GlueLUT: Efficient Lookup Table Arguments over Residue Rings

Yuanju Wei[1,2], Zhelei Zhou[3,✉], Xinxuan Zhang[1,2,✉], Songyu Wu[1,2], Binwu Xiang[4], Cheng Hong[3], Yi Deng[5,✉], and Xiaofeng Chen[5]

[1] State Key Laboratory of Cyberspace Security Defense
Institute of Information Engineering, Chinese Academy of Sciences
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] Ant Group
[4] East China Normal University
[5] School of Cryptology, Xidian University

{weiyuanju, zhangxinxuan, wusongyu}@iie.ac.cn
{zhouzhelei.zzl, vince.hc}@antgroup.com
bwxiang@sc.ecnu.edu.cn, ydeng.cas@gmail.com, xfchen@xidian.edu.cn

Abstract. Lookup table polynomial interactive oracle proofs (LUT PIOPs) make SNARK arithmetizations concise, yet almost all efficient constructions assume field arithmetic. We show that directly instantiating them over a composite residue ring $\mathbb{Z}_Q$ can be unsound: component-wise set checks preserve the residue multiset in each CRT component but lose the alignment across components. We formalize this obstruction, which we call the *CRT alignment ambiguity*.

To overcome this ambiguity, we present GlueLUT, a family of efficient LUT PIOPs over residue rings. GlueLUT runs the lookup over an auxiliary field while retaining the arithmetic proof over $\mathbb{Z}_Q$, and then proves consistency between the two witness representations. Our first construction, GlueLUT-4Sq, introduces a cross-modulus consistency (CMC) PIOP, which proves that witnesses over two coprime moduli encode the same integer vector. We construct the CMC PIOP via a range-check PIOP over the product ring using the Lagrange four-square decompositions, structured Johnson- Lindenstrauss projections, and a GKR-style consistency check. For a witness of size n, a table of size m, and security parameter λ, GlueLUT-4Sq has $O(n+m)$ algebraic prover work after witness generation and $O(\mathrm{poly}(\lambda, \log n, \log m))$ verifier work and proof size. Our second construction, GlueLUT-Fold, uses random rank-one folded consistency checks to achieve better concrete prover efficiency and has $O(n+m)$ prover work, at the cost of $O(\sqrt{n}+\mathrm{poly}(\lambda, \log n, \log m))$ verifier work; its proof size is $O(\log n + \log m)$. We implement GlueLUT-4Sq and GlueLUT-Fold as stand-alone PIOPs and report prototype results that corroborate our theoretical efficiency analysis.

Keywords: Lookup-table · PIOPs · Residue rings.

1 Introduction

Lookup Table (LUT) argument replace expensive algebraic constraints with membership in a prescribed table and have become a standard component of modern SNARK arithmetizations [5, 22, 26, 51, 57]. Informally, given a witness vector and a public table, a lookup argument allows the prover to convince the verifier that every witness entry occurs in the table. This abstraction replaces operations such as bit decomposition, range checking, and exponentiation by table-membership constraints, and is now widely used in succinct arguments of knowledge [4, 29, 39, 45].

Why lookups matter even more for SNARKs over rings. While most deployed SNARKs operate over large prime fields, an increasingly important set of applications is natively *ring-based*, with arithmetic carried out over residue rings $\mathbb{Z}_Q$. Examples include verifiable computation for homomorphic encryption, verifiable CPU computations, and efficient fixed-/floating-point arithmetic [27, 35, 37]. Motivated by these targets, recent work has begun to build SNARKs and polynomial commitments directly over residue rings (or Galois rings) [15, 27, 37, 55, 56]. However, despite this progress on *ring-native algebraic proof systems*, we still lack efficient and general-purpose LUT arguments over $\mathbb{Z}_Q$.

Ring barriers. Over prime fields, lookup arguments admit several highly optimized approaches. Hafezi et al. [33] organize them into multiset-equality methods [5, 26, 59], LogUp and logarithmic-derivative methods [8, 11, 12, 21–23, 32, 46, 60], accumulator-based methods [13, 25, 34], and subvector-extraction or matrix–vector methods [49–51, 57, 58]. Bünz et al. [9] recently introduced a further approach based on permutation checks.

Almost all of these techniques rely on *set checks*, such as multiset checks or bit checks, as discussed in Section 3. We identify a structural obstruction to directly instantiating such a set check over $\mathbb{Z}_Q$ when Q has multiple coprime factors. To see this, write $Q = \prod_{j=0}^{t-1} Q_j$ where the Q_j's are pairwise coprime. By the Chinese Remainder Theorem (CRT), an algebraic check over $\mathbb{Z}_Q$ decomposes into checks over the component rings $\mathbb{Z}_{Q_j}$. Such component-wise checks can establish the intended set relation in every component, while they do not reveal how residues from different components are aligned into elements of $\mathbb{Z}_Q$. We call this loss of joint information the *CRT alignment ambiguity*.

For example, consider $\mathbb{Z}_{15} \cong \mathbb{Z}_3 \times \mathbb{Z}_5$. Under this isomorphism, the table $\{0, 1\}$ corresponds to $\{(0, 0), (1, 1)\}$. Realigning the second components gives $\{(0, 1), (1, 0)\}$, which corresponds to $\{6, -5\} \subseteq \mathbb{Z}_{15}$. The two tables have identical multiset projections modulo 3 and modulo 5, but they are distinct over $\mathbb{Z}_{15}$.

Thus, component-wise set checks do not imply the corresponding set relation over $\mathbb{Z}_Q$. Section 3 formalizes this ambiguity and explains why almost all existing field-based lookup techniques cannot be instantiated over $\mathbb{Z}_Q$ in a black-box manner. This raises the central question of this work:

Can we construct sound and efficient LUT arguments over $\mathbb{Z}_Q$ despite the CRT alignment ambiguity?

1.1 Our Results

We answer the above question affirmatively by presenting GlueLUT, a family of efficient LUT PIOPs over residue rings, and evaluating its concrete performance. Our contributions are as follows.

The CRT alignment ambiguity. We formalize the CRT alignment ambiguity and show that it is a fundamental barrier to black-box, component-wise instantiations of existing lookup techniques over $\mathbb{Z}_Q$.

GlueLUT: LUT PIOPs over residue rings. We construct a set of LUT PIOPs over the residue ring $\mathbb{Z}_Q$ (cf. Table 1) with different asymptotic costs. To achieve these results, we develop several novel techniques and summary them as follows.

GlueLUT-4Sq: based on a new primitive called CMC PIOP. To bypass the CRT alignment ambiguity, our main idea is to introduce an auxiliary field $\mathbb{F}_P$, and run the LUT PIOP over $\mathbb{F}_P$ while retaining the arithmetic proof over $\mathbb{Z}_Q (P > Q)$. To soundly connect the two witness representations, we introduce *cross-modulus consistency* (CMC) PIOP, a new primitive that certifies that witnesses over two coprime moduli encode the same centered integer vector. Our key observation is that CMC PIOP can be reduced, through an implicit CRT bridge, to a range-check PIOP over the product ring $\mathbb{Z}_{PQ}$.

To construct an efficient range-check PIOP over $\mathbb{Z}_{PQ}$, we explore how to combine the structured Johnson–Lindenstrauss projections [40] and the four-square decomposition technique [48]. We further optimize the efficiency by introducing a GKR-style consistency check [30]. Given the above techniques, we propose our first LUT PIOP GlueLUT-4Sq over the residue ring.

GlueLUT-Fold: based on random rank-one folded consistency checks. Our second LUT PIOP GlueLUT-Fold shares the similar idea with GlueLUT-4Sq, i.e., running the LUT PIOP over an auxiliary field $\mathbb{F}_P$ while retaining the arithmetic proof over $\mathbb{Z}_Q$. The difference is that, to achieve better concrete prover efficiency, GlueLUT-Fold avoids the explicit CMC PIOP construction and ensures the witness consistency via our new technique called *random rank-one folded consistency checks*. We note that, this design substantially improves concrete prover efficiency at the cost of higher asymptotic verifier complexity.

RingLUT-Perm: based on permutation-check based LUT PIOPs. Along the way to build our range-check PIOP over the product ring, we also discover how to construct a Booleanity PIOP over the residue rings.

As a by product, we extend the permutation-check based LUT PIOP by Bünz et al. [9] over a *field* into the ring setting by combining our Booleanity PIOP over $\mathbb{Z}_Q$, which is denoted as RingLUT-Perm.

Table 1: Asymptotic costs of our lookup PIOPs over $\mathbb{Z}_Q$.[a]

Protocol	Prover	Verifier	Size	Extra oracles[b]
GlueLUT-4Sq	$O(n+m)$[c]	$O(\mathrm{poly}(\lambda, \log n, \log m))$	$O(\mathrm{poly}(\lambda, \log n, \log m))$	9
GlueLUT-Fold	$O(n+m)$	$O(\sqrt{n} + \mathrm{poly}(\lambda, \log n, \log m))$	$O(\log n + \log m)$	1
RingLUT-Perm	$O(\iota n + m)$	$O(\mathrm{poly}(\lambda, \log n, \log m))$	$O(\mathrm{poly}(\lambda, \log n, \log m))$	ι

[a] Let $n = 2^\mu$ be the witness size, $m = 2^\iota$ the table size, and $\ell_Q = \lceil \log_2 Q \rceil$.
[b] Extra oracles count committed witness oracles introduced beyond the native ring witness.
[c] This is the algebraic PIOP cost after the eight four-square root oracles have been generated. Their expected generation cost is $O\big(n(\log Q)^2 / \log\log Q\big)$ basic integer operations.

Implementation and evaluation. We implement GlueLUT-4Sq and GlueLUT-Fold as stand-alone PIOPs in a single-threaded prototype over $Q = pq$, where p and q are distinct 64-bit primes. The concrete performance results are consistent with our theoretical analysis.

1.2 Technical Overview

Bypassing CRT alignment: lookup over an auxiliary field. We begin by making precise the *CRT alignment ambiguity* and then explain how it motivates our auxiliary-field approach. Suppose that the moduli are $Q = \prod_{j=0}^{t-1} Q_j$, where $t \geq 2$ and the Q_j's are pairwise coprime, and let $\pi_j : \mathbb{Z}_Q \to \mathbb{Z}_{Q_j}$ denote the canonical projection. For a multiset A over $\mathbb{Z}_Q$, a component-wise set check reveals only the projected multisets $\pi_j(A)$. This information does not determine A: residues in different CRT components can be permuted independently, changing how they are combined into elements of $\mathbb{Z}_Q$ while leaving every projected multiset unchanged. In other words, the map

$$A \longmapsto \big(\pi_0(A), \ldots, \pi_{t-1}(A)\big)$$

is not injective. We call this loss of information the *CRT alignment ambiguity*.

Consequently, a field-based LUT PIOP whose set check decomposes across CRT components may accept an invalid lookup over $\mathbb{Z}_Q$, and cannot be instantiated over $\mathbb{Z}_Q$ in a black-box manner. Our starting point is therefore to separate the lookup from the native computation. Let g denote the witness used by the arithmetic proof over $\mathbb{Z}_Q$. We choose an auxiliary prime $P > Q$, interpret each table entry as an integer, and map it to $\mathbb{F}_P$. The prover commits to an auxiliary-field witness f and uses a field-based lookup argument to prove that every entry of f belongs to the resulting table over $\mathbb{F}_P$.

It remains to show that the auxiliary-field witness f and the native ring witness g encode the same integer at every position. If this holds, the table membership proved for f also holds for g. We capture this requirement in a separate primitive called *cross-modulus consistency* (CMC).

Gluing the two witnesses: cross-modulus consistency. CMC asks whether witnesses committed over two coprime moduli encode the same integer vector. At each coordinate, $f(b) \in \mathbb{Z}_P$ and $g(b) \in \mathbb{Z}_Q$ should be the reductions of the

same integer modulo P and modulo Q, respectively. CMC is also of independent interest: beyond lookup arguments, it provides a general mechanism for binding bounded integer witnesses represented modulo two coprime moduli.

Our key observation is that proving CMC reduces to proving a range statement over a single product ring. Throughout the paper, we represent each element of $\mathbb{Z}_Q$ by the unique integer in $[-\lfloor Q/2 \rfloor, \lceil Q/2 \rceil)$. Set $R = PQ (P > Q)$. By CRT, each pair $(f(b), g(b))$ determines a unique value $h(b) \in \mathbb{Z}_R$ satisfying

$$h(b) \equiv f(b) \pmod{P}, \qquad h(b) \equiv g(b) \pmod{Q}.$$

If $h(b)$ lies in $[-\lfloor Q/2 \rfloor, \lceil Q/2 \rceil)$, then $f(b)$ and $g(b)$ encode this same integer modulo P and modulo Q, respectively. Therefore, CMC is reduced to proving that the CRT combination $h(b)$ stays within this range at every coordinate. The prover does not need to commit to h separately, since it is already determined by the two committed witnesses by CRT.

The remaining challenge is to prove this range statement over the composite ring $\mathbb{Z}_R$. Over fields, range proofs are often implemented using LUT PIOPs. However, this approach would be circular in our setting, if a LUT PIOP over rings is required to implement the range proof. We instead develop a range-check PIOP over $\mathbb{Z}_R$ based on a four-square representation.

The core of CMC: a range-check PIOP over the product ring. In the following, we will show how to construct a range-check PIOP over the product ring. Note that, to express the required range in a non-negative form, we let z be the integer between roughly $-R/2$ and $R/2$ that equals h modulo R, and shift it by $\tau_Q = \lfloor Q/2 \rfloor$. Define

$$D = z + \tau_Q, \qquad A = Q - 1 - D.$$

Then z lies in the valid range for $\mathbb{Z}_Q$ exactly when $D \in [0, Q)$. Equivalently, it suffices to certify that D and its complementary slack $A = Q - 1 - D$ are both non-negative.

The challenge: reducing range-check to proof of shortness. We certify the non-negativity of D and A by the four-square decomposition lemma, i.e., we represent each of them as a sum of four squares. Over the integers, these decompositions immediately imply $D, A \geq 0$. Over $\mathbb{Z}_R$, however, the corresponding identities hold only modulo R. Therefore, the challenge is *how to certify that the square-decomposition witnesses are sufficiently short* to rule out modular wraparound, thereby lifting the modular four-square identities to equalities over the integers.

Existing range proofs based on square decomposition address this challenge through commitment-specific arguments. CKLR [20] and SHARP [19] certify short rational openings using bounded-response Σ-protocols tailored to Pedersen commitments, whereas DewTwo [10] and Dew++ [24] rely on proofs of integer-exponent relations in groups of unknown order. These techniques do not directly provide a range proof for polynomial oracles over $\mathbb{Z}_R$, and their soundness arguments are tied to the underlying commitment schemes. In contrast, our goal is to certify shortness directly for polynomia oracles over $\mathbb{Z}_R$, which yields a

range-check PIOP that is independent of a particular commitment construction and can be compiled with any compatible PCS.

Concretely, for every $b \in B_\mu$, the prover supplies eight auxiliary values $d_0(b), \ldots, d_3(b)$ and $e_0(b), \ldots, e_3(b)$. Define

$$D_b := \sum_{i=0}^{3} d_i(b)^2, \qquad A_b := \sum_{i=0}^{3} e_i(b)^2.$$

The protocol checks modulo R that D_b represents the shifted value $D(b)$ and that $D_b + A_b \equiv Q - 1 \pmod{R}$.

Since both D_b and A_b are sums of squares, it remains to show that their sum does not wrap around modulo R. Once $D_b + A_b < R/2$, the congruence above becomes an equality over the integers, giving $0 \le D_b \le Q - 1$.

To establish this condition simultaneously for all $b \in B_\mu$, we arrange the evaluations of the eight auxiliary oracles as a $2^\mu \times 8$ matrix and concatenate them into a single vector $\mathbf{u}$. For an honest decomposition,

$$\|\mathbf{u}\|_2^2 = \sum_{b \in B_\mu} (D_b + A_b) = 2^\mu (Q - 1).$$

It remains to certify that the decomposition witnesses are sufficiently short. In particular, it suffices to establish the global norm bound $\|\mathbf{u}\|_2^2 < R/2$, since this implies, for every $b \in B_\mu$, $D_b + A_b \le \|\mathbf{u}\|_2^2 < R/2$.

This global norm bound rules out wraparound at every point and lifts the checked congruences to the corresponding integer range statements. The remaining question is how to certify this norm bound with linear prover complexity and polylogarithmic verifier complexity and transcript size.

Certifying shortness via structured JL projection and GKR-style protocols. To certify this norm bound, we first introduce the modular JL projection technique used in LaBRADOR [3]. Let $M_0 = 2^{\mu+3}$ be the length of $\mathbf{u}$, and write $\mathbf{w}^{(0)} = \mathbf{u} \in \mathbb{Z}_R^{M_0}$. Informally, the JL projection lemma states that applying a random projection matrix Π gives a shorter vector whose norm, except with negligible probability, lies in $[\alpha\|\mathbf{w}^{(0)}\|_2, \beta\|\mathbf{w}^{(0)}\|_2]$, where α, β are some parameters. The verifier can therefore infer a bound on $\|\mathbf{w}^{(0)}\|_2$ by checking the norm of the projected vector. However, the projection matrix has $O(M_0)$ entries, resulting in linear verifier complexity.

The subsequent lattice-based SNARK Rok and Roll [40] reduces this cost using structured JL projections. Instead of one large projection, it recursively applies $L = O(\log M_0)$ structured projections, each generated using a constant number of random coins (depending only on the security parameter). The projection chain is

$$\mathbf{w}^{(i+1)} = \Pi_i \mathbf{w}^{(i)}, \qquad i = 0, \ldots, L - 1.$$

After L projections, the initial vector is reduced to a short vector $\mathbf{w}^{(L)}$, whose norm the verifier can check directly. Consequently, sampling the projections and checking the final norm require only $\mathrm{poly}(\lambda, \log M_0)$ verifier work.

It remains to prove that $\mathbf{w}^{(L)}$ was computed correctly. Rok and Roll [40] makes this chain of intermediate commitments efficient by exploiting the algebraic structure of its lattice-based argument. This mechanism does not carry over directly to a hash-based PCS over rings [15, 37, 55, 56], where committing to every intermediate vector would incur substantial proof overhead.

Our key observation is that the repeated-block structure of the structured JL projections allows us to certify the entire projection chain using a GKR-style protocol, without committing to the intermediate vectors. Specifically, each structured projection applies the same small random matrix J_i to every block, and hence has the repeated-block form $\Pi_i = I \otimes J_i$, where I is an identity matrix of the appropriate dimension. Let $M_i = 2^{m_i}$ denote the length of $\mathbf{w}^{(i)}$, and let $\widetilde{w}^{(i)}$ and $\widetilde{\Pi}_i$ denote the multilinear extensions of $\mathbf{w}^{(i)}$ and Π_i, respectively. Write the row and column variables of $\widetilde{\Pi}_i$ as $y = (y_b, y_a)$ and $x = (x_b, x_a)$, where the subscripts b and a denote the block and within-block coordinates.

Each projection can be expressed as a sumcheck instance:

$$\widetilde{w}^{(i+1)}(r) = \sum_{x \in B_{m_i}} \widetilde{\Pi}_i(r, x)\widetilde{w}^{(i)}(x).$$

At the end of sumcheck, for a verifier-chosen point s, this claim is reduced to the two evaluations $\widetilde{w}^{(i)}(s)$ and $\widetilde{\Pi}_i(r, s)$. The latter can be evaluated locally by the verifier. Indeed, the repeated-block structure gives

$$\widetilde{\Pi}_i((y_b, y_a), (x_b, x_a)) = \mathrm{eq}(y_b, x_b)\, \widetilde{J}_i(y_a, x_a),$$

where eq is the multilinear equality polynomial and $\widetilde{J}_i$ is the multilinear extension of J_i. Since J_i is small and determined by the verifier's randomness, $\widetilde{\Pi}_i(r, s)$ can be evaluated in polylogarithmic time. Consequently, after evaluating $\widetilde{\Pi}_i(r, s)$ locally, the verifier reduces an evaluation claim on $\widetilde{w}^{(i+1)}$ at r to an evaluation claim on $\widetilde{w}^{(i)}$ at s. Applying this reduction recursively, with one sumcheck per projection layer, transforms the claim on $\widetilde{w}^{(L)}$ into a single evaluation claim on the committed $\widetilde{w}^{(0)}$. The prover therefore sends only $\mathbf{w}^{(L)}$, with no commitments to intermediate vectors. Since the vector lengths decrease geometrically, the protocol has linear prover complexity and polylogarithmic verifier complexity and transcript size.

