E}_{\Gamma_{i+1}} [\mathbb{E}_{S_i} [K_U(q - \Gamma_{i+1} - S_i)]] = \mathbb{E}_{S_{i+1}} [K_U(q - S_{i+1})],$$

where the last equality follows from the independence of S_i and Γ_{i+1} and $S_{i+1} = S_i + \Gamma_{i+1}$. This proves the claim for $i + 1$ and completes the induction. $\square$

With the above theorems, we now prove our main result.

Proof of Theorem 5. By hypothesis U is (e, m) -far from $\mathcal{C}_\ell^{(r)}$. Hence $\rho_j(U) < m$ for every $j \in \{0, \dots, e\}$. If the folded word is accepted, namely $d(U^{(\log r)}, \mathcal{C}_\ell) \leq e$, then $\rho_e(U^{(\log r)}) = \ell$ and hence $K_{U^{(\log r)}}(\ell) \geq 1$. Therefore, $\Pr[\text{accept}] \leq \mathbb{E} [K_{U^{(\log r)}}(\ell)]$. Applying Theorem 7 with $i = \log r$ and $q = \ell$ gives $\Pr[\text{accept}] \leq \mathbb{E}_{S_{\log r}} [K_U(\ell - S_{\log r})]$. By the definition of the crossing count,

$$\mathbb{E}_{S_{\log r}} [K_U(\ell - S_{\log r})] = \sum_{j=0}^e \Pr[S_{\log r} \geq \ell - \rho_j(U)].$$

Since $\rho_j(U) \leq m - 1$ for every j , we have $\ell - \rho_j(U) \geq \ell - m + 1$. Therefore,

$$\Pr[\text{accept}] \leq \sum_{j=0}^e \Pr[S_{\log r} \geq \ell - m + 1] = (e + 1)\Phi(\ell - m + 1, \log r),$$

where the final equality follows from Lemma 1. $\square$

6 A Commitment Scheme for Rings

In this section, we construct a Brakedown-style commitment scheme supporting tensor evaluations of vectors over $\mathbb{Z}_{2^m}$, using the proximity gap from Section 5.

6.1 Definition of Ring Commitment Scheme

Syntax. We first formally define the ring commitment scheme.

Definition 7 (Ring commitment). *A ring commitment scheme is a tuple of algorithms $\Pi = (\text{Setup}, \text{Commit}, \text{Open}, \text{Prove}, \text{Verify})$ with the following syntax.*

- $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$. On input the security parameter λ and a size parameter $t = O(\log \lambda)$, $\Pi.\text{Setup}$ samples public parameters pp , including (among other things) the knowledge domain $\mathbb{Z}_{2^k}$, the evaluation domain $\mathbb{Z}_{2^m}$, and a split $t = t_0 + t_1$ with $t_0, t_1 \geq 1$. Define $r_0 := 2^{t_0}$, $r_1 := 2^{t_1}$ and $T := r_0 r_1 = 2^t$.
- $(c, u) \leftarrow \Pi.\text{Commit}(\text{pp}, v)$. On input a vector $v \in \mathbb{Z}_{2^m}^T$, $\Pi.\text{Commit}$ returns a commitment c to v , and an opening hint u .
- $b \leftarrow \Pi.\text{Open}(\text{pp}, c; \bar{v}, u)$. On input a commitment c , a claimed low-precision opening $\bar{v} \in \mathbb{Z}_{2^k}^T$, and an opening hint u , $\Pi.\text{Open}$ verifies whether $\bar{v}$ is a valid low-precision opening of c using u . We set $\Pi.\text{Open}(\text{pp}, c; \perp, u) := 0$.
- $\pi \leftarrow \Pi.\text{Prove}(\text{pp}, c, s, \eta; v, u)$. On input a commitment c , a purported evaluation $s \in \mathbb{Z}_{2^m}$, an evaluation point $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$, a vector $v \in \mathbb{Z}_{2^m}^T$, and an opening hint u , $\Pi.\text{Prove}$ generates an evaluation proof π for the claim $\text{Tensor}(\eta) \cdot v = s$.
- $b \leftarrow \Pi.\text{Verify}(\text{pp}, c, s, \eta, \pi)$. On input a commitment c , a purported evaluation $s \in \mathbb{Z}_{2^m}$, an evaluation point $\eta \in \mathbb{Z}_{2^m}^{2t}$, and a proof π , $\Pi.\text{Verify}$ outputs a success bit $b \in \{0, 1\}$.

The requirement $t = O(\log \lambda)$ ensures that the length $T = 2^t$ of the committed vectors is polynomial in λ . The scheme Π is *complete* if the obvious correctness properties hold: for honestly generated $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$ and $(c, u) \leftarrow \Pi.\text{Commit}(\text{pp}, v)$, it holds that $\Pi.\text{Open}(\text{pp}, c; \pi_k(v), u) = 1$, and for every $\eta \in \mathbb{Z}_{2^m}^{2t}$, writing $s := \text{Tensor}(\eta) \cdot v$, the proof $\pi \leftarrow \Pi.\text{Prove}(\text{pp}, c, s, \eta; v, u)$ satisfies $\Pi.\text{Verify}(\text{pp}, c, s, \eta, \pi) = 1$ with probability 1.

Security properties. We further define security properties for the ring commitment scheme: *low-precision binding*, *evaluation binding*, and *low-precision knowledge soundness*. The key point is that the scheme is binding and extractable only at the lower precision $\mathbb{Z}_{2^k}$, whereas evaluations are proved over $\mathbb{Z}_{2^m}$. Below we give an intuition of these properties for Protocol 3, which are formalized afterwards.

Low-precision binding requires that a commitment admit at most one valid low-precision opening over $\mathbb{Z}_{2^k}$. This follows directly from the distance of the underlying codes. If two openings are both accepted, their corresponding encodings can differ only on the union of their disagreement sets. Since the size of this union is strictly smaller than the code distance, the two encodings must be identical. Consequently, the openings themselves must also be identical.

Evaluation binding requires that all accepted evaluation proofs for a given commitment be explained by the same vector $w \in \mathbb{Z}_{2^m}^T$. The proximity test first guarantees, except with negligible probability, that the committed matrix is close to the unique codeword over $\mathbb{Z}_{2^m}$. Let w denote the message underlying this codeword. Now suppose that the prover sends a row response that differs from the corresponding linear

combination of the rows of w . By the distance of the code, the encoding of this response must differ from the expected encoding on many columns. Consequently, all accepted evaluation proofs are consistent with the same vector w . Note that the evaluation binding only requires the existence of a common w and in particular such a vector need not be unique. For later usage, we establish a stronger property that every w consistent with all accepted evaluations is also consistent with the commitment c . More precisely, its reduction $\pi_k(w)$ must constitute a valid low-precision opening of the same commitment.

Finally, we establish *low-precision extractability*. Recall that over a field, extractability requires an efficient emulator that runs in expected polynomial time and recovers the full witness underlying the commitment. Following [DP23], our emulator repeatedly rewinds $\mathcal{A}$ to collect accepted evaluation proofs at freshly sampled challenges and then attempts to recover the witness by solving the resulting linear system. The analysis of evaluation binding ensures that the reduced row responses in these proofs give linear equations in the same message w . The main difficulty arises from selective answering, where the adversary only responds to part of challenges. In particular, if all answered challenges lie in a proper submodule, the resulting linear system would remain underdetermined regardless of how many accepted proofs are collected.

Such a strategy is easy to realize over $\mathbb{Z}_{2^m}$. For instance, the adversary may answer only those challenges vanishing on some nonzero vector divisible by a high power of two. A random challenge satisfies this condition with high probability, since it only has to supply the few factors of two still missing for the inner product to vanish. The adversary thus retains most of its success probability while preventing extraction of the full witness over $\mathbb{Z}_{2^m}$.

To mitigate this problem, we turn to the smaller ring $\mathbb{Z}_{2^k}$ and only require that the collected system determine the witness modulo 2^k . Suppose that it did not. Then there would be a vector that is nonzero modulo 2^k on which every recorded challenge vanishes over $\mathbb{Z}_{2^m}$. The adversary would thus have to answer only challenges vanishing on this vector, since every other would contribute new information about it to the system. Since the vector has fewer than k factors of two, a challenge vanishing on it must itself supply all the remaining factors needed to vanish over $\mathbb{Z}_{2^m}$. Given the gap between the two rings, this can happen only rarely for a random challenge. Therefore, the adversary cannot restrict its answers to such challenges while retaining a nonnegligible success probability. Hence, with high probability, sufficiently many accepted proofs determine the witness uniquely modulo 2^k .

We now formalize these three properties. The first is a binding property for Π .Open. Unlike the standard definition, which rules out the possibility of opening the commitment to two distinct vectors over $\mathbb{Z}_{2^m}$, our notion only requires uniqueness after reduction to $\mathbb{Z}_{2^k}$.

Definition 8. For each ring commitment scheme Π , security parameter λ , size parameter t , and PPT adversary $\mathcal{A}$, we define the low-precision binding experiment $\text{LPBind}_{\mathcal{A}}^{\Pi,t}(\lambda)$ as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$.
2. The adversary outputs a commitment together with two openings, $(c, \bar{v}_0, u_0, \bar{v}_1, u_1) \leftarrow \mathcal{A}(\text{pp})$, where $\bar{v}_0, \bar{v}_1 \in \mathbb{Z}_{2^k}^T$.
3. The output of the experiment is defined to be 1 if $\Pi.\text{Open}(\text{pp}, c; \bar{v}_0, u_0) = \Pi.\text{Open}(\text{pp}, c; \bar{v}_1, u_1) = 1$ and $\bar{v}_0 \neq \bar{v}_1$; otherwise, it is defined to be 0.

We say that Π is low-precision binding if, for each PPT adversary $\mathcal{A}$, each $\lambda \in \mathbb{N}$ and each $t = O(\log \lambda)$, $\Pr[\text{LPBind}_{\mathcal{A}}^{\Pi,t}(\lambda) = 1] \leq \text{negl}(\lambda)$.

The second property is evaluation binding. Intuitively, it requires that all accepted evaluation proofs for the same commitment arise from a single vector over $\mathbb{Z}_{2^m}$, and that the reduction of this vector to $\mathbb{Z}_{2^k}$ be a valid opening of that commitment.

Definition 9. For each ring commitment scheme Π , security parameter λ , size parameter t , PPT query sampler $\mathcal{Q}$, stateful PPT adversary $\mathcal{A}$, and polynomially bounded $B = B(\lambda) \geq 1$, we define the evaluation binding experiment $\text{EvalBind}_{\mathcal{Q}, \mathcal{A}}^{\Pi, t, B}(\lambda)$ as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$ and $\mathcal{Q}$.
2. The adversary outputs a commitment $c \leftarrow \mathcal{A}(\text{pp})$.
3. For every $i \in [B]$, the query sampler independently outputs $\eta_i \leftarrow \mathcal{Q}(\text{pp})$, the adversary is reset to its state immediately after outputting c , and it outputs $(s_i, \pi_i) \leftarrow \mathcal{A}(\eta_i)$.
4. The output of the experiment is defined to be 1 if $\Pi.\text{Verify}(\text{pp}, c, s_i, \eta_i, \pi_i) = 1$ for every $i \in [B]$ and there exist no vector $v \in \mathbb{Z}_{2^m}^T$ and opening hint u with $\Pi.\text{Open}(\text{pp}, c; \pi_k(v), u) = 1$ and $\text{Tensor}(\eta_i) \cdot v = s_i$ for every $i \in [B]$; otherwise, it is defined to be 0.

We say that Π is evaluation binding with respect to $\mathcal{Q}$ if, for each stateful PPT adversary $\mathcal{A}$, each polynomially bounded $B \geq 1$, each $\lambda \in \mathbb{N}$ and each $t = O(\log \lambda)$, $\Pr[\text{EvalBind}_{\mathcal{Q}, \mathcal{A}}^{\Pi, t, B}(\lambda) = 1] \leq \text{negl}(\lambda)$.

The third property is the knowledge notion. Unlike the field case, we only consider a weaker notion which we call *low-precision extractability*. Intuitively, the emulator only extracts the lower part of the witness in $\mathbb{Z}_{2^k}$, instead of the full witness in $\mathbb{Z}_{2^m}$.

Definition 10. For each ring commitment scheme Π , security parameter λ , size parameter t , PPT query sampler $\mathcal{Q}$, stateful PPT adversary $\mathcal{A}$, expected PPT emulator $\mathcal{E}$, and PPT distinguisher $\mathcal{D}$, we define two random variables $\text{Real}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$ and $\text{Emul}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$, each valued in $\{0, 1\}$, as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$, $\mathcal{Q}$ and $\mathcal{E}$.
2. The adversary outputs a commitment $c \leftarrow \mathcal{A}(\text{pp})$.
3. The query sampler outputs $\eta \leftarrow \mathcal{Q}(\text{pp})$.
4. The experimenter proceeds in one of two separate ways:
 - $\text{Real}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$: Run $(s, \pi) \leftarrow \mathcal{A}(\eta)$. Output the single bit $\mathcal{D}(c, \eta, s, \pi)$.
 - $\text{Emul}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$: Run $(s, \pi; \hat{v}, u) \leftarrow \mathcal{E}^{\mathcal{A}}(\eta)$, where the emulator may repeatedly reset $\mathcal{A}$ to its state immediately after outputting c , and where $\hat{v} \in \mathbb{Z}_{2^k}^T \cup \{\perp\}$. Output the single bit

$$\mathcal{D}(c, \eta, s, \pi) \wedge (\Pi.\text{Verify}(\text{pp}, c, s, \eta, \pi) = 1 \implies \Pi.\text{Open}(\text{pp}, c; \hat{v}, u) = 1).$$

We say that Π is low-precision knowledge sound with respect to $\mathcal{Q}$ if, for each stateful PPT adversary $\mathcal{A}$, there exists an expected PPT emulator $\mathcal{E}$ such that, for each PPT distinguisher $\mathcal{D}$, the ensembles $\text{Real}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$ and $\text{Emul}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$, indexed by $\lambda \in \mathbb{N}$ and $t = O(\log \lambda)$, are statistically close.

The following definition is drawn from [DP23; CHMMVW20], which requires that $\mathcal{Q}$ sample uniformly.

Definition 11. The query sampler $\mathcal{Q}$ is admissible if, for each λ and t , and each $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, which contains the description of the evaluation domain $\mathbb{Z}_{2^m}$, the output $\eta \leftarrow \mathcal{Q}(\text{pp})$ is uniform over $\mathbb{Z}_{2^m}^{2t}$.

The emulator of Definition 10 and the explaining vector of Definition 9 are obtained separately, and nothing so far relates them. The compilation of Theorem 8 needs the low-precision opening produced by the emulator to be the reduction of a vector that explains the accepted evaluation proofs. The following notion makes this precise.

Definition 12. For each ring commitment scheme Π , security parameter λ , size parameter t , PPT query sampler $\mathcal{Q}$, stateful PPT adversary $\mathcal{A}$, expected PPT emulator $\mathcal{E}$, and polynomially bounded $B = B(\lambda) \geq 1$, we define the consistency experiment $\text{Consist}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}}^{\Pi, t, B}(\lambda)$ as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$, $\mathcal{Q}$ and $\mathcal{E}$.
2. The adversary outputs a commitment $c \leftarrow \mathcal{A}(\text{pp})$.
3. The query sampler outputs $\eta_0 \leftarrow \mathcal{Q}(\text{pp})$, and the experimenter runs $(s_0, \pi_0; \hat{v}, u) \leftarrow \mathcal{E}^{\mathcal{A}}(\eta_0)$, where the emulator may repeatedly reset $\mathcal{A}$ to its state immediately after outputting c .
4. For every $i \in \{1, \dots, B\}$, the query sampler independently outputs $\eta_i \leftarrow \mathcal{Q}(\text{pp})$, the adversary is reset to its state immediately after outputting c , and it outputs $(s_i, \pi_i) \leftarrow \mathcal{A}(\eta_i)$.
5. The output of the experiment is defined to be 1 if $\Pi.\text{Verify}(\text{pp}, c, s_i, \eta_i, \pi_i) = 1$ for every $i \in \{0, \dots, B\}$ and there exists no vector $w \in \mathbb{Z}_{2^m}^T$ with $\pi_k(w) = \hat{v}$ and $\text{Tensor}(\eta_i) \cdot w = s_i$ for every $i \in \{0, \dots, B\}$; otherwise, it is defined to be 0.

We say that Π has consistent extraction with respect to $\mathcal{Q}$ if, for each stateful PPT adversary $\mathcal{A}$, there exists an expected PPT emulator $\mathcal{E}$ as in Definition 10 such that, for each polynomially bounded $B \geq 1$, each $\lambda \in \mathbb{N}$ and each $t = O(\log \lambda)$, $\Pr[\text{Consist}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}}^{\Pi, t, B}(\lambda) = 1] \leq \text{negl}(\lambda)$.

Compilation. We close this part by proving that a ring commitment instantiates a ring oracle. Let $\Pi_{\text{IOP}} = (\mathcal{P}, \mathcal{V})$ be an IOP with ring oracles for an indexed relation $\mathcal{R}$ over $\mathbb{Z}_{2^k}$. Assume that the index is explicit, the instance consists of the oracles $\llbracket v_0 \rrbracket, \dots, \llbracket v_{r-1} \rrbracket$ of vectors $v_0, \dots, v_{r-1} \in \mathbb{Z}_{2^k}^T$, each of which holds some lift of its vector to $\mathbb{Z}_{2^m}$, and the prover sends explicit elements of $\mathbb{Z}_{2^m}$ and ring oracles over $\mathbb{Z}_{2^m}$. Let Π be a ring commitment. The compiled argument Π_{IOP}^Π for $\mathcal{R}$ proceeds as follows.

1. The prover commits to the canonical lift $\hat{v}_j$ of each v_j with Π , obtaining a commitment c_j and an opening hint u_j , and sends $c_0, \dots, c_{r-1}$ to the verifier; if two instance oracles hold the same vector, one commitment serves both.
2. The parties run Π_{IOP} . Explicit messages are sent as they are, and every ring oracle $\llbracket w \rrbracket$ sent by $\mathcal{P}$ is replaced by a commitment to w with Π .
3. A tensor query of $\mathcal{V}$ to an oracle at a point η is answered by a value s together with a proof generated by $\Pi.\text{Prove}$ for the corresponding commitment, which the verifier checks with $\Pi.\text{Verify}$. The verifier accepts if $\mathcal{V}$ accepts and every evaluation proof verifies.

The compiled argument makes several queries to the same commitment, at points chosen by $\mathcal{V}$. We therefore assume the security of Π with respect to the query samplers induced by $\mathcal{V}$, that is, for each query of $\mathcal{V}$, the sampler that outputs the point of that query. The knowledge soundness of the compiled argument is stated for the contents of the instance oracles: the extractor has to output vectors $\hat{v}_0, \dots, \hat{v}_{r-1} \in \mathbb{Z}_{2^k}^T$ with $(i, (\llbracket \hat{v}_0 \rrbracket, \dots, \llbracket \hat{v}_{r-1} \rrbracket)) \in \mathcal{L}(\mathcal{R})$ whenever the verifier accepts.

Theorem 8. Let $\Pi_{\text{IOP}} = (\mathcal{P}, \mathcal{V})$ be an IOP with ring oracles for an indexed relation $\mathcal{R}$ over $\mathbb{Z}_{2^k}$ with soundness error ϵ_{IOP} . Suppose that the index is explicit, the instance consists of the ring oracles $\llbracket v_0 \rrbracket, \dots, \llbracket v_{r-1} \rrbracket$ of vectors over $\mathbb{Z}_{2^k}$, and the prover sends explicit elements and ring oracles. Let Π be a complete ring commitment that is evaluation binding and has consistent extraction with respect to the query samplers induced by $\mathcal{V}$, and let ϵ_R bound the statistical distance between Real and Emul in Definition 10 for its emulator. Then the compiled argument Π_{IOP}^Π is a complete and knowledge-sound argument for $\mathcal{R}$ with knowledge error $\epsilon_{\text{IOP}} + r \cdot \epsilon_R + \text{negl}(\lambda)$.