One subtlety requires particular care: a JL projection introduces both completeness and soundness slack. Suppose that the initial vector satisfies $\|\mathbf{w}^{(0)}\|_2 = E$. Except with negligible probability, after one projection, $\alpha E \leq \|\mathbf{w}^{(1)}\|_2 \leq \beta E$. Applying this argument recursively, after L successful projections, $\alpha^L E \leq \|\mathbf{w}^{(L)}\|_2 \leq \beta^L E$.

Now suppose that the desired norm bound for $\mathbf{w}^{(0)}$ is B. To preserve completeness, the verifier should accept a final vector satisfying $\|\mathbf{w}^{(L)}\|_2 \leq \beta^L B$. We call $\beta^L B$ the completeness slack. Conversely, from this final bound, the verifier can infer only $\|\mathbf{w}^{(L-1)}\|_2 \leq \frac{\beta^L}{\alpha} B$. Applying the same argument backwards through all L projection layers gives $\|\mathbf{w}^{(0)}\|_2 \leq \left(\frac{\beta}{\alpha}\right)^L B$, which we call the soundness slack.

For an honest four-square decomposition, $B^2 = 2^\mu(Q-1)$. For completeness, the norms entering the projection layers need to remain within the admissible range of the JL lemma. For soundness, the inferred upper bound $\left(\frac{\beta}{\alpha}\right)^L B$ on $\|\mathbf{w}^{(0)}\|_2$ needs to remain within the centered range; equivalently, $\left(\frac{\beta}{\alpha}\right)^{2L} B^2 < R/2$. Since $B^2 = 2^\mu(Q-1)$, the latter condition becomes $2\left(\frac{\beta}{\alpha}\right)^{2L} 2^\mu(Q-1) < R$. Thus, the construction applies when R is sufficiently larger than $2^\mu Q$ to accommodate the accumulated JL slack. The complete parameter requirements are given in Section 5.2.

Under these conditions, an accepting proof implies $\|\mathbf{u}\|_2^2 < R/2$, completing the four-square range PIOP. Using this range PIOP to instantiate CMC, we obtain GlueLUT-4Sq, a lookup table PIOP over residue rings. Together with the auxiliary-field witness f, the eight root oracles account for the 9 extra oracles of GlueLUT-4Sq.

A by-product: Booleanity PIOP and its application to LUT PIOP. We find that the norm-certification technique above also provides a Booleanity PIOP over residue rings. For a polynomial $u : B_\mu \to \mathbb{F}$, Booleanity over a field is captured by $u(x)(u(x)-1) = 0$ for every $x \in B_\mu$. Over $\mathbb{Z}_R$, however, CRT alignment allows values outside $\{0,1\}$ to satisfy the same equations. For example, over $\mathbb{Z}_{15}$, the realigned values 6 and -5 both satisfy $x(x-1) = 0$, although neither is a bit.

Following the insight of LaBRADOR [3], we combine the Boolean equation with a low-norm check. Informally, let $u : B_\mu \to \mathbb{Z}_R$. The Boolean equation checks $u(x)(u(x)-1) \equiv 0 \pmod R$ for every $x \in B_\mu$, while the low-norm check establishes $\|u\|_2 < \sqrt{R/2}$, where $\|u\|_2 = \sqrt{\sum_{x\in B_\mu} u(x)^2}$. This bound gives $|u(x)| < \sqrt{R/2}$, and hence

$$|u(x)(u(x)-1)| \le |u(x)|(|u(x)|+1) < R.$$

The modular Boolean equation can therefore hold only if $u(x)(u(x)-1) = 0$ over the integers, which implies $u(x) \in \{0,1\}$. We certify the required low-norm condition using the structured JL projections and GKR-style consistency check described above.

As a byproduct, we combine this Booleanity PIOP with the permutation-based lookup of Bünz et al. [9] to obtain RingLUT-Perm, a lookup table PIOP over residue rings. We defer its construction and analysis to Appendix C.

Improving the prover's efficiency via random rank-one folded consistency checks. The range-check PIOP above instantiates the general CMC PIOP with a constant number of auxiliary oracles, but it still incurs the concrete prover cost of proving a pointwise range statement for the implicit CRT combination. This generality is useful when consistency between the two representations is needed independently of a lookup-table argument.

The lookup setting admits a further optimization, since the auxiliary field can be chosen independently of the native computation over $\mathbb{Z}_Q$. We exploit this flexibility by applying the same random linear form to the evaluation vectors

of the auxiliary-field witness f and the native ring witness g, while choosing P large enough to prevent wraparound on the field side. The verifier interprets the resulting field sum as an integer, reduces it modulo Q, and compares it with the corresponding ring-side sum, thereby testing the consistency between f and g.

More concretely, a natural way to obtain such a random linear form is to evaluate the multilinear extensions of f and g at the same random point. However, this assigns to each entry a coefficient that is a product of μ equality weights. The integer values of these coefficients can grow rapidly with μ, requiring a much larger auxiliary field to rule out wraparound. We instead reshape each witness into a $2^{\mu/2} \times 2^{\mu/2}$ matrix and use a rank-one random weight matrix. Writing $b = (x, y)$, where $x, y \in B_{\mu/2}$, let q be the smallest prime divisor of Q, and let $S := \big[-\lfloor q/2 \rfloor, \lceil q/2 \rceil\big) \cap \mathbb{Z}$ be an exceptional set of $\mathbb{Z}_Q$. The verifier independently samples two coefficient tables $r, t : B_{\mu/2} \to S$.

The parties use sumcheck to establish the folded sums

$$a = \sum_{x,y \in B_{\mu/2}} g(x, y) r(x) t(y) \in \mathbb{Z}_Q, \qquad s = \sum_{x,y \in B_{\mu/2}} f(x, y) r(x) t(y) \in \mathbb{F}_P.$$

Let $f_{\mathbb{Z}}(x, y)$ denote the centered integer representative of $f(x, y)$. Crucially, the field lookup itself certifies the required range for f: every accepted $f(x, y)$ is a lifted table entry, and hence $|f_{\mathbb{Z}}(x, y)| \le \lfloor Q/2 \rfloor$. The choice of challenges independently gives $|r(x)|, |t(y)| \le \lfloor q/2 \rfloor$. Hence the corresponding integer weighted sum

$$s_{\mathbb{Z}} := \sum_{x,y \in B_{\mu/2}} f_{\mathbb{Z}}(x, y) r(x) t(y)$$

satisfies $|s_{\mathbb{Z}}| \le 2^{\mu} \lfloor Q/2 \rfloor \lfloor q/2 \rfloor^2$. We therefore choose $P > \max\big\{Q, 2^{\mu+1} \lfloor Q/2 \rfloor \lfloor q/2 \rfloor^2\big\}$. Then $|s_{\mathbb{Z}}| < P/2$, so the field element s uniquely determines $s_{\mathbb{Z}}$ as its centered integer representative. The verifier reduces $s_{\mathbb{Z}}$ modulo Q and compares it with a. Thus, enlarging the auxiliary field allows us to test folded agreement without a separate pointwise CMC range statement.

For soundness, suppose that f and g disagree at some position after their values are compared modulo Q. The difference between the two folded sums is then a nonzero polynomial of total degree two in the sampled coefficients. The challenge space above is an exceptional set over $\mathbb{Z}_Q$, so the generalized Schwartz–Zippel lemma bounds the probability that this difference vanishes by $2/q$. Repeating the check k times reduces this error to $(2/q)^k$.

The resulting PIOP GlueLUT-Fold avoids both the explicit CMC layer and its pointwise range proof, leading to a substantially smaller concrete prover time. Counting arithmetic over $\mathbb{F}_P$ and $\mathbb{Z}_Q$ in field and ring operations, respectively, GlueLUT-Fold has prover complexity $O(n + m)$, verifier complexity $O(\sqrt{n} + \mathrm{poly}(\lambda, \log n, \log m))$, and proof size $O(\log n + \log m)$ field or ring elements.[6] The $O(\sqrt{n})$ verifier term is in the same asymptotic regime as the two-

[6] In the interactive description, the verifier sends the coefficient tables r and t. After applying the Fiat–Shamir transform, both parties derive these tables from the transcript through the random oracle, so the tables need not be included in the transcript and are not counted toward its size.

dimensional tensor-IOP instantiation underlying Brakedown [31], and remains sublinear in n, hence retaining succinct verification under the standard sublinear notion. The auxiliary-field witness f is its only extra polynomial oracle. The full construction and its parameter requirements are given in Section 6.

1.3 Related Works

LUT arguments. LUT arguments have been extensively studied in the context of SNARKs over prime fields. A large body of work has proposed highly efficient constructions based on different algebraic principles. Early and influential approaches are based on *multiset equality*, which encodes table membership as an equality between product polynomials [5, 26, 59]. Subsequent works developed alternative paradigms, including *accumulator-based* techniques [13, 25, 34], *subvector extraction (matrix–vector) methods* [49–51, 57, 58], and *LogUp-style rational-function arguments* [8, 11, 12, 21–23, 32, 46, 60]. Hafezi et al. [33] provide a comprehensive systematization and comparison of these techniques. More recently, Bünz et al. [9] proposed a new LUT argument construction based on permutation checks. Their approach relies on sumcheck, which makes it algebraically more flexible.

All of the above techniques, however, were developed over fields and are typically instantiated over large prime fields. Consequently, they do not directly yield sound LUT arguments over residue rings $\mathbb{Z}_Q$. Our work differs in focus: rather than proposing a new LUT argument over fields, we study how LUT arguments can be realized efficiently over *residue rings*.

Cascudo et al. [15] constructed range proofs over residue rings in the context of verifiable homomorphic encryption. Their construction directly instantiates over a residue ring a field-based lookup argument based on a multiset check. As shown in subsection 3.2, such a direct ring instantiation preserves only the component-wise multisets and cannot distinguish nontrivial CRT realignments. Consequently, an additional mechanism is needed to resolve CRT alignment when applying this approach over $\mathbb{Z}_Q$.

Concurrent work. Concurrent work by Bootle et al. [7] also studies LUT arguments over rings, but in a different algebraic setting. We consider residue rings $\mathbb{Z}_Q$, whereas Bootle et al. consider the polynomial ring $\mathbb{Z}_q[X]/(X^d + 1)$, where q is prime and d is a power of two. Both works independently identify CRT-component swapping as an obstruction to directly instantiating field-based LUT arguments over rings. The constructions are also different: we use auxiliary-field consistency, whereas Bootle et al. develop ring-native variants of Plookup and LogUp. In summary, these two works address different ring settings and use different techniques.

SNARKs over rings. A recent line of work has begun constructing SNARKs over rings. In [18], the sumcheck and GKR protocols were generalized to finite commutative rings, where the verifier's challenges are required to be sampled from an exceptional subset of the ring. In [6], a sumcheck protocol over rings was developed and further applied to solve the Rank-1 Constraint System (R1CS)

problem over rings. Moreover, the sumcheck protocol was extended to infinite non-commutative rings in [53].

The Rinocchio protocol [27] was the first complete SNARK designed for ring-based arithmetic circuits, albeit requiring a trusted setup. Subsequent works focused on transparent SNARK constructions over (Galois) rings [36,37,55,56]. These systems establish the feasibility of efficient arithmetic proofs over residue rings, but do not address LUT arguments.

Our work complements this line of research by constructing efficient LUT arguments over the integer residue ring $\mathbb{Z}_Q$, thereby filling a key missing component in the design of SNARK systems over rings.

Besides constructing SNARKs directly over rings, a different line of work considers simulating ring arithmetic within field-based proof systems. Garreta et al. [28] propose Zinc, which realizes SNARKs for computations over $\mathbb{Z}$ and $\mathbb{Q}$ by reducing integer relations to relations over randomly sampled prime fields. This framework also supports computations over residue rings $\mathbb{Z}_Q$ by expressing them as integer relations. A subsequent work, Zinc+ [1], generalizes this framework to statements over polynomial rings and related quotient structures. Zaratan [14] provides succinct arguments for computations over $\mathbb{Z}$ by lifting a prime-field proof system to the integers. Limber [17] constructs a practical integer mod-PCS from any standard field PCS, further reducing the commitment overhead of SNARKs for integer arithmetic.

The Chinese Remainder Theorem in SNARKs. Several recent works employ the Chinese Remainder Theorem (CRT) to address the Non-Native Arithmetic (NNA) problem in SNARKs [2,41,43]. The NNA problem concerns proving correctness of arithmetic operations that are not natively supported by the algebraic domain in which the SNARK is defined. These works use the CRT within a single underlying algebraic structure to express other arithmetic operations.

In contrast, our use of the CRT serves a fundamentally different purpose. We use the CRT as a consistency mechanism to connect witnesses represented over different algebraic structures. Concretely, we combine the CRT with a range-proof PIOP over rings to enforce that two representations correspond to the same underlying integer values. This enables lookup arguments over rings by soundly transferring membership checks to a field-based lookup protocol and then transporting the result back to the ring via consistency constraints. This role of the CRT is orthogonal to NNA-focused applications and is not captured by prior CRT-based SNARK constructions.

2 Preliminaries

2.1 Notions

We let λ be the security parameter and use PPT to denote probabilistic algorithms with expected polynomial running time. We write $x \leftarrow S$ when x is sampled uniformly from a finite set S. For integers $a \leq b$, let $[a, b] := \{a, a+1, \ldots, b\}$, and for $a \in \mathbb{N}$, let $[a] := \{0, 1, \ldots, a-1\}$. We write $B_\mu := \{0,1\}^\mu$ for the μ-dimensional Boolean hypercube.

2.2 Background on Rings

We use $\mathfrak{R}$ to denote a ring. For every integer modulus $Q \geq 2$, define the centered interval $\mathcal{I}_Q := \big[-\lfloor Q/2 \rfloor, \lceil Q/2 \rceil\big) \cap \mathbb{Z}$. In this work, we use the centered interval $\mathcal{I}_Q$ to represent elements of $\mathbb{Z}_Q$ in a symmetric way around zero. For $x \in \mathbb{Z}_Q$, we write $\mathrm{rep}_Q(x) \in \mathcal{I}_Q$ for its unique centered representative. In particular, when Q is even, the endpoint convention chooses $-Q/2$ rather than $Q/2$.

We write $\mathfrak{R}[X_0, \ldots, X_{\ell-1}]$ for the polynomial ring over $\mathfrak{R}$. Its polynomials of individual degree at most d form $\mathfrak{R}^{(\leq d)}[X_0, \ldots, X_{\ell-1}]$.

Definition 1 (Exceptional sets [27]). *An exceptional set is a subset $S \subset \mathfrak{R}$ with the property that $a - b$ is an invertible element of $\mathfrak{R}$ for each pair of distinct elements $a, b \in S$.*

Remark 1. For a residue ring $\mathbb{Z}_Q$, let q be the smallest prime factor of Q. Then $\mathcal{I}_q$, embedded naturally in $\mathbb{Z}_Q$, is an exceptional set.

Lemma 1 (Generalized Schwartz–Zippel lemma [27]). *Let $f \in \mathfrak{R}[X_1, \ldots, X_\ell]$ be a non-zero polynomial of total degree d. Then for an exceptional set $S \subset \mathfrak{R}$,*

$$\Pr_{(r_1,\ldots,r_\ell) \leftarrow S^\ell}\big[f(r_1, \ldots, r_\ell) = 0\big] \leq \frac{d}{|S|}.$$

Definitions of SNARKs and polynomial commitment schemes over rings are recalled in Appendices A.1 and A.2, respectively. Ring-based sumcheck and ZeroCheck PIOPs are recalled in Appendices A.3 and A.4, respectively.

2.3 Modular Johnson–Lindenstrauss Projections

We recall the modular Johnson–Lindenstrauss (JL) projection lemmas used in our range-check PIOPs. Let χ be the distribution over $\{-1, 0, 1\}$ with

$$\Pr[\chi = 0] = \frac{1}{2}, \qquad \Pr[\chi = 1] = \Pr[\chi = -1] = \frac{1}{4}.$$

Lemma 2 (Johnson–Lindenstrauss projection [3]). *For every $\kappa_{\mathrm{rp}} \in (0, 1)$, there exist parameters $n_{\mathrm{rp}} = n_{\mathrm{rp}}(\kappa_{\mathrm{rp}}), \alpha_{\mathrm{rp}} = \alpha_{\mathrm{rp}}(\kappa_{\mathrm{rp}}), \beta_{\mathrm{rp}} = \beta_{\mathrm{rp}}(\kappa_{\mathrm{rp}}), c_{\mathrm{rp}} = c_{\mathrm{rp}}(\kappa_{\mathrm{rp}})$ such that, for every $m_{\mathrm{rp}} \in \mathbb{N}$, every $Q \in \mathbb{N}$, and every nonzero $\mathbf{w} \in \mathbb{Z}^{m_{\mathrm{rp}}}$ with $\|\mathbf{w}\|_2 = B$ and $B \leq Q/c_{\mathrm{rp}}$,*

$$\Pr\big[\|J\mathbf{w} \bmod Q\|_2 \leq \alpha_{\mathrm{rp}} B \;\big|\; J \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}\big] \leq \kappa_{\mathrm{rp}},$$
$$\Pr\big[\|J\mathbf{w} \bmod Q\|_2 \geq \beta_{\mathrm{rp}} B \;\big|\; J \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}\big] \leq \kappa_{\mathrm{rp}}.$$

Here, reduction modulo Q is represented in the centered interval $\mathcal{I}_Q$. For $\kappa_{\mathrm{rp}} = 2^{-128}$, we use the concrete parameters $n_{\mathrm{rp}} = 256, \alpha_{\mathrm{rp}} = \sqrt{30}, \beta_{\mathrm{rp}} = \sqrt{337}, c_{\mathrm{rp}} = 125$. Asymptotically, one expects $n_{\mathrm{rp}} = \Theta(\log(1/\kappa_{\mathrm{rp}}))$ and $\alpha_{\mathrm{rp}} = \Theta(\sqrt{\log(1/\kappa_{\mathrm{rp}})}), c_{\mathrm{rp}} = \Theta(\sqrt{\log(1/\kappa_{\mathrm{rp}})})$.

Lemma 3 (Structured Johnson–Lindenstrauss projection [40]). *Let $m_{\mathrm{rp}} \mid M$, let $\mathbf{w} \in \mathbb{Z}^M$, and partition $\mathbf{w}$ into M/m_{rp} blocks of length m_{rp}. Let*

$$J \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}, \qquad \Pi := I_{M/m_{\mathrm{rp}}} \otimes J,$$

where J is sampled as in Lemma 2. For every nonzero $\mathbf{w} \in \mathbb{Z}^M$ with $\|\mathbf{w}\|_2 = B$ and $B \le Q/c_{\mathrm{rp}}$,

$$\Pr\left[\|\Pi\mathbf{w} \bmod Q\|_2 \le \alpha_{\mathrm{rp}} B \;\middle|\; J \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}\right] \le \kappa_{\mathrm{rp}} \frac{M}{m_{\mathrm{rp}}},$$
$$\Pr\left[\|\Pi\mathbf{w} \bmod Q\|_2 \ge \beta_{\mathrm{rp}} B \;\middle|\; J \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}\right] \le \kappa_{\mathrm{rp}} \frac{M}{m_{\mathrm{rp}}}.$$

The corresponding one-sided variants are stated in Appendix A.5.

3 The CRT Alignment Ambiguity

A direct instantiation of a field-based LUT argument over $\mathbb{Z}_Q$ decomposes into independent checks over the CRT components. Such checks retain only component-wise information and do not determine how residues from different components are aligned into elements of $\mathbb{Z}_Q$. We call this loss of joint information the *CRT alignment ambiguity*. Subsection 3.1 formalizes this ambiguity, and Subsection 3.2 examines its consequences for existing field-based LUT arguments.

3.1 Non-Injectivity of Component-Wise CRT Encodings

We now formalize the CRT alignment ambiguity described above. We first describe what a component-wise lookup check sees, and then show that this information does not uniquely determine a multiset over $\mathbb{Z}_Q$.

Let $Q = \prod_{j=0}^{t-1} Q_j$ where $t \ge 2$ and the Q_j's are pairwise coprime. Let $\pi_j : \mathbb{Z}_Q \to \mathbb{Z}_{Q_j}$ be the canonical projection. For a multiset A over $\mathbb{Z}_Q$, write $\pi_j(A) := \{\pi_j(a) : a \in A\}$ as a multiset over $\mathbb{Z}_{Q_j}$.

Definition 2 (CRT-component equivalence). *Two multisets A, B over $\mathbb{Z}_Q$ are* CRT-component equivalent*, written $A \equiv_{\mathsf{CRT}} B$, if $\pi_j(A) = \pi_j(B)$ as multisets for every $j \in [t]$.*

CRT-component equivalence is the information retained by a check that works independently in each CRT component. Over a field, the corresponding multiset information determines the table. Over a residue ring with a nontrivial CRT decomposition, it doesn't determine the joint alignment across components.

Theorem 1 (CRT alignment ambiguity). *Let $Q = \prod_{j=0}^{t-1} Q_j$ with $t \ge 2$ and pairwise coprime Q_j's. Let $m \ge 2$. For each $j \in [t]$, let*

$$A_j = \{a_{j,0}, \ldots, a_{j,m-1}\} \subseteq \mathbb{Z}_{Q_j}$$

be a set of size m. For every tuple of permutations $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{t-1}) \in (S_m)^{t-1}$, define a multiset

$$A^{\boldsymbol{\sigma}} = \{a_0^{\boldsymbol{\sigma}}, \ldots, a_{m-1}^{\boldsymbol{\sigma}}\} \subseteq \mathbb{Z}_Q$$

by the CRT conditions

$$\pi_0(a_i^{\boldsymbol{\sigma}}) = a_{0,i}, \qquad \pi_j(a_i^{\boldsymbol{\sigma}}) = a_{j,\sigma_j(i)} \quad \text{for } j = 1, \ldots, t-1.$$

Then all multisets $A^{\boldsymbol{\sigma}}$ are CRT-component equivalent. Moreover, if $\boldsymbol{\sigma} \neq \boldsymbol{\tau}$, then $A^{\boldsymbol{\sigma}} \neq A^{\boldsymbol{\tau}}$. Hence CRT-component equivalence is not injective whenever Q has at least two coprime factors.

The proof of Theorem 1 is given in Appendix B.1. Theorem 1 is the structural source of the failure. A CRT-component check sees the same projected table A_j in each component, but it does not see which element of A_0 is aligned with which element of A_j for $j \geq 1$. This is a multi-component ambiguity: for $t = 2$ it is a pairing ambiguity, while for $t > 2$ it is a joint alignment ambiguity.

We call B a *CRT realignment* of A if $A \equiv_{\mathsf{CRT}} B$, and call the realignment nontrivial if $A \neq B$. Thus Theorem 1 constructs many nontrivial CRT realignments by changing the joint alignment while preserving every projected multiset.

Over a field, a standard way to arithmetize a multiset A is to encode it by its product polynomial $\prod_{a \in A}(X - a)$. The next corollary shows why the same encoding is not injective over $\mathbb{Z}_Q$: CRT-component equivalent multisets have the same product polynomial.

Corollary 1 (Product of polynomials cannot distinguish CRT realignments). *Let A, B be CRT-component equivalent multisets over $\mathbb{Z}_Q$. Then*

$$\prod_{a \in A}(X - a) = \prod_{b \in B}(X - b) \quad \text{in } \mathbb{Z}_Q[X].$$

Consequently, product encodings are not injective over $\mathbb{Z}_Q$ when Q has at least two coprime factors.

The proof of Corollary 1 is given in Appendix B.2. The failure is not specific to the product-polynomial encoding. Theorem 1 also implies that component-wise membership alone is insufficient to establish ring-level membership.

Corollary 2 (Component-wise membership does not imply ring-level membership). *Let $Q = \prod_{j=0}^{t-1} Q_j$ with $t \geq 2$ and pairwise coprime factors. Let $T \subseteq \mathbb{Z}_Q$ be a finite lookup table, and suppose that T admits a nontrivial CRT realignment $T' \subseteq \mathbb{Z}_Q$; that is, $T' \equiv_{\mathsf{CRT}} T$ but $T' \neq T$. Then there exists a witness value $w \in T' \setminus T$ such that $\pi_j(w) \in \pi_j(T)$, for every $j \in [t]$, even though $w \notin T$. Consequently, the component-wise membership checks do not by themselves soundly certify the ring-level lookup relation $w \in T$.*

The formal proof of Corollary 2 is given in Appendix B.3. We present a simple example of this corollary in the following: In $\mathbb{Z}_{15} \cong \mathbb{Z}_3 \times \mathbb{Z}_5$, the Boolean table

$\{0,1\}$ corresponds to the pairs $\{(0,0),(1,1)\}$. Swapping the $\mathbb{Z}_5$-components gives $\{(0,1),(1,0)\}$, which corresponds to the centered representatives $\{6,-5\} \subseteq \mathcal{I}_{15}$. The two tables are CRT-component equivalent. Taking $w = 6$, we have $w \notin \{0,1\}$, but $w \bmod 3 = 0 \in \{0,1\}, w \bmod 5 = 1 \in \{0,1\}$. Thus the invalid ring-level lookup passes in both CRT components. The same realignment also explains the product-polynomial collision from Corollary 1:

$$X(X-1) = (X-6)(X+5) \qquad \text{in } \mathbb{Z}_{15}[X].$$

In summary, component-wise checks retain only the coordinate projections of ring elements and lose their joint CRT alignment. A sound ring-level lookup argument therefore needs an additional mechanism beyond independent component-wise checks to preserve or recover this alignment. The next subsection examines how this obstruction manifests across existing field-based LUT techniques.

3.2 Implications for Existing LUT Techniques

Following the taxonomy of Hafezi et al. [33], we consider multiset-check, LogUp, subvector, and accumulator techniques. We also include the recent permutation-check approach of Bünz et al. [9]. We discuss multiset-check and LogUp together because both derive lookup soundness from identities over unordered multisets.

Multiset Checks and LogUp. Multiset checks are a basic ingredient in many lookup arguments, including grand-product constructions [5, 26, 42, 59]. Over a field, they can certify multiset equality by comparing product polynomials, often at a random verifier challenge. Their soundness uses the fact that the product polynomial uniquely determines the multiset.

This implication fails over $\mathbb{Z}_Q$. By Corollary 1, CRT realignments have the same product polynomial, so a direct ring instantiation can establish equality of the component views without establishing equality of the ring-level multisets.

LogUp-style arguments [8, 11, 12, 21–23, 32, 46, 60] replace product comparison with rational-function identities derived from the same unordered multiset data. CRT-component equivalent multisets induce the same identities in every component. Hence these identities alone also cannot distinguish a table from a nontrivial CRT realignment.

Subvector and Matrix–Vector Techniques. In this family, lookup membership is expressed through a sparse selector or a memory-checking relation. Caulk [57], Caulk+ [49], and Baloo [58], for example, rely on KZG commitments [38] and pairing-based algebra. These ingredients do not directly instantiate over the residue rings considered here.

Lasso [51] supports more flexible polynomial commitments and expresses sparse selection through a memory-checking technique. This technique relies on multiset checks. Under a direct instantiation over $\mathbb{Z}_Q$, these checks are enforced independently in each CRT component, so the alignment ambiguity from Theorem 1 reappears. Shout [50] instead uses Spark-naive polynomial commitments. Existing realizations of this property rely on group-based assumptions such as

DDH-type structures or pairings, and we are not aware of an efficient Spark-naive commitment scheme over the residue rings considered here.

Accumulator-Based Techniques. Accumulator-based lookup arguments use cryptographic accumulators together with commit-and-prove batch openings. Known constructions rely either on pairing-based accumulators [25] or on RSA accumulators [13,34], where table elements are encoded as primes. These assumptions and encodings are not native to arithmetic over $\mathbb{Z}_Q$, so these constructions do not yield a black-box ring lookup argument or by themselves address CRT realignment.

Permutation-Check-Based Techniques. Bünz et al. [9] construct the BiPerm and MulPerm permutation checks and derive lookup arguments from them. BiPerm relies on Spark-naive polynomial commitments, whereas MulPerm can use an arbitrary polynomial commitment and is closer to a native ring instantiation.

The remaining issue is the reduction from an unindexed lookup to an indexed one. Let $g : B_\mu \to \mathbb{Z}_Q$ encode the queried values and let $T : B_\iota \to \mathbb{Z}_Q$ encode the table. For an unindexed lookup, the prover supplies a selector $\rho : B_\mu \to B_\iota$ satisfying $g(x) = T(\rho(x))$ for every $x \in B_\mu$, and proves that every coordinate of ρ is Boolean. Over a field, this is enforced by $h(h-1) = 0$. Over $\mathbb{Z}_Q$, the same equation allows CRT-realigned values: in the $\mathbb{Z}_{15}$ example above, both 6 and -5 satisfy the equation although neither lies in $\{0, 1\}$. Thus the field-level Booleanity reduction is unsound over $\mathbb{Z}_Q$.

4 GlueLUT-CMC: A General Framework for Constructing LUT PIOP via Cross-Modulus Consistency

As shown in Section 3, a direct component-wise instantiation of a field-based LUT PIOP over $\mathbb{Z}_Q$ cannot certify the joint alignment of the CRT components. To bypass this barrier, our core idea is to run the lookup over an auxiliary field and glue its witness to the native $\mathbb{Z}_Q$ witness. Let $T : B_\iota \to \mathbb{Z}_Q$ be a lookup table, and let $g : B_\mu \to \mathbb{Z}_Q$ be the committed ring witness. For a prime $P > Q$, we lift the centered representatives of the table entries to $\mathbb{F}_P$. The prover commits to an auxiliary-field witness $f : B_\mu \to \mathbb{F}_P$ and proves that f belongs to the lifted table using a field-based lookup argument.

The LUT PIOP over a field alone does not bind f to g. We therefore require the two witnesses to encode the same centered integer vector. We first formalize this requirement as Cross-Modulus Consistency (CMC) and construct a CMC PIOP using a range-check PIOP as a black box. We then combine the CMC PIOP with a LUT PIOP over a auxiliary field to obtain a LUT PIOP over the residue ring $\mathbb{Z}_Q$.

Definition 3 (Lookup relation over $\mathbb{Z}_Q$). *Let $Q \geq 2$ be an integer. We define the LUT relation over $\mathbb{Z}_Q$, denoted by $\mathcal{R}_{\mathsf{LUT},Q}$, as the set of all tuples*

$$(x, w) = \big((Q, [[g]]_Q, T, [[T]]_Q); (g)\big),$$

where $g \in \mathbb{Z}_Q^{(\leq 1)}[X_0, \ldots, X_{\mu-1}]$ and $T \in \mathbb{Z}_Q^{(\leq 1)}[X_0, \ldots, X_{\iota-1}]$, such that for every $b \in B_\mu$, there exists $u \in B_\iota$ satisfying $g(b) = T(u)$ over $\mathbb{Z}_Q$.

4.1 Cross-Modulus Consistency

We begin with the general answer to the gluing problem. The goal is not only to support LUT arguments, but to provide a reusable PIOP for *cross-modulus consistency* (CMC): given two prescribed coprime moduli P and Q, where $P > Q$, the verifier is convinced that two committed residue witnesses encode the same centered integer vector. This formulation is stronger than what is strictly needed for lookup, but it is useful whenever a protocol maintains a native ring witness together with a companion representation over another modulus. We first define the CMC relation.

Definition 4 (Cross-modulus consistency relation). *Let $P, Q \geq 2$ be coprime integers with $P > Q$. We define the CMC relation, denoted by $\mathcal{R}_{\mathsf{CMC},P,Q}$, as the set of all tuples*

$$(x, w) = \big((P, Q, [[f]]_P, [[g]]_Q); (f, g)\big),$$

where $f \in \mathbb{Z}_P^{(\leq 1)}[X_0, \ldots, X_{\mu-1}]$ and $g \in \mathbb{Z}_Q^{(\leq 1)}[X_0, \ldots, X_{\mu-1}]$, such that

$$\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b)) \qquad \text{for every } b \in B_\mu.$$

We write $\mathsf{CMC}_{P,Q}(f, g) = 1$ for the condition above.

We construct the CMC PIOP based on the following canonical bridge.

Fact 1 (CMC via a centered CRT bridge) *Let P, Q be coprime integers with $P > Q$, and set $R := PQ$. Let $x \in \mathcal{I}_P$, $y \in \mathcal{I}_Q$, and $z \in \mathcal{I}_R$. If*

$$z \equiv x \pmod{P}, \qquad z \equiv y \pmod{Q}, \qquad \text{and} \qquad z \in \mathcal{I}_Q,$$

then $x = y = z$ as integers in $\mathcal{I}_Q$.

The proof of Fact 1 is given in Appendix B.4. We now turn this characterization into a PIOP for $\mathsf{CMC}_{P,Q}(f, g) = 1$. Let $R := PQ$, and define an *implicit* oracle $h : B_\mu \to \mathbb{Z}_R$ pointwise by CRT, so that $h(b) \equiv f(b) \pmod{P}$ and $h(b) \equiv g(b) \pmod{Q}$. By Fact 1, CMC reduces to proving $\mathrm{rep}_R(h(b)) \in \mathcal{I}_Q$ for every b.

We use a range-check PIOP for this last statement as a black box. The range-check PIOP takes the commitments to f and g, views h as their implicit CRT lift, and proves $\mathrm{rep}_R(h(b)) \in \mathcal{I}_Q$, $\forall b \in B_\mu$. Section 5 instantiates this range-check PIOP using a four-square decomposition together with a structured JL small-norm check.

Theorem 2 (CMC from a centered range-check PIOP). *Let P, Q be coprime integers with $P > Q$, and let $R = PQ$. If the range PIOP in Fig. 1 has completeness probability at least $1 - \varepsilon_{\mathsf{Range}}$ and knowledge-soundness error at most δ_{Range}, then the resulting CMC PIOP has completeness probability at least $1 - \varepsilon_{\mathsf{Range}}$ and knowledge-soundness error at most δ_{Range}. Its prover complexity, verifier complexity, and proof size are those of the range-check PIOP.*

The proof of Theorem 2 is given in Appendix B.5.

CMC PIOP from a centered range-check PIOP.

- **Common input:** coprime moduli $Q < P$, $R = PQ$, and oracle commitments $[[f]]_P, [[g]]_Q$ for multilinear polynomials $f : B_\mu \to \mathbb{Z}_P$ and $g : B_\mu \to \mathbb{Z}_Q$.
- **Implicit CRT bridge:** for every $b \in B_\mu$, let $h(b) \in \mathbb{Z}_R$ be determined by

$$h(b) \equiv f(b) \pmod{P}, \qquad h(b) \equiv g(b) \pmod{Q}.$$

- **Range-check:** P and V run a range-check PIOP to prove

$$\mathrm{rep}_R(h(b)) \in \mathcal{I}_Q \qquad \forall b \in B_\mu.$$

Fig. 1: CMC PIOP from a range-check PIOP for the implicit CRT bridge.

4.2 Constructing LUT PIOP via CMC PIOP

We now apply the CMC PIOP to construct the LUT PIOP over the residure ring. The table is lifted from $\mathbb{Z}_Q$ to an auxiliary field $\mathbb{F}_P$. The prover commits to an auxiliary-field witness f, proves that f belongs to the lifted table, and then proves $\mathsf{CMC}_{P,Q}(f, g) = 1$. The CMC check identifies the centered representatives of f and the native witness g; therefore, the field lookup implies the original lookup over $\mathbb{Z}_Q$. We refer to this generic composition as the GlueLUT-CMC framework. It is given in Fig. 2.

Theorem 3 (GlueLUT-CMC framework). *Let $Q \geq 2$ be an integer, let $P > Q$ be prime, and let $n = 2^\mu$ and $m = 2^\iota$. Let $T : B_\iota \to \mathbb{Z}_Q$ be a lookup table and let $\widetilde{T} : B_\iota \to \mathbb{F}_P$ be its canonical lift. Assume:*

- *the LUT PIOP over $\mathbb{F}_P$ for vector length n and table size m has perfect completeness, knowledge-soundness error $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$, prover complexity $C^{\mathsf{LUT}}_{\mathsf{P}}$, verifier complexity $C^{\mathsf{LUT}}_{\mathsf{V}}$, and proof size C^{LUT}_{π};*
- *the CMC PIOP has completeness probability at least $1 - \varepsilon_{\mathsf{CMC}}$, knowledge-soundness error δ_{CMC}, prover complexity $C^{\mathsf{CMC}}_{\mathsf{P}}$, verifier complexity $C^{\mathsf{CMC}}_{\mathsf{V}}$, and proof size C^{CMC}_{π}.*

Then the generic composition in Fig. 2:

1. *has completeness probability at least $1 - \varepsilon_{\mathsf{CMC}}$;*
2. *has knowledge-soundness error at most $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}} + \delta_{\mathsf{CMC}}$;*
3. *has prover complexity $C^{\mathsf{LUT}}_{\mathsf{P}} + C^{\mathsf{CMC}}_{\mathsf{P}}$, verifier complexity $C^{\mathsf{LUT}}_{\mathsf{V}} + C^{\mathsf{CMC}}_{\mathsf{V}}$, and proof size $C^{\mathsf{LUT}}_{\pi} + C^{\mathsf{CMC}}_{\pi}$, excluding oracle commitment costs.*

The proof of Theorem 3 is given in Appendix B.6.

GlueLUT-CMC framework

- **Common input:** a modulus Q, a prime $P > Q$, a table $T : B_\iota \to \mathbb{Z}_Q$ of size $m = 2^\iota$, and oracle access to a committed ring witness $g : B_\mu \to \mathbb{Z}_Q$, where $n = 2^\mu$.
- **Canonical lift of the table:** define $\widetilde{T} : B_\iota \to \mathbb{F}_P$ by
$$\widetilde{T}(u) = \mathrm{rep}_Q(T(u)) \bmod P \in \mathbb{F}_P \qquad \forall u \in B_\iota.$$
Since $P > Q$, the centered integer $\mathrm{rep}_Q(T(u)) \in \mathcal{I}_Q \subseteq \mathcal{I}_P$ is represented without ambiguity modulo P.

Commitment. P commits to a multilinear polynomial $f : B_\mu \to \mathbb{F}_P$, intended to be the canonical lift of g, and sends the commitment $[[f]]_P$ to V.

Lookup over the auxiliary field. P and V run a LUT PIOP over a field $\mathbb{F}_P$ to prove $f(B_\mu) \subseteq \widetilde{T}(B_\iota)$.

Cross-modulus consistency. P and V run the CMC PIOP of Fig. 1 to prove
$$\mathsf{CMC}_{P,Q}(f, g) = 1.$$

Fig. 2: GlueLUT-CMC framework.

5 GlueLUT-4Sq: Range Checking via Four Squares Decomposition

Section 4.1 reduces the CMC PIOP to proving that the centered representative of every entry of the implicit CRT bridge lies in $\mathcal{I}_Q$. This section constructs such a range-check PIOP using four-square representations of a shifted value and its complementary slack. We first reduce the range statement to a global norm bound on the decomposition witnesses, and then certify this bound using structured JL projections together with a GKR-style consistency-check protocol.