Proof. Completeness follows from that of Π_{IOP} and Π : the honest prover commits to the canonical lifts, every evaluation proof verifies, and the answers are exactly the answers of the oracles of Π_{IOP} holding these lifts.

For knowledge soundness, fix a PPT prover $\mathcal{A}$ for the compiled argument. The extractor $\mathcal{E}$ runs $\mathcal{A}$ against the honest verifier, records the transcript together with the verdict of the verifier, and then, for each instance commitment c_j , runs the emulator $\mathcal{E}_j$ guaranteed by the low-precision knowledge soundness of Π , which may rewind $\mathcal{A}$ to its state immediately after sending $c_0, \dots, c_{r-1}$, obtaining $(\hat{v}_j, u_j)$ with $\hat{v}_j \in \mathbb{Z}_{2^k}^T \cup \{\perp\}$. It outputs $\hat{v} := (\hat{v}_0, \dots, \hat{v}_{r-1})$. Since each $\mathcal{E}_j$ runs in expected polynomial time, so does $\mathcal{E}$.

We bound the probability that the verifier accepts while $(i, \llbracket \hat{v} \rrbracket) \notin \mathcal{L}(\mathcal{R})$. Consider the following three events.

- *Extraction fails.* For some $j \in [r]$, every evaluation proof for c_j in the transcript verifies, but $\Pi.\text{Open}(\text{pp}, c_j; \hat{v}_j, u_j) = 0$. Restricting the experiment to c_j and to the distinguisher that always outputs 1, the corresponding Real outputs 1 with probability 1, so the statistical closeness of Definition 10 bounds this event by ϵ_R for each j , and by $r \cdot \epsilon_R$ in total.
- *Evaluation binding fails.* For some commitment c sent in step 2, the accepted evaluation proofs for c are not all explained by a single vector, that is, there is no $w_c \in \mathbb{Z}_{2^m}^T$ with $\text{Tensor}(\eta) \cdot w_c = s$ for every accepted answer (η, s) to c . The number of queries to c is polynomial, so evaluation binding with respect to the induced samplers bounds this event by $\text{negl}(\lambda)$.
- *Inconsistent extraction.* For some $j \in [r]$, no vector explaining the accepted proofs for c_j reduces to $\hat{v}_j$, that is, there is no $w \in \mathbb{Z}_{2^m}^T$ with $\pi_k(w) = \hat{v}_j$ and $\text{Tensor}(\eta) \cdot w = s$ for every accepted answer (η, s) to c_j . By consistent extraction with respect to the induced samplers, this event has probability $\text{negl}(\lambda)$.

Condition on none of these events occurring and on the verifier accepting. For each instance commitment c_j , fix a vector w_{c_j} that explains its accepted answers and satisfies $\pi_k(w_{c_j}) = \hat{v}_j$, and for each commitment c sent in step 2, fix a vector w_c that explains its accepted answers. Then every accepted answer to a query to c_j equals $\text{Tensor}(\eta) \cdot w_{c_j}$, where w_{c_j} is a lift of $\hat{v}_j$, and every accepted answer to c equals $\text{Tensor}(\eta) \cdot w_c$. The transcript is therefore a transcript of Π_{IOP} against some prover strategy, in which the instance oracle $\llbracket \hat{v}_j \rrbracket$ holds the lift w_{c_j} , a choice allowed by the model, and the oracles sent by the prover hold the vectors w_c . If $(i, \llbracket \hat{v} \rrbracket) \notin \mathcal{L}(\mathcal{R})$, the soundness of Π_{IOP} , which holds for every choice of the lifts, implies that $\mathcal{V}$ accepts with probability at most ϵ_{IOP} . A union bound over the three events gives the claimed knowledge error. $\square$

6.2 Instantiation of Ring Commitment Scheme

We show our ring commitment scheme in Protocol 3, which closely follows Brakedown [GLSTW23; DP23]. Since the proximity gap requires a uniformly random tensor challenge over $\mathbb{Z}_{2^\ell}$, whereas the evaluation point is uniform over $\mathbb{Z}_{2^m}$, we let the verifier independently sample the high parts of the relevant coordinates. This makes the resulting $\tilde{L}$ uniformly random over $\mathbb{Z}_{2^\ell}$.

PROTOCOL 3. Ring Commitment Scheme

We define $\Pi = (\text{Setup}, \text{Commit}, \text{Open}, \text{Prove}, \text{Verify})$ as follows.

- $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$. On input 1^λ and t , choose $t_0, t_1 \geq 1$ with $t_0 + t_1 = t$, set $r_0 := 2^{t_0}$, $r_1 := 2^{t_1}$ and $T := r_0 r_1 = 2^t$, and return t_0 and t_1 , the knowledge domain $\mathbb{Z}_{2^k}$, the evaluation domain $\mathbb{Z}_{2^m}$ and the challenge domain $\mathbb{Z}_{2^\ell}$, where $m \geq k + \lambda$ and $\ell \geq m + \lambda$, together with a binary $[N, r_1, d]$ -code $C \subseteq \mathbb{F}_2^N$ for which $N = 2^{O(t)}$ and $d = \Omega(N)$, and a repetition parameter $\gamma = \Theta(\lambda)$.
- $(c, u) \leftarrow \Pi.\text{Commit}(\text{pp}, v)$. On input $v \in \mathbb{Z}_{2^m}^T$, write its canonical lift $\tilde{v} \in \mathbb{Z}_{2^\ell}^T$ as an $r_0 \times r_1$ matrix $(\tilde{v}_i)_{i=0}^{r_0-1}$, and compute the $r_0 \times N$ matrix $(u_i)_{i=0}^{r_0-1} := (\text{Enc}_\ell(\tilde{v}_i))_{i=0}^{r_0-1}$. Output a Merkle commitment c to the columns $(u_{i,j})_{i=0}^{r_0-1}$, $j \in [N]$, together with the opening hint $u := (u_i)_{i=0}^{r_0-1}$.
- $b \leftarrow \Pi.\text{Open}(\text{pp}, c; \bar{v}, u)$. On input the root c , a low-precision opening $\bar{v} \in \mathbb{Z}_{2^k}^T$, and an opening hint u consisting of distinct Merkle paths against c , missing the columns $\mathcal{M} \subseteq [N]$, say, write $\bar{v}$ as an $r_0 \times r_1$ matrix $(\bar{v}_i)_{i=0}^{r_0-1}$ and check $|\Delta_k((u_i)_{i=0}^{r_0-1}, (\text{Enc}_\ell(\bar{v}_i))_{i=0}^{r_0-1}) \cup \mathcal{M}| \stackrel{?}{<} \frac{d}{2}$, where the disagreement set ranges over the columns present in u .

We define $\Pi.\text{Prove}$ and $\Pi.\text{Verify}$ by applying the Fiat–Shamir heuristic to the following interactive protocol, where $\mathcal{P}$ has v , its lift $(\tilde{v}_i)_{i=0}^{r_0-1}$ and $(u_i)_{i=0}^{r_0-1}$, and $\mathcal{P}$ and $\mathcal{V}$ have c , s , and $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$.

- $\mathcal{V}$ samples $\alpha_i^h, \beta_i^h \leftarrow \mathbb{Z}_{2^{\ell-m}}$ for $i \in [t_0]$ and sends them to $\mathcal{P}$. Both parties set $\tilde{\alpha}_i := \alpha_i + 2^m \alpha_i^h$ and $\tilde{\beta}_i := \beta_i + 2^m \beta_i^h$ in $\mathbb{Z}_{2^\ell}$, and the lifted challenge $\tilde{L} := \bigotimes_{i=0}^{t_0-1} (\tilde{\alpha}_i, \tilde{\beta}_i) \in \mathbb{Z}_{2^\ell}^{r_0}$.
- $\mathcal{P}$ sends $\mathcal{V}$ the row combination $y' := \tilde{L} \cdot (\tilde{v}_i)_{i=0}^{r_0-1} \in \mathbb{Z}_{2^\ell}^{r_1}$ in the clear.
- For each $i \in \{0, \dots, \gamma - 1\}$, $\mathcal{V}$ samples $j_i \leftarrow [N]$. $\mathcal{V}$ sends $\mathcal{P}$ the set $J := \{j_0, \dots, j_{\gamma-1}\}$.
- $\mathcal{P}$ sends $\mathcal{V}$ the columns $\{(u_{i,j})_{i=0}^{r_0-1}\}_{j \in J}$, each featuring an accompanying Merkle path against c .
- $\mathcal{V}$ computes $\text{Enc}_\ell(y')$. For each $j \in J$, $\mathcal{V}$ verifies the Merkle path attesting to $(u_{i,j})_{i=0}^{r_0-1}$, and moreover requires that $\tilde{L} \cdot (u_{i,j})_{i=0}^{r_0-1} \stackrel{?}{=} \text{Enc}_\ell(y')_j$. Finally, $\mathcal{V}$ requires $s \stackrel{?}{=} \pi_m(y') \cdot \bigotimes_{i=t_0}^{t-1} (\alpha_i, \beta_i)$.

We now analyze Protocol 3. We first fix the notation and the error terms, verify completeness, and recall Merkle extraction in the random oracle model. We then prove low-precision binding (Theorem 9), evaluation binding (Theorem 10) and low-precision knowledge soundness (Theorem 11), with the auxiliary lemmas placed where they are used, and close with consistent extraction (Corollary 2).

Notation. A vector $v \in \mathbb{Z}_{2^m}^T$, and likewise a vector over $\mathbb{Z}_{2^k}$ or $\mathbb{Z}_{2^\ell}$, is identified with the $r_0 \times r_1$ matrix $(v_i)_{i=0}^{r_0-1}$ that lists its entries in row-major order, so that $v_i \in \mathbb{Z}_{2^m}^{r_1}$ is its i -th row and $(v_{i,j})_{i=0}^{r_0-1}$ its j -th column. For a vector $L \in \mathbb{Z}_{2^m}^{r_0}$, we write $L \cdot (v_i)_{i=0}^{r_0-1} := \sum_{i=0}^{r_0-1} L_i v_i \in \mathbb{Z}_{2^m}^{r_1}$ for the corresponding combination of the rows, and $(\text{Enc}_\ell(v_i))_{i=0}^{r_0-1}$ for the row-wise encoding of the matrix. For an evaluation point $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$, we write $\text{Tensor}(\eta) := \bigotimes_{i=0}^{t-1} (\alpha_i, \beta_i)$. With $r_1 = 2^{t_1}$ and $r_0 = 2^{t_0}$, the low t_1 bits of an index into v select the column and the high t_0 bits select the row, and the first t_0 pairs of an

evaluation point correspond to these high bits, so the tensor of the point splits along the two parts of the point:

$$\text{Tensor}(\eta) \cdot v = \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i) \cdot (v_i)_{i=0}^{r_0-1} \cdot \otimes_{i=t_0}^{t-1} (\alpha_i, \beta_i). \quad (7)$$

For an evaluation at the point η with row response y' , we call $L := \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i)$, which equals $\pi_m(\tilde{L})$, its *row challenge*, and $y := \pi_m(y')$ its *reduced row response*. In the following, we set the proximity parameter as $e := \lfloor \frac{d-1}{3} \rfloor$, so that $3e < d$.

Completeness. For an honest prover, $u_i = \text{Enc}_\ell(\tilde{v}_i)$ for every i , and Enc_ℓ is $\mathbb{Z}_{2^\ell}$ -linear, so for every column j , $\tilde{L} \cdot (u_{i,j})_{i=0}^{r_0-1} = \text{Enc}_\ell(\tilde{L} \cdot (\tilde{v}_i)_{i=0}^{r_0-1})_j = \text{Enc}_\ell(y')_j$. Moreover, $\pi_m(\tilde{\alpha}_i) = \alpha_i$ and $\pi_m(\tilde{\beta}_i) = \beta_i$, so $\pi_m(\tilde{L}) = \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i)$ and $\pi_m(y') = \pi_m(\tilde{L}) \cdot (v_i)_{i=0}^{r_0-1}$. By (7), the final check of the verifier therefore holds exactly when $s = \text{Tensor}(\eta) \cdot v$. Finally, $\pi_k(u_i) = \text{Enc}_k(\pi_k(v_i))$ for every i , so the honest opening hint, which contains all Merkle paths, has no missing column and no disagreement, and $\Pi.\text{Open}$ accepts $\pi_k(v)$. The requirement $N = 2^{O(t)}$ is needed merely for C to be efficiently encodable, and $d = \Omega(N)$ says that C has constant relative distance.

Merkle trees. We use Merkle trees [Mer89] in the random oracle model following [BCS16; DP23], with a random oracle H of output length λ . The Merkle commitment to $(u_i)_{i=0}^{r_0-1} \in \mathbb{Z}_{2^\ell}^{r_0 \times N}$ hashes each column $(u_{i,j})_{i=0}^{r_0-1}$ into a leaf, and each internal node is the hash of its two children. A Merkle path for position j consists of the column $(u_{i,j})_{i=0}^{r_0-1}$ and the siblings of the nodes on the path from the j -th leaf to the root, and it is valid against c if recomputing the path from the leaf yields c .

Fix an adversary $\mathcal{A}$ that makes at most $q_H = \text{poly}(\lambda)$ oracle queries, and consider the moment at which it outputs a root c . The following algorithm `TreeBuilder`, taken from [DP23; BCS16], reads the oracle queries made by $\mathcal{A}$ so far and rebuilds from them the part of the tree that $\mathcal{A}$ has actually computed. It starts from the empty set $I := \emptyset$ of reconstructed positions and is invoked as `TreeBuilder(c, log N, 0)`. Its arguments are a node value h , its height k and the index j of its leftmost leaf.

- 1: **procedure** `TREEBUILDER`(h, k, j)
- 2: **if** $k = 0$ and $h = H((x_i)_{i=0}^{r_0-1})$ for some column $(x_i)_{i=0}^{r_0-1} \in \mathbb{Z}_{2^\ell}^{r_0}$ queried by $\mathcal{A}$ **then**
- 3: set $(u_{i,j})_{i=0}^{r_0-1} := (x_i)_{i=0}^{r_0-1}$ and add j to I
- 4: **else if** $k > 0$ and $h = H(h_0 \parallel h_1)$ for some pair $(h_0, h_1) \in \{0, 1\}^\lambda \times \{0, 1\}^\lambda$ queried by $\mathcal{A}$ **then**
- 5: `TREEBUILDER`($h_0, k - 1, 2j$)
- 6: `TREEBUILDER`($h_1, k - 1, 2j + 1$)
- 7: **end if**
- 8: **end procedure**

The conditions in lines 2 and 4 require the queried preimage to be well formed, namely a column in $\mathbb{Z}_{2^\ell}^{r_0}$ at a leaf and a pair of λ -bit strings at an inner node. If h has several distinct well-formed preimages among the queries, which can only happen if $\mathcal{A}$ has found a collision of H , then `TreeBuilder` picks one of them arbitrarily. The output is the partial matrix $(u_{i,j})_{i \in [r_0], j \in I}$ together with a valid Merkle path against c for each of its columns, and we write $\mathcal{M} := [N] \setminus I$ for the set of missing columns. `TreeBuilder` is straight-line and runs in polynomial time, since it inspects each of the at most q_H queries at most once per node of the tree. Its guarantee is the following.

Lemma 8 (Merkle extraction [BCS16, §A.1]). *Let $\delta_{\text{mt}} := (q_H^2 + 1)/2^\lambda$. Except with probability δ_{mt} over the random oracle, every valid Merkle path against c that $\mathcal{A}$ outputs, together with c or in any later execution following the commitment, is for a position $j \in I$ and carries the column $(u_{i,j})_{i=0}^{r_0-1}$ reconstructed by `TreeBuilder`. In particular, $\mathcal{A}$ produces no valid path for a position in $\mathcal{M}$, and any two valid paths for the same position carry the same column.*

We call the event of the lemma *good Merkle extraction* and condition on it throughout the analysis, charging δ_{mt} to the error. For intuition, a valid path for position j is a chain of oracle evaluations ending in c . If $\mathcal{A}$ queried all of them before outputting c , then TreeBuilder follows the chain and reconstructs the column at position j , unless two of its queries collide. Otherwise $\mathcal{A}$ has to hit a value that the chain has already fixed with an oracle answer it has not yet seen, that is, to guess a uniformly random λ -bit string. Both events are accounted for in δ_{mt} .

We begin with low-precision binding, which follows from Merkle extraction and the distance of the lifted code alone.

Theorem 9. *The scheme of Protocol 3 is low-precision binding.*

Proof. Fix a PPT adversary $\mathcal{A}$ making at most $q_H = \text{poly}(\lambda)$ oracle queries, and suppose that it outputs $(c, \bar{v}^0, u^0, \bar{v}^1, u^1)$ such that $\Pi.\text{Open}$ accepts both openings. We run TreeBuilder on the oracle queries of $\mathcal{A}$ and condition on good Merkle extraction, which fails with probability at most δ_{mt} . For $b \in \{0, 1\}$, let $\mathcal{M}_b$ be the set of columns missing from u^b , write $(u_{i,j}^b)_{i \in [r_0], j \notin \mathcal{M}_b}$ for the columns carried by the Merkle paths in u^b , and define

$$\Delta_b := \Delta_k^{(r_0)} \left((u_{i,j}^b)_{i \in [r_0], j \notin \mathcal{M}_b}, (\text{Enc}_k(\bar{v}_i^b))_{i=0}^{r_0-1} \right).$$

Let $X := \Delta_0 \cup \mathcal{M}_0 \cup \Delta_1 \cup \mathcal{M}_1$. By hypothesis, $|X| < d$. For every column $j \notin X$, both hints contain a valid Merkle path for position j against c , so by Lemma 8 both paths carry the column reconstructed by TreeBuilder, that is, $(u_{i,j}^0)_{i=0}^{r_0-1} = (u_{i,j}^1)_{i=0}^{r_0-1}$. Moreover, since $j \notin \Delta_0 \cup \Delta_1$, this column reduces modulo 2^k both to $(\text{Enc}_k(\bar{v}_i^0))_{i=0}^{r_0-1}$ and to $(\text{Enc}_k(\bar{v}_i^1))_{i=0}^{r_0-1}$. Hence $(\text{Enc}_k(\bar{v}_i^0 - \bar{v}_i^1))_{i=0}^{r_0-1}$ is supported inside X , and since $|X| < d$, Lemma 5 forces $\text{Enc}_k(\bar{v}_i^0) = \text{Enc}_k(\bar{v}_i^1)$ for every i . Finally, $\bar{v}^0 = \bar{v}^1$ as Enc_k is injective. The experiment thus outputs 1 with probability at most δ_{mt} . $\square$

We next establish what an accepted evaluation proof tells the verifier about the committed matrix. Fix an adversary $\mathcal{A}$ that has output a commitment c , let $(u_{i,j})_{i \in [r_0], j \in I}$ be the partial matrix reconstructed by TreeBuilder with missing set $\mathcal{M} = [N] \setminus I$, and condition on good Merkle extraction, so that every column opened in an accepted evaluation at a position $j \in I$ equals $(u_{i,j})_{i=0}^{r_0-1}$, and no position in $\mathcal{M}$ is ever opened. We call $w \in \mathbb{Z}_{2^m}^T$ a *quotient message* of the commitment if

$$\left| \Delta_m^{(r_0)} \left((u_{i,j})_{i \in [r_0], j \in I}, (\text{Enc}_m(w_i))_{i=0}^{r_0-1} \right) \cup \mathcal{M} \right| \leq e. \quad (8)$$

That is, the reconstructed matrix, reduced to $\mathbb{Z}_{2^m}$, agrees with the encoding of the message at all but at most e columns, with the missing columns included. The following lemma shows that an accepted evaluation proof ensures, except with negligible probability, that a quotient message exists, and that it is unique.

Lemma 9. *The commitment has at most one quotient message. If $\mathcal{Q}$ is admissible and the commitment has no quotient message, then $\mathcal{A}$ can produce an accepting evaluation proof with probability at most $\delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma$, where $\delta_{\text{pgap}} := (e + 1) \Phi(\ell - m + 1, t_0)$ is the error of the proximity gap in Theorem 5.*

Proof. For the first claim, let w and w' both be quotient messages of the commitment, and write X for the union of $\mathcal{M}$ with their two disagreement sets, so that $|X| \leq 2e < d$. At every column $j \notin X$ both encodings agree with the reconstructed column $\pi_m((u_{i,j})_{i=0}^{r_0-1})$, hence with each other, so the interleaved codeword $(\text{Enc}_m(w_i - w'_i))_{i=0}^{r_0-1}$ is supported inside X . If it were nonzero, one of its rows would be a nonzero codeword of $\mathcal{C}_m$ of weight less than d , contradicting the minimum distance of the code. Hence $\text{Enc}_m(w_i) = \text{Enc}_m(w'_i)$ for every i , which further implies $w = w'$ as Enc_m is injective.

For the second claim, we first argue that we may assume $|\mathcal{M}| \leq e$. Indeed, if $|\mathcal{M}| > e$, then the sampled set J avoids $\mathcal{M}$ with probability at most $(1 - \frac{d}{3N})^\gamma$, and $\mathcal{A}$ can pass in case $J \cap \mathcal{M} \neq \emptyset$ only by producing a valid Merkle path for a missing position, which good Merkle extraction excludes.

We thus assume $|\mathcal{M}| \leq e$, write $e' := e - |\mathcal{M}|$, and puncture C at $\mathcal{M}$. The punctured binary code has dimension r_1 , length $|I|$ and distance $d' \geq d - |\mathcal{M}|$, so that $3e' = 3e - 3|\mathcal{M}| < d - |\mathcal{M}| \leq d'$; its lifts are the puncturings of the codes $\mathcal{C}_q$, which we denote by $\overline{\mathcal{C}_q}$. A message w violates (8) exactly when $\pi_m((u_{i,j})_{i \in [r_0], j \in I})$ differs from $(\text{Enc}_m(w_i))_{i=0}^{r_0-1}$ at more than e' columns of I . By hypothesis, therefore, $d_m((u_{i,j})_{i \in [r_0], j \in I}, \overline{\mathcal{C}_\ell}) > e'$, that is, the reconstructed matrix is (e', m) -far from $\overline{\mathcal{C}_\ell}$.

By admissibility of $\mathcal{Q}$, the row half $(\alpha_0, \beta_0, \dots, \alpha_{t_0-1}, \beta_{t_0-1})$ of η is uniform over $\mathbb{Z}_{2^m}^{2t_0}$, and the high bits α_i^h, β_i^h are uniform over $\mathbb{Z}_{2^{\ell-m}}$ and independent of everything else. Therefore, the completed coefficients $\tilde{\alpha}_i, \tilde{\beta}_i, i \in [t_0]$, are independent and uniform over $\mathbb{Z}_{2^\ell}$. Using Theorem 5, we have $d(\tilde{L} \cdot (u_{i,j})_{i \in [r_0], j \in I}, \overline{\mathcal{C}_I}) > e'$ holds except with probability at most $(e' + 1)\Phi(\ell - m + 1, t_0) \leq \delta_{\text{pgap}}$. Fix a challenge $\tilde{L}$ for which it holds. Whatever row response y' the prover sends, the restriction of $\text{Enc}_\ell(y')$ to the positions in I is a codeword of $\mathcal{C}_{\ell,I}$, so there are at least $e' + 1$ positions $j \in I$ at which $\tilde{L} \cdot (u_{i,j})_{i=0}^{r_0-1} \neq \text{Enc}_\ell(y')_j$; at each of them the column check of the verifier fails, since good Merkle extraction forces any column opened at j to equal $(u_{i,j})_{i=0}^{r_0-1}$. Together with the $|\mathcal{M}|$ missing positions, at least $e' + 1 + |\mathcal{M}| = e + 1$ positions cannot pass, and J avoids all of them with probability at most $(1 - \frac{d}{3N})^\gamma$. Adding the two probabilities proves the second claim. $\square$

We next show that, once the quotient message w exists, the row response y' sent by the adversary must be consistent with it.

Lemma 10. *Suppose that the quotient message w exists, and consider an evaluation at the point η with the lifted challenge $\tilde{L}$, so that $\pi_m(\tilde{L}) = \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i)$. If the row response y' sent by $\mathcal{A}$ satisfies $\pi_m(y') \neq \pi_m(\tilde{L}) \cdot (w_i)_{i=0}^{r_0-1}$, then $\mathcal{A}$ passes with probability at most $(1 - \frac{2d}{3N})^\gamma$. Consequently, if the evaluation proof is accepted and $\pi_m(y') = \pi_m(\tilde{L}) \cdot (w_i)_{i=0}^{r_0-1}$, then $s = \text{Tensor}(\eta) \cdot w$.*

Proof. Put $z := \pi_m(y') - \pi_m(\tilde{L}) \cdot (w_i)_{i=0}^{r_0-1} \neq 0$. By Lemma 5, $\text{Enc}_m(z)_j \neq 0$ for at least d positions j . On the other hand, by hypothesis, the set

$$\Delta_m^{(r_0)}((u_{i,j})_{i \in [r_0], j \in I}, (\text{Enc}_m(w_i))_{i=0}^{r_0-1}) \cup \mathcal{M}$$

has at most e elements. Hence, there are at least $d - e$ positions j outside this set for which $\text{Enc}_m(z)_j \neq 0$. Fix any such position j . Since $j \notin \mathcal{M}$, the column $(u_{i,j})_{i=0}^{r_0-1}$ is present. Moreover, since $j \notin \Delta_m^{(r_0)}((u_{i,j})_{i \in [r_0], j \in I}, (\text{Enc}_m(w_i))_{i=0}^{r_0-1})$, it reduces to $(\text{Enc}_m(w_i))_{i=0}^{r_0-1}$ modulo 2^m . Therefore, by linearity of the encoding and the identity $\pi_m \circ \text{Enc}_\ell = \text{Enc}_m \circ \pi_m$,

$$\begin{aligned} \pi_m(\text{Enc}_\ell(y')_j - \tilde{L} \cdot (u_{i,j})_{i=0}^{r_0-1}) &= \text{Enc}_m(\pi_m(y') - \pi_m(\tilde{L}) \cdot (w_i)_{i=0}^{r_0-1})_j \\ &= \text{Enc}_m(z)_j \neq 0. \end{aligned}$$

Thus, $\text{Enc}_\ell(y')_j - \tilde{L} \cdot (u_{i,j})_{i=0}^{r_0-1} \neq 0$ in $\mathbb{Z}_{2^\ell}$, and hence the column check fails at position j . To make the check pass at such a position, the adversary would therefore have to open a column different from $(u_{i,j})_{i=0}^{r_0-1}$, which is ruled out by good Merkle extraction. The same argument applies to each of the at least $d - e$ positions j satisfying $\text{Enc}_m(z)_j \neq 0$ outside $\Delta_m^{(r_0)}((u_{i,j})_{i \in [r_0], j \in I}, (\text{Enc}_m(w_i))_{i=0}^{r_0-1}) \cup \mathcal{M}$. Consequently, the verifier can accept only if all γ independent uniform positions in J avoid this set, which occurs with probability at most $(1 - \frac{d-e}{N})^\gamma \leq (1 - \frac{2d}{3N})^\gamma$.

For the second claim, suppose that the evaluation proof is accepted and that $\pi_m(y') = \pi_m(\tilde{L}) \cdot (w_i)_{i=0}^{r_0-1}$. Then we have $s = \pi_m(y') \cdot \otimes_{i=t_0}^{t-1} (\alpha_i, \beta_i)$. Substituting the above expression for $\pi_m(y')$ and using, we obtain $s = \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i) \cdot (w_i)_{i=0}^{r_0-1} \cdot \otimes_{i=t_0}^{t-1} (\alpha_i, \beta_i) = \text{Tensor}(\eta) \cdot w$. $\square$

Theorem 10. *If the query sampler $\mathcal{Q}$ is admissible, then the scheme of Protocol 3 is evaluation binding with respect to $\mathcal{Q}$.*

Proof. Fix a stateful PPT adversary $\mathcal{A}$ and a polynomially bounded $B \geq 1$, and consider the experiment of Definition 9. After $\mathcal{A}$ outputs c , we run TreeBuilder to obtain the partial matrix $(u_{i,j})_{i \in [r_0], j \in I}$ and $\mathcal{M}$, together with the opening hint u . In the following, we assume good Merkle extraction happens, which fails with probability at most δ_{mt} .

Suppose first that no quotient message exists. Since $B \geq 1$, the experiment can output 1 only if the first evaluation proof is accepted. By Lemma 9, this happens with probability at most $\delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma$.

Now suppose that the quotient message w exists. By Lemma 9, it is unique. For each $i \in [B]$, Lemma 10 implies that π_i is accepted while $s_i \neq \text{Tensor}(\eta_i) \cdot w$ with probability at most $(1 - \frac{2d}{3N})^\gamma$. Hence, by a union bound, except with probability $B(1 - \frac{2d}{3N})^\gamma$, every accepted evaluation proof satisfies $s_i = \text{Tensor}(\eta_i) \cdot w$. It remains to see that $\pi_k(w)$ is the correct opening to c with the hint u . By construction, u contains valid Merkle paths for the columns $(u_{i,j})_{i=0}^{r_0-1}$ with $j \in I$. Moreover, reduction to $\mathbb{Z}_{2^k}$ can only eliminate disagreements. We therefore have

$$\begin{aligned} & \Delta^{(r_0)} \left(\pi_k \left((u_{i,j})_{i \in [r_0], j \in I} \right), (\text{Enc}_k(\pi_k(w_i)))_{i=0}^{r_0-1} \right) \\ & \subseteq \Delta^{(r_0)} \left(\pi_m \left((u_{i,j})_{i \in [r_0], j \in I} \right), (\text{Enc}_m(w_i))_{i=0}^{r_0-1} \right). \end{aligned}$$

By (8), the union of the disagreement set in the right-hand side with $\mathcal{M}$ has size at most $e < d/2$. Thus, $\Pi.\text{Open}(\text{pp}, c; \pi_k(w), u) = 1$. Therefore, if all B evaluation proofs are accepted and satisfy $s_i = \text{Tensor}(\eta_i) \cdot w$, then $(\pi_k(w), u)$ is a valid opening of c and the corresponding message w satisfies all B evaluation claims.

Combining the two cases and accounting for failure of Merkle extraction, the probability that the experiment outputs 1 is at most $\delta_{\text{mt}} + \delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma + B(1 - \frac{2d}{3N})^\gamma$, which is negligible. $\square$

We now prove low-precision knowledge soundness. We start by collecting the two algebraic facts we need. The first is the ring analogue of the Schwartz–Zippel Lemma and a special case of Lemma 3: the probability that a random row challenge vanishes on a fixed vector z with $\text{Po2}_m(z) < k$ is governed by the tail Φ .

Lemma 11. *Let $z \in \mathbb{Z}_{2^m}^{r_0}$ with $\text{Po2}_m(z) < k$. If $\eta \leftarrow \mathbb{Z}_{2^m}^{2t}$ is uniform and L is the row challenge determined by η , then $\Pr[L \cdot z = 0] \leq \delta_{\text{ext}} := \Phi(m - k + 1, t_0)$.*

Proof. The row challenge $L = \otimes_{i=0}^{t_0-1} (\alpha_i, \beta_i)$ is the tensor of t_0 independent uniform pairs $(\alpha_i, \beta_i) \in \mathbb{Z}_{2^m}^2$. Put $a := \text{Po2}_m(z) \leq k-1$. Lemma 3 with t_0 folds gives $\Pr[L \cdot z = 0] \leq \Phi(m-a, t_0) \leq \Phi(m-k+1, t_0) = \delta_{\text{ext}}$, where the second inequality holds because $\Phi(S, t_0)$, a tail probability, is nonincreasing in S . $\square$

The second fact follows from [DP23, Lemma 4.11]. Since the emulator records evaluations conditioned on acceptance, the resulting query distribution may differ significantly from uniform. Let V denote the event that the adversary submits an accepting proof, and let E be any bad event of negligible probability. The following lemma shows that, provided $\Pr[V]$ is non-negligible, conditioning on V preserves the negligibility of E .

Lemma 12. *Let V and E be events with $\Pr[V] = \varepsilon > 0$. If $\Pr[V \wedge E] \leq \delta$, then $\Pr[E \mid V] \leq \delta/\varepsilon$. In particular, if $\varepsilon > \sqrt{\delta}$, then $\Pr[E \mid V] < \sqrt{\delta}$.*

Proof. We have $\Pr[E \mid V] = \Pr[V \wedge E] / \Pr[V] \leq \delta/\varepsilon$, and $\delta/\varepsilon < \delta/\sqrt{\delta} = \sqrt{\delta}$ when $\varepsilon > \sqrt{\delta}$. $\square$

Since the emulator only needs the solution in $\mathbb{Z}_{2^k}$, it suffices that the kernel of the recorded challenges be trivial modulo 2^k . The following lemma bounds the probability that this condition fails for the challenges arising from $t_{\text{ext}} := (m - k + 1)r_0$ accepted evaluation proofs.

Lemma 13. *Suppose that $\mathcal{Q}$ is admissible. Let $\varepsilon > 0$ be the probability that the adversary outputs an accepted evaluation proof, and let $L_0, \dots, L_{t_{\text{ext}}-1} \in \mathbb{Z}_{2^m}^{r_0}$ be the row challenges of t_{ext} accepted evaluation proofs. Then*

$$\Pr\left[\exists z \in \mathbb{Z}_{2^m}^{r_0} \text{ with } \text{Po2}_m(z) < k : L_i \cdot z = 0 \text{ for every } i \in [t_{\text{ext}}]\right] \leq \frac{t_{\text{ext}} \delta_{\text{ext}}}{\varepsilon}.$$

Proof. For $i \in \{0, \dots, t_{\text{ext}}\}$, let

$$\mathcal{K}_i := \{\text{Po2}_m(z) \geq k : L_{i'} \cdot z = 0 \text{ for every } i' < i\}.$$

Thus, $\mathcal{K}_i$ is the subgroup of $2^{k-1}\mathbb{Z}_{2^m}^{r_0}$ vanishing on the first i recorded challenges. In particular, $\mathcal{K}_0 = 2^{k-1}\mathbb{Z}_{2^m}^{r_0}$ and $|\mathcal{K}_0| = 2^{t_{\text{ext}}}$. Whenever $\mathcal{K}_i$ contains a vector z with $\text{Po2}_m(z) < k$, choose one deterministically as a function of the first i recorded evaluations and denote it by z_i .

We claim that, if there exists $z \in \mathbb{Z}_{2^m}^{r_0}$ with $\text{Po2}_m(z) < k$ such that $L_i \cdot z = 0$ for every $i \in [t_{\text{ext}}]$, then $L_i \cdot z_i = 0$ for some $i \in [t_{\text{ext}}]$. Indeed, suppose such a z exists, and let $\mu := \text{Po2}_m(z)$. Then $\tilde{z} := 2^{k-1-\mu}z$ satisfies the same equations and lies in $2^{k-1}\mathbb{Z}_{2^m}^{r_0}$. Moreover, $\text{Po2}_m(\tilde{z}) = k - 1 < k$, so $\tilde{z} \in \mathcal{K}_{t_{\text{ext}}}$. Since $\mathcal{K}_{t_{\text{ext}}} \subseteq \mathcal{K}_i$ for every $i \in [t_{\text{ext}}]$, the previously defined z_i exists for every $i \in [t_{\text{ext}}]$.

Suppose, toward a contradiction, that $L_i \cdot z_i \neq 0$ for every $i \in [t_{\text{ext}}]$. Then, for each such i , $\mathcal{K}_{i+1} = \{z \in \mathcal{K}_i : L_i \cdot z = 0\}$ is a proper subgroup of $\mathcal{K}_i$. Since $\mathcal{K}_i$ is a finite 2-group, we have $|\mathcal{K}_{i+1}| \leq |\mathcal{K}_i|/2$. Iterating over all t_{ext} indices gives $|\mathcal{K}_{t_{\text{ext}}}| \leq |\mathcal{K}_0|/2^{t_{\text{ext}}} = 1$. Thus $\mathcal{K}_{t_{\text{ext}}} = \{0\}$, contradicting $\tilde{z} \in \mathcal{K}_{t_{\text{ext}}}$. Therefore $L_i \cdot z_i = 0$ for some $i \in [t_{\text{ext}}]$.

It remains to bound the probability of this event. Fix $i \in [t_{\text{ext}}]$ and consider the recorded evaluation of index i , whose challenge is L_i . Before conditioning on acceptance, its evaluation point is a fresh sample from $\mathcal{Q}(\text{pp})$, independent of the first i recorded evaluations and hence of z_i . By Lemma 11, its challenge L satisfies $L \cdot z_i = 0$ with probability at most δ_{ext} . The joint event that the evaluation is accepted and $L \cdot z_i = 0$ is contained in this event. Hence, by Lemma 12, conditioned on acceptance, $L_i \cdot z_i = 0$ holds with probability at most $\delta_{\text{ext}}/\varepsilon$. Consequently, applying the union bound over $i \in [t_{\text{ext}}]$ gives the desired result. $\square$

Theorem 11. *If the query sampler $\mathcal{Q}$ is admissible, then the scheme of Protocol 3 is low-precision knowledge sound with respect to $\mathcal{Q}$.*

Proof. Fix a stateful PPT adversary $\mathcal{A}$. We construct an emulator $\mathcal{E}$. Given access to $\mathcal{A}$, and on inputs pp , c and η , $\mathcal{E}$ proceeds as follows.

1. From the oracle queries made by $\mathcal{A}$ before outputting c , $\mathcal{E}$ runs TreeBuilder to recover the partial matrix $(u_{i,j})_{i \in [r_0], j \in I}$ together with the Merkle paths of its columns, where $\mathcal{M} = [N] \setminus I$ denotes the missing columns. It sets the opening hint u to the collection of these paths.
2. $\mathcal{E}$ runs $\mathcal{A}$ on input η and acts honestly in the evaluation protocol until $\mathcal{A}$ outputs s , obtaining transcript π . If $\Pi.\text{Verify}(\text{pp}, c, s, \eta, \pi) = 0$, then $\mathcal{E}$ outputs $(s, \pi; \perp, \perp)$ and terminates. Otherwise, it records the proof as $(\eta_0, s_0, \pi_0) := (\eta, s, \pi)$.

3. $\mathcal{E}$ repeats the following until t_{ext} accepted proofs have been recorded: rewind $\mathcal{A}$ to its state after outputting c , sample a fresh $\eta' \leftarrow \mathcal{Q}(\text{pp})$ and runs the evaluation protocol at η' using fresh verifier randomness. Whenever the proof is accepted with value s' and transcript π' , it records (η', s', π') as the next proof (η_i, s_i, π_i) .
4. For each recorded proof $i \in [t_{\text{ext}}]$, let L_i denote its row challenge and y_i its reduced row response. $\mathcal{E}$ solves the linear system $L_i \cdot (x_a)_{a=0}^{r_0-1} = y_i$, $i \in [t_{\text{ext}}]$, over $\mathbb{Z}_{2^m}$. If the system has no solution, or if its solutions do not all agree modulo 2^k , then $\mathcal{E}$ outputs $(s, \pi; \perp, u)$. Otherwise, it outputs $(s, \pi; \hat{v}, u)$, where $\hat{v} \in \mathbb{Z}_{2^k}^T$ is the common reduction of the solutions modulo 2^k .