5.1 Reducing Range Checking to Proof of Shortness

We construct the required range-check PIOP using the Lagrange's four-square theorem. The construction uses a constant number of auxiliary root oracles, independently of Q. Set $\tau_Q := \lfloor Q/2 \rfloor$. A centered integer $z \in \mathcal{I}_Q$ is equivalent to a shifted integer $D := z + \tau_Q \in [0, Q)$. The latter range condition is in turn equivalent to the nonnegativity of both D and its slack $A := Q - 1 - D$. Lagrange's four-square theorem lets the prover witness these two nonnegative integers as
$$D = \sum_{j=0}^{3} d_j^2, \qquad A = \sum_{j=0}^{3} e_j^2.$$
The verifier can check the two identities modulo R, but these modular checks alone do not exclude wraparound. To rule it out, we concatenate the centered integer evaluations of the eight root oracles into a vector $\mathbf{w}^{(0)}$. It remains to prove a global small-norm bound on this vector. If $\|\mathbf{w}^{(0)}\|_2^2 < R/2$, then at every point b, $\sum_{j=0}^{3} d_j(b)^2 + \sum_{j=0}^{3} e_j(b)^2 < R/2$. Consequently, its congruence to $Q-1$

modulo R is an equality over the integers. This proves $D(b) \in [0, Q)$, and hence places the implicit CRT bridge in $\mathcal{I}_Q$.

This construction uses eight auxiliary root oracles, independently of Q. The additional cost of constructing the four-square witnesses follows from the algorithmic result below.

Fact 2 (Lagrange's four-square theorem) *Every nonnegative integer is the sum of four integer squares.*

Theorem 4 (Efficient four-square decomposition [48]). *Given a positive integer z, there is an unconditional randomized algorithm that outputs integers x_0, x_1, x_2, x_3 satisfying $z = x_0^2 + x_1^2 + x_2^2 + x_3^2$. As z grows, the algorithm uses an expected $O\left(\frac{(\log z)^2}{\log \log z}\right)$ basic operations on integers of size at most $z^{O(1)}$.*

For each $b \in B_\mu$, our prover applies Theorem 4 to $D(b)$ and $Q - 1 - D(b)$; a zero input is represented by four zeros. Generating all eight root oracles therefore takes an expected $O(n(\log Q)^2/\log\log Q)$ basic integer operations. Once these oracles have been generated, the PIOP requires only $O(n)$ ring operations and a constant number of auxiliary oracle commitments. We report these two costs separately below.

Fact 3 (Centered range-check from four-square witnesses) *Let $R = PQ$, let $h \in \mathbb{Z}_R$, and set $\tau_Q := \lfloor Q/2 \rfloor$. For integers $d_0, \ldots, d_3, e_0, \ldots, e_3 \in \mathcal{I}_R$, define $D := \sum_{j=0}^{3} d_j^2$, $A := \sum_{j=0}^{3} e_j^2$. Suppose that*

$$h + \tau_Q \equiv D \pmod{R}, \qquad D + A \equiv Q - 1 \pmod{R}, \qquad D + A < R/2.$$

Then $\mathrm{rep}_R(h) \in \mathcal{I}_Q$.

The proof of Fact 3 is given in Appendix B.7.

Remark 2 (Three-square variant). Applying Legendre's three-square theorem [47] to $4D + 1$ and $4A + 1$ reduces the number of square-root oracles from eight to six and changes the squared-norm target to $4Q - 2$. The protocol below then applies with adjusted norm bounds and projection dimensions. We use four-square decompositions for simplicity.

5.2 Certifying Shortness via Structured JL Projections and GKR-Style Protocols

We now certify this global bound using JL projection techniques, as in several lattice-based SNARKs [3, 40]. Informally, a JL projection reduces a norm-bound claim on $\mathbf{w}^{(0)}$ to one on its compressed image $\Pi\mathbf{w}^{(0)}$, where Π is a random projection matrix. As discussed in Section 1.2, we use the structured JL projection from Lemma 3 for efficiency. It applies the same small random matrix to each block and, after $O(\log M_0)$ layers, reduces $\mathbf{w}^{(0)}$ to $\mathrm{poly}(\lambda)$ entries.

However, structured JL projections introduce a consistency-check problem: the final norm check constrains the initial vector $\mathbf{w}^{(0)}$ only if the final vector is obtained by applying every verifier-sampled projection to the committed root oracles. In [40], the authors make the corresponding chain of intermediate commitments efficient by exploiting the algebraic structure of its lattice-based argument. This mechanism does not carry over directly to hash-based PCSs over rings [15, 37, 55, 56], where committing to every intermediate vector would incur substantial proof overhead.

Our key observation is that the repeated-block structure of the structured JL projections makes the entire projection chain a public layered linear computation. We therefore use a GKR-style protocol to certify this chain without committing to the intermediate vectors. Each layer is verified by a sumcheck PIOP that reduces an evaluation claim on the output vector to one on the input vector. Chaining these reductions leaves only evaluation claims on the eight committed root oracles. Together, these ingredients give a range-check PIOP based on square decomposition for the implicit CRT lift h. Using this protocol to instantiate the generic construction of Theorem 2 yields a CMC PIOP proving $\mathsf{CMC}_{P,Q}(f,g)=1$. The complete protocol is shown in Figs. 3–4.

Theorem 5 (CMC PIOP Based on Four-Square Decomposition). *Let P,Q be coprime integers with $Q<P$, set $R:=PQ$, and let $n=2^{\mu}$. Set $M_0:=8n$ and $M_{i+1}:=M_i n_{\mathrm{rp}}/m_{\mathrm{rp}}$ for $i=0,\dots,L-1$, where $m_{\mathrm{rp}}\mid M_i$, and let $m_i:=\log M_i$. Choose L so that $M_L=\mathrm{poly}(\lambda)$. Define*

$$B_0:=\sqrt{n(Q-1)},\qquad B_i:=\beta_{\mathrm{rp}}^i B_0,\qquad \kappa_{\mathrm{JL}}^{\mathsf{4Sq}}:=\sum_{i=0}^{L-1}\kappa_{\mathrm{rp}}\frac{M_i}{m_{\mathrm{rp}}}.$$

Let S_R be the exceptional set used by the projection-consistency sumchecks over an extension of $\mathbb{Z}_R$, and assume

$$R>\max\left\{c_{\mathrm{rp}}\frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}}\sqrt{n(Q-1)},\,2\left(\frac{\beta_{\mathrm{rp}}}{\alpha_{\mathrm{rp}}}\right)^{2L}n(Q-1)\right\}.$$

Let $T_{\mathsf{4Sq}}(Q)$ denote the expected number of basic integer operations required to find a four-square representation of an integer in $[0,Q)$. The unconditional randomized algorithm of Pollack and Treviño [48] gives $T_{\mathsf{4Sq}}(Q)=O\left(\frac{(\log Q)^2}{\log\log Q}\right)$ for asymptotically growing Q. Assume:

- *the size-M_i, degree-2 projection-consistency sumcheck, evaluated in the extension ring containing S_R, has knowledge-soundness error at most $\kappa_{\mathrm{SC},M_i}^2$, linear prover complexity, logarithmic verifier complexity and proof size;*
- *the size-n, degree-2 ZeroCheck PIOP over $\mathbb{Z}_R$ has perfect completeness and knowledge-soundness error at most $\delta_{\mathbb{Z}_R,\mathsf{Zero}}^{n,2}$. Its prover complexity is linear, and its verifier complexity and proof size are logarithmic.*

Define $\delta_{P,Q,\mathsf{4SqCMC}}^{n,Q}:=\kappa_{\mathrm{JL}}^{\mathsf{4Sq}}+\frac{m_L}{|S_R|}+\sum_{i=0}^{L-1}\kappa_{\mathrm{SC},M_i}^2+2\delta_{\mathbb{Z}_R,\mathsf{Zero}}^{n,2}$. Then the CMC PIOP in Figs. 3–4:

CMC PIOP Based on Four-Square Decomposition.

- **Common input:** coprime moduli $Q < P$, $R = PQ$, and oracle commitments $[[f]]_P, [[g]]_Q$ for multilinear polynomials $f : B_\mu \to \mathbb{Z}_P$ and $g : B_\mu \to \mathbb{Z}_Q$, where $n = 2^\mu$.
- **Public parameters:** set $\tau_Q := \lfloor Q/2 \rfloor$, $M_0 := 8n$, and $B_0 := \sqrt{n(Q-1)}$. For $i = 0, \ldots, L-1$, set $M_{i+1} := M_i n_{\mathrm{rp}}/m_{\mathrm{rp}}$, assuming $m_{\mathrm{rp}} \mid M_i$. For $i = 0, \ldots, L$, set $B_i := \beta_{\mathrm{rp}}^i B_0$.
- **Statement:** $\mathsf{CMC}_{P,Q}(f, g) = 1$.
- **Auxiliary oracles:** P commits over $\mathbb{Z}_R$ to multilinear oracles $d_0, \ldots, d_3, e_0, \ldots, e_3 : B_\mu \to \mathbb{Z}_R$. Their intended centered integer values satisfy, for every $b \in B_\mu$,

$$\sum_{j=0}^{3} d_j(b)^2 = \mathrm{rep}_R(h(b)) + \tau_Q, \qquad \sum_{j=0}^{3} e_j(b)^2 = Q - 1 - \sum_{j=0}^{3} d_j(b)^2,$$

  where h is the implicit CRT lift of (f, g).

Small-norm check. Identify B_3 with $[8]$ in the natural binary order, and define the concatenated oracle $u : B_3 \times B_\mu \to \mathbb{Z}_R$ by

$$u(j, b) := d_j(b) \quad (j \in \{0, 1, 2, 3\}), \qquad u(4 + j, b) := e_j(b) \quad (j \in \{0, 1, 2, 3\}).$$

Let

$$\mathbf{w}^{(0)} := \big(\mathrm{rep}_R(u(s, b))\big)_{(s,b) \in B_3 \times B_\mu} \in \mathcal{I}_R^{M_0}.$$

Projection. After receiving the eight oracle commitments, V independently samples

$$J_i \leftarrow \chi^{n_{\mathrm{rp}} \times m_{\mathrm{rp}}}, \qquad i = 0, \ldots, L-1,$$

and sends $J_0, \ldots, J_{L-1}$ to P. Both parties set $\Pi_i := I_{M_i/m_{\mathrm{rp}}} \otimes J_i$.
For $i = 0, \ldots, L-1$, P computes

$$\mathbf{w}^{(i+1)} := \mathrm{rep}_R\Big(\Pi_i \mathbf{w}^{(i)} \bmod R\Big),$$

where rep_R is applied coordinatewise. The prover sends only $\mathbf{w}^{(L)}$. The verifier rejects unless $\|\mathbf{w}^{(L)}\|_2 \le B_L = \beta_{\mathrm{rp}}^L B_0$.

Fig. 3: CMC PIOP based on four-square decomposition (Part I).

1. *has completeness probability at least* $1 - \kappa_{\mathrm{JL}}^{\mathsf{4Sq}}$;
2. *has knowledge-soundness error at most* $\delta_{P,Q,\mathsf{4SqCMC}}^{n,Q}$;
3. *has PIOP complexity, excluding oracle commitment costs:*
   - *prover complexity consisting of expected* $O(nT_{\mathsf{4Sq}}(Q))$ *basic integer operations for witness generation and* $O(n)$ *ring operations for the subsequent PIOP execution;*
   - *verifier complexity* $O(\mathrm{poly}(\lambda, \log n))$ *ring operations;*
   - *proof size* $O(\mathrm{poly}(\lambda, \log n))$ *ring elements.*
   - *eight auxiliary witness oracles.*

The proof of Theorem 5 is given in Appendix B.8.

Corollary 3 (GlueLUT-4Sq). *Instantiate the GlueLUT-CMC framework of Theorem 3 with CMC PIOP from Theorem 5. If the LUT PIOP over the auxiliary*

CMC PIOP Based on Four-Square Decomposition (Continued).
Projection consistency. Let $\widetilde{w}^{(i)}$ and $\widetilde{\Pi}_i$ denote the multilinear extensions of $\mathbf{w}^{(i)}$ and Π_i, respectively, and let S_R be the exceptional set in the extension ring used for the sumcheck challenges. The verifier samples $r_L \leftarrow S_R^{m_L}$ and computes

$$y_L := \widetilde{w}^{(L)}(r_L)$$

from the explicit final vector. For $i = L, L-1, \ldots, 1$, the parties run sumcheck for

$$y_i = \sum_{x \in B_{m_{i-1}}} \widetilde{\Pi}_{i-1}(r_i, x)\widetilde{w}^{(i-1)}(x).$$

The reduction produces $r_{i-1} \leftarrow S_R^{m_{i-1}}$ and the claim $y_{i-1} = \widetilde{w}^{(i-1)}(r_{i-1})$. At the end of the layer, V evaluates $\widetilde{\Pi}_{i-1}(r_i, r_{i-1})$ locally and checks the terminal sumcheck value against $\widetilde{\Pi}_{i-1}(r_i, r_{i-1})\, y_{i-1}$.
After the last reduction, write $r_0 = (r_{\mathsf{sel}}, r_b)$, where $r_{\mathsf{sel}} \in S_R^3$ and $r_b \in S_R^{\mu}$. For $a \in B_3$, let $\mathrm{eq}(r_{\mathsf{sel}}, a)$ be the multilinear equality polynomial. The prover opens $\widetilde{d}_j(r_b)$ and $\widetilde{e}_j(r_b)$ for $j = 0, \ldots, 3$. The verifier checks these openings against the eight oracle commitments and verifies

$$y_0 = \sum_{j=0}^{3} \Big(\mathrm{eq}(r_{\mathsf{sel}}, \langle j \rangle_3)\widetilde{d}_j(r_b) + \mathrm{eq}(r_{\mathsf{sel}}, \langle 4+j \rangle_3)\widetilde{e}_j(r_b) \Big).$$

where $\langle j \rangle_3 \in B_3$ is the three-bit representation of j. This completes the proof that the final vector is obtained by applying the sampled projection chain to the committed root oracles.
Shifted reconstruction. Recall that $h : B_\mu \to \mathbb{Z}_R$ is the implicit CRT lift of f and g, whose evaluations are obtained by CRT-combining the corresponding evaluations of f and g. The parties run a degree-2 ZeroCheck over $\mathbb{Z}_R$ to prove

$$h(b) + \tau_Q = \sum_{j=0}^{3} d_j(b)^2 \qquad \forall b \in B_\mu.$$

Complementary four-square check. The parties run a degree-2 ZeroCheck over $\mathbb{Z}_R$ to prove

$$\sum_{j=0}^{3} \big(d_j(b)^2 + e_j(b)^2 \big) = Q - 1 \qquad \forall b \in B_\mu.$$

Fig. 4: CMC PIOP based on four-square decomposition (Part II).

field has $O(n+m)$ prover complexity and $L_{n,m} = \mathrm{poly}(\lambda, \log n, \log m)$ verifier complexity and proof size, the resulting LUT PIOP over $\mathbb{Z}_Q$, denoted GlueLUT-4Sq, has $O(n+m)$ algebraic prover complexity after witness generation, $O(L_{n,m})$ verifier complexity and proof size, and nine extra committed witness oracles. Generating the four-square witnesses takes expected time $O(nT_{4\mathsf{Sq}}(Q))$.

The proof of Corollary 3 is given in Appendix B.9.

A byproduct: Booleanity and RingLUT-Perm. Inspired by LaBRADOR's low-norm check for integer Booleanity [3], our norm-certification technique yields a Booleanity PIOP over residue rings: combined with $u(u-1) = 0$, its small-norm

bound excludes CRT-realigned nonbits. Combining it with the permutation-based LUT PIOP of Bünz et al. [9] gives RingLUT-Perm; see Appendix C.

6 GlueLUT-Fold: Check the Consistency via Random Rank-One Folding

The GlueLUT-CMC framework of Subsection 4.2 gives a general, modular solution to the gluing problem. Its CMC primitive certifies consistency between the auxiliary-field and native ring witnesses through a pointwise range-check PIOP for their implicit CRT bridge.

For LUT relations, we can specialize this gluing step because the auxiliary field can be chosen independently of the native computation over $\mathbb{Z}_Q$. We exploit this flexibility by applying the same random linear form to the evaluation vectors of the two witnesses, choosing P large enough to avoid wraparound on the field side, and comparing the resulting folded sums modulo Q.

We retain the notation of Subsection 4.2: $g : B_\mu \to \mathbb{Z}_Q$ is the committed ring witness, $f : B_\mu \to \mathbb{F}_P$ is the auxiliary-field witness, and $\widetilde{T}(u) = \mathrm{rep}_Q(T(u)) \bmod P$ is the centered lift of the lookup table. After the field lookup proves $f(B_\mu) \subseteq \widetilde{T}(B_\iota)$, it remains to establish $\mathrm{rep}_P(f(b)) \equiv g(b) \pmod{Q}$ for every $b \in B_\mu$.

A natural approach is to compare the two representations by evaluating their multilinear extensions. This direct random linear combination is unsuitable here because the equality polynomial $\mathrm{eq}(r, b) = \prod_{i=0}^{\mu-1} \big(r_i b_i + (1-r_i)(1-b_i)\big)$ contains μ multiplicative factors. The problem is that the absolute value of an equality weight can grow exponentially in μ, so preventing wraparound would require a much larger auxiliary modulus. We instead fold the witness by a rank-one random tensor. For simplicity, assume that μ is even. Write $b = (x, y)$ with $x, y \in B_{\mu/2}$. Let q be the smallest prime divisor of Q; then $\mathcal{I}_q \subset \mathbb{Z}_Q$ is an exceptional set. The verifier samples two tables

$$r : B_{\mu/2} \to \mathcal{I}_q, \qquad t : B_{\mu/2} \to \mathcal{I}_q,$$

and lets $\phi^{(Q)}$ and $\phi^{(P)}$ denote the multilinear extensions of these integer tables over $\mathbb{Z}_Q$ and $\mathbb{F}_P$, respectively, after reducing their entries modulo the corresponding modulus. The parties prove the two folded sums

$$a = \sum_{x,y \in B_{\mu/2}} g(x, y)\phi^{(Q)}(0, x)\phi^{(Q)}(1, y) \in \mathbb{Z}_Q,$$
$$s = \sum_{x,y \in B_{\mu/2}} f(x, y)\phi^{(P)}(0, x)\phi^{(P)}(1, y) \in \mathbb{F}_P.$$

To compare these folded sums across the two moduli, we need to ensure that the field-side sum does not wrap modulo P. Here the field lookup provides the required bound. Let $f_{\mathbb{Z}}(x, y) = \mathrm{rep}_P(f(x, y))$. Since every accepted value of f is a lifted table entry, $f_{\mathbb{Z}}(x, y) \in \mathcal{I}_Q$ and hence $|f_{\mathbb{Z}}(x, y)| \le \lfloor Q/2 \rfloor$. The corresponding signed integer sum $S := \sum_{x,y \in B_{\mu/2}} f_{\mathbb{Z}}(x, y)r(x)t(y)$ satisfies $|S| \le 2^\mu \lfloor Q/2 \rfloor \lfloor q/2 \rfloor^2$.

GlueLUT-Fold: Check the consistency via random rank-one folding

- **Common input:** a table $T : B_\iota \to \mathbb{Z}_Q$, and oracle access to a committed ring witness $g : B_\mu \to \mathbb{Z}_Q$. Let $n = 2^\mu$, $m = 2^\iota$, and let q be the smallest prime divisor of Q.
- **Public parameters:** choose a prime $P > \max\left\{Q,\, 2^{\mu+1}\lfloor Q/2\rfloor\lfloor q/2\rfloor^2\right\}$, and define $\widetilde{T}(u) = \mathrm{rep}_Q(T(u)) \bmod P \in \mathbb{F}_P$.

Auxiliary oracle. P commits to a multilinear polynomial $f : B_\mu \to \mathbb{F}_P$, intended to be the centered lift of g, and sends the commitment $[[f]]_P$ to V.