We first show that $\mathcal{E}$ runs in expected polynomial time. We write ε for the probability that $\mathcal{A}$ produces an accepting evaluation proof. Then, $\mathcal{E}$ proceeds beyond step 2 with probability exactly ε , and each iteration of step 3 records an accepting proof with probability exactly ε , so step 3 performs $(t_{\text{ext}} - 1)/\varepsilon$ evaluations in expectation. The expected total number of evaluations is therefore $1 + \varepsilon \cdot (t_{\text{ext}} - 1)/\varepsilon = t_{\text{ext}}$, which is polynomial in λ . Furthermore, TreeBuilder is straight-line and polynomial time, and the linear system of step 4 has $t_{\text{ext}} r_1$ scalar equations in the T entries of x over $\mathbb{Z}_{2^m}$, which is polynomial in λ . Writing every element of $\mathbb{Z}_{2^m}$ as a power of two times a unit, Gaussian elimination solves the system in polynomial time provided that each pivot is a remaining entry with the smallest power of two. The elimination also exhibits the solutions as a particular solution plus an arbitrary element of the kernel of the system, whose generators it produces, so $\mathcal{E}$ can decide in polynomial time whether all solutions agree modulo 2^k . Hence $\mathcal{E}$ runs in expected polynomial time.

We now analyze the distribution of the output of $\mathcal{E}$. The distribution of (c, η, s, π) is identical in $\text{Real}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$ and in $\text{Emul}_{\mathcal{Q}, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$, since $\mathcal{E}$ runs the initial evaluation honestly. It thus suffices to show that it holds in at most a negligible proportion of executions that, simultaneously, $\text{II.Verify}(\text{pp}, c, s, \eta, \pi) = 1$ and $\text{II.Open}(\text{pp}, c; \hat{v}, u) = 0$. Recall that II.Open rejects $\hat{v} = \perp$.

In the following, we condition on good Merkle extraction, whose failure contributes δ_{mt} . Fix the state of $\mathcal{A}$ immediately after it outputs c . Suppose first that no quotient message exists for c . By Lemma 9, $\varepsilon \leq \delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma$. Hence the probability that the initial evaluation is accepted, and therefore that $\mathcal{E}$ is required to produce a valid low-precision opening, is at most this quantity. Averaging over all post-commitment states for which no quotient message exists, their total contribution to the failure probability is at most $\delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma$. We may therefore assume from now on that the quotient message w exists. By Lemma 9, it is unique.

Set $\delta := t_{\text{ext}}((1 - \frac{2d}{3N})^\gamma + \delta_{\text{ext}})$. By assumption, δ is negligible in λ , and hence so is $\sqrt{\delta}$. States for which $\varepsilon \leq \sqrt{\delta}$ contribute at most $\sqrt{\delta}$ to the failure probability, since the initial evaluation is accepted with probability at most $\sqrt{\delta}$. We may thus restrict attention to states satisfying $\varepsilon > \sqrt{\delta}$ and condition on the initial evaluation being accepted. Under this conditioning, all t_{ext} recorded evaluations, including the initial one, are independent fresh evaluations conditioned on acceptance.

It remains to show that $\mathcal{E}$ produces a valid opening whenever neither bad event occurs. Since every recorded evaluation satisfies $y_i = L_i \cdot (w_a)_{a=0}^{r_0-1}$, the quotient message w is a solution to the system in step 4. Hence the system is consistent, and $\mathcal{E}$ outputs $\hat{v} = \pi_k(x)$ for some solution x . Fix $j \in [r_1]$. Since both x and w satisfy the recorded equations, their j -th columns satisfy $L_i \cdot (x_{a,j} - w_{a,j})_{a=0}^{r_0-1} = 0$ for every $i \in [t_{\text{ext}}]$. Thus $z := (x_{a,j} - w_{a,j})_{a=0}^{r_0-1}$ lies in the kernel of the recorded row challenges. Since the second bad event does not occur, $\text{Po2}_m(z) \geq k$, and hence $z \in 2^k \mathbb{Z}_{2^m}^{r_0}$. Therefore the j -th columns of x and w agree modulo 2^k . As this holds for every $j \in [r_1]$, we have $\pi_k(x) = \pi_k(w)$. Hence $\hat{v} = \pi_k(w)$. Using the similar argument at the end of the proof of Theorem 10, we conclude $\text{II.Open}(\text{pp}, c; \hat{v}, u) = 1$.

Combining the preceding bounds, the probability that the initial evaluation is accepted while $\text{II.Open}(\text{pp}, c; \hat{v}, u) =$

0 is at most $\delta_{\text{mt}} + \delta_{\text{pgap}} + (1 - \frac{d}{3N})^\gamma + 3\sqrt{\delta}$, which is negligible. This completes the proof of the theorem. $\square$

Finally, the emulator constructed above is consistent with the vector that explains the accepted evaluation proofs, which is the property used by Theorem 8.

Corollary 2. *If the query sampler $\mathcal{Q}$ is admissible, then the scheme of Protocol 3 has consistent extraction with respect to $\mathcal{Q}$.*

Proof. Fix a stateful PPT adversary $\mathcal{A}$, let $\mathcal{E}$ be the emulator of Theorem 11, and consider the experiment of Definition 12. We condition on good Merkle extraction and on $\mathcal{E}$ outputting a valid opening $\hat{v}$ whenever its initial proof is accepted; both fail with negligible probability. Suppose that all $B + 1$ proofs are accepted. By Lemma 9, the commitment has a quotient message w , since otherwise even the initial proof is accepted with negligible probability. By Lemma 10, a proof whose reduced row response differs from the corresponding combination of the rows of w is accepted with probability at most $(1 - \frac{2d}{3N})^\gamma$, so by a union bound over the $B + 1$ proofs, except with negligible probability every accepted proof satisfies $s_i = \text{Tensor}(\eta_i) \cdot w$. Thus w explains all accepted proofs. The proofs recorded by $\mathcal{E}$ satisfy the same identity, so w solves the linear system set up in its last step, and since $\mathcal{E}$ outputs the common reduction of all solutions modulo 2^k , we have $\pi_k(w) = \hat{v}$. Hence the experiment outputs 0 except with negligible probability. $\square$

Complexity. Let $T_{\text{enc}}(r)$ denote the time required to encode a message of length r . For the commitment, the prover performs $O(r_0 \cdot T_{\text{enc}}(r_1)) \mathbb{Z}_{2^\ell}$ operations and constructs a Merkle tree over N columns. For the evaluation proof, the prover computes the row combination using $O(T) \mathbb{Z}_{2^\ell}$ operations and provides γ Merkle openings; the verifier performs $O(\gamma r_0 + r_1) \mathbb{Z}_{2^\ell}$ operations and verifies γ Merkle paths, and the proof contains $O(\gamma r_0 + r_1) \mathbb{Z}_{2^\ell}$ elements.

Remark 3 (Non-uniform tensor query). *The security of Protocol 3 holds only when the query η is sampled uniformly at random. When the evaluation points are not uniform (such as the evaluation point for the first vector in the IOP from Section 4), we add an extra testing phase as in Brakedown [GLSTW23]. Specifically, the parties execute the evaluation protocol, except that the verifier samples a uniformly random $\tilde{\eta} \in \mathbb{Z}_{2^\ell}^{2t_0}$. Both parties set $\tilde{L} := \text{Tensor}(\tilde{\eta}) \in \mathbb{Z}_{2^\ell}^{T_0}$, and run the remaining protocol with $\tilde{L}$, omitting the final evaluation check.*

7 Permutation Checks over Rings

In this section, we present a permutation check over rings.

7.1 Permutation Relations over $\mathbb{Z}_{2^k}$ and $\mathbb{F}_{2^k}$

We first define the permutation relation over $\mathbb{Z}_{2^k}$ and its counterpart over $\mathbb{F}_{2^k}$.

Definition 13. *The ring permutation relation $\mathcal{R}_{\text{perm}}^{\text{R},k}$ is the set of all tuples $(\mathbf{i}, \mathbf{x}; \mathbf{w}) = (\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket); (v_0, v_1))$, where σ is a permutation of $[T]$, $v_0, v_1 \in \mathbb{Z}_{2^k}^T$, and $v_1 = \sigma(v_0)$.*

Definition 14. *The field permutation relation $\mathcal{R}_{\text{perm}}^{\text{F},k}$ is the set of all tuples $(\mathbf{i}, \mathbf{x}; \mathbf{w}) = (\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket); (v_0, v_1))$, where σ is a permutation of $[T]$, $v_0, v_1 \in \mathbb{F}_{2^k}^T$, and $v_1 = \sigma(v_0)$.*

Since we want to reduce the proof of the ring permutation relation to its field counterpart, we generalize the notion of an oracle so that its underlying entries can be interpreted both as ring elements and as field elements. In particular, the generalized oracle supports queries over both the ring and the field.

Fix a basis $\zeta_0, \dots, \zeta_{k-1}$ of $\mathbb{F}_{2^k}$ over $\mathbb{F}_2$. For every $x \in \mathbb{Z}_{2^k}$, we use $\text{Bin}(x) = (x_0, \dots, x_{k-1}) \in \{0, 1\}^k$ to denote the bit decomposition of x . We then associate each $x \in \mathbb{Z}_{2^k}$ with an element of $\mathbb{F}_{2^k}$ via the map $\varphi(x) := \sum_{a=0}^{k-1} x_a \zeta_a$.

Definition 15. A compatible oracle $\llbracket u \rrbracket$ stores a vector as follows. If $u \in \mathbb{Z}_{2^k}^T$, then $\llbracket u \rrbracket$ stores a lift $\hat{u} \in \mathbb{Z}_{2^m}^T$ of u ; if $u \in \mathbb{Z}_{2^m}^T$, then it simply stores $\hat{u} = u$. The oracle supports the following two types of queries:

- a ring query at $\eta \in \mathbb{Z}_{2^m}^{2t}$, to which it responds with $\text{Tensor}(\eta) \cdot \hat{u}$.
- a field query at $\eta' \in \mathbb{F}_{2^m}^{2t}$, to which it responds with $\text{Tensor}(\eta') \cdot \varphi(\pi_k(\hat{u}))$.

We refer to an oracle that supports only field queries as a *field oracle*. The compatible oracle thus serves as a bridge between the ring oracle and the field oracle. Under this setup, we show that an IOP for the corresponding field permutation relation can be used to prove the ring permutation relation.

Theorem 12. Let Π_{perm} be an IOP for $\mathcal{R}_{\text{perm}}^{\text{F},k}$ that uses field oracles and has soundness error ϵ_{perm} . Then there exists an IOP for $\mathcal{R}_{\text{perm}}^{\text{R},k}$ that uses compatible oracles and has the same soundness error ϵ_{perm} . Additionally, if Π_{perm} is perfectly complete, then so is the resulting IOP.

Proof. We construct the IOP for $\mathcal{R}_{\text{perm}}^{\text{R},k}$ by running Π_{perm} and answering the field queries using the corresponding compatible oracles.

Concretely, consider an instance $(\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket))$ of $\mathcal{R}_{\text{perm}}^{\text{R},k}$. For each $b \in \{0, 1\}$, a field query at η' to the compatible oracle $\llbracket v_b \rrbracket$ is answered by $\text{Tensor}(\eta') \cdot \varphi(\pi_k(\hat{v}_b)) = \text{Tensor}(\eta') \cdot \varphi(v_b)$, which is exactly the answer returned by the field oracle $\llbracket \varphi(v_b) \rrbracket$. Therefore, for any prover, the view of the verifier of Π_{perm} on $(\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket))$ is identical to its view on the corresponding field instance $(\sigma, (\llbracket \varphi(v_0) \rrbracket, \llbracket \varphi(v_1) \rrbracket))$.

Suppose that $(\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket)) \notin \mathcal{L}(\mathcal{R}_{\text{perm}}^{\text{R},k})$. Since φ is bijective, we must have $(\sigma, (\llbracket \varphi(v_0) \rrbracket, \llbracket \varphi(v_1) \rrbracket)) \notin \mathcal{L}(\mathcal{R}_{\text{perm}}^{\text{F},k})$. Hence, by the soundness of Π_{perm} , every prover is accepted with probability at most ϵ_{perm} . Thus, the resulting protocol has soundness error ϵ_{perm} .

Finally, suppose that Π_{perm} is perfectly complete and that $(\sigma, (\llbracket v_0 \rrbracket, \llbracket v_1 \rrbracket)) \in \mathcal{L}(\mathcal{R}_{\text{perm}}^{\text{R},k})$. This implies we have $(\sigma, (\llbracket \varphi(v_0) \rrbracket, \llbracket \varphi(v_1) \rrbracket)) \in \mathcal{L}(\mathcal{R}_{\text{perm}}^{\text{F},k})$. The honest prover can therefore run the honest prover of Π_{perm} on $\varphi(v_0)$ and $\varphi(v_1)$, and the verifier accepts with probability 1. Hence, the resulting protocol is also perfectly complete. $\square$

Remark 4 (Lookup Relations). *The same approach extends straightforwardly to lookup relations, which roughly require the witness vector v to be contained in a given public table. In particular, by applying φ to the involved vectors, a lookup relation over $\mathbb{Z}_{2^k}$ can be reduced to the corresponding lookup relation over $\mathbb{F}_{2^k}$. The compatible oracles can then be used to answer the field queries of any IOP for the field lookup relation, exactly as in the permutation case.*

7.2 Definition of Compatible Commitment

A compatible commitment scheme instantiates the compatible oracles of Definition 15. It takes the form $\Pi = (\text{Setup}, \text{Commit}, \text{ROpen}, \text{RProve}, \text{RVerify}, \text{FOpen}, \text{FProve}, \text{FVerify})$ and consists of two components that share the algorithms Setup and Commit.

Definition 16. A compatible commitment is a tuple $\Pi = (\text{Setup}, \text{Commit}, \text{ROpen}, \text{RProve}, \text{RVerify}, \text{FOpen}, \text{FProve}, \text{FVerify})$ of algorithms with the following syntax.

- $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$. On input the security parameter λ and a size parameter $t = O(\log \lambda)$, $\Pi.\text{Setup}$ samples public parameters $\text{pp} := (\text{pp}_R, \text{pp}_F)$. pp_R includes (among other things) the knowledge domain $\mathbb{Z}_{2^k}$ and the evaluation domain. pp_F includes (among other things) the base field $\mathbb{F}_{2^k}$ and the extension field $\mathbb{F}_{2^\mu}$.
- $(c, u) \leftarrow \Pi.\text{Commit}(\text{pp}, v)$. On input a vector $v \in \mathbb{Z}_{2^m}^T$, $\Pi.\text{Commit}$ returns a commitment c to v and an opening hint u .
- $b \leftarrow \Pi.\text{ROpen}(\text{pp}, c; \bar{v}, u)$. On input a commitment c , a claimed low-precision opening $\bar{v} \in \mathbb{Z}_{2^k}^T$ and an opening hint u , $\Pi.\text{ROpen}$ verifies whether $\bar{v}$ is a valid low-precision opening of c , using u . We set $\Pi.\text{ROpen}(\text{pp}, c; \perp, u) := 0$.
- $\pi \leftarrow \Pi.\text{RProve}(\text{pp}, c, s, \eta; v, u)$. On input a commitment c , a purported evaluation $s \in \mathbb{Z}_{2^m}$, an evaluation point $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$, a vector $v \in \mathbb{Z}_{2^m}^T$, and an opening hint u , $\Pi.\text{RProve}$ generates an evaluation proof π for the claim $\text{Tensor}(\eta) \cdot v = s$.
- $b \leftarrow \Pi.\text{RVerify}(\text{pp}, c, s, \eta, \pi)$. On input a commitment c , a purported evaluation $s \in \mathbb{Z}_{2^m}$, an evaluation point $\eta \in \mathbb{Z}_{2^m}^{2t}$, and a proof π , $\Pi.\text{RVerify}$ outputs a success bit $b \in \{0, 1\}$.
- $b \leftarrow \Pi.\text{FOpen}(\text{pp}, c; w, u)$. On input a commitment c , a claimed opening $w \in \mathbb{F}_{2^k}^T$ and an opening hint u , $\Pi.\text{FOpen}$ verifies whether w is a valid opening of c . We set $\Pi.\text{FOpen}(\text{pp}, c; \perp, u) := 0$.
- $\pi \leftarrow \Pi.\text{FProve}(\text{pp}, c, s, \eta'; w, u)$. On input a commitment c , a purported evaluation $s \in \mathbb{F}_{2^\mu}$, an evaluation point $\eta' = (1 - \theta_0, \theta_0, \dots, 1 - \theta_{t-1}, \theta_{t-1}) \in \mathbb{F}_{2^\mu}^{2t}$, a vector $w \in \mathbb{F}_{2^k}^T$, and an opening hint u , $\Pi.\text{FProve}$ generates an evaluation proof π for the claim $\text{Tensor}(\eta') \cdot w = s$.
- $b \leftarrow \Pi.\text{FVerify}(\text{pp}, c, s, \eta', \pi)$. On input a commitment c , a purported evaluation $s \in \mathbb{F}_{2^\mu}$, an evaluation point $\eta' \in \mathbb{F}_{2^\mu}^{2t}$, and a proof π , $\Pi.\text{FVerify}$ outputs a success bit $b \in \{0, 1\}$.

The scheme is complete if, for an honestly committed vector v with $\bar{v} := \pi_k(v)$ and $w := \varphi(\bar{v})$, $\Pi.\text{ROpen}$ accepts $\bar{v}$, $\Pi.\text{FOpen}$ accepts w , and $\Pi.\text{RVerify}$ and $\Pi.\text{FVerify}$ accept honestly generated proofs at every evaluation point.

Intuitively, the compatible commitment should act as a commitment scheme over both the ring and the field. More precisely, the ring-side algorithms (Setup, Commit, ROpen, RProve, RVerify) are required to satisfy the security requirements of Section 6. For the field-side algorithms (Setup, Commit, FOpen, FProve, FVerify), following [DP25], we require the resulting scheme to be binding and extractable.

Definition 17 ([DP25]). For every compatible commitment Π , security parameter λ , size parameter t , PPT query sampler $\mathcal{Q}_F$ and stateful PPT adversary $\mathcal{A}$, we define the field binding experiment $\text{FieldBind}_{\mathcal{A}}^{\Pi, t}(\lambda)$ as follows

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$.
2. The adversary outputs $(c, w_0, u_0, w_1, u_1) \leftarrow \mathcal{A}(\text{pp})$, where c is a commitment, $w_0, w_1 \in \mathbb{F}_{2^k}^T$.
3. The output of the experiment is defined to be 1 if $\Pi.\text{FOpen}(\text{pp}, c; w_0, u_0) = \Pi.\text{FOpen}(\text{pp}, c; w_1, u_1) = 1$ and $w_0 \neq w_1$. Otherwise, it is defined to 0.

We say that Π is field binding if, for each stateful PPT adversary $\mathcal{A}$, each $\lambda \in \mathbb{N}$ and each $t = O(\log \lambda)$, $\Pr[\text{FieldBind}_{\mathcal{A}}^{\Pi, t}(\lambda) = 1] \leq \text{negl}(\lambda)$.

Definition 18 ([DP25]). For each compatible commitment scheme Π , security parameter λ , size parameter t , PPT query sampler $\mathcal{Q}_F$, stateful PPT adversary $\mathcal{A}$, expected PPT emulator $\mathcal{E}$, and PPT distinguisher $\mathcal{D}$, we define two random variables $\text{Real}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$ and $\text{Emul}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$, each valued in $\{0, 1\}$, as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$, $\mathcal{Q}_F$ and $\mathcal{E}$.
2. The adversary outputs a commitment $c \leftarrow \mathcal{A}(\text{pp})$.
3. The query sampler outputs $\eta' \leftarrow \mathcal{Q}_F(\text{pp})$.
4. The experimenter proceeds in one of two separate ways:
 - $\text{Real}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$: Run $(s', \pi') \leftarrow \mathcal{A}(\eta')$. Output the single bit $\mathcal{D}(c, \eta', s', \pi')$.
 - $\text{Emul}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$: Run $(s', \pi'; \hat{w}, u) \leftarrow \mathcal{E}^{\mathcal{A}}(\eta)$, where the emulator may repeatedly reset $\mathcal{A}$ to its state immediately after outputting c , and where $\hat{w} \in \mathbb{F}_{2^k}^T \cup \{\perp\}$. Output the single bit $\mathcal{D}(c, \eta', s', \pi') \wedge (\Pi.\text{FVerify}(\text{pp}, c, s', \eta', \pi') = 1 \implies \Pi.\text{FOpen}(\text{pp}, c; \hat{w}, u) = 1 \wedge \text{Tensor}(\eta') \cdot \hat{w} = s')$

We say that Π is field knowledge sound with respect to $\mathcal{Q}_F$ if, for each stateful PPT adversary $\mathcal{A}$, there exists an expected PPT emulator $\mathcal{E}$ such that, for each PPT distinguisher $\mathcal{D}$, the ensembles $\text{Real}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$ and $\text{Emul}_{\mathcal{Q}_F, \mathcal{A}, \mathcal{E}, \mathcal{D}}^{\Pi, t}(\lambda)$, indexed by $\lambda \in \mathbb{N}$ and $t = O(\log \lambda)$, are statistically close.

These notions ensure that the ring side and the field side each form a secure commitment scheme on their own. However, this alone is insufficient for a compatible commitment, which requires the ring and field queries to be answered consistently with respect to the underlying vector. We therefore introduce a compatibility property to enforce this consistency: the vector extracted by the field emulator must be the image under φ of the vector extracted by the ring emulator.

Definition 19. For every compatible commitment Π , security parameter λ , size parameter t , PPT query samplers $\mathcal{Q}_R$ and $\mathcal{Q}_F$, stateful PPT adversary $\mathcal{A}$, ring emulator $\mathcal{E}_R$ w.r.t $\mathcal{Q}_R$, and field emulator $\mathcal{E}_F$ w.r.t $\mathcal{Q}_F$, we define the experiment $\text{Compat}_{\mathcal{Q}_R, \mathcal{Q}_F, \mathcal{A}, \mathcal{E}_R, \mathcal{E}_F}^{\Pi, t}(\lambda)$ as follows:

1. The experimenter samples $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$, and gives pp to $\mathcal{A}$, $\mathcal{Q}_R$, $\mathcal{Q}_F$, $\mathcal{E}_R$ and $\mathcal{E}_F$.
2. The adversary outputs a commitment $c \leftarrow \mathcal{A}(\text{pp})$.
3. The query samplers output $\eta \leftarrow \mathcal{Q}_R(\text{pp})$ and $\eta' \leftarrow \mathcal{Q}_F(\text{pp})$.
4. The experimenter runs $(s, \pi; \hat{v}, u) \leftarrow \mathcal{E}_R^{\mathcal{A}}(\eta)$ and $(s', \pi'; \hat{w}, u') \leftarrow \mathcal{E}_F^{\mathcal{A}}(\eta')$, where each emulator may repeatedly reset $\mathcal{A}$ to its state immediately after outputting c .
5. The output of the experiment is defined to be 1 if $\Pi.\text{RVerify}(\text{pp}, c, s, \eta, \pi) = 1$, $\Pi.\text{FVerify}(\text{pp}, c, s', \eta', \pi') = 1$ and $\hat{w} \neq \varphi(\hat{v})$. Otherwise, it is defined to be 0.

We say that Π is compatible with respect to $(\mathcal{Q}_R, \mathcal{Q}_F)$ if, for each stateful PPT adversary $\mathcal{A}$, there exist a ring emulator $\mathcal{E}_R$ and a field emulator $\mathcal{E}_F$ for $\mathcal{A}$ such that, for each $\lambda \in \mathbb{N}$ and each $t = O(\log \lambda)$, $\Pr[\text{Compat}_{\mathcal{Q}_R, \mathcal{Q}_F, \mathcal{A}, \mathcal{E}_R, \mathcal{E}_F}^{\Pi, t}(\lambda) = 1] \leq \text{negl}(\lambda)$.

We then demonstrate that if the ring- and field-side algorithms satisfy their respective security properties, and every valid ring opening implies the corresponding valid field opening, the resulting commitment is compatible.

Lemma 14. *Suppose that, for all pp , c , $\bar{v}$ and u , an accepted ring opening is an accepted field opening, that is, $\Pi.\text{ROpen}(\text{pp}, c; \bar{v}, u) = 1$ implies $\Pi.\text{FOpen}(\text{pp}, c; \varphi(\bar{v}), u) = 1$. If Π is low-precision knowledge sound with respect to $\mathcal{Q}_R$, field knowledge sound with respect to $\mathcal{Q}_F$ and field binding, then Π is compatible with respect to $(\mathcal{Q}_R, \mathcal{Q}_F)$.*

Proof. Fix a stateful PPT adversary $\mathcal{A}$, and let $\mathcal{E}_R$ and $\mathcal{E}_F$ be a ring emulator and a field emulator for $\mathcal{A}$, respectively. Consider the experiment of Definition 19.

We first bound the probability that either emulator produces an opening inconsistent with an accepting evaluation proof. Restricting the first four steps of the experiment to the ring emulator gives the experiment Emul of Definition 10 with the distinguisher that always outputs 1. Since the corresponding experiment Real outputs 1 with probability 1, statistical closeness implies that

$$\Pr [\Pi.\text{RVerify}(\text{pp}, c, s, \eta, \pi) = 1 \wedge \Pi.\text{ROpen}(\text{pp}, c; \hat{v}, u) = 0] \leq \text{negl}(\lambda).$$

Similarly,

$$\Pr [\Pi.\text{FVerify}(\text{pp}, c, s', \eta', \pi') = 1 \wedge \Pi.\text{FOpen}(\text{pp}, c; \hat{w}, u') = 0] \leq \text{negl}(\lambda).$$

Condition on neither of these events occurring. If the compatibility experiment outputs 1, then $(\hat{v}, u)$ is an accepted ring opening of c and $(\hat{w}, u')$ is an accepted field opening of c , while $\hat{w} \neq \varphi(\hat{v})$. By hypothesis, the accepted ring opening $(\hat{v}, u)$ induces an accepted field opening $(\varphi(\hat{v}), u)$ of the same commitment c . Hence $(\varphi(\hat{v}), u)$ and $(\hat{w}, u')$ are two distinct accepted field openings of c .

We can therefore construct an adversary against field binding that runs $\mathcal{A}$, $\mathcal{E}_R$, and $\mathcal{E}_F$ exactly as in the compatibility experiment and outputs $(c, \varphi(\hat{v}), u, \hat{w}, u')$. Whenever the compatibility experiment outputs 1 and neither of the two negligible events above occurs, this adversary wins the field-binding experiment.

The resulting adversary runs in expected polynomial time. Let $\tau(\lambda)$ bound its expected running time. Suppose, toward a contradiction, that it wins with probability at least $1/p(\lambda)$ for some polynomial p and infinitely many λ . Truncating its execution after $2p(\lambda)\tau(\lambda)$ steps yields a PPT adversary. By Markov's inequality, the truncation occurs with probability at most $1/(2p(\lambda))$, so the truncated adversary still wins with probability at least $1/(2p(\lambda))$ for infinitely many λ , contradicting field binding. Therefore, the probability that the compatibility experiment outputs 1 is negligible in λ . $\square$

We now make precise the claim that a compatible commitment instantiates a compatible oracle. Let $\Pi_{\text{IOP}} = (\mathcal{P}, \mathcal{V})$ be an IOP with compatible oracles for an indexed relation $\mathcal{R}$ whose index is explicit, whose instance consists of the compatible oracles $\llbracket v_0 \rrbracket, \dots, \llbracket v_{r-1} \rrbracket$ of vectors $v_0, \dots, v_{r-1} \in \mathbb{Z}_{2^k}^T$, and whose prover sends explicit elements, ring oracles over $\mathbb{Z}_{2^m}$, and field oracles over $\mathbb{F}_{2^\mu}$. Let Π be a compatible commitment, Π_R a ring commitment, and Π_F a field commitment over $\mathbb{F}_{2^\mu}$. The *compiled argument* Π_{IOP}^Π for $\mathcal{R}$ proceeds as follows.

1. The prover commits to the canonical lift $\hat{v}_j$ of each v_j with Π , obtaining a commitment c_j and an opening hint u_j , and sends $c_0, \dots, c_{r-1}$ to the verifier; if two instance oracles hold the same vector, one commitment serves both. The ring opening of c_j is v_j and its field opening is $\varphi(v_j)$.
2. The parties run Π_{IOP} . Explicit messages are sent as they are, every ring oracle sent by $\mathcal{P}$ is replaced by a commitment with Π_R , and every field oracle sent by $\mathcal{P}$ by a commitment with Π_F .
3. A ring query of $\mathcal{V}$ to an instance oracle at a point η is answered by a value s together with a proof generated by $\Pi.\text{RProve}$, which the verifier checks with $\Pi.\text{RVerify}$; a field query at a point η' is answered by a value s' together with a proof generated by $\Pi.\text{FProve}$, which the verifier checks with $\Pi.\text{FVerify}$. Queries to the oracles sent by $\mathcal{P}$ are answered and checked with Π_R and Π_F in the same way. The verifier accepts if $\mathcal{V}$ accepts and every evaluation proof verifies.

As in Section 6.1, $\mathcal{Q}_R$ and $\mathcal{Q}_F$ denote the query samplers induced by the ring and field queries of $\mathcal{V}$, and knowledge soundness is stated for the contents of the instance oracles. The extractor has to output $\hat{v}_0, \dots, \hat{v}_{r-1} \in \mathbb{Z}_{2^k}^T$ with $(\mathbf{i}, (\llbracket \hat{v}_0 \rrbracket, \dots, \llbracket \hat{v}_{r-1} \rrbracket)) \in \mathcal{L}(\mathcal{R})$ whenever the verifier accepts. On the ring side, the emulator only recovers the low-precision opening, so we again need consistent extraction (Definition 12) to tie it to a vector explaining the ring answers. On the field side, the emulator of Definition 18 recovers a vector that explains the answer itself, and field binding makes this vector the same for all field queries to a commitment. Compatibility (Definition 19) then identifies the two sides.

Theorem 13. *Let $\Pi_{\text{IOP}} = (\mathcal{P}, \mathcal{V})$ be an IOP with compatible oracles for an indexed relation $\mathcal{R}$ over $\mathbb{Z}_{2^k}$ with soundness error ϵ_{IOP} . Suppose that the index is explicit, the instance consists of the compatible oracles $\llbracket v_0 \rrbracket, \dots, \llbracket v_{r-1} \rrbracket$ of vectors over $\mathbb{Z}_{2^k}$, and the prover sends explicit elements, ring oracles, and field oracles. Let Π be a complete compatible commitment whose ring side is evaluation binding and has consistent extraction with respect to $\mathcal{Q}_R$, whose field side is field binding and field knowledge sound with respect to $\mathcal{Q}_F$, and which is compatible with respect to $(\mathcal{Q}_R, \mathcal{Q}_F)$ for the ring emulator of consistent extraction. Let ϵ_R bound the statistical distance between Real and Emul in Definition 10 for this emulator, and ϵ_C the probability that the experiment of Definition 19 outputs 1. Let Π_R be a complete ring commitment that is evaluation binding with respect to $\mathcal{Q}_R$, and Π_F a complete field commitment that is field binding and field knowledge sound with respect to $\mathcal{Q}_F$. Then the compiled argument Π_{IOP}^Π is a complete and knowledge-sound argument for $\mathcal{R}$ with knowledge error $\epsilon_{\text{IOP}} + r \cdot (\epsilon_R + \epsilon_C) + \text{negl}(\lambda)$.*

Proof. Completeness follows from that of Π_{IOP} , Π , Π_R and Π_F : the honest prover commits to the canonical lifts, every evaluation proof verifies, and the answers are exactly the answers of the oracles of Π_{IOP} , since the field opening of c_j is $\varphi(v_j) = \varphi(\pi_k(\hat{v}_j))$.

For knowledge soundness, fix a PPT prover $\mathcal{A}$ for the compiled argument, and let $\mathcal{E}_R$ and $\mathcal{E}_F$ be the ring and field emulators of the hypotheses. The extractor $\mathcal{E}$ runs $\mathcal{A}$ against the honest verifier, records the transcript together with the verdict of the verifier, and then, for each instance commitment c_j , runs $\mathcal{E}_R$ on the point of a ring query of $\mathcal{V}$ to c_j , rewinding $\mathcal{A}$ to its state immediately after sending $c_0, \dots, c_{r-1}$, obtaining $(\hat{v}_j, u_j)$ with $\hat{v}_j \in \mathbb{Z}_{2^k}^T \cup \{\perp\}$. It outputs $\hat{v} := (\hat{v}_0, \dots, \hat{v}_{r-1})$ and runs in expected polynomial time. For the analysis, we also run $\mathcal{E}_F$ on the point of every field query to an instance commitment or to a commitment of Π_F , rewinding $\mathcal{A}$ in the same way, and write $(\hat{w}, u')$ for its output; these runs are not part of $\mathcal{E}$.

We bound the probability that the verifier accepts while $(\mathbf{i}, \llbracket \hat{v} \rrbracket) \notin \mathcal{L}(\mathcal{R})$. Consider the following events.

- *Ring extraction fails.* For some $j \in [r]$, every ring proof for c_j in the transcript verifies, but $\Pi.\text{ROpen}(\text{pp}, c_j; \hat{v}_j, u_j) = 0$. As in the proof of Theorem 8, Definition 10 bounds this event by $r \cdot \epsilon_R$.
- *Field extraction fails.* For some field query, to an instance commitment or to a commitment of Π_F , the proof verifies but the run of $\mathcal{E}_F$ on its point does not return an accepted field opening $\hat{w}$ with $\text{Tensor}(\eta') \cdot \hat{w} = s'$. By Definition 18 and the field knowledge soundness of Π_F , each run fails with negligible probability, and there are polynomially many runs.
- *Field openings disagree.* For some commitment, two runs of $\mathcal{E}_F$ return accepted field openings $\hat{w} \neq \hat{w}'$. Outputting the two openings together with their hints gives an expected polynomial-time adversary against field binding, and truncating its execution as in the proof of Lemma 14 turns a nonnegligible success probability into a PPT adversary. Hence this event has probability $\text{negl}(\lambda)$.
- *Compatibility fails.* For some $j \in [r]$ that receives a field query, the ring proof used by $\mathcal{E}_R$ and the field proof used by one run of $\mathcal{E}_F$ verify, but $\hat{w} \neq \varphi(\hat{v}_j)$. This is the experiment of Definition 19 for c_j , so the event has probability at most $r \cdot \epsilon_C$.

- *Evaluation binding fails.* For some commitment c of Π_R sent in step 2, there is no $w_c \in \mathbb{Z}_{2^m}^T$ with $\text{Tensor}(\eta) \cdot w_c = s$ for every accepted answer (η, s) to c . Evaluation binding of Π_R bounds this event by $\text{negl}(\lambda)$.
- *Inconsistent extraction.* For some $j \in [r]$, there is no $w \in \mathbb{Z}_{2^m}^T$ with $\pi_k(w) = \hat{v}_j$ and $\text{Tensor}(\eta) \cdot w = s$ for every accepted ring answer (η, s) to c_j . Consistent extraction bounds this event by $\text{negl}(\lambda)$.

Condition on none of these events occurring and on the verifier accepting. For each instance commitment c_j , fix a vector w_{c_j} that explains its accepted ring answers and satisfies $\pi_k(w_{c_j}) = \hat{v}_j$. Every accepted field answer (η', s') to c_j satisfies $s' = \text{Tensor}(\eta') \cdot \hat{w}$ for the common field opening $\hat{w} = \varphi(\hat{v}_j) = \varphi(\pi_k(w_{c_j}))$. Hence the ring and field answers to c_j are exactly the answers of the compatible oracle $\llbracket \hat{v}_j \rrbracket$ holding the lift w_{c_j} , a choice allowed by the model. Likewise, the answers to a commitment of Π_R are the answers of a ring oracle holding w_c , and the answers to a commitment of Π_F are those of a field oracle holding its common field opening. The transcript is therefore a transcript of Π_{IOP} against some prover strategy. If $(i, \llbracket \hat{v} \rrbracket) \notin \mathcal{L}(\mathcal{R})$, the soundness of Π_{IOP} , which holds for every choice of the lifts, implies that $\mathcal{V}$ accepts with probability at most ϵ_{IOP} . A union bound over the events above gives the claimed knowledge error. $\square$

7.3 Instantiation of Compatible Commitment

We now show our instantiation of the compatible commitment in Protocol 4 based on our ring commitment scheme in Protocol 3. The key idea is to commit to the binary decomposition of the vector with our ring commitment scheme, as the binary components admit two natural reconstructions corresponding to the ring and field cases. On the ring side, they are recombined using the corresponding powers of two to recover the original ring vector. On the field side, since the underlying code is defined over $\mathbb{F}_2$, reducing the encoded components modulo 2 recovers the corresponding binary codewords, which can then be recombined using the basis $\zeta_0, \dots, \zeta_{k-1}$ to obtain the field representation over $\mathbb{F}_{2^k}$.

In more detail, recall that the ring commitment takes as input a vector over $\mathbb{Z}_{2^m}$. For a vector $v \in \mathbb{Z}_{2^m}^T$, we let $v^{(0)}, \dots, v^{(k-1)} \in \mathbb{Z}_{2^m}^T$ denote a bit decomposition of v satisfying $\pi_k(v) = \sum_{a=0}^{k-1} 2^a \pi_k(v^{(a)})$. For the honest prover, we assume that the input $v \in \mathbb{Z}_{2^m}^T$ is the canonical lift of some vector over $\mathbb{Z}_{2^k}$ and therefore $v^{(0)}, \dots, v^{(k-1)} \in \{0, 1\}^T$. Following the notation of Section 6, we write $(v_i)_{i=0}^{r_0-1}$ for the matrix representation of v . We define the *binary matrix* of v by arranging the matrix representations of its binary components as $(v_i^{(a)})_{i \in [r_0], a \in [k]} \in \{0, 1\}^{r_0 \times kr_1}$. Similarly, for a vector $w \in \mathbb{F}_{2^k}^T$, let $w^{(0)}, \dots, w^{(k-1)} \in \{0, 1\}^T$ denote its binary components with respect to the basis $\zeta_0, \dots, \zeta_{k-1}$, such that $w = \sum_{a=0}^{k-1} \zeta_a w^{(a)}$. We write $(w_i^{(a)})_{i=0}^{r_0-1}$ for the matrix representation of $w^{(a)}$. Clearly, we have $(v_i)_{i=0}^{r_0-1} = \sum_{a=0}^{k-1} 2^a (v_i^{(a)})_{i=0}^{r_0-1}$, and $\varphi\left((v_i)_{i=0}^{r_0-1}\right) = \sum_{a=0}^{k-1} \zeta_a (v_i^{(a)})_{i=0}^{r_0-1}$.

We now prove that Protocol 4 is a complete and secure compatible commitment. For completeness, we check the three components of the protocol in turn: the ring and field openings, the ring and field evaluations, and the Booleanity certification.

Theorem 14 (Completeness). *If the prover commits to a canonical lift v and follows Protocol 4, then $\Pi.\text{ROpen}$ accepts $\pi_k(v)$, $\Pi.\text{FOpen}$ accepts $\varphi(\pi_k(v))$, the certification is accepted, and every ring and field evaluation at every evaluation point is accepted.*

Proof. Let $\bar{v} := \pi_k(v)$. Since v is the canonical lift of $\bar{v}$, they have the same binary components. For every $a \in [k]$ and $i \in [r_0]$, the honest opening hint satisfies $u_i^{(a)} = \text{Enc}_\ell(v_i^{(a)})$. Reducing modulo 2^k gives $\pi_k(u_i^{(a)}) = \text{Enc}_k(\bar{v}_i^{(a)})$. Moreover, the honest opening hint contains all committed columns, so there are no missing columns or disagreements. Hence $\Pi.\text{ROpen}$ accepts $\bar{v}$.