Consistency-check via folding. The verifier samples k independent pairs of coefficient tables

$$r_i, t_i : B_{\mu/2} \to \mathcal{I}_q, \qquad i = 0, \ldots, k-1,$$

and views them over each prescribed modulus. For $X \in \{Q, P\}$, let $\phi_i^{(X)} : \{0,1\} \times B_{\mu/2} \to \mathbb{Z}_X$, where $\mathbb{Z}_P = \mathbb{F}_P$, be the multilinear extension satisfying

$$\phi_i^{(X)}(0, z) = r_i(z) \bmod X, \qquad \phi_i^{(X)}(1, z) = t_i(z) \bmod X.$$

For each i, the parties run a sumcheck PIOP over $\mathbb{Z}_Q$ to prove

$$a_i = \sum_{x,y \in B_{\mu/2}} g(x,y)\phi_i^{(Q)}(0,x)\phi_i^{(Q)}(1,y),$$

and a sumcheck PIOP over $\mathbb{F}_P$ to prove

$$s_i = \sum_{x,y \in B_{\mu/2}} f(x,y)\phi_i^{(P)}(0,x)\phi_i^{(P)}(1,y).$$

The verifier rejects unless $a_i \equiv \mathrm{rep}_P(s_i) \pmod Q$ for every i.

Lookup over a auxiliary-field. P and V run a LUT PIOP over $\mathbb{F}_P$ to prove $f(B_\mu) \subseteq \widetilde{T}(B_\iota)$.

Fig. 5: GlueLUT-Fold Protocol.

Hence a prime $P > \max\left\{Q,\, 2^{\mu+1}\lfloor Q/2\rfloor\lfloor q/2\rfloor^2\right\}$ prevents wraparound in both directions: the centered representative of $s \in \mathbb{F}_P$ is exactly S. The verifier can therefore compare $\mathrm{rep}_P(s) \bmod Q$ with a. Repeating the test k times gives the folded-consistency error $(2/q)^k$, before adding the soundness errors of sumcheck PIOP and the LUT PIOP over a field. The full protocol is given in Fig. 5.

Theorem 6 (GlueLUT-Fold). *Let $n = 2^\mu$, $m = 2^\iota$, and let q be the smallest prime divisor of Q. Assume $q > 2$, choose k such that $(2/q)^k \le 2^{-\lambda}$, and choose a prime $P > \max\left\{Q,\, 2^{\mu+1}\lfloor Q/2\rfloor\lfloor q/2\rfloor^2\right\}$. Assume:*

- *The sumcheck PIOP over a ring $\mathfrak{R}$, for μ variables and individual degree at most d, has perfect completeness and knowledge-soundness error $\delta^{\mu,d}_{\mathfrak{R},\mathsf{Sum}}$;*
- *The LUT PIOP over $\mathbb{F}_P$ for vector length n and table size m has perfect completeness and knowledge-soundness error $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$, prover complexity $O(n{+}m)$, verifier complexity $O(\log n{+}\log m)$, and proof size $O(\log n{+}\log m)$.*

Then the protocol in Fig. 5 is a PIOP for LUT over $\mathbb{Z}_Q$ with perfect completeness and knowledge-soundness error at most $\left(\frac{2}{q}\right)^k + k \cdot \delta^{\mu,2}_{\mathbb{Z}_Q,\mathsf{Sum}} + k \cdot \delta^{\mu,2}_{\mathbb{F}_P,\mathsf{Sum}} + \delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$. *Excluding oracle commitment costs, its PIOP complexity is*

- *prover time* $O(kn + m)$*;*
- *verifier time* $O(k\sqrt{n} + \log n + \log m)$*;*
- *proof size* $O(k \log n + \log m)$ *ring or field elements.*

The proof of Theorem 6 is given in Appendix B.10.

Brakedown optimization. The folded checks above use sumcheck to remain independent of the underlying PCS. With Brakedown [31], or any PCS supporting tensor-style linear queries, the prover can instead answer matrix queries of the form $r^\top U$, which the verifier compares coordinatewise modulo Q. This eliminates the folded-consistency sumchecks at the cost of transmitting the queried vectors; see Appendix D.

7 Implementation and Evaluation

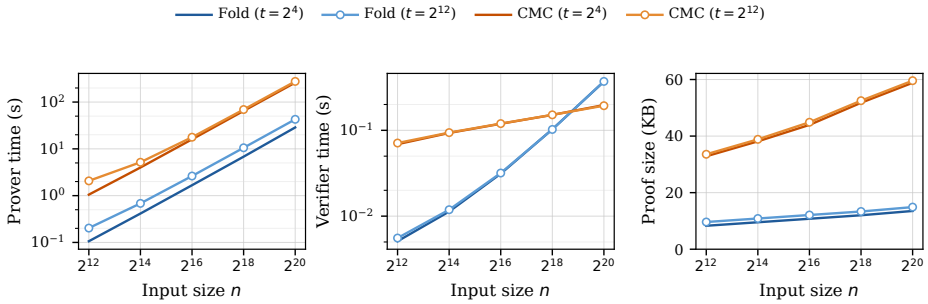
We provide a prototype implementation of our proposed LUT PIOPs in C++ using a single thread, on top of NTL [52] and EMP-toolkit [54] libraries. All experiments were conducted on a cloud instance equipped with an Intel Xeon Platinum 8575 CPU at 2.80 GHz and 128 GB of RAM. Each experiment is repeated ten times, and we report the average.

Our experimental evaluation aims to validate the practical efficiency of the proposed LUT PIOPs as stand-alone proof systems at the PIOP layer. In particular, we measure prover and verifier time and the PIOP transcript size after applying the Fiat–Shamir transform. Here, transcript size is the concrete byte length corresponding to the proof-size measure in our asymptotic analysis. PCS commitment generation and PCS opening verification are excluded from both the running time and transcript size. We do not build an end-to-end application (or a full SNARK system) on top of our protocols, since doing so would require committing to a particular PCS and engineering choices that are largely orthogonal to the contributions of this work.

Implemented PIOPs and parameter choices. We implement two of our main constructions, GlueLUT-4Sq and GlueLUT-Fold. Both target moduli $Q = pq$, where p and q are distinct 64-bit primes. All parameter sets achieve at least 100-bit computational security. Each experiment is repeated 10 times and we report the mean. We instantiate the FS random oracle with SHA-256. Both GlueLUT-4Sq and GlueLUT-Fold invoke a LUT PIOP over a *field* as a subroutine; we instantiate it with the multilinear variant of the LogUp-style LUT PIOP of [22].

For the structured-JL chain in GlueLUT-4Sq, we set the per-projection failure probability to $\kappa_{\mathrm{rp}} = 2^{-128}$ and use the concrete parameters $n_{\mathrm{rp}} = 256$, $\alpha_{\mathrm{rp}} = \sqrt{30}$, $\beta_{\mathrm{rp}} = \sqrt{337}$, and $c_{\mathrm{rp}} = 125$. Our implementation uses blocks of $m_{\mathrm{rp}} = 512$ coordinates, so every projection maps 512 coordinates to 256 and the chain stops once 512 coordinates remain. For the Galois-ring arithmetic used by sumcheck, we set the extension degree to 2.

Performance of our LUT PIOPs over a ring. Let n denote the witness size and let m denote the table size. We report the prover time, the verifier time, and the transcript size of both protocols for $m \in \{2^4, 2^{12}\}$ in Figure 6. It is easy to see that changing the table size from 2^4 to 2^{12} has only a small effect on all three metrics.



(a) Prover time (b) Verifier time (c) Proof size

Fig. 6: Performance of our LUT PIOPs over $\mathbb{Z}_{pq}$.

When the modulus parameters are fixed, the prover time of both protocols scales approximately linearly in n. As expected, GlueLUT-Fold is the faster construction. At $n = 2^{20}$, it takes 28.67–42.74 seconds, whereas GlueLUT-4Sq takes approximately 260–276 seconds. The additional cost of GlueLUT-4Sq comes primarily from constructing the CMC PIOP.

The verifier time curves exhibit a crossover. When $t = 2^4$ and $n = 2^{12}$, GlueLUT-4Sq takes about 0.069 seconds, whereas GlueLUT-Fold takes only about 0.005 seconds. This initial gap is caused by the relatively large parameters of the structured-JL chain: sampling and checking the fixed-size JL blocks introduces a visible startup cost. As n grows, the GlueLUT-4Sq verifier increases much more slowly, while the folded consistency check in GlueLUT-Fold has $O(\sqrt{n})$ verifier complexity. The verifier time curve of GlueLUT-Fold therefore overtakes that of GlueLUT-4Sq between $n = 2^{18}$ and $n = 2^{20}$. Both verifiers nevertheless remain below one second. Transcript sizes also grow only moderately as n grows.

Comparison with LUT PIOP over a field. To provide a better understanding of our proposed LUT PIOPs over the rings, we compare our protocols with the multilinear LogUp-style LUT PIOP over a 128-bit prime *field*, which is denoted as LUT-Field. Table 2 reports the comparison when n is set as 2^{20}.

Table 2: Comparison with a LUT PIOP over a 128-bit field at $n = 2^{20}$.

	$m = 2^4$			$m = 2^{12}$		
Protocol	Prover (s)	Verifier (s)	Trans. (KB)	Prover (s)	Verifier (s)	Trans. (KB)
LUT-Field	3.40	0.01	1.92	3.87	0.01	2.54
GlueLUT-Fold	28.67	0.37	13.52	42.74	0.37	14.89
GlueLUT-4Sq	260.02	0.20	58.84	276.17	0.20	59.57

As expected, LUT-Field has faster running time and a smaller transcript than our protocols. At $n = 2^{20}$, the resulting overhead relative to LUT-Field is about 8–76× for the prover, 20–37× for the verifier, and 6–31× for the PIOP transcript size, depending on the construction and table size. This is because our protocols

over the rings additionally resolve CRT-alignment ambiguity as discussed in Section 3, which requires much more effort. Furthermore, both protocols require arithmetic over extension rings to obtain a sufficiently large challenge space. Such arithmetic is much more expensive than the native-field arithmetic used by LUT-Field. To quantify this cost, we benchmarked 2^{20} multiplications using NTL: the computation takes 0.044 seconds over the field $\mathbb{F}_P$, compared with 0.436 seconds over the degree-2 extension ring $\mathsf{GR}(Q, 2)$, where both P and Q are 128-bit. This overhead is not specific to our constructions: Jia et al. [37] likewise report that their implementation of SNARK over Galois-rings is several orders of magnitude slower than finite-field counterparts, largely because currently available Galois-ring arithmetic lacks comparable low-level optimizations.

We emphasize that, even with our unoptimized single-threaded prototype, GlueLUT-Fold and GlueLUT-4Sq can prove a million-entry lookup in under 30 seconds and approximately 4.3 minutes, respectively. Both protocols have sub-second verifiers, while their PIOP transcript sizes remain below 15 KB and 60 KB, respectively. We therefore view both protocols as practical building blocks for ring-native applications.

Use of AI. The authors used GPT-5.6-sol to assist with revising and editing author-written text and with developing and debugging the experimental code. The authors reviewed all AI-assisted content and independently verified the implementation, experimental methodology, and reported data.

References

1. Abdugafarov, A., Garreta, A., Kumar, A., Osadnik, M., Vesely, P., Vlasov, I., Zheng, K.Z.: Zinc+: SNARKs for polynomial rings. Cryptology ePrint Archive, Paper 2026/855 (2026), https://eprint.iacr.org/2026/855
2. Ambrona, M., Firsov, D., Querejeta-Azurmendi, I.: Efficient foreign-field arithmetic in PLONK. IACR Cryptol. ePrint Arch. p. 695 (2025), https://eprint.iacr.org/2025/695
3. Beullens, W., Seiler, G.: Labrador: Compact proofs for R1CS from module-sis. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14085, pp. 518–548. Springer (2023). https://doi.org/10.1007/978-3-031-38554-4_17, https://doi.org/10.1007/978-3-031-38554-4_17
4. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Goldwasser, S. (ed.) Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012. pp. 326–349. ACM (2012), https://doi.org/10.1145/2090236.2090263
5. Bootle, J., Cerulli, A., Groth, J., Jakobsen, S.K., Maller, M.: Arya: Nearly linear-time zero-knowledge proofs for correct program execution. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11272, pp. 595–626. Springer (2018).

https://doi.org/10.1007/978-3-030-03326-2_20, https://doi.org/10.1007/978-3-030-03326-2_20
6. Bootle, J., Chiesa, A., Sotiraki, K.: Sumcheck arguments and their applications. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12825, pp. 742–773. Springer (2021), https://doi.org/10.1007/978-3-030-84242-0_26
7. Bootle, J., Guskind, J., Patranabis, S., Sotiraki, K.: Lookup arguments over rings and applications to batch-verification of RAM programs. IACR Cryptol. ePrint Arch. **2026**, 471 (2026), https://eprint.iacr.org/2026/471
8. Bünz, B., Chen, J.: Proofs for deep thought: Accumulation for large memories and deterministic computations. In: Chung, K., Sasaki, Y. (eds.) Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part V. Lecture Notes in Computer Science, vol. 15488, pp. 269–301. Springer (2024). https://doi.org/10.1007/978-981-96-0935-2_9, https://doi.org/10.1007/978-981-96-0935-2_9
9. Bünz, B., Chen, J., DeStefano, Z.: Linear*-time permutation check. IACR Cryptol. ePrint Arch. p. 1850 (2025), https://eprint.iacr.org/2025/1850
10. Bünz, B., Mopuri, T., Shirzad, A., Sridhar, S.: Dewtwo: A transparent PCS with quasi-linear prover, logarithmic verifier and 4.5kb proofs from falsifiable assumptions. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 16005, pp. 549–583. Springer (2025). https://doi.org/10.1007/978-3-032-01887-8_18, https://doi.org/10.1007/978-3-032-01887-8_18
11. Campanelli, M., Faonio, A., Fiore, D., Li, T., Lipmaa, H.: Lookup arguments: Improvements, extensions and applications to zero-knowledge decision trees. In: Tang, Q., Teague, V. (eds.) Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14602, pp. 337–369. Springer (2024). https://doi.org/10.1007/978-3-031-57722-2_11, https://doi.org/10.1007/978-3-031-57722-2_11
12. Campanelli, M., Fiore, D., Gennaro, R.: Natively compatible super-efficient lookup arguments and how to apply them. J. Cryptol. **38**(1), 14 (2025). https://doi.org/10.1007/S00145-024-09535-0, https://doi.org/10.1007/s00145-024-09535-0
13. Campanelli, M., Fiore, D., Han, S., Kim, J., Kolonelos, D., Oh, H.: Succinct zero-knowledge batch proofs for set accumulators. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 455–469. ACM (2022). https://doi.org/10.1145/3548606.3560677, https://doi.org/10.1145/3548606.3560677
14. Campanelli, M., Hall-Andersen, M.: General techniques for building SNARKs over the integers. In: Public-Key Cryptography - PKC 2026, Part III. Lecture Notes in Computer Science, vol. 16553, pp. 197–230. Springer (2026). https://doi.org/10.1007/978-3-032-26737-5_7, https://doi.org/10.1007/978-3-032-26737-5_7
15. Cascudo, I., Costache, A., Cozzo, D., Fiore, D., Guimarães, A., Soria-Vazquez, E.: Verifiable computation for approximate homomorphic encryption schemes. In:

Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VII. Lecture Notes in Computer Science, vol. 16006, pp. 643–677. Springer (2025). https://doi.org/10.1007/978-3-032-01907-3_21, https://doi.org/10.1007/978-3-032-01907-3_21
16. Chen, B., Bünz, B., Boneh, D., Zhang, Z.: Hyperplonk: Plonk with linear-time prover and high-degree custom gates. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14005, pp. 499–530. Springer (2023). https://doi.org/10.1007/978-3-031-30617-4_17, https://doi.org/10.1007/978-3-031-30617-4_17
17. Chen, J., Xia, L., Nguyen, W., Bünz, B.: Limber: Low overhead SNARKs for integers from any PCS. Cryptology ePrint Archive, Paper 2026/1635 (2026), https://eprint.iacr.org/2026/1635
18. Chen, S., Cheon, J.H., Kim, D., Park, D.: Verifiable computing for approximate computation. IACR Cryptol. ePrint Arch. p. 762 (2019), https://eprint.iacr.org/2019/762
19. Couteau, G., Goudarzi, D., Klooß, M., Reichle, M.: Sharp: Short relaxed range proofs. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 609–622. ACM (2022). https://doi.org/10.1145/3548606.3560628, https://doi.org/10.1145/3548606.3560628
20. Couteau, G., Klooß, M., Lin, H., Reichle, M.: Efficient range proofs with transparent setup from bounded integer commitments. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part III. Lecture Notes in Computer Science, vol. 12698, pp. 247–277. Springer (2021). https://doi.org/10.1007/978-3-030-77883-5_9, https://doi.org/10.1007/978-3-030-77883-5_9
21. Dore, D.: Tassle: Lasso for the commitment-phobic. IACR Cryptol. ePrint Arch. p. 1075 (2024), https://eprint.iacr.org/2024/1075
22. Eagen, L., Fiore, D., Gabizon, A.: cq: Cached quotients for fast lookups. IACR Cryptol. ePrint Arch. p. 1763 (2022), https://eprint.iacr.org/2022/1763
23. Eagen, L., Kanjalkar, S., Ruffing, T., Nick, J.: Bulletproofs++: Next generation confidential transactions via reciprocal set membership arguments. In: Joye, M., Leander, G. (eds.) 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 V. Lecture Notes in Computer Science, vol. 14655, pp. 249–279. Springer (2024). https://doi.org/10.1007/978-3-031-58740-5_9, https://doi.org/10.1007/978-3-031-58740-5_9
24. Fu, S.: Improved constant-sized polynomial commitment schemes without trusted setup. IACR Commun. Cryptol. **2**(2), 27 (2025). https://doi.org/10.62056/ANUDY76BM, https://doi.org/10.62056/anudy76bm
25. Gabizon, A., Khovratovich, D.: flookup: Fractional decomposition-based lookups in quasi-linear time independent of table size. IACR Cryptol. ePrint Arch. p. 1447 (2022), https://eprint.iacr.org/2022/1447
26. Gabizon, A., Williamson, Z.J.: plookup: A simplified polynomial protocol for lookup tables. IACR Cryptol. ePrint Arch. p. 315 (2020), https://eprint.iacr.org/2020/315

27. Ganesh, C., Nitulescu, A., Soria-Vazquez, E.: Rinocchio: Snarks for ring arithmetic. J. Cryptol. **36**(4), 41 (2023), https://doi.org/10.1007/s00145-023-09481-3
28. Garreta, A., Waldner, H., Vlasov, I., Hristova, K., Dall'Ava, L., Čupić, M., Klein, M.: Zinc: Succinct arguments with small arithmetization overheads from iops of proximity to the integers. In: Annual International Cryptology Conference. pp. 259–291. Springer (2025)
29. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011. pp. 99–108. ACM (2011), https://doi.org/10.1145/1993636.1993651
30. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: Interactive proofs for muggles. J. ACM **62**(4), 27:1–27:64 (2015), https://doi.org/10.1145/2699436
31. Golovnev, A., Lee, J., Setty, S.T.V., Thaler, J., Wahby, R.S.: Brakedown: Linear-time and field-agnostic snarks for R1CS. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14082, pp. 193–226. Springer (2023). https://doi.org/10.1007/978-3-031-38545-2_7, https://doi.org/10.1007/978-3-031-38545-2_7
32. Haböck, U.: Multivariate lookups based on logarithmic derivatives. IACR Cryptol. ePrint Arch. p. 1530 (2022), https://eprint.iacr.org/2022/1530
33. Hafezi, H., Anthoine, G., Campanelli, M., Fiore, D.: Sok: Lookup table arguments. IACR Cryptol. ePrint Arch. p. 1876 (2025), https://eprint.iacr.org/2025/1876
34. Han, S., Yoon, G., Oh, H., Kim, J.: DUPLEX: scalable zero-knowledge lookup arguments over RSA group. In: Proceedings of the 20th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2025, Hanoi, Vietnam, August 25-29, 2025. pp. 72–86. ACM (2025). https://doi.org/10.1145/3708821.3733863, https://doi.org/10.1145/3708821.3733863
35. Hao, M., Chen, H., Li, H., Weng, C., Zhang, Y., Yang, H., Zhang, T.: Scalable zero-knowledge proofs for non-linear functions in machine learning. IACR Cryptol. ePrint Arch. p. 507 (2025), https://eprint.iacr.org/2025/507
36. Huang, M.M., Mao, X., Zhang, J.: Sublinear proofs over polynomial rings. IACR Cryptol. ePrint Arch. p. 199 (2025), https://eprint.iacr.org/2025/199
37. Jia, Y., Li, S., Xing, C., Yao, Y., Yuan, C.: Polynomial commitments for galois rings and applications to snarks over $\mathbb {Z}_{2^k}$. In: Kalai, Y.T., Kamara, S.F. (eds.) Advances in Cryptology - CRYPTO 2025 - 45th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2025, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 16005, pp. 515–548. Springer (2025). https://doi.org/10.1007/978-3-032-01887-8_17, https://doi.org/10.1007/978-3-032-01887-8_17
38. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: Abe, M. (ed.) Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6477, pp. 177–194. Springer (2010), https://doi.org/10.1007/978-3-642-17373-8_11
39. Kilian, J.: A note on efficient zero-knowledge proofs and arguments (extended abstract). In: Kosaraju, S.R., Fellows, M., Wigderson, A., Ellis, J.A. (eds.) Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May