PROTOCOL 4. *Compatible Commitment Scheme*

We define $\Pi = (\text{Setup}, \text{Commit}, \text{ROpen}, \text{RProve}, \text{RVerify}, \text{FOpen}, \text{FProve}, \text{FVerify})$ as follows.

- $\text{pp} \leftarrow \Pi.\text{Setup}(1^\lambda, t)$. On input 1^λ and t , choose $t_0, t_1 \geq 1$, set $r_0 := 2^{t_0}$, $r_1 := 2^{t_1}$ and $T := r_0 r_1 = 2^t$. Choose the rings $\mathbb{Z}_{2^k}$, $\mathbb{Z}_{2^m}$, and $\mathbb{Z}_{2^\ell}$ for the ring commitment, and the fields $\mathbb{F}_{2^k}$ and $\mathbb{F}_{2^\mu}$ for the field commitment, where $\mathbb{F}_{2^\mu}$ is an extension field of $\mathbb{F}_{2^k}$. In addition, choose a binary $[N, r_1, d]$ -code $C \subseteq \mathbb{F}_2^N$ with $N = 2^{O(t)}$ and $d = \Omega(N)$, and set the repetition parameter $\gamma = \Theta(\lambda)$. Return all these parameters as the public parameters pp .
- $(c, u) \leftarrow \Pi.\text{Commit}(\text{pp}, v)$. On input $v \in \mathbb{Z}_{2^m}^T$, let $\tilde{v} \in \mathbb{Z}_{2^\ell}^T$ be its canonical lift to $\mathbb{Z}_{2^\ell}$. Compute its bit decomposition $(\tilde{v}^{(0)}, \dots, \tilde{v}^{(k-1)})$. For every $a \in [k]$, compute $(u_i^{(a)})_{i=0}^{r_0-1} = (\text{Enc}_\ell(\tilde{v}_i^{(a)}))_{i=0}^{r_0-1}$. For each $j \in [N]$, set the column block $(u_{i,j}^{(a)})_{i \in [r_0], a \in [k]}$. Output a Merkle commitment c to these N column blocks. Finally, run the IOP in Section 4 to check $\pi_k(\tilde{v}^{(a)} \circ (\tilde{v}^{(a)} - 1)) = 0$ for every $a \in [k]$.
- $b \leftarrow \Pi.\text{ROpen}(\text{pp}, c; \bar{v}, u)$. On input c , $\bar{v} \in \mathbb{Z}_{2^k}^T$, and a hint u of distinct Merkle paths against c missing the column blocks $\mathcal{M} \subseteq [N]$, check $|\Delta_k((u_i^{(a)})_{i=0}^{r_0-1}, (\text{Enc}_k(\bar{v}_i^{(a)}))_{i=0}^{r_0-1}) \cup \mathcal{M}| \stackrel{?}{<} \frac{d}{2}$ for every $a \in [k]$.
- $b \leftarrow \Pi.\text{FOpen}(\text{pp}, c; w, u)$. On input c , $w \in \mathbb{F}_{2^k}^T$, and a hint u of distinct Merkle paths against c missing the column blocks $\mathcal{M} \subseteq [N]$, check $|\Delta_1((u_i^{(a)})_{i=0}^{r_0-1}, (\text{Enc}_1(w_i^{(a)}))_{i=0}^{r_0-1}) \cup \mathcal{M}| \stackrel{?}{<} \frac{d}{2}$ for every $a \in [k]$.

Ring evaluations. $\Pi.\text{RProve}$ and $\Pi.\text{RVerify}$ apply the Fiat–Shamir heuristic to the following interactive protocol on c , $s \in \mathbb{Z}_{2^m}$ and $\eta = (\alpha_0, \beta_0, \dots, \alpha_{t-1}, \beta_{t-1}) \in \mathbb{Z}_{2^m}^{2t}$, in which $\mathcal{P}$ also holds v and u .

- $\mathcal{V}$ samples $\alpha_i^h, \beta_i^h \leftarrow \mathbb{Z}_{2^{\ell-m}}$ for $i \in [t_0]$ and sends them to $\mathcal{P}$; both parties set $\tilde{\alpha}_i := \alpha_i + 2^m \alpha_i^h$, $\tilde{\beta}_i := \beta_i + 2^m \beta_i^h$ and $\tilde{L} := \bigotimes_{i=0}^{t_0-1} (\tilde{\alpha}_i, \tilde{\beta}_i)$.
- $\mathcal{P}$ sends $\mathcal{V}$ the row combination $y^{(a)} := \tilde{L} \cdot (v_i^{(a)})_{i=0}^{r_0-1} \in \mathbb{Z}_{2^\ell}^{r_1}$ for every $a \in [k]$.
- $\mathcal{V}$ samples $j_l \leftarrow [N]$ for $l \in [\gamma]$ and sends $J := \{j_0, \dots, j_{\gamma-1}\}$; $\mathcal{P}$ sends the column blocks $(u_{i,j}^{(a)})_{i \in [r_0], a \in [k]}$ for $j \in J$, each with a Merkle path against c .
- $\mathcal{V}$ verifies the paths and requires $\tilde{L} \cdot (u_{i,j}^{(a)})_{i=0}^{r_0-1} \stackrel{?}{=} \text{Enc}_\ell(y^{(a)})_j$ for every $j \in J$ and $a \in [k]$, and $s \stackrel{?}{=} \sum_{a=0}^{k-1} 2^a \cdot \pi_m(y^{(a)}) \cdot R$, where $R = \bigotimes_{i=t_0}^{t-1} (\alpha_i, \beta_i)$.

Field evaluations. $\Pi.\text{FProve}$ and $\Pi.\text{FVerify}$ apply the Fiat–Shamir heuristic to the following interactive protocol on c , $s' \in \mathbb{F}_{2^\mu}$ and $\eta' = (1 - \theta_0, \theta_0, \dots, 1 - \theta_{t-1}, \theta_{t-1}) \in \mathbb{F}_{2^\mu}^{2t}$, in which $\mathcal{P}$ also holds v and u .

- Both parties set $L' = \bigotimes_{i=0}^{t_0-1} (1 - \theta_i, \theta_i)$. $\mathcal{P}$ sends $\mathcal{V}$ the row combinations $y^{(a)} := L' \cdot \pi_1((v_i^{(a)})_{i \in [r_0]}) \in \mathbb{F}_{2^\mu}^{r_1}$ for every $a \in [k]$.
- $\mathcal{V}$ samples $j_l \leftarrow [N]$ for $l \in [\gamma]$ and sends $J := \{j_0, \dots, j_{\gamma-1}\}$; $\mathcal{P}$ sends the column blocks $(u_{i,j}^{(a)})_{i \in [r_0], a \in [k]}$ for $j \in J$, each with a Merkle path against c .
- $\mathcal{V}$ verifies the paths and requires $L' \cdot \pi_1((u_{i,j}^{(a)})_{i=0}^{r_0-1}) \stackrel{?}{=} \text{Enc}_1(y^{(a)})_j$ for every $j \in J$ and $a \in [k]$, and $s' \stackrel{?}{=} \sum_{a=0}^{k-1} \zeta_a \cdot y^{(a)} \cdot R$, where $R := \bigotimes_{i=t_0}^{t-1} (1 - \theta_i, \theta_i)$.

Let $w := \varphi(\bar{v})$. By the definition of φ , the binary components of w with respect to the basis $\zeta_0, \dots, \zeta_{k-1}$ are exactly those of $\bar{v}$. Furthermore, $\pi_1(u_i^{(a)}) = \text{Enc}_1(w_i^{(a)})$, since $\pi_1 \circ \text{Enc}_\ell = \text{Enc}_1 \circ \pi_1$. Thus there are again no missing columns or disagreements, and $\Pi.\text{FOpen}$ accepts $w = \varphi(\bar{v})$.

We next consider a ring evaluation. For every $a \in [k]$, the honest prover sends $y^{(a)} = \tilde{L} \cdot (v_i^{(a)})_{i=0}^{r_0-1}$. Then we have $\tilde{L} \cdot (u_{i,j}^{(a)})_{i=0}^{r_0-1} = \text{Enc}_\ell(y^{(a)})_j$ for every opened position j and every $a \in [k]$. Hence all column checks are accepted. Moreover, reducing $\tilde{L}$ modulo 2^m gives $L := \bigotimes_{i=0}^{t_0-1} (\alpha_i, \beta_i)$. Therefore, $\sum_{a=0}^{k-1} 2^a \pi_m(y^{(a)}) \cdot R = L \cdot (v_i)_{i=0}^{r_0-1} \cdot R = \text{Tensor}(\eta) \cdot v = s$, where $R := \bigotimes_{i=t_0}^{t-1} (\alpha_i, \beta_i)$. Thus every honest ring evaluation is accepted.

For a field evaluation, let $L' := \bigotimes_{i=0}^{t_0-1} (1 - \theta_i, \theta_i)$ and $R' := \bigotimes_{i=t_0}^{t-1} (1 - \theta_i, \theta_i)$. For every $a \in [k]$, the honest prover sends $y^{(a)} = L' \cdot (v_i^{(a)})_{i=0}^{r_0-1}$. Since $\pi_1(u_i^{(a)}) = \text{Enc}_1(v_i^{(a)})$, we have $L' \cdot (\pi_1(u_{i,j}^{(a)}))_{i=0}^{r_0-1} = \text{Enc}_1(y^{(a)})_j$ for every opened position j and every $a \in [k]$. Hence all column checks are accepted. The value check also holds, since $\sum_{a=0}^{k-1} \zeta_a \cdot y^{(a)} \cdot R' = \text{Tensor}(\eta') \cdot \varphi(\bar{v}) = s'$. Thus every honest field evaluation is accepted.

Finally, each $v^{(a)}$ is binary by definition. Hence $v^{(a)} \circ (v^{(a)} - 1) = 0$ for every $a \in [k]$, and the Booleanity certification is accepted by the completeness of the IOP of Section 4. Therefore all checks in Protocol 4 are accepted. $\square$

For security, the ring side of Protocol 4 is the ring commitment of Section 6 applied to the binary components, and the field side is the small-field polynomial commitment of [DP25], so each half inherits the security of the underlying scheme. Compatibility then follows from Lemma 14 once we show that every accepted ring opening induces an accepted field opening.

Theorem 15 (Security). *Protocol 4 is a secure compatible commitment. In particular, it satisfies the security properties of the underlying ring and field commitments with respect to admissible ring and field queries, and is compatible with respect to every pair $(\mathcal{Q}_R, \mathcal{Q}_F)$ of admissible query samplers.*

Proof. The security of the compatible commitment follows directly from the underlying commitment schemes. On the ring side, Protocol 4 is exactly the ring commitment of Section 6 applied to the matrix consisting of the binary components $(v_i^{(a)})_{i,a}$. Hence its low-precision binding, evaluation binding, and low-precision knowledge soundness follow from the corresponding properties established therein. On the field side, this is exactly the small-field polynomial commitment of [DP25], and hence inherits its binding and knowledge soundness.

It remains to establish compatibility between the two sides. The Boolean check ensures that the low-precision witness extracted by the ring emulator is binary, and hence defines a valid binary decomposition of some $\bar{v} \in \mathbb{Z}_{2^k}^T$. We next show that every accepted ring opening induces the corresponding accepted field opening. Suppose that $\Pi.\text{ROpen}(\text{pp}, c; \bar{v}, u) = 1$. By definition, after reduction modulo 2^k , the opening hint u is sufficiently close to the encodings of the binary components $\bar{v}^{(a)}$. Reducing further modulo 2 can only remove disagreements, and $\pi_1 \circ \text{Enc}_k = \text{Enc}_1 \circ \pi_1$. Moreover, the binary components of $\varphi(\bar{v})$ with respect to the basis $\zeta_0, \dots, \zeta_{k-1}$ are precisely $\bar{v}^{(0)}, \dots, \bar{v}^{(k-1)}$. Therefore, $\Pi.\text{FOpen}(\text{pp}, c; \varphi(\bar{v}), u) = 1$. Therefore, compatibility of Protocol 4 follows from Lemma 14. $\square$

Complexity. Let $T_{\text{enc}}(r)$ denote the time required to encode a message of length r . For the commitment, the prover performs $O(k \cdot r_0 \cdot T_{\text{enc}}(r_1)) \mathbb{Z}_{2^\ell}$ operations and constructs a Merkle tree over the resulting N column blocks, while the Boolean check requires $O(k \cdot T) \mathbb{Z}_{2^m}$ operations for the prover and $O(\log(k \cdot T)) \mathbb{Z}_{2^m}$ operations for the verifier. For the ring proof, the prover computes one row combination for each binary component using $O(kT) \mathbb{Z}_{2^\ell}$ operations and provides γ Merkle openings; the verifier performs $O(k(\gamma r_0 + r_1)) \mathbb{Z}_{2^\ell}$ operations and verifies γ Merkle paths, and the proof contains $O(k(\gamma r_0 + r_1)) \mathbb{Z}_{2^\ell}$

elements. The field proof has the same asymptotic complexity, with $\mathbb{F}_{2^\mu}$ operations and elements in place of $\mathbb{Z}_{2^\ell}$ operations and elements.

Remark 5. *The above complexity can be highly optimized. In Section 7.4 we give some optimizations based on the fact that honest inputs are binary.*

7.4 Optimizations of the Compatible Commitment

In this subsection, we present several optimizations for compatible commitment.

7.4.1 Encoding over a Smaller Ring

Although the encoding algorithm is defined over $\mathbb{Z}_{2^\ell}$, performing the entire encoding over $\mathbb{Z}_{2^\ell}$ is unnecessarily. Since honest inputs are binary, for codes with shallow addition depth, such as RAA codes [BCFRRZ25] or EA codes [Boy+22; BFKTWZ24; LXLYH26], the intermediate and output values remain much smaller than the modulus of $\mathbb{Z}_{2^\ell}$. We can therefore represent the inputs over a smaller ring, such as $\mathbb{Z}_{2^k}$, perform the encoding entirely over $\mathbb{Z}_{2^k}$, and lift the resulting values to $\mathbb{Z}_{2^\ell}$ only when necessary. The column blocks can likewise be hashed in this compact representation. Moreover, the γ column blocks opened during an evaluation can be transmitted as elements of the smaller ring, further reducing the proof size.

7.4.2 Boolean Check with Lookup Tables

Recall that during the commitment phase of Protocol 4, the prover and verifier engage in an IOP to check that the committed vector is binary. Due to the bit decomposition, however, this vector is k times longer than the original vector, so directly applying the IOP to the decomposed vector would incur substantial overhead. In this part, we describe a more efficient approach for performing the Boolean check on a vector $v \in \mathbb{Z}_{2^k}^{k \cdot T}$.

Specifically, the prover and verifier need to verify that $v \circ (1 - v) = 0^{k \cdot T}$. The key idea is to employ the higher-arity folding technique from Remark 1 in the first round of the reduction, while exploiting the binary structure of the underlying vector to accelerate this step.

Let $\nu = \log(k \cdot T)$. As in Protocol 1, the verifier first samples $(\theta_i^{(0)}, \theta_i^{(1)}) \leftarrow \mathbb{Z}_{2^m}^2$ for each $i \in [\nu]$ and sends them to the prover. Both parties then set $\theta := \bigotimes_{i=0}^{\nu-1} (\theta_i^{(0)}, \theta_i^{(1)})$. The resulting batched instance consists of the vectors $x := \theta \circ v$ and $y := 1 - v$, together with the claimed value $z_0 := 0$. Let $B = 2^b$ for some $b \leq \nu$. We decompose θ as $\theta = \vartheta \otimes \theta'$ with $\vartheta := \bigotimes_{i=0}^{b-1} (\theta_i^{(0)}, \theta_i^{(1)}) \in \mathbb{Z}_{2^m}^B$ and $\theta' := \bigotimes_{i=b}^{\nu-1} (\theta_i^{(0)}, \theta_i^{(1)}) \in \mathbb{Z}_{2^m}^{kT/B}$.

In the first round, we reduce the vector length by a factor of B . Specifically, we partition v into B contiguous blocks $v^{(0)}, \dots, v^{(B-1)} \in \mathbb{Z}_{2^m}^{kT/B}$. Accordingly, the vectors $x = \theta \circ v$ and $y = 1 - v$ decompose into blocks

$$x^{(j)} := \vartheta_j (\theta' \circ v^{(j)}), \quad y^{(j)} := 1 - v^{(j)}, \quad j \in [B].$$

To derive the claim for the reduced instance, the prover should compute the B^2 cross-term claims $z_0^{(j,j')} := x^{(j)} \cdot y^{(j')}$. Our first optimization is that the B diagonal terms need not be computed. Indeed, if v is binary, then each block $v^{(j)}$ is binary, and hence $x^{(j)} \cdot y^{(j)} = \vartheta_j \langle \theta', (v^{(j)} \circ (1 - v^{(j)})) \rangle = 0$. Thus, the verifier directly sets all diagonal claims to zero and the prover computes only the off-diagonal cross terms. It remains to show how the prover can efficiently compute the $B(B-1)$ off-diagonal claims.

By definition, for every $j \neq j'$, $x^{(j)} \cdot y^{(j')} = \vartheta_j \langle \theta', v^{(j)} \circ (1 - v^{(j')}) \rangle$. Since v is binary, the vector $v^{(j)} \circ (1 - v^{(j')})$ is also binary, with its i -th entry equal to 1 precisely when $v_i^{(j)} = 1$ and $v_i^{(j')} = 0$. Thus,

up to the scalar ϑ_j , each off-diagonal claim is a subset sum of the entries of θ'

$$x^{(j)} \cdot y^{(j')} = \vartheta_j \sum_{i: v_i^{(j)}=1, v_i^{(j')}=0} \theta'_i.$$

Importantly, the subset appearing in this sum is determined solely by the bits of v .

To exploit this structure, observe that whether an index i contributes to a particular claim depends only on the set of blocks whose i -th entries are equal to 1. For each $i \in [kT/B]$, define its *pattern* $P_i := \{j \in [B] : v_i^{(j)} = 1\}$. The condition $v_i^{(j)} = 1$ and $v_i^{(j')} = 0$ is then equivalent to $j \in P_i$ and $j' \notin P_i$. Grouping indices according to their patterns gives

$$\begin{aligned} x^{(j)} \cdot y^{(j')} &= \vartheta_j \sum_{i: j \in P_i, j' \notin P_i} \theta'_i \\ &= \vartheta_j \sum_{\substack{P \subseteq [B] \\ j \in P, j' \notin P}} \sum_{i: P_i = P} \theta'_i \\ &= \vartheta_j \sum_{\substack{P \subseteq [B] \\ j \in P, j' \notin P}} A[P], \end{aligned}$$

where $A[P] := \sum_{i: P_i = P} \theta'_i$, for every $P \subseteq [B]$. Hence, all $B(B-1)$ off-diagonal claims are determined by the table A of 2^B ring elements. This table can be constructed in a single pass over the kT/B indices, using essentially one ring addition per index. For fixed $j \neq j'$, exactly 2^{B-2} patterns contain j but not j' . Thus, computing each claim directly from A requires summing 2^{B-2} table entries, for a total of $B(B-1)2^{B-2}$ ring additions across all off-diagonal claims, independent of ν apart from the cost of constructing A .