4-6, 1992, Victoria, British Columbia, Canada. pp. 723–732. ACM (1992), https://doi.org/10.1145/129712.129782
40. Klooß, M., Lai, R.W.F., Nguyen, N.K., Osadnik, M.: Rok and roll - verifier-efficient random projection for $o\tilde{}(\lambda)$-size lattice arguments - (extended abstract). In: Hanaoka, G., Yang, B. (eds.) Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part III. Lecture Notes in Computer Science, vol. 16247, pp. 297–329. Springer (2025). https://doi.org/10.1007/978-981-95-5099-9_10, https://doi.org/10.1007/978-981-95-5099-9_10
41. Kohrita, T., Towa, P., Williamson, Z.J.: One-shot native proofs of non-native operations in incrementally verifiable computations. IACR Cryptol. ePrint Arch. **2024**, 1651 (2024), https://eprint.iacr.org/2024/1651
42. Lipton, R.J.: Efficient checking of computations. In: Choffrut, C., Lengauer, T. (eds.) STACS 90, 7th Annual Symposium on Theoretical Aspects of Computer Science, Rouen, France, February 22-24, 1990, Proceedings. Lecture Notes in Computer Science, vol. 415, pp. 207–215. Springer (1990). https://doi.org/10.1007/3-540-52282-4_44, https://doi.org/10.1007/3-540-52282-4_44
43. Lubarov, D., Melé, J.B.: Casting out primes: Bignum arithmetic for zero-knowledge proofs. IACR Cryptol. ePrint Arch. p. 1470 (2022), https://eprint.iacr.org/2022/1470
44. Lund, C., Fortnow, L., Karloff, H., Nisan, N.: Algebraic methods for interactive proof systems. Journal of the ACM (JACM) **39**(4), 859–868 (1992)
45. Micali, S.: CS proofs (extended abstracts). In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 436–453. IEEE Computer Society (1994), https://doi.org/10.1109/SFCS.1994.365746
46. Papini, S., Haböck, U.: Improving logarithmic derivative lookups using GKR. IACR Cryptol. ePrint Arch. p. 1284 (2023), https://eprint.iacr.org/2023/1284
47. Pollack, P., Schorn, P.: Dirichlet's proof of the three-square theorem: An algorithmic perspective. Math. Comput. **88**(316), 1007–1019 (2019). https://doi.org/10.1090/MCOM/3349, https://doi.org/10.1090/mcom/3349
48. Pollack, P., Treviño, E.: Finding the four squares in lagrange's theorem. Integers **18A**, A15 (2018), http://math.colgate.edu/%7Eintegers/sjs15/sjs15.Abstract.html
49. Posen, J., Kattis, A.A.: Caulk+: Table-independent lookup arguments. IACR Cryptol. ePrint Arch. p. 957 (2022), https://eprint.iacr.org/2022/957
50. Setty, S.T.V., Thaler, J.: Twist and shout: Faster memory checking arguments via one-hot addressing and increments. IACR Cryptol. ePrint Arch. p. 105 (2025), https://eprint.iacr.org/2025/105
51. Setty, S.T.V., Thaler, J., Wahby, R.S.: Unlocking the lookup singularity with lasso. In: Joye, M., Leander, G. (eds.) Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part VI. Lecture Notes in Computer Science, vol. 14656, pp. 180–209. Springer (2024). https://doi.org/10.1007/978-3-031-58751-1_7, https://doi.org/10.1007/978-3-031-58751-1_7
52. Shoup, V.: NTL: a library for doing numbery theory. https://github.com/libntl/ntl

53. Soria-Vazquez, E.: Doubly efficient interactive proofs over infinite and non-commutative rings. In: Kiltz, E., Vaikuntanathan, V. (eds.) Theory of Cryptography - 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13747, pp. 497–525. Springer (2022), https://doi.org/10.1007/978-3-031-22318-1_18
54. Wang, X., Malozemoff, A.J., Katz, J.: EMP-toolkit: Efficient MultiParty computation toolkit. https://github.com/emp-toolkit
55. Wei, Y., Zhang, X., Deng, Y.: Transparent snarks over galois rings. In: Jager, T., Pan, J. (eds.) 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. Lecture Notes in Computer Science, vol. 15674, pp. 418–451. Springer (2025). https://doi.org/10.1007/978-3-031-91820-9_14, https://doi.org/10.1007/978-3-031-91820-9_14
56. Wu, Z., Zhang, X., Deng, Y., Wei, Y., Zhang, Z., Yang, L.: Polylogarithmic polynomial commitment scheme over galois rings. In: Nicomette, V., Benzekri, A., Boulahia-Cuppens, N., Vaidya, J. (eds.) Computer Security - ESORICS 2025 - 30th European Symposium on Research in Computer Security, Toulouse, France, September 22-24, 2025, Proceedings, Part II. Lecture Notes in Computer Science, vol. 16054, pp. 400–420. Springer (2025). https://doi.org/10.1007/978-3-032-07891-9_21, https://doi.org/10.1007/978-3-032-07891-9_21
57. Zapico, A., Buterin, V., Khovratovich, D., Maller, M., Nitulescu, A., Simkin, M.: Caulk: Lookup arguments in sublinear time. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022. pp. 3121–3134. ACM (2022). https://doi.org/10.1145/3548606.3560646, https://doi.org/10.1145/3548606.3560646
58. Zapico, A., Gabizon, A., Khovratovich, D., Maller, M., Ràfols, C.: Baloo: Nearly optimal lookup arguments. IACR Cryptol. ePrint Arch. p. 1565 (2022), https://eprint.iacr.org/2022/1565
59. Zcash Developers: Lookup argument. https://zcash.github.io/halo2/design/proving-system/lookup.html (2025), accessed: 2025-06-23
60. Zhang, Y., Sun, S., Gu, D.: Efficient kzg-based univariate sum-check and lookup argument. In: Tang, Q., Teague, V. (eds.) Public-Key Cryptography - PKC 2024 - 27th IACR International Conference on Practice and Theory of Public-Key Cryptography, Sydney, NSW, Australia, April 15-17, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14602, pp. 400–425. Springer (2024). https://doi.org/10.1007/978-3-031-57722-2_13, https://doi.org/10.1007/978-3-031-57722-2_13

Appendix

A Additional Preliminaries

A.1 SNARKs

Let $\mathcal{R}$ be an efficiently computable binary relation which consists of pairs of the form (x, w) where x is a statement and w is a witness. Let $\mathcal{L}$ be the language associated with the relation $\mathcal{R}$, i.e. $\mathcal{L} = \{x | \exists w, \text{ s.t. } \mathcal{R}(x, w) = 1\}$.

A proof or argument system for $\mathcal{R}$ consists of a triple of PPT algorithms $\Pi = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ defined as follows:

- $\mathsf{Setup}(1^\lambda) \to (\sigma, \mathrm{vk})$: take a security parameter λ and outputs a common (structured) reference string σ together with private verification information vk.
- $\mathsf{Prove}(\sigma, x, w) \to \pi$: on input σ, a statement x and witness w, outputs an argument π.
- $\mathsf{Verify}(\sigma, \mathrm{vk}, x, \pi) \to 1/0$: on input σ, the private verification key vk, a statement x and a proof π, it outputs either 1 indicating accepting the argument or 0 for rejecting it.

Definition 5 (SNARK [27]). *A triple of polynomial time algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *is a SNARK for an NP relation $\mathcal{R}$, if the following properties are satisfied:*

- *Completeness. For all $(x, w) \in \mathcal{R}$, the following holds:*

$$Pr\left[\begin{array}{c}(\sigma, vk) \leftarrow \mathsf{Setup}(1^\lambda); \pi \leftarrow \mathsf{Prove}(\sigma, x, w) : \\ \mathsf{Verify}(\sigma, vk, x, \pi) = 1\end{array}\right] = 1.$$

- *Knowledge Soundness. For any PPT adversary $\mathcal{A}$, there exists a PPT algorithm $\mathcal{E}_\mathcal{A}$ such that the following probability is negligible in λ:*

$$Pr\left[\begin{array}{c}(\sigma, vk) \leftarrow \mathsf{Setup}(1^\lambda); ((\tilde{x}, \tilde{\pi}); w') \leftarrow \mathcal{A} | \mathcal{E}_\mathcal{A}(\sigma) : \\ \mathsf{Verify}(\sigma, vk, \tilde{x}, \tilde{\pi}) = 1 \wedge \mathcal{R}(\tilde{x}, w') = 0\end{array}\right]$$

- *A non-interactive argument of knowledge satisfies knowledge is succinct if the size of proof π and the time to* Verify *are sublinear in the size of the statement proven.*

If a SNARK does not require a private verification key vk, then the SNARK scheme is referred to as a public-key SNARK. If a SNARK does not require a CRS σ, then the SNARK scheme is referred to as transparent.

A.2 Polynomial Commitment Schemes over Rings

Definition 6 (Polynomial Commitment Scheme [31].). *A polynomial commitment scheme for multilinear polynomial over Ring $\mathfrak{R}$ is a tuple of four protocols $PC =$* $(\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$*:*

- $pp \leftarrow \mathsf{Gen}(1^\lambda, \mu)$ *takes as input* μ *(the number of variables in a multilinear polynomial); produces public parameters* pp.
- $\mathcal{C} \leftarrow \mathsf{Commit}(pp, \mathcal{G}, \gamma)$*: takes any input a* $\mu-$*variate multilinear polynomial over a Ring* $\mathcal{G} \in \mathrm{Poly}(\mathfrak{R})$ *and a random string* γ *produces a commitment* $\mathcal{C}$.
- $b \leftarrow \mathsf{Open}(pp, \mathcal{C}, \mathcal{G}, \gamma)$*: verifies the opening of commitment* $\mathcal{C}$ *to the* $\mu-$*variate multilinear polynomial* $\mathcal{G} \in \mathrm{Poly}(\mathfrak{R})$*; outputs* $b \in \{0, 1\}$.
- $b \leftarrow \mathsf{Eval}(pp, \mathcal{C}, z, v, \mu, \mathcal{G})$ *is a protocol between a PPT prover* $\mathcal{P}$ *and verifier* $\mathcal{V}$*. Both* $\mathcal{V}$ *and* $\mathcal{P}$ *hold a commitment* $\mathcal{C}$*, the number of variables* μ*, a scalar* $v \in \mathfrak{R}$ *and* $z \in \mathfrak{R}^\mu$*.* $\mathcal{P}$ *additionally knows a* $\mu-$*variate multilinear polynomial* $\mathcal{G} \in \mathrm{Poly}(\mathfrak{R})$*.* $\mathcal{P}$ *attempts to convince* $\mathcal{V}$ *that* $\mathcal{G}(z) = v$*. At the end of the protocol,* $\mathcal{V}$ *outputs* $b \in \{0, 1\}$.

A tuple of four protocol (Gen,Commit,Open,Eval) is an extractable polynomial commitment scheme for multilinear polynomials over Ring $\mathfrak{R}$ *if the following conditions hold.*

- ***Completeness.*** *For any multilinear polynomial* $\mathcal{G} \in$Poly$(\mathfrak{R})$,

$$\Pr\left[\begin{matrix} pp \rightarrow Gen(1^\lambda, \mu); \mathcal{C} \leftarrow Commit(pp, \mathcal{G}, \gamma) : \\ Eval(pp, \mathcal{C}, r, v, \mu, \mathcal{G}) = 1 \wedge v = \mathcal{G}(r) \end{matrix}\right] \geq 1 - negl(\lambda)$$

- ***Binding.*** *For any PPT adversary* $\mathcal{A}$*, size parameter* $\mu \geq 1$,

$$\Pr\left[\begin{matrix} pp \leftarrow Gen(1^\lambda, \mu); \mathcal{C}, \mathcal{G}_0, \mathcal{G}_1 \leftarrow \mathcal{A}(pp); \\ b_0 \leftarrow Open(pp, \mathcal{C}, \mathcal{G}_0, \gamma); b_1 \leftarrow Open(pp, \mathcal{C}, \mathcal{G}_1, \gamma); \\ b_0 = b_1 \neq 0 \wedge \mathcal{G}_0 \neq \mathcal{G}_1 \end{matrix}\right] \leq negl(\lambda)$$

- ***Knowledge soundness.*** *Eval is a succinct argument of knowledge for the following NP relation given* $pp \leftarrow Gen(1^\lambda, \mu)$.

$$\mathcal{R}_{Eval}(pp) = \{\langle(\mathcal{C}, z, v), (\mathcal{G})\rangle : \mathcal{G} \in GR(p^s, r)[\mu] \wedge \mathcal{G}(z) = v \wedge Open(pp, \mathcal{C}, \mathcal{G}, \gamma) = 1\}$$

There are several works focus on constructing SNARKs over Galois rings [36, 37, 55, 56] and residue rings [15]. These schemes support opening of multilinear polynomials at extension ring elements.

A.3 Sumcheck over Rings

We recall the ring-based sumcheck and ZeroCheck protocols of Cascudo et al. [15]. The sumcheck protocol [44] allows a prover to convince a verifier that the sum of evaluations of a low-degree polynomial $g : \mathfrak{R}^\mu \rightarrow \mathfrak{R}$ over the Boolean hypercube $\{0, 1\}^\mu$ is equal to the claimed value $H \in \mathfrak{R}$.

- P sends H claiming $H = \sum_{x \in \{0,1\}^\mu} g(x)$.
- For $i = 0, \ldots, \mu - 1$, P sends a univariate $g_i(X)$, intended as

$$g_i(X) = \sum_{x_{i+1}, \ldots, x_{\mu-1} \in \{0,1\}} g(r_0, \ldots, r_{i-1}, X, x_{i+1}, \ldots, x_{\mu-1}).$$

- V checks $\deg(g_i) \le \deg_i(g)$ and

$$H = g_0(0) + g_0(1), \qquad g_{i-1}(r_{i-1}) = g_i(0) + g_i(1)\ (i \ge 1),$$

  then samples $r_i \leftarrow S \subseteq R$.
- V queries $g(r_0, \ldots, r_{\mu-1})$ and accepts iff $g(r_0, \ldots, r_{\mu-1}) = g_{\mu-1}(r_{\mu-1})$.

A.4 ZeroCheck over Rings

ZeroCheck [16] is a protocol that allows a prover to convince a verifier that a given low-degree polynomial $g : R^\mu \to R$ is identically zero over the Boolean hypercube $\{0,1\}^\mu$.

- The verifier samples $\mathbf{r} \leftarrow S^\mu$ from an exceptional set $S \subset R$ and sends $\mathbf{r}$ to the prover.
- The prover and verifier execute sumcheck to prove $\sum_{\mathbf{x}\in\{0,1\}^\mu} \mathrm{eq}(\mathbf{r},\mathbf{x})\cdot g(\mathbf{x}) = 0$, where $\mathrm{eq}(\mathbf{r},\mathbf{x})$ is the standard multilinear equality polynomial.

A.5 One-Sided Modular JL Projections

Lemma 4 (One-Side Johnson–Lindenstrauss projection [3]). *Let the parameters be as in Lemma 2. For every $m_{\mathrm{rp}} \in \mathbb{N}$, every $Q \in \mathbb{N}$, and every $\mathbf{w} \in \mathbb{Z}^{m_{\mathrm{rp}}}$ with $\|\mathbf{w}\|_2 > B$ and $B \le Q/c_{\mathrm{rp}}$,*

$$\Pr\left[\|J\mathbf{w} \bmod Q\|_2 \le \alpha_{\mathrm{rp}} B \;\middle|\; J \leftarrow \chi^{n_{\mathrm{rp}}\times m_{\mathrm{rp}}}\right] \le \kappa_{\mathrm{rp}}.$$

Lemma 5 (One-Side Structured Johnson–Lindenstrauss projection [40]). *Let $m_{\mathrm{rp}} \mid M$, let $\mathbf{w} \in \mathbb{Z}^M$, and define $\Pi := I_{M/m_{\mathrm{rp}}} \otimes J$ as in Lemma 3. For every $\mathbf{w} \in \mathbb{Z}^M$ with $\|\mathbf{w}\|_2 > B$ and $B \le Q/c_{\mathrm{rp}}$,*

$$\Pr\left[\|\Pi\mathbf{w} \bmod Q\|_2 \le \alpha_{\mathrm{rp}} B \;\middle|\; J \leftarrow \chi^{n_{\mathrm{rp}}\times m_{\mathrm{rp}}}\right] \le \kappa_{\mathrm{rp}} \frac{M}{m_{\mathrm{rp}}}.$$

B Deferred Proofs

This appendix contains the proofs deferred from the main body.

B.1 Proof of Theorem 1

Proof. Each $a_i^{\boldsymbol{\sigma}}$ exists and is unique by the Chinese Remainder Theorem. For component 0, the projected multiset is always

$$\pi_0(A^{\boldsymbol{\sigma}}) = \{a_{0,0}, \ldots, a_{0,m-1}\}.$$

For component $j \ge 1$, the projection is

$$\pi_j(A^{\boldsymbol{\sigma}}) = \{a_{j,\sigma_j(0)}, \ldots, a_{j,\sigma_j(m-1)}\} = A_j,$$

which is independent of $\boldsymbol{\sigma}$. Thus all $A^{\boldsymbol{\sigma}}$'s are CRT-component equivalent.

It remains to show that different alignments give different multisets. Since the elements $a_{0,0}, \ldots, a_{0,m-1}$ are pairwise distinct, every multiset $A^{\boldsymbol{\sigma}}$ contains a unique element whose first component is $a_{0,i}$. The j-th component of that element is $a_{j,\sigma_j(i)}$. If $A^{\boldsymbol{\sigma}} = A^{\boldsymbol{\tau}}$, then for every i and every $j \geq 1$ we have

$$a_{j,\sigma_j(i)} = a_{j,\tau_j(i)}.$$

Since each A_j is a set, this implies $\sigma_j(i) = \tau_j(i)$ for all i and j. Hence $\boldsymbol{\sigma} = \boldsymbol{\tau}$, proving the claim. □

B.2 Proof of Corollary 1

Proof. For every component j, reducing the two product polynomials modulo Q_j gives

$$\prod_{a \in A} (X - \pi_j(a)) \qquad \text{and} \qquad \prod_{b \in B} (X - \pi_j(b)) \quad \text{in } \mathbb{Z}_{Q_j}[X].$$

These polynomials are equal because $\pi_j(A) = \pi_j(B)$ as multisets. Hence the two product polynomials have the same reduction modulo Q_j for every j. Applying CRT coefficient-wise gives equality in $\mathbb{Z}_Q[X]$. Non-injectivity follows from Theorem 1. □

B.3 Proof of Corollary 2

Proof. CRT-component equivalence implies $|T'| = |T|$. Because $T' \neq T$, there exists $w \in T' \setminus T$. For every j, membership of w in T' gives $\pi_j(w) \in \pi_j(T')$. Since $T' \equiv_{\mathsf{CRT}} T$, we have $\pi_j(T') = \pi_j(T)$, and hence $\pi_j(w) \in \pi_j(T)$. Thus all component-wise membership checks hold even though $w \notin T$. □

B.4 Proof of Fact 1

Proof. Recall that $\mathcal{I}_N = [-\lfloor N/2 \rfloor, \lceil N/2 \rceil) \cap \mathbb{Z}$. Since $P > Q$, we have $\lfloor Q/2 \rfloor \leq \lfloor P/2 \rfloor$ and $\lceil Q/2 \rceil \leq \lceil P/2 \rceil$, and hence $\mathcal{I}_Q \subseteq \mathcal{I}_P$. In particular, the assumption $z \in \mathcal{I}_Q$ implies $z \in \mathcal{I}_P$.