This cost can be reduced further using a superset-sum transform. Define $S[Q] := \sum_{P \supseteq Q} A[P]$, $Q \subseteq [B]$. The entire table S can be computed in place from A using the standard dynamic programming, which requires B passes of 2^{B-1} ring additions, for a total of $B2^{B-1}$ additions. Moreover,

$$\sum_{\substack{P \subseteq [B] \\ j \in P, j' \notin P}} A[P] = S[\{j\}] - S[\{j, j'\}],$$

because the patterns containing j but not j' are precisely those containing j after excluding those that also contain j' . Consequently,

$$z_0^{(j,j')} = \vartheta_j (S[\{j\}] - S[\{j, j'\}]).$$

Once S has been computed, each off-diagonal claim therefore requires only one subtraction, together with the multiplication by ϑ_j . In particular, the $B(B-1)2^{B-2}$ additions required by the direct approach are replaced by $B2^{B-1}$ additions for the transform, plus $B(B-1)$ subtractions for extracting all claims.

The verifier then samples $(\alpha_i^{(0)}, \alpha_i^{(1)}), (\beta_i^{(0)}, \beta_i^{(1)}) \leftarrow \mathbb{Z}_{2^m}^2$ for each $i \in [b]$. Both parties set $\alpha := \bigotimes_{i=0}^{b-1} (\alpha_i^{(0)}, \alpha_i^{(1)})$ and $\beta := \bigotimes_{i=0}^{b-1} (\beta_i^{(0)}, \beta_i^{(1)})$, and fold the blocks as

$$x \leftarrow \sum_{j=0}^{B-1} \alpha_j x^{(j)}, \quad y \leftarrow \sum_{j=0}^{B-1} \beta_j y^{(j)}.$$

Computing these two folded vectors naively would require B ring multiplications per coordinate. The pattern representation introduced above allows these computations to be reduced to table lookups.

Indeed, substituting the definition of $x^{(j)}$ into the i -th coordinate of the folded vector x gives

$$x_i = \sum_{j=0}^{B-1} \alpha_j x_i^{(j)} = \sum_{j=0}^{B-1} \alpha_j \vartheta_j \theta'_i v_i^{(j)} = \theta'_i \sum_{j \in P_i} \alpha_j \vartheta_j,$$

where the last equality follows from the fact that $v_i^{(j)} = 1$ exactly when $j \in P_i$. Thus, the common factor θ'_i can be pulled out, and the remaining quantity depends on i only through its pattern P_i . Similarly,

$$y_i = \sum_{j=0}^{B-1} \beta_j y_i^{(j)} = \sum_{j=0}^{B-1} \beta_j (1 - v_i^{(j)}) = \sum_{j=0}^{B-1} \beta_j - \sum_{j \in P_i} \beta_j.$$

Hence, apart from the factor θ'_i in x_i , both folded coordinates depend on i only through the pattern P_i .

Once α and β are known, the prover computes the tables

$$X[P] := \sum_{j \in P} \alpha_j \vartheta_j \quad \text{and} \quad Y[P] := \sum_{j=0}^{B-1} \beta_j - \sum_{j \in P} \beta_j \quad (P \subseteq [B]),$$

and then obtains the folded vectors as

$$x_i = \theta'_i X[P_i], \quad y_i = Y[P_i] \quad (i \in [kT/B]),$$

with one lookup and one multiplication per entry of x and one lookup per entry of y .

After the first folding step, the vectors x and y are no longer Boolean, so the protocol proceeds exactly as in Protocol 1 on vectors of length kT/B . This incurs an additional prover overhead of $O(kT/B)$.

Cost. Using the algorithm described here, the prover overhead is roughly dominated by $kT/B + B(B-1)2^{B-2}$ ring additions and $O(kT/B)$ ring multiplications for the remaining folds.

7.4.3 Row Combinations from Lookup Tables

The row combinations for the ring evaluations in Protocol 4 can be accelerated using a similar idea. For each $a \in [k]$, the prover computes $y^{(a)} = \tilde{L} \cdot (v_i^{(a)})_{i=0}^{r_0-1} \in \mathbb{Z}_{2^\ell}^{r_1}$ for all $a \in [k]$, which naively requires kT multiplications in $\mathbb{Z}_{2^\ell}$. Since the entries of $v^{(a)}$ are bits, each coordinate of $y^{(a)}$ is simply a subset sum of the entries of $\tilde{L}$: $y_j^{(a)} = \sum_{i: v_{i,j}^{(a)}=1} \tilde{L}_i$. Moreover, the corresponding subset is determined entirely by the r_0 bits in the j -th column. To exploit this structure, the prover partitions the r_0 rows into groups G of B consecutive rows. For each group G , it precomputes the subset-sum table $T_G[P] := \sum_{i \in P} \tilde{L}_i$ for all $P \subseteq G$. Each table contains 2^B entries and can be constructed using $O(2^B)$ additions in $\mathbb{Z}_{2^\ell}$. For every $a \in [k]$, group G , and column j , define the pattern $P_{G,j}^{(a)} := \{i \in G : v_{i,j}^{(a)} = 1\}$. This pattern can be read directly from the bits of the corresponding column. The prover can then assemble each coordinate as $y_j^{(a)} = \sum_G T_G[P_{G,j}^{(a)}]$.

There are r_0/B groups, so constructing all subset-sum tables requires $O\left(\frac{2^B r_0}{B}\right)$ additions in $\mathbb{Z}_{2^\ell}$. Combining the table entries across groups for all a and j requires $O\left(\frac{kT}{B}\right)$ additional additions. Importantly, this procedure requires no multiplications in $\mathbb{Z}_{2^\ell}$, replacing the naive cost of kT ring multiplications by $O\left(\frac{2^B r_0}{B} + \frac{kT}{B}\right)$ ring additions.

PROTOCOL 5. IOP for the Plonkish Relation

Let $\text{gp} = (k, T, n_p, n_w, n_q, g)$ with $T = 2^t$, $n_p = 2^{t_p}$, $n_w = 2^{t_w}$ and $n_q = 2^{t_q}$, and let Π_{perm} be an IOP for $\mathcal{R}_{\text{perm}}^{\text{F},k}$ with field oracles on vectors of length $n_w T$. $\mathcal{P}$ holds $((q, \sigma), (p, w); w)$, and the oracles hold the canonical lifts $\hat{q}_j, \hat{w}_j \in \mathbb{Z}_{2^m}^T$ of q_j and w_j . $\mathcal{V}$ holds σ and p , has ring-query access to the ring oracles $\llbracket q_0 \rrbracket_{\mathbb{R}}, \dots, \llbracket q_{n_q-1} \rrbracket_{\mathbb{R}}$, and has ring- and field-query access to the compatible oracles $\llbracket w_0 \rrbracket_{\mathbb{C}}, \dots, \llbracket w_{n_w-1} \rrbracket_{\mathbb{C}}$. Let $\hat{p} \in \mathbb{Z}_{2^m}^{n_p}$ be the canonical lift of p , and let $n := t - t_p$.

1. *Gate identity.* $\mathcal{P}$ computes $h \in \mathbb{Z}_{2^m}^T$ with $\hat{g}(\hat{q}_0, \dots, \hat{q}_{n_q-1}, \hat{w}_0, \dots, \hat{w}_{n_w-1}) = 2^k h$ and sends $\llbracket h \rrbracket_{\mathbb{R}}$. $\mathcal{P}$ and $\mathcal{V}$ then run Protocol 1 on the lifted instance $(\hat{g}, ((\hat{q}_0, \dots, \hat{q}_{n_q-1}, \hat{w}_0, \dots, \hat{w}_{n_w-1}), h))$, where the tensor queries to $\hat{q}_j, \hat{w}_j$ and h are the ring queries to $\llbracket q_j \rrbracket_{\mathbb{R}}, \llbracket w_j \rrbracket_{\mathbb{C}}$ and $\llbracket h \rrbracket_{\mathbb{R}}$.
2. *Wiring identity.* $\mathcal{P}$ and $\mathcal{V}$ run Π_{perm} on the instance $(\sigma, (\llbracket \varphi(w) \rrbracket, \llbracket \varphi(w) \rrbracket))$, where a field query to $\llbracket \varphi(w) \rrbracket$ at $\eta' = (\vartheta, \theta)$ with $\vartheta \in \mathbb{F}_{2^\mu}^{2t_w}$ and $\theta \in \mathbb{F}_{2^\mu}^{2t}$ is answered by $\sum_{j \in [n_w]} \text{Tensor}(\vartheta)_j \cdot a_j$, where a_j is the answer of the field query to $\llbracket w_j \rrbracket_{\mathbb{C}}$ at θ .

3. *Public input.* $\mathcal{V}$ samples $\theta \leftarrow \mathbb{Z}_{2^m}^{2t_p}$, queries $\llbracket w_0 \rrbracket_{\mathbb{C}}$ at $\langle 0 \rangle_n \parallel \theta$, obtaining a , and checks $a \stackrel{?}{=} \text{Tensor}(\theta) \cdot \hat{p}$.

$\mathcal{V}$ accepts if all three checks accept.

7.4.4 Field Evaluations over $\mathbb{F}_{2^k}$

In a field evaluation, Protocol 4 processes the k binary components separately, with the prover sending one row combination $y^{(a)} \in \mathbb{F}_{2^\mu}^{r_1}$ per component and the verifier testing every opened column block against every $y^{(a)}$ and recombining the results as $s' = \sum_{a=0}^{k-1} \zeta_a y^{(a)} \cdot R'$. The computation can instead be carried out directly over $\mathbb{F}_{2^k}$ and $\mathbb{F}_{2^\mu}$, which eliminates the apparent factor- k overhead. Let $w_i := \sum_{a=0}^{k-1} \zeta_a \pi_1(v_i^{(a)}) \in \mathbb{F}_{2^k}^{r_1}$ be the rows of $\varphi(\pi_k(v))$, viewed as an $r_0 \times r_1$ matrix over $\mathbb{F}_{2^k}$. Since Enc_1 is $\mathbb{F}_2$ -linear, it extends linearly to vectors over $\mathbb{F}_{2^k}$ and $\mathbb{F}_{2^\mu}$ and commutes with the recombination, that is, $\sum_a \zeta_a \text{Enc}_1(y^{(a)}) = \text{Enc}_1(\sum_a \zeta_a y^{(a)})$. The prover therefore sends the single row combination $y' := \sum_{a=0}^{k-1} \zeta_a y^{(a)} = L' \cdot (w_i)_{i=0}^{r_0-1} \in \mathbb{F}_{2^\mu}^{r_1}$, and the verifier recombines every opened column block into the $\mathbb{F}_{2^k}$ elements $w_{i,j} := \sum_{a=0}^{k-1} \zeta_a \pi_1(u_{i,j}^{(a)})$ and checks $L' \cdot (w_{i,j})_{i=0}^{r_0-1} \stackrel{?}{=} \text{Enc}_1(y')_j$ for $j \in J$ and $s' \stackrel{?}{=} y' \cdot R'$. This is the evaluation protocol of the small-field polynomial commitment of [DP25] for the matrix $(w_i)_{i=0}^{r_0-1}$ over $\mathbb{F}_{2^k}$, whose rows are encoded by the extension of the binary code to $\mathbb{F}_{2^k}$, and the algorithm $\Pi.\text{FOpen}$ is adjusted to this matrix accordingly. The prover computes one row combination over $\mathbb{F}_{2^\mu}$ of an $r_0 \times r_1$ matrix over $\mathbb{F}_{2^k}$ instead of k row combinations of binary matrices, and the verifier performs one encoding and γ column checks over $\mathbb{F}_{2^\mu}$ instead of k of each.

8 Rotor: A SNARK for Power-of-Two Rings

We now present Rotor, a SNARK for general circuits over $\mathbb{Z}_{2^k}$. At its core, Rotor adapts the Plonkish constraint system [CBBZ23] to the setting of $\mathbb{Z}_{2^k}$.

8.1 Plonkish Relations over $\mathbb{Z}_{2^k}$

We define the indexed relation $\mathcal{R}_{\text{plonk}}^k$ over $\mathbb{Z}_{2^k}$.

Public parameters. Fix $\text{gp} := (k, T, n_p, n_w, n_q, g)$, where $T = 2^t$ is the number of constraints, $n_p = 2^{t_p} \leq T$ is the length of the public input, $n_w = 2^{t_w}$ and $n_q = 2^{t_q}$ are the numbers of wires and selectors per constraint, and g is an $(n_q + n_w)$ -variable polynomial over $\mathbb{Z}_{2^k}$ of total degree $d \geq 2$. A selector assignment is a vector $q \in \mathbb{Z}_{2^k}^{n_q T}$, written as the concatenation $q = (q_0 \parallel \dots \parallel q_{n_q-1})$ of n_q vectors $q_j \in \mathbb{Z}_{2^k}^T$, and a wire assignment is a vector $w \in \mathbb{Z}_{2^k}^{n_w T}$, written as the concatenation $w = (w_0 \parallel \dots \parallel w_{n_w-1})$ of n_w vectors $w_j \in \mathbb{Z}_{2^k}^T$. $\llbracket q \rrbracket$ and $\llbracket w \rrbracket$ denote the lists of oracles $(\llbracket q_0 \rrbracket, \dots, \llbracket q_{n_q-1} \rrbracket)$ and $(\llbracket w_0 \rrbracket, \dots, \llbracket w_{n_w-1} \rrbracket)$.

Definition 20. The Plonkish relation $\mathcal{R}_{\text{plonk}}^k$ is the set of all tuples $(i, x; w) = ((\llbracket q \rrbracket, \sigma), (p, \llbracket w \rrbracket); w)$, where $q \in \mathbb{Z}_{2^k}^{n_q T}$, σ is a permutation of $[n_w T]$, $p \in \mathbb{Z}_{2^k}^{n_p}$ and $w \in \mathbb{Z}_{2^k}^{n_w T}$, such that

- the gate identity holds, i.e., $g(q_0, \dots, q_{n_q-1}, w_0, \dots, w_{n_w-1}) = 0^T$, or equivalently $(g, (\llbracket q_0 \rrbracket, \dots, \llbracket w_{n_w-1} \rrbracket); (q, w)) \in \mathcal{R}_{\text{arith}}^k$;
- the wiring identity holds, i.e., $(\sigma, (\llbracket w \rrbracket, \llbracket w \rrbracket); (w, w)) \in \mathcal{R}_{\text{perm}}^{R,k}$;
- the public input is consistent with the witness, i.e., $w_{0,i} = p_i$ for every $i \in [n_p]$.

8.2 An IOP for the Plonkish Relation

We now give an IOP for $\mathcal{R}_{\text{plonk}}^k$ with ring oracles and compatible oracles.

Oracle choices. We first specify the oracle type used for each vector. For clarity, we use $\llbracket \cdot \rrbracket_R$ and $\llbracket \cdot \rrbracket_C$ to denote ring oracles and compatible oracles, respectively. Since compatible oracles are needed only for the permutation check, we encode the witness vectors as compatible oracles and set $\llbracket w \rrbracket_C = (\llbracket w_0 \rrbracket_C, \dots, \llbracket w_{n_w-1} \rrbracket_C)$. The remaining vectors are used only in the gate check and are therefore encoded as ring oracles and set $\llbracket q \rrbracket_R = (\llbracket q_0 \rrbracket_R, \dots, \llbracket q_{n_q-1} \rrbracket_R)$.

Public input. To ensure the consistency of committed witness with public input $w_p := (w_{0,0}, \dots, w_{0,n_p-1})$ of the first wire to equal p . The claim is checked by evaluating both sides at a random query point. For $\theta \in \mathbb{Z}_{2^m}^{2t_p}$ and $n := t - t_p$, let $\langle 0 \rangle_n \parallel \theta$ denote the point that prepends n pairs $(1, 0)$ to θ . Therefore, $\text{Tensor}(\langle 0 \rangle_n \parallel \theta) \cdot \hat{w}_0 = \text{Tensor}(\theta) \cdot \hat{w}_p$.

We give a full description of our construction in Protocol 5.

Theorem 16. Let Π_{perm} be a perfectly complete IOP for $\mathcal{R}_{\text{perm}}^{F,k}$ with field oracles on vectors of length $n_w T$, with soundness error ϵ_{perm} . Then Protocol 5 is an IOP for $\mathcal{R}_{\text{plonk}}^k$ that is perfectly complete and has soundness error $\epsilon_{\text{plonk}} := \max\{\Phi(m - k + 1, (d + 1)t), \epsilon_{\text{perm}}, \Phi(m - k + 1, t_p)\}$.

Proof. Completeness. Let $((\llbracket q \rrbracket_R, \sigma), (p, \llbracket w \rrbracket_C); w) \in \mathcal{R}_{\text{plonk}}^k$, and let $\hat{q}_j$ and $\hat{w}_j$ be the canonical lifts held by the oracles. Since $g(q_0, \dots, w_{n_w-1}) = 0^T$, the honest prover finds h with $\hat{g}(\hat{q}_0, \dots, \hat{w}_{n_w-1}) = 2^k h$, as in the proof of Theorem 1, and Protocol 1 accepts by Theorem 2. Since $w = \sigma(w)$ and φ is a bijection, $\varphi(w) = \sigma(\varphi(w))$, and Π_{perm} accepts by its perfect completeness, as every field query to $\varphi(w)$ at a point η' is answered with $\text{Tensor}(\eta') \cdot \varphi(w)$ by the identity above. Finally, since $w_p = p$ and both lifts are canonical, $\hat{w}_p = \hat{p}$, so the answer $a = \text{Tensor}(\theta) \cdot \hat{w}_p$ of the prefix query equals $\text{Tensor}(\theta) \cdot \hat{p}$ and the third check accepts.

Soundness. Let $((\llbracket q \rrbracket_R, \sigma), (p, \llbracket w \rrbracket_C)) \notin \mathcal{L}(\mathcal{R}_{\text{plonk}}^k)$, with arbitrary lifts $\hat{q}_j$ and $\hat{w}_j$ held by the oracles, and put $w_j := \pi_k(\hat{w}_j)$. Since the only candidate witness is w , at least one of the three conditions of Definition 20 fails for w . The challenges of the three checks are independent, so for every fixed choice of the challenges of two of the checks, a prover strategy for Protocol 5 induces a prover strategy for the third; the soundness bound of each check therefore applies within the whole protocol, and it suffices to bound the acceptance probability of the check corresponding to a failing condition.

If the gate identity fails, then $(g, (\llbracket q_0 \rrbracket, \dots, \llbracket w_{n_w-1} \rrbracket)) \notin \mathcal{L}(\mathcal{R}_{\text{arith}}^k)$, and the oracles $\llbracket q_j \rrbracket_R$ and $\llbracket w_j \rrbracket_C$ hold lifts of q_j and w_j and answer ring queries as the oracles of Definition 1 do. The first check runs the IOP Π_{arith} on this instance, whose soundness error is $\Phi(m - k + 1, (d + 1)t)$ for every choice of the lifts by Theorems 1 and 2. Hence the first check accepts with at most this probability.

If the wiring identity fails, then $w \neq \sigma(w)$, hence $\varphi(w) \neq \sigma(\varphi(w))$ and $(\sigma, (\llbracket \varphi(w) \rrbracket, \llbracket \varphi(w) \rrbracket)) \notin \mathcal{L}(\mathcal{R}_{\text{perm}}^{F,k})$. Every field query of Π_{perm} to $\varphi(w)$ is answered with $\text{Tensor}(\eta') \cdot \varphi(w)$ by the identity above, so the view of the verifier of Π_{perm} in the second check is identical to its view on this field instance, and the second check accepts with probability at most ϵ_{perm} .