The congruence $x \equiv z \pmod{P}$ implies that $x - z = kP$ for some $k \in \mathbb{Z}$. Since both x and z lie in $\mathcal{I}_P$, whose integer diameter is $P-1$, we have $|x-z| < P$. The only multiple of P with absolute value strictly less than P is zero, so $k = 0$ and therefore $x = z$.

Similarly, $y, z \in \mathcal{I}_Q$, and the congruence $y \equiv z \pmod{Q}$ gives $y - z = \ell Q$ for some $\ell \in \mathbb{Z}$. Because $|y - z| < Q$, we obtain $\ell = 0$, and hence $y = z$. Therefore $x = y = z$; moreover, their common integer value lies in $\mathcal{I}_Q$. □

B.5 Proof of Theorem 2

Proof. For completeness, suppose that $\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b))$ for every $b \in B_\mu$. This common integer lies in $\mathcal{I}_Q$ and is the centered representative of the implicit CRT lift $h(b)$. Hence the range statement is valid.

For knowledge soundness, except with probability δ_{Range}, an accepting range proof establishes $\mathrm{rep}_R(h(b)) \in \mathcal{I}_Q$ for every b. Fact 1 then implies

$$\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b)) \qquad \forall b \in B_\mu.$$

The CMC PIOP adds no step beyond the range-check PIOP, so the complexity bounds are unchanged.

B.6 Proof of Theorem 3

Proof. **Completeness.** Suppose $g(b) \in T(B_\iota)$ for every $b \in B_\mu$. The prover sets

$$f(b) := \mathrm{rep}_Q(g(b)) \bmod P.$$

Then $f(B_\mu) \subseteq \widetilde{T}(B_\iota)$ and $\mathsf{CMC}_{P,Q}(f, g) = 1$. The field lookup is perfectly complete, while the CMC proof accepts with probability at least $1 - \varepsilon_{\mathsf{CMC}}$.
Knowledge soundness. Except with probability $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$, the field lookup extractor outputs an auxiliary witness f such that, for every $b \in B_\mu$, there exists $u_b \in B_\iota$ satisfying

$$f(b) = \widetilde{T}(u_b) = \mathrm{rep}_Q(T(u_b)) \bmod P.$$

Except with probability δ_{CMC}, the CMC extractor also establishes

$$\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b)) \qquad \forall b \in B_\mu.$$

Since $\mathrm{rep}_Q(T(u_b)) \in \mathcal{I}_Q \subseteq \mathcal{I}_P$, the field-lookup equality implies

$$\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(T(u_b)).$$

Hence $\mathrm{rep}_Q(g(b)) = \mathrm{rep}_Q(T(u_b))$, and therefore $g(b) = T(u_b)$ in $\mathbb{Z}_Q$. Thus the extracted witness satisfies the claimed ring lookup relation. The extractor fails with probability at most $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}} + \delta_{\mathsf{CMC}}$.
Complexity. The protocol consists of one auxiliary-field lookup proof and one CMC proof. The stated bounds follow by summing the costs of the two proving layers.

B.7 Proof of Fact 3

Proof. Since $P > Q$, both $D + A$ and $Q - 1$ lie in $[0, R/2) \subseteq \mathcal{I}_R$. Their congruence modulo R therefore implies $D + A = Q - 1$ over the integers. Since $D, A \geq 0$, we have $D \in [0, Q)$. Let $z := D - \tau_Q$. Then

$$z \in [-\lfloor Q/2 \rfloor, \lceil Q/2 \rceil) = \mathcal{I}_Q.$$

The first congruence gives $h \equiv z \pmod{R}$. Because $\mathcal{I}_Q \subseteq \mathcal{I}_R$, uniqueness of the centered representative implies $\mathrm{rep}_R(h) = z \in \mathcal{I}_Q$. □

B.8 Proof of Theorem 5

Proof. **Completeness.** Suppose $\mathsf{CMC}_{P,Q}(f,g) = 1$. For each $b \in B_\mu$, let

$$z_b := \mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b)) \in \mathcal{I}_Q, \quad D_b := z_b + \tau_Q, \quad A_b := Q - 1 - D_b.$$

Then $D_b, A_b \geq 0$. By Fact 2, the prover can choose integers $d_0(b), \ldots, d_3(b)$ and $e_0(b), \ldots, e_3(b)$ such that

$$D_b = \sum_{j=0}^{3} d_j(b)^2, \qquad A_b = \sum_{j=0}^{3} e_j(b)^2.$$

These values define the eight multilinear witness oracles. Both ZeroCheck statements hold, and the initial vector $\mathbf{w}^{(0)}$ satisfies

$$\|\mathbf{w}^{(0)}\|_2^2 = \sum_{b \in B_\mu} (D_b + A_b) = n(Q-1) = B_0^2.$$

Applying Lemma 3 at every projection layer and taking a union bound shows that the final norm check fails with probability at most $\kappa_{\mathrm{JL}}^{4\mathsf{Sq}}$. The two ZeroChecks are perfectly complete, so the protocol has completeness probability at least $1 - \kappa_{\mathrm{JL}}^{4\mathsf{Sq}}$.

Knowledge soundness. We use the standard soundness-to-knowledge lifting for polynomial oracle relations [16, Lemma 2.3]. Since the witness oracles are multilinear and can be recovered from their evaluations on the Boolean hypercube, the same extraction argument applies over $\mathbb{Z}_R$. It therefore suffices to identify the relation extracted from an accepting transcript and bound the probability that it is invalid.

Let the extracted root oracles define the concatenated oracle u from Fig. 3, and set $\mathbf{w}_u^{(0)} := \big(\mathrm{rep}_R(u(s,b))\big)_{(s,b) \in B_3 \times B_\mu}$. Since the eight root oracles are fixed before the verifier samples the projection matrices, $\mathbf{w}_u^{(0)}$ is independent of $J_0, \ldots, J_{L-1}$. For the sampled matrices, define the canonical projection chain determined by u as

$$\mathbf{w}_u^{(i+1)} := \mathrm{rep}_R\Big(\Pi_i \mathbf{w}_u^{(i)} \bmod R\Big), \qquad i = 0, \ldots, L-1.$$

Let $\mathbf{w}_m^{(L)} \in \mathcal{I}_R^{M_L}$ be the final vector sent by the prover, and define the event

$$E := \Big\{ \mathbf{w}_m^{(L)} = \mathbf{w}_u^{(L)} \Big\}.$$

Thus, E is the event that the vector sent by the prover equals the endpoint of the canonical projection chain. We first bound the probability that the verifier accepts the projection-consistency phase when E does not occur. If E does not occur, then the multilinear extensions $\widetilde{w}_m^{(L)}$ and $\widetilde{w}_u^{(L)}$ are distinct. By the generalized Schwartz–Zippel lemma, for $r_L \leftarrow S_R^{m_L}$,

$$\Pr\Big[\widetilde{w}_m^{(L)}(r_L) = \widetilde{w}_u^{(L)}(r_L)\Big] \leq \frac{m_L}{|S_R|}.$$

We next consider the sumcheck at layer i, for $i = L, \ldots, 1$. At the beginning of this layer, the verifier holds a value y_i that purportedly equals $\widetilde{w}_u^{(i)}(r_i)$. Relative to the canonical projection chain, the parties use sumcheck to verify

$$y_i = \sum_{x \in B_{m_{i-1}}} \widetilde{\Pi}_{i-1}(r_i, x)\widetilde{w}_u^{(i-1)}(x).$$

At the end of this sumcheck, the verifier evaluates $\widetilde{\Pi}_{i-1}(r_i, r_{i-1})$ locally, while the prover provides the claimed value y_{i-1} for $\widetilde{w}_u^{(i-1)}(r_{i-1})$. If the layer-$i$ statement is false and the prover provides the correct value of $\widetilde{w}_u^{(i-1)}(r_{i-1})$, then sumcheck knowledge soundness bounds the probability that the verifier accepts by $\kappa^2_{\mathrm{SC},M_{i-1}}$. If the prover instead provides an incorrect value, the false claim is propagated to layer $i-1$; after the last layer, such a false claim is rejected by the oracle-opening check for y_0. When E does not occur and the two final-vector MLEs do not collide at r_L, the initial layer-L claim is false relative to the canonical projection chain. Hence, a union bound over the layers gives

$$\Pr[\neg E \ \wedge \ \mathsf{V} \text{ accepts the projection-consistency phase}] \leq \frac{m_L}{|S_R|} + \sum_{i=0}^{L-1} \kappa^2_{\mathrm{SC},M_i}.$$

Except with the probability above, an accepting transcript therefore satisfies E. Starting from the extracted $\mathbf{w}_u^{(0)}$, the extractor can compute the entire canonical projection chain using the public matrices. Since E occurs and the verifier accepts the final norm check, $\|\mathbf{w}_u^{(L)}\|_2 = \|\mathbf{w}_m^{(L)}\|_2 \leq B_L$.

It remains to derive a norm bound for $\mathbf{w}_u^{(0)}$. Define the backward thresholds

$$C_i := \frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}^{L-i}} B_0, \qquad i = 0, \ldots, L.$$

In particular, $C_L = B_L$ and $C_{i+1} = \alpha_{\mathrm{rp}} C_i$. To bound the probability that the backward inference fails, define the following bad events in the original random experiment, without conditioning on E:

$$\mathcal{B}_i := \left\{ \|\mathbf{w}_u^{(i)}\|_2 > C_i \ \wedge \ \|\mathbf{w}_u^{(i+1)}\|_2 \leq C_{i+1} \right\}, \qquad i = 0, \ldots, L-1.$$

For every i, the vector $\mathbf{w}_u^{(i)}$ depends only on the extracted oracle u and the previously sampled projection matrices. Conditioned on $J_0, \ldots, J_{i-1}$, the vector $\mathbf{w}_u^{(i)}$ is fixed while the independently sampled J_i remains distributed according to χ. Thus, the one-sided structured JL lemma applies to $\mathcal{B}_i$ with threshold C_i.

Since $C_{i+1} = \alpha_{\mathrm{rp}} C_i$ and $\alpha_{\mathrm{rp}} > 1$, the sequence $(C_i)_{i=0}^{L-1}$ is increasing. The first modulus condition gives, for every $i < L$,

$$C_i \leq C_{L-1} = \frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}} B_0 < \frac{R}{c_{\mathrm{rp}}}.$$

Lemma 5 therefore gives

$$\Pr[\mathcal{B}_i] \le \kappa_{\mathrm{rp}} \frac{M_i}{m_{\mathrm{rp}}}.$$

Let $\mathcal{E}_{\mathrm{JL}} := \bigcap_{i=0}^{L-1} \neg\mathcal{B}_i$. A union bound over all layers gives

$$\Pr[\neg\mathcal{E}_{\mathrm{JL}}] \le \kappa_{\mathrm{JL}}^{\mathsf{4Sq}}.$$

If $\mathcal{E}_{\mathrm{JL}}$ occurs, the final norm bound $\|\mathbf{w}_u^{(L)}\|_2 \le C_L$ and the complements of $\mathcal{B}_{L-1}, \dots, \mathcal{B}_0$ imply recursively that

$$\|\mathbf{w}_u^{(0)}\|_2 \le C_0 = \left(\frac{\beta_{\mathrm{rp}}}{\alpha_{\mathrm{rp}}}\right)^L \sqrt{n(Q-1)}.$$

By the second modulus condition,

$$\|\mathbf{w}_u^{(0)}\|_2^2 < R/2. \tag{$*$}$$

Finally, except with probability at most $2\delta^{n,2}_{\mathbb{Z}_R,\mathsf{Zero}}$, the two accepted ZeroChecks establish, for every $b \in B_\mu$, the shifted-reconstruction and complementary four-square identities over $\mathbb{Z}_R$. Fix such a point b, take the centered integer representatives of the extracted root values, and define

$$D_b := \sum_{j=0}^{3} d_j(b)^2, \qquad A_b := \sum_{j=0}^{3} e_j(b)^2.$$

The two identities imply

$$h(b) + \tau_Q \equiv D_b \pmod{R}, \qquad D_b + A_b \equiv Q - 1 \pmod{R}.$$

Moreover, $(*)$ gives

$$D_b + A_b \le \|\mathbf{w}_u^{(0)}\|_2^2 < R/2.$$

Fact 3 therefore yields $\mathrm{rep}_R(h(b)) \in \mathcal{I}_Q$. Applying Fact 1 to the CRT residues $f(b)$ and $g(b)$ shows

$$\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b)).$$

Since this holds for every b, the extracted oracles satisfy $\mathsf{CMC}_{P,Q}(f,g) = 1$. Summing the projection-consistency, JL, and ZeroCheck errors, the extractor fails with probability at most

$$\kappa_{\mathrm{JL}}^{\mathsf{4Sq}} + \frac{m_L}{|S_R|} + \sum_{i=0}^{L-1} \kappa^2_{\mathrm{SC},M_i} + 2\delta^{n,2}_{\mathbb{Z}_R,\mathsf{Zero}} = \delta^{n,Q}_{P,Q,\mathsf{4SqCMC}},$$

which proves the stated knowledge-soundness bound.

Complexity. For each $b \in B_\mu$, the prover computes four-square representations of D_b and $A_b = Q-1-D_b$. Thus, generating all eight root oracles takes expected time

$$O\big(nT_{\mathsf{4Sq}}(Q)\big) = O\left(n\frac{(\log Q)^2}{\log\log Q}\right)$$

basic integer operations, using the unconditional randomized algorithm of Pollack and Treviño [48]. This cost is separate from the algebraic cost of proving the resulting oracle relations.

The concatenated oracle has length $8n$. Since the vector lengths decrease geometrically and the small projection dimensions depend only on λ, the projection layers and their consistency sumchecks require $O(n)$ prover work and $O(\mathrm{poly}(\lambda, \log n))$ verifier work and communication for the JL small-norm check. The protocol additionally invokes two degree-2 ZeroChecks of size n; together they contribute $O(n)$ prover work and $O(\log n)$ verifier work and communication. Only eight auxiliary oracles are used, independently of Q. This proves the stated complexity bounds. □

B.9 Proof of Corollary 3

Proof. The claim follows by combining Theorems 3 and 5; the extra oracles are the auxiliary-field witness and the eight four-square root oracles.

B.10 Proof of Theorem 6

Proof. **Completeness.** If $\mathrm{rep}_P(f(b)) = \mathrm{rep}_Q(g(b))$ for every $b \in B_\mu$, then the auxiliary-field lookup accepts and every folded sum over $\mathbb{F}_P$ is the signed integer lift of the corresponding folded sum over $\mathbb{Z}_Q$. The choice of P places every such integer sum in $\mathcal{I}_P$, so no wraparound occurs in either direction and the verifier's modulo-Q comparisons hold.

Knowledge soundness. As in the PIOP soundness-to-knowledge lifting used in HyperPlonk [16, Lemma 2.3], it suffices to bound the probability that an invalid oracle tuple passes all checks.

Condition on the event that the auxiliary-field lookup is valid. Then $\mathrm{rep}_P(f(b)) \in \mathcal{I}_Q$ for every $b \in B_\mu$. Suppose that there exists $(x^*, y^*) \in B_{\mu/2} \times B_{\mu/2}$ such that

$$\mathrm{rep}_P(f(x^*, y^*)) \not\equiv g(x^*, y^*) \pmod{Q}.$$

For a single folded check, define the polynomial over $\mathbb{Z}_Q$

$$H(r, t) = \sum_{x,y \in B_{\mu/2}} \big(\mathrm{rep}_P(f(x, y)) - g(x, y)\big) r_x t_y,$$

where $r = (r_x)_x$ and $t = (t_y)_y$ range over the verifier's coefficient tables. The modulus condition ensures that the centered representative of the field-side folded sum equals its signed integer lift. Consequently, the verifier's comparison is equivalent to $H(r, t) = 0$ in $\mathbb{Z}_Q$. The coefficient of $r_{x^*} t_{y^*}$ is nonzero in $\mathbb{Z}_Q$, so H is a nonzero polynomial of total degree two. By Lemma 1, because $\mathcal{I}_q$ is an exceptional set in $\mathbb{Z}_Q$ and H has total degree two,

$$\Pr[H(r, t) = 0] \le \frac{2}{q}.$$

Thus k independent folded checks miss an actual inconsistency with probability at most $(2/q)^k$.

The sumcheck protocols may also accept false claims about the folded sums. By a union bound over the k ring sumchecks and the k field sumchecks, this contributes at most

$$k \cdot \delta^{\mu,2}_{\mathbb{Z}_Q,\mathsf{Sum}} + k \cdot \delta^{\mu,2}_{\mathbb{F}_P,\mathsf{Sum}}.$$

If the auxiliary-field lookup itself is false, it is accepted with probability at most $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$. Summing these events gives the stated soundness bound.

Complexity. The protocol runs $2k$ sumcheck instances, each over an n-point hypercube and with individual degree at most two. Their prover cost is $O(kn)$. The verifier evaluates the sampled half-size coefficient tables at the final sumcheck points; this costs $O(\sqrt{n})$ per repetition and dominates the logarithmic verifier work of sumcheck. Hence the folded layer costs $O(kn)$ prover time, $O(k\sqrt{n})$ verifier time, and $O(k \log n)$ proof elements.

The auxiliary-field lookup contributes $O(n+m)$ prover time, $O(\log n + \log m)$ verifier time, and $O(\log n + \log m)$ proof elements. Combining the two layers gives the claimed bounds. □

C RingLUT-Perm: A Permutation-Based LUT PIOP over Residue Rings

This appendix presents the Booleanity PIOP obtained from our norm-certification technique and then uses it to give the complete construction and analysis of RingLUT-Perm.

C.1 Booleanity via Structured JL and GKR-Style Protocols

We begin by formalizing the Booleanity condition used to construct RingLUT-Perm.

Definition 7 (Ring Booleanity). *Let $M = 2^m$, and let $h : \{0,1\}^m \to \mathbb{Z}_Q$. We say that h is Boolean over $\mathbb{Z}_Q$ if, for every $b \in \{0,1\}^m$, $\mathrm{rep}_Q(h(b)) \in \{0,1\}$.*

Let $M = 2^m$, and let $\mathbf{w}^{(0)} := \big(\mathrm{rep}_Q(h(b))\big)_{b\in\{0,1\}^m} \in \mathcal{I}_Q^M$ be the evaluation vector of h, written using centered representatives.

Inspired by lattice-based SNARKs, we observe that in LabRADOR [3], Booleanity over the integers can be enforced by combining arithmetic constraints with a low-norm check. We use the same fact to address a different obstruction: the norm bound rules out CRT-realigned representatives that satisfy the Boolean equation componentwise but do not correspond to bits in $\mathbb{Z}_Q$.

Fact 4 *Let $Q > 2$ and $M = 2^m$. Let $h : \mathbb{Z}_Q^m \to \mathbb{Z}_Q$ be multilinear. If*

$$h(b)(h(b)-1) \equiv 0 \pmod{Q} \qquad \forall b \in \{0,1\}^m$$

and $\|\mathbf{w}^{(0)}\|_2 < \sqrt{Q/2}$, then $\mathrm{rep}_Q(h(b)) \in \{0,1\}$ for every $b \in \{0,1\}^m$.

Proof. Fix $b \in \{0,1\}^m$, and write $z := \mathrm{rep}_Q(h(b)) \in \mathcal{I}_Q$. Since $\|\mathbf{w}^{(0)}\|_2 < \sqrt{Q/2}$, we have $|z| < \sqrt{Q/2}$. The modular equation implies that Q divides $z(z-1)$, while $|z(z-1)| < \sqrt{Q/2} \cdot (\sqrt{Q/2}+1) < Q$. Hence $z(z-1) = 0$ over the integers, and so $\mathrm{rep}_Q(h(b)) = z \in \{0,1\}$. □

The Boolean equation itself can be checked by a standard ZeroCheck over $\mathbb{Z}_Q$. To certify $\|\mathbf{w}^{(0)}\|_2 < \sqrt{Q/2}$ succinctly, we instantiate the structured-JL norm-certification framework of Subsection 5.2 for the evaluation vector of h. The prover sends only the final projected vector. As in the four-square construction, one sumcheck per projection layer reduces an evaluation claim on $\widetilde{w}^{(i)}$ to one on $\widetilde{w}^{(i-1)}$; chaining these reductions leaves a single opening claim for the committed polynomial h. The repeated-block form $\Pi_i = I \otimes J_i$ allows the verifier to evaluate each transition succinctly.

For simplicity, we assume that all projection lengths are powers of two and that $m_{\mathrm{rp}} \mid M_i$ at every layer. Arbitrary lengths can be handled by padding with zeros. As in the main construction, the projection-consistency sumchecks are run over an extension ring $\mathbb{Z}_Q^{(k)} := \mathbb{Z}_Q[X]/(F(X))$. Let S denote its exceptional set. If F remains irreducible modulo every prime divisor $p \mid Q$, then $|S| = q^k$, where q is the smallest prime divisor of Q. Such a polynomial can be obtained by choosing irreducible degree-k polynomials over each $\mathbb{F}_p$ and combining their coefficients via the CRT. The resulting protocol is given in Fig. 7.

The recursive projection chain accumulates both the JL failure probability and the norm distortion across its L layers. The following theorems make these losses explicit in the completeness and knowledge-soundness bounds.

Theorem 7 (Booleanity completeness). *Let $M = 2^m$, let $M_0 = M$, and let $M_{i+1} = M_i n_{\mathrm{rp}}/m_{\mathrm{rp}}$ for $i = 0, \ldots, L-1$, where $m_{\mathrm{rp}} \mid M_i$. Define $B_0 := \sqrt{M}$ and $B_i := \beta_{\mathrm{rp}}^i B_0$ for $i = 1, \ldots, L$. Set $\kappa_{\mathrm{JL}} := \sum_{i=0}^{L-1} \kappa_{\mathrm{rp}} \cdot M_i/m_{\mathrm{rp}}$. Assume $Q > c_{\mathrm{rp}} B_{L-1}$. Then, the protocol in Figure 7 satisfies completeness with probability at least $1 - \kappa_{\mathrm{JL}}$.*

Proof. If h is a Boolean multilinear polynomial, then its ℓ_2-norm is at most $\sqrt{M}$, where $M = 2^m$ is the size of the hypercube. According to Lemma 3, after each projection, the probability that the ℓ_2-norm of the resulting vector exceeds B_i is at most $\kappa_{\mathrm{rp}} \cdot M_i/m_{\mathrm{rp}}$. By a union bound, after L projections, the probability that the ℓ_2-norm of the resulting vector exceeds B_L is at most $\sum_{i=0}^{L-1} \kappa_{\mathrm{rp}} \cdot M_i/m_{\mathrm{rp}} = \kappa_{\mathrm{JL}}$. Therefore, the probability that the ℓ_2-norm of $\mathbf{w}^{(L)}$ exceeds B_L is at most κ_{JL}.

All other components of the protocol have perfect completeness, since both sumcheck and ZeroCheck are complete. Consequently, the completeness probability of the entire protocol is at least $1 - \kappa_{\mathrm{JL}}$. □

Theorem 8 (Booleanity knowledge soundness). *Let $M = 2^m$, let $M_0 = M$, and let $M_{i+1} = M_i n_{\mathrm{rp}}/m_{\mathrm{rp}}$ for $i = 0, \ldots, L-1$, where $m_{\mathrm{rp}} \mid M_i$. Let $m_i := \log M_i$, and let S be the exceptional set used in the projection-consistency phase. Define*

$$B_0 := \sqrt{M}, \qquad B_i := \beta_{\mathrm{rp}}^i B_0 \quad (i = 1, \ldots, L),$$

Booleanity PIOP over $\mathbb{Z}_Q$.

- **Prover input:** A multilinear polynomial $h : \mathbb{Z}_Q^m \to \mathbb{Z}_Q$.
- **Verifier input:** Oracle access to $[[h]]_Q$.

Projection. Set $M_0 = M = 2^m$, $m_i = \log M_i$, and $\mathbf{w}^{(0)} = (\mathrm{rep}_Q(h(b)))_{b\in\{0,1\}^m}$. For $i = 0, \ldots, L-1$, define $M_{i+1} = M_i n_{\mathrm{rp}}/m_{\mathrm{rp}}$, assuming $m_{\mathrm{rp}} \mid M_i$.

- V samples $J_i \leftarrow \chi^{n_{\mathrm{rp}}\times m_{\mathrm{rp}}}$ and both parties set $\Pi_i := I_{M_i/m_{\mathrm{rp}}} \otimes J_i$.
- P computes $\mathbf{w}^{(i+1)} := \mathrm{rep}_Q(\Pi_i \mathbf{w}^{(i)} \bmod Q)$, where rep_Q is applied coordinate-wise. After round $L-1$, P sends $\mathbf{w}^{(L)}$ to V.
- V rejects unless $\|\mathbf{w}^{(L)}\|_2 \le B_L := \beta_{\mathrm{rp}}^L \sqrt{M}$.

Projection consistency. Let $\widetilde{w}^{(i)}$ and $\widetilde{\Pi}_i$ denote the multilinear extensions of $\mathbf{w}^{(i)}$ and Π_i. Let S be the exceptional set of the extension ring $\mathbb{Z}_Q^{(k)}$.

- V samples $r_L \leftarrow S^{m_L}$ and computes $y_L := \widetilde{w}^{(L)}(r_L)$ from the explicit vector $\mathbf{w}^{(L)}$.
- For $i = L, L-1, \ldots, 1$, P and V run sumcheck for

$$y_i = \sum_{x\in\{0,1\}^{m_{i-1}}} \widetilde{\Pi}_{i-1}(r_i, x)\widetilde{w}^{(i-1)}(x).$$

  The reduction outputs $r_{i-1} \leftarrow S^{m_{i-1}}$ and a new claim $y_{i-1} := \widetilde{w}^{(i-1)}(r_{i-1})$; V evaluates $\widetilde{\Pi}_{i-1}(r_i, r_{i-1})$ locally and checks the terminal sumcheck value against $\widetilde{\Pi}_{i-1}(r_i, r_{i-1}) \cdot y_{i-1}$.
- P opens $\widetilde{w}^{(0)}(r_0)$. V checks that the opening equals y_0 and is consistent with $[[h]]_Q$.

Boolean equation.

- P and V run ZeroCheck to prove $h(b)(h(b)-1) \equiv 0 \pmod Q$ for every $b \in \{0,1\}^m$.

Fig. 7: Booleanity PIOP over $\mathbb{Z}_Q$.

and set

$$\kappa_{\mathrm{JL}} := \sum_{i=0}^{L-1} \kappa_{\mathrm{rp}} \frac{M_i}{m_{\mathrm{rp}}}.$$

Assume

$$Q > \max\left\{ c_{\mathrm{rp}} \frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}} \sqrt{M},\ 2\left(\frac{\beta_{\mathrm{rp}}}{\alpha_{\mathrm{rp}}}\right)^{2L} M \right\}.$$

Assume that the size-N, degree-d sumcheck and ZeroCheck protocols have knowledge-soundness errors at most $\kappa_{\mathrm{SC},N}^d$ and $\kappa_{\mathrm{ZC},N}^d$, respectively. Then the protocol in Figure 7 has knowledge-soundness error at most

$$\kappa_{\mathrm{JL}} + \frac{m_L}{|S|} + \sum_{i=0}^{L-1} \kappa_{\mathrm{SC},M_i}^2 + \kappa_{\mathrm{ZC},M}^2.$$

Proof. Let

$$\mathbf{w}_h^{(0)} := \left(\mathrm{rep}_Q(h(b))\right)_{b\in\{0,1\}^m}.$$

Since the witness oracle h is fixed before the verifier samples the projection matrices, $\mathbf{w}_h^{(0)}$ is independent of $J_0, \ldots, J_{L-1}$. For the sampled projection matrices, define the corresponding projection chain by

$$\mathbf{w}_h^{(i+1)} := \mathrm{rep}_Q\Big(\Pi_i \mathbf{w}_h^{(i)} \bmod Q\Big), \qquad i = 0, \ldots, L-1.$$

Let $\mathbf{w}_m^{(L)}$ be the vector sent by the prover, and define the event

$$E := \Big\{\mathbf{w}_m^{(L)} = \mathbf{w}_h^{(L)}\Big\}.$$

Thus, E is the event that the vector sent by the prover equals the endpoint of the projection chain determined by h and the sampled matrices. We first bound the probability that the verifier accepts the projection-consistency phase when E does not occur. If E does not occur, then $\mathbf{w}_m^{(L)} \neq \mathbf{w}_h^{(L)}$, and hence their multilinear extensions $\widetilde{w}_m^{(L)}$ and $\widetilde{w}_h^{(L)}$ are distinct. By the generalized Schwartz–Zippel lemma, for a uniformly sampled $r_L \leftarrow S^{m_L}$,

$$\Pr\Big[\widetilde{w}_m^{(L)}(r_L) = \widetilde{w}_h^{(L)}(r_L)\Big] \leq \frac{m_L}{|S|}.$$

We next consider the sumcheck statement at layer i, for $i = L, \ldots, 1$. At the beginning of this layer, the verifier holds a value y_i that purportedly equals $\widetilde{w}_h^{(i)}(r_i)$, where $r_i \in S^{m_i}$. Relative to the correct projection chain, the parties use sumcheck to verify

$$y_i = \sum_{x \in \{0,1\}^{m_{i-1}}} \widetilde{\Pi}_{i-1}(r_i, x)\widetilde{w}_h^{(i-1)}(x).$$

At the end of the sumcheck, the verifier evaluates $\widetilde{\Pi}_{i-1}(r_i, r_{i-1})$ locally, while the prover provides the claimed value for $\widetilde{w}_h^{(i-1)}(r_{i-1})$. If the layer-$i$ statement is false and the prover provides the correct value of $\widetilde{w}_h^{(i-1)}(r_{i-1})$, then sumcheck knowledge soundness bounds the probability that the verifier accepts this false statement by $\kappa^2_{\mathrm{SC},M_{i-1}}$. If the prover instead provides an incorrect value, the false claim is propagated to layer $i-1$. Therefore, if the layer-L statement is false, a union bound over all layers bounds the probability that the verifier accepts the GKR-style consistency check by $\sum_{i=0}^{L-1} \kappa^2_{\mathrm{SC},M_i}$.

When E does not occur, either

$$\widetilde{w}_m^{(L)}(r_L) = \widetilde{w}_h^{(L)}(r_L),$$

or

$$y_L = \widetilde{w}_m^{(L)}(r_L) \neq \widetilde{w}_h^{(L)}(r_L) = \sum_{x \in \{0,1\}^{m_{L-1}}} \widetilde{\Pi}_{L-1}(r_L, x)\widetilde{w}_h^{(L-1)}(x).$$

In the latter case, the layer-L sumcheck statement is false with respect to the correct projection chain. If the prover supplies an incorrect evaluation of the

preceding-layer polynomial, the false claim is propagated to layer $L-1$. Combining the two bounds above gives

$$\Pr\left[\neg E \ \wedge \ \mathsf{V} \text{ accepts the projection-consistency phase}\right] \leq \frac{m_L}{|S|} + \sum_{i=0}^{L-1} \kappa_{\mathrm{SC},M_i}^2.$$

Except with the probability above, an accepting transcript satisfies E. We now consider such an accepting transcript. The extractor, with oracle access to h, queries its evaluations on the Boolean hypercube and recovers $\mathbf{w}_h^{(0)}$. These evaluations uniquely determine the multilinear witness polynomial h over $\mathbb{Z}_Q$. Starting from this extracted vector, it uses the public projection matrices to compute $\mathbf{w}_h^{(1)}, \ldots, \mathbf{w}_h^{(L)}$ recursively as

$$\mathbf{w}_h^{(i+1)} := \mathrm{rep}_Q\left(\Pi_i \mathbf{w}_h^{(i)} \bmod Q\right), \qquad i = 0, \ldots, L-1.$$

The extractor therefore obtains the entire projection chain from $\mathbf{w}_h^{(0)}$; the proof does not require the prover to send or commit to any intermediate vector. Since E occurs and the verifier accepts the transcript, the verifier's final norm check gives $\|\mathbf{w}_h^{(L)}\|_2 = \|\mathbf{w}_m^{(L)}\|_2 \leq B_L$.

It remains to derive a norm bound for the extracted vector $\mathbf{w}_h^{(0)}$ from this final bound. Define the soundness thresholds

$$C_i := \frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}^{L-i}} \sqrt{M}, \qquad i = 0, \ldots, L.$$

In particular, $C_L = B_L$ and $C_{i+1} = \alpha_{\mathrm{rp}} C_i$. To bound the probability that the backward inference fails, we define the following bad events in the original random experiment, without conditioning on E. For each $i = 0, \ldots, L-1$, let

$$\mathcal{B}_i := \left\{ \|\mathbf{w}_h^{(i)}\|_2 > C_i \ \wedge \ \|\mathbf{w}_h^{(i+1)}\|_2 \leq C_{i+1} \right\}.$$

We apply Lemma 5 to $\mathcal{B}_i$ with input vector $\mathbf{w}_h^{(i)}$ and threshold C_i. By definition of the honest projection chain, $\mathbf{w}_h^{(i)}$ depends only on h and $J_0, \ldots, J_{i-1}$. Since h is fixed before the projection matrices are sampled and the matrices are sampled independently, conditioned on $J_0, \ldots, J_{i-1}$, the vector $\mathbf{w}_h^{(i)}$ is fixed while J_i remains distributed according to χ. Revealing all projection matrices simultaneously does not affect this distributional argument.

It remains to verify that $C_i \leq Q/c_{\mathrm{rp}}$. Since $C_{i+1} = \alpha_{\mathrm{rp}} C_i$ and $\alpha_{\mathrm{rp}} > 1$, the sequence $(C_i)_{i=0}^{L-1}$ is increasing. The modulus condition therefore gives, for every $i = 0, \ldots, L-1$,

$$C_i \leq C_{L-1} = \frac{\beta_{\mathrm{rp}}^L}{\alpha_{\mathrm{rp}}} \sqrt{M} < \frac{Q}{c_{\mathrm{rp}}}.$$

Lemma 5 therefore gives

$$\Pr[\mathcal{B}_i] \leq \kappa_{\mathrm{rp}} \frac{M_i}{m_{\mathrm{rp}}}.$$

Let $\mathcal{E}_{\mathrm{JL}} := \bigcap_{i=0}^{L-1} \neg\mathcal{B}_i$. A union bound over all layers gives

$$\Pr[\neg\mathcal{E}_{\mathrm{JL}}] \le \kappa_{\mathrm{JL}}.$$

If $\mathcal{E}_{\mathrm{JL}}$ also occurs, then the final norm bound above gives $\|\mathbf{w}_h^{(L)}\|_2 \le B_L = C_L$. The complement of $\mathcal{B}_{L-1}$ then gives $\|\mathbf{w}_h^{(L-1)}\|_2 \le C_{L-1}$. Applying the same argument backwards for $i = L-2, \ldots, 0$ yields

$$\|\mathbf{w}_h^{(0)}\|_2 \le C_0 = \left(\frac{\beta_{\mathrm{rp}}}{\alpha_{\mathrm{rp}}}\right)^L \sqrt{M} < \sqrt{Q/2},$$

where the strict inequality follows from the second part of the modulus condition.

Finally, except with probability at most $\kappa_{\mathrm{ZC},M}^2$, the accepted ZeroCheck establishes

$$h(b)(h(b)-1) \equiv 0 \pmod{Q} \qquad \forall b \in \{0,1\}^m.$$

Together with the norm bound above, Fact 4 implies that $\mathrm{rep}_Q(h(b)) \in \{0,1\}$ for every $b \in \{0,1\}^m$. Therefore, outside the failure events above, the extractor outputs a multilinear polynomial h satisfying

$$\mathrm{rep}_Q(h(b)) \in \{0,1\} \qquad \forall b \in \{0,1\}^m.$$

Hence, h is a valid witness for the Booleanity oracle relation. Summing the projection-consistency, JL, and ZeroCheck errors, the extractor fails with probability at most

$$\kappa_{\mathrm{JL}} + \frac{m_L}{|S|} + \sum_{i=0}^{L-1} \kappa_{\mathrm{SC},M_i}^2 + \kappa_{\mathrm{ZC},M}^2,$$

which proves the stated knowledge-soundness bound. □

Lemma 6 (Booleanity PIOP complexity). *Let $M = 2^m$. The JL parameters n_{rp} and m_{rp} depend only on the security parameter λ, and each projection layer shrinks the vector length by the constant factor $\rho := n_{\mathrm{rp}}/m_{\mathrm{rp}} < 1$. Let $M_i = M\rho^i$, and choose L such that $M_L = \mathrm{poly}(\lambda)$. Assume that the ZeroCheck for $h(h-1) = 0$ has linear prover complexity and logarithmic verifier and communication complexity. Then the Booleanity PIOP in Fig. 7, excluding oracle commitment and opening costs, has:*

- *prover complexity $O(M)$ ring operations;*
- *verifier complexity $O(\mathrm{poly}(\lambda, \log M))$ ring operations;*
- *communication complexity $O(\mathrm{poly}(\lambda, \log M))$ ring elements.*

Proof. The projection phase computes $\mathbf{w}^{(i+1)} = (I_{M_i/m_{\mathrm{rp}}} \otimes J_i)\mathbf{w}^{(i)}$ for $i = 0, \ldots, L-1$. Since n_{rp} and m_{rp} are fixed once λ is fixed, the prover spends $O(M_i)$ ring operations in layer i. The same bound covers the prover work in the corresponding projection-consistency sumcheck. Because the lengths shrink geometrically,

$$\sum_{i=0}^{L-1} M_i = M \sum_{i=0}^{L-1} \rho^i = O(M).$$

Together with the linear ZeroCheck for $h(h-1)=0$, this gives total prover complexity $O(M)$.

The verifier receives the final vector $\mathbf{w}^{(L)}$, checks its norm, and evaluates its multilinear extension at r_L; this costs $\mathrm{poly}(\lambda)$ because $M_L = \mathrm{poly}(\lambda)$. Each projection-consistency sumcheck has $m_i = \log M_i$ rounds, and evaluating the structured transition $\widetilde{\Pi}_i$ costs $\mathrm{poly}(\lambda, \log M)$ using the factorization $I \otimes J_i$. Since $L = O(\log M)$, all verifier work is bounded by $O(\mathrm{poly}(\lambda, \log M))$. The final ZeroCheck contributes only $O(\log M)$ verifier work, so it is absorbed in the same bound.

The communication consists of three parts. First, the prover sends the final vector $\mathbf{w}^{(L)}$, which contains $M_L = \mathrm{poly}(\lambda)$ ring elements. Second, in each projection-consistency sumcheck, the prover sends one constant-degree univariate polynomial per round, and the verifier returns one challenge per round. Thus layer i communicates $O(m_i)$ ring elements, and over all layers this is

$$O\left(\sum_{i=0}^{L-1} m_i\right) \subseteq O(\mathrm{poly}(\lambda, \log M)).$$

Third, the ZeroCheck for $h(h-1)=0$ communicates $O(\log M)$ ring elements. Summing these three terms gives the stated communication complexity.

C.2 The RingLUT-Perm Construction