If the public input is inconsistent, then $w_p \neq p$. Let $\Xi := \hat{w}_p - \hat{p} \in \mathbb{Z}_{2^m}^{n_p}$, which is determined by the instance. Then $\pi_k(\Xi) = w_p - p \neq 0$, so $\text{Po2}_m(\Xi) < k$. The challenge θ is sampled independently of Ξ , and $\text{Tensor}(\theta)$ is the tensor of t_p independent uniform pairs, so Lemma 3 gives $\Pr[\text{Tensor}(\theta) \cdot \Xi = 0] \leq \Phi(m - \text{Po2}_m(\Xi), t_p) \leq \Phi(m - k + 1, t_p)$, where the second inequality holds because $\Phi(S, t_p)$ is nonincreasing in S . Since the prefix query is answered by the oracle, $a - \text{Tensor}(\theta) \cdot \hat{p} = \text{Tensor}(\theta) \cdot \Xi$, so the third check accepts with at most this probability. $\square$

Complexity. Let M be the number of monomials in g . In the first check, the prover and verifier perform $O(d2^d MT)$ and $O(2^d \log T + Md)$ $\mathbb{Z}_{2^m}$ operations, respectively; the prover sends $2^d \log T$ $\mathbb{Z}_{2^m}$ elements over $\log T$ rounds, and the verifier makes at most $d(n_q + n_w + 1)$ ring queries. The second check inherits the complexity of Π_{perm} on vectors of length $n_w T$, with each field query to Π_{perm} requiring n_w field queries. The third check requires $O(n_p)$ $\mathbb{Z}_{2^m}$ operations for the verifier, no prover communication, and one ring query. By Lemma 4, choosing $m - k = \lambda + O(\log T \log \lambda)$ yields soundness error at most $2^{-\lambda}$.

Remark 6 (Compatibility with binary-field SNARKs). *Rotor admits a direct integration with binary-field SNARKs [DP25; Hu+25; BRW26; Set26], thereby enabling efficient proofs for arithmetic operations expressed directly over $\mathbb{F}_{2^k}$, such as XOR and AND. Specifically, given an IOP for a relation over $\mathbb{F}_{2^k}$ that uses field oracles, we can simply replace each field oracle with the corresponding compatible oracle and answer the resulting field queries directly using these compatible oracles.*

AI Disclosure

The main results of this work were developed primarily by the authors through human trial and errors. However, we also engaged in extensive discussions with GPT-5.5 Pro to validate the results, simplify some of the arguments, and assist in drafting parts of the proofs. GPT-5.6 Sol was used for coding and benchmarking.

The authors take full responsibility for the accuracy, integrity, and content of the manuscript, including all material developed with AI assistance.

References

- [a1626] a16z crypto. *Jolt: The Simplest and Most Extensible zkVM*. <https://jolt.a16zcrypto.com/>. Project website. 2026.

- [Abd+26] A. Abdugafarov, A. Garreta, A. Kumar, M. Osadnik, P. Vesely, I. Vlasov, and K. Z. Zheng. *Zinc+: SNARKs for Polynomial Rings*. Cryptology ePrint Archive, Paper 2026/855. 2026. URL: <https://eprint.iacr.org/2026/855>.
- [AER24] G. Angeris, A. Evans, and G. Roh. “A Note on Liger and Logarithmic Randomness”. In: *IACR Cryptol. ePrint Arch.* 2024 (2024), p. 1399. URL: <https://eprint.iacr.org/2024/1399>.
- [AHIV17] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. “Liger: Lightweight Sublinear Arguments Without a Trusted Setup”. In: *Proceedings of the 24th ACM Conference on Computer and Communications Security*. CCS ’17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 2087–2104.
- [AST24] A. Arun, S. T. V. Setty, and J. Thaler. “Jolt: SNARKs for Virtual Machines via Lookups”. In: *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI*. Ed. by M. Joye and G. Leander. Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 3–33. DOI: [10.1007/978-3-031-58751-1_1](https://doi.org/10.1007/978-3-031-58751-1_1). URL: https://doi.org/10.1007/978-3-031-58751-1_1.
- [BBBPWM18] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. “Bulletproofs: Short Proofs for Confidential Transactions and More”. In: *Proceedings of the 39th IEEE Symposium on Security and Privacy*. S&P ’18. 2018, pp. 315–334.
- [BBMHS22] C. Baum, L. Braun, A. Munch-Hansen, and P. Scholl. “Moz $\mathbb{Z}_2^k$ arella: Efficient Vector-OLE and Zero-Knowledge Proofs over $\mathbb{Z}_2^k$ ”. In: *Advances in Cryptology – CRYPTO 2022: 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15–18, 2022, Proceedings, Part IV*. Santa Barbara, CA, USA: Springer-Verlag, 2022, 329–358. ISBN: 978-3-031-15984-8. DOI: [10.1007/978-3-031-15985-5_12](https://doi.org/10.1007/978-3-031-15985-5_12). URL: https://doi.org/10.1007/978-3-031-15985-5_12.
- [BBMRS21] C. Baum, L. Braun, A. Munch-Hansen, B. Razet, and P. Scholl. “Appenzeller to Brie: Efficient Zero-Knowledge Proofs for Mixed-Mode Arithmetic and $\mathbb{Z}_2^k$ ”. In: *CCS ’21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*. Ed. by Y. Kim, J. Kim, G. Vigna, and E. Shi. ACM, 2021, pp. 192–211. DOI: [10.1145/3460120.3484812](https://doi.org/10.1145/3460120.3484812). URL: <https://doi.org/10.1145/3460120.3484812>.
- [BCD25] B. Bünz, J. Chen, and Z. DeStefano. “Linear*-Time Permutation Check”. In: *IACR Cryptol. ePrint Arch.* 2025 (2025), p. 1850. URL: <https://eprint.iacr.org/2025/1850>.
- [BCFRRZ25] M. Brehm, B. Chen, B. Fisch, N. Resch, R. D. Rothblum, and H. Zeilberger. “Blaze: Fast SNARKs from Interleaved RAA Codes”. In: *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part IV*. Ed. by S. Fehr and P. Fouque. Vol. 15604. Lecture Notes in Computer Science. Springer, 2025, pp. 123–152. DOI: [10.1007/978-3-031-91134-7_5](https://doi.org/10.1007/978-3-031-91134-7_5). URL: https://doi.org/10.1007/978-3-031-91134-7_5.
- [BCS16] E. Ben-Sasson, A. Chiesa, and N. Spooner. “Interactive Oracle Proofs”. In: *Proceedings of the 14th Theory of Cryptography Conference*. TCC ’16-B. 2016, pp. 31–60.
- [BFKRWZ24] A. R. Block, Z. Fang, J. Katz, J. Thaler, H. Waldner, and Y. Zhang. “Field-Agnostic SNARKs from Expand-Accumulate Codes”. In: *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, Part X*. Ed. by L. Reyzin and D. Stebila. Vol. 14929. Lecture Notes in Computer Science. Springer, 2024, pp. 276–307. DOI: [10.1007/978-3-031-68403-6_9](https://doi.org/10.1007/978-3-031-68403-6_9). URL: https://doi.org/10.1007/978-3-031-68403-6_9.
- [BFS20] B. Bünz, B. Fisch, and A. Szeponiec. “Transparent SNARKs from DARK Compilers”. In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’20. 2020, pp. 677–706.

- [Boy+22] E. Boyle, G. Couteau, N. Gilboa, Y. Ishai, L. Kohl, N. Resch, and P. Scholl. “Correlated Pseudo-randomness from Expand-Accumulate Codes”. In: *Proceedings of the 42nd Annual International Cryptology Conference*. CRYPTO ’22. 2022, pp. 603–633.
- [Bre26] Brevis Network. *Pico: A Modular and Performant zkVM*. <https://pico-docs.brevis.network/>. Project website. 2026.
- [BRW26] B. Bünz, R. Rothblum, and W. Wang. *Flock: Fast Proving for Batch Boolean Computations*. Cryptology ePrint Archive, Paper 2026/1329. 2026. URL: <https://eprint.iacr.org/2026/1329>.
- [BSGJOST23] L. Braun, C. D. de Saint Guilhem, R. Jadoul, E. Orsini, N. P. Smart, and T. Tanguy. “ZK-for-Z2K: MPC-in-the-Head Zero-Knowledge Proofs for $\mathbb{Z}_{2^k}$ ”. In: *Cryptography and Coding - 19th IMA International Conference, IMACC 2023, London, UK, December 12-14, 2023, Proceedings*. Ed. by E. A. Quaglia. Vol. 14421. Lecture Notes in Computer Science. Springer, 2023, pp. 137–157. DOI: [10.1007/978-3-031-47818-5_8](https://doi.org/10.1007/978-3-031-47818-5_8). URL: https://doi.org/10.1007/978-3-031-47818-5_8.
- [CBBZ23] B. Chen, B. Bünz, D. Boneh, and Z. Zhang. “HyperPlonk: Plonk with Linear-Time Prover and High-Degree Custom Gates”. In: *Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’23. Cham: Springer Nature Switzerland, 2023, pp. 499–530.
- [CDESX18] R. Cramer, I. Damgård, D. Escudero, P. Scholl, and C. Xing. “SPD $\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority”. In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*. Ed. by H. Shacham and A. Boldyreva. Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 769–798. DOI: [10.1007/978-3-319-96881-0_26](https://doi.org/10.1007/978-3-319-96881-0_26). URL: https://doi.org/10.1007/978-3-319-96881-0_26.
- [CHA26] M. Campanelli and M. Hall-Andersen. “General Techniques for Building SNARKs over the Integers”. In: *Public-Key Cryptography – PKC 2026: 29th IACR International Conference on Practice and Theory of Public-Key Cryptography, West Palm Beach, FL, USA, May 25–28, 2026, Proceedings, Part III*. West Palm Beach, FL, USA: Springer-Verlag, 2026, 197–230. ISBN: 978-3-032-26736-8. DOI: [10.1007/978-3-032-26737-5_7](https://doi.org/10.1007/978-3-032-26737-5_7). URL: https://doi.org/10.1007/978-3-032-26737-5_7.
- [CHMMVW20] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, and N. Ward. “Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS”. In: *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’20. 2020.
- [DP23] B. E. Diamond and J. Posen. “Proximity testing with logarithmic randomness”. In: *Cryptology ePrint Archive* (2023).
- [DP25] B. E. Diamond and J. Posen. “Succinct Arguments over Towers of Binary Fields”. In: *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part IV*. Ed. by S. Fehr and P. Fouque. Vol. 15604. Lecture Notes in Computer Science. Springer, 2025, pp. 93–122. DOI: [10.1007/978-3-031-91134-7_4](https://doi.org/10.1007/978-3-031-91134-7_4). URL: https://doi.org/10.1007/978-3-031-91134-7_4.
- [DP26] B. E. Diamond and J. Posen. “Polylogarithmic Proofs for Multilinears over Binary Towers”. In: *Proceedings of the 45th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT ’26. 2026, pp. 3–32. DOI: [10.1007/978-3-032-25336-1_1](https://doi.org/10.1007/978-3-032-25336-1_1).

- [EHLXY23] D. Escudero, C. Hong, H. Liu, C. Xing, and C. Yuan. “Degree-D Reverse Multiplication-Friendly Embeddings: Constructions and Applications”. In: *Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part I*. Ed. by J. Guo and R. Steinfeld. Vol. 14438. Lecture Notes in Computer Science. Springer, 2023, pp. 106–138. DOI: [10.1007/978-981-99-8721-4_4](https://doi.org/10.1007/978-981-99-8721-4_4). URL: https://doi.org/10.1007/978-981-99-8721-4_4.
- [Gar+25] A. Garreta, H. Waldner, I. Vlasov, K. Hristova, L. Dall’Ava, M. Čupić, and M. Klein. “Zinc: Succinct Arguments with Small Arithmetization Overheads from IOPs of Proximity to the Integers”. In: *Advances in Cryptology – CRYPTO 2025: 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–21, 2025, Proceedings, Part VII*. Santa Barbara, CA, USA: Springer-Verlag, 2025, 259–291. ISBN: 978-3-032-01906-6. DOI: [10.1007/978-3-032-01907-3_9](https://doi.org/10.1007/978-3-032-01907-3_9). URL: https://doi.org/10.1007/978-3-032-01907-3_9.
- [GLSTW23] A. Golovnev, J. Lee, S. T. V. Setty, J. Thaler, and R. S. Wahby. “Brakedown: Linear-Time and Field-Agnostic SNARKs for R1CS”. In: *Proceedings of the 43rd Annual International Cryptology Conference*. CRYPTO ’23. Cham: Springer Nature Switzerland, 2023, pp. 193–226.
- [GNSV23] C. Ganesh, A. Nitulescu, and E. Soria-Vazquez. “Rinocchio: SNARKs for Ring Arithmetic”. In: *J. Cryptol.* 36.4 (Oct. 2023). ISSN: 0933-2790. DOI: [10.1007/s00145-023-09481-3](https://doi.org/10.1007/s00145-023-09481-3). URL: <https://doi.org/10.1007/s00145-023-09481-3>.
- [Gro16] J. Groth. “On the Size of Pairing-Based Non-interactive Arguments”. In: *Proceedings of the 35th Annual International Conference on Theory and Applications of Cryptographic Techniques*. EURO-CRYPT ’16. 2016, pp. 305–326.
- [GSV21] Y. Gvili, S. Scheffler, and M. Varia. “Booligero: Improved sublinear zero knowledge proofs for Boolean circuits”. In: *Financial Cryptography and Data Security: 25th International Conference, FC 2021, Virtual Event, March 1–5, 2021, Revised Selected Papers, Part I* 25. Springer. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 476–496.
- [GW20] A. Gabizon and Z. J. Williamson. “plookup: A simplified polynomial protocol for lookup tables”. In: *IACR Cryptol. ePrint Arch.* (2020).
- [GWC19] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. *PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge*. ePrint 2019/953. 2019.
- [GWHW24] R. Gao, Z. Wan, Y. Hu, and H. Wang. “A Succinct Range Proof for Polynomial-based Vector Commitment”. In: *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’24. 2024, pp. 3152–3166. DOI: [10.1145/3658644.3670324](https://doi.org/10.1145/3658644.3670324).
- [Hab22] U. Haböck. “Multivariate lookups based on logarithmic derivatives”. In: *IACR Cryptol. ePrint Arch.* (2022).
- [Hu+25] Y. Hu, C. Li, Z. Qiu, T. Xie, Y. Ying, J. Zhang, and Z. Zhang. *GKR for Boolean Circuits with Sub-linear RAM Operations*. Cryptology ePrint Archive, Paper 2025/717. 2025. URL: <https://eprint.iacr.org/2025/717>.
- [Irr25] Irreducible. *Integer Multiplication in Binius*. <https://www.irreducible.com/posts/integer-multiplication-in-binius>. Blog post. 2025.
- [JLXY25] Y. Jia, S. Li, C. Xing, Y. Yao, and C. Yuan. “Polynomial Commitments for Galois Rings and Applications to SNARKs Over $\mathbb{Z}_{2^k}$ ”. In: *Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI*. Ed. by Y. T. Kalai and S. F. Kamara. Vol. 16005. Lecture Notes in Computer Science. Springer, 2025, pp. 515–548. DOI: [10.1007/978-3-032-01887-8_17](https://doi.org/10.1007/978-3-032-01887-8_17). URL: https://doi.org/10.1007/978-3-032-01887-8_17.

- [Li+24] Y. Li, D. Escudero, Y. Duan, Z. Huang, C. Hong, C. Zhang, and Y. Song. “Sublinear Distributed Product Checks on Replicated Secret-Shared Data over $\mathbb{Z}_{2^k}$ without Ring Extensions”. In: *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 2024, pp. 825–839. DOI: [10.1145/3658644.3690260](https://doi.org/10.1145/3658644.3690260).
- [LXLYH26] C. Li, R. Xu, Y. Li, Y. Yu, and Y. Hu. *EA Codes Approaching Singleton Bound (with Application to Field-Agnostic SNARKs)*. Cryptology ePrint Archive, Paper 2026/1782. 2026. URL: <https://eprint.iacr.org/2026/1782>.
- [LXY24] F. Lin, C. Xing, and Y. Yao. “More Efficient Zero-Knowledge Protocols over $\mathbb{Z}_{2^k}$ via Galois Rings”. In: *Advances in Cryptology – CRYPTO 2024: 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2024, Proceedings, Part IX*. Santa Barbara, CA, USA: Springer-Verlag, 2024, 424–457. ISBN: 978-3-031-68399-2. DOI: [10.1007/978-3-031-68400-5_13](https://doi.org/10.1007/978-3-031-68400-5_13). URL: https://doi.org/10.1007/978-3-031-68400-5_13.
- [LZZHZ25] T. Liu, Z. Zhang, Y. Zhang, W. Hu, and Y. Zhang. “Ceno: Non-uniform, Segment and Parallel Zero-Knowledge Virtual Machine”. In: *J. Cryptol.* 38.2 (2025), p. 17. DOI: [10.1007/s00145-024-09533-2](https://doi.org/10.1007/s00145-024-09533-2). URL: <https://doi.org/10.1007/s00145-024-09533-2>.
- [Mer89] R. C. Merkle. “A certified digital signature”. In: *Proceedings of the 9th Annual International Cryptology Conference*. CRYPTO ’89. 1989, pp. 218–238.
- [Ope26] OpenVM Contributors. *OpenVM: A Performant and Modular zkVM Framework*. <https://openvm.dev/>. Project website. 2026.
- [RIS26] RISC Zero. *RISC Zero: A Zero-Knowledge Verifiable General Computing Platform*. <https://risczero.com/>. Project website. 2026.
- [Sch80] J. T. Schwartz. “Fast Probabilistic Algorithms for Verification of Polynomial Identities”. In: *Journal of the ACM* 27.4 (1980), pp. 701–717.
- [Set26] S. Setty. *BinarySpartan: Spartan over binary fields*. Cryptology ePrint Archive, Paper 2026/1656. 2026. URL: <https://eprint.iacr.org/2026/1656>.
- [STW24] S. T. V. Setty, J. Thaler, and R. S. Wahby. “Unlocking the Lookup Singularity with Lasso”. In: *Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26–30, 2024, Proceedings, Part VI*. Ed. by M. Joye and G. Leander. Vol. 14656. Lecture Notes in Computer Science. Springer, 2024, pp. 180–209. DOI: [10.1007/978-3-031-58751-1_7](https://doi.org/10.1007/978-3-031-58751-1_7). URL: https://doi.org/10.1007/978-3-031-58751-1_7.
- [STZ26] S. Setty, J. Thaler, and M. Zhu. “Twist and Shout: Faster Memory Checking Arguments via One-Hot Addressing and Increments”. In: *Advances in Cryptology – CRYPTO 2026: 46th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17–20, 2026, Proceedings, Part IX*. Santa Barbara, CA, USA: Springer-Verlag, 2026, 267–296. ISBN: 978-3-032-35426-6. DOI: [10.1007/978-3-032-35424-2_10](https://doi.org/10.1007/978-3-032-35424-2_10). URL: https://doi.org/10.1007/978-3-032-35424-2_10.
- [Suc26] Succinct Labs. *SP1: A Zero-Knowledge Virtual Machine for RISC-V Programs*. <https://www.succinct.xyz/>. Project website. 2026.
- [WZD25] Y. Wei, X. Zhang, and Y. Deng. “Transparent SNARKs over Galois Rings”. In: *Public-Key Cryptography – PKC 2025: 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12–15, 2025, Proceedings, Part I*. Røros, Norway: Springer-Verlag, 2025, 418–451. ISBN: 978-3-031-91819-3. DOI: [10.1007/978-3-031-91820-9_14](https://doi.org/10.1007/978-3-031-91820-9_14). URL: https://doi.org/10.1007/978-3-031-91820-9_14.
- [XZZPS19] T. Xie, J. Zhang, Y. Zhang, C. Papamanthou, and D. Song. “Libra: Succinct Zero-Knowledge Proofs with Optimal Prover Computation”. In: *Proceedings of the 39th Annual International Cryptology Conference*. CRYPTO ’19. 2019, pp. 733–764.

- [YQGZ26] X. Yang, W. Qu, Y. Guo, and J. Zhang. “Rarus: A Succinct and Efficient Range Proof for Polynomial-based Vector Commitment”. In: *Proceedings of the 35th USENIX Security Symposium*. USENIX Security '26. Full version at <https://eprint.iacr.org/2026/1440>. 2026.
- [Zip79] R. Zippel. “Probabilistic algorithms for sparse polynomials”. In: *Proceedings of the 1979 International Symposium on Symbolic and Algebraic Computation*. EUROSAM '79. 1979, pp. 216–226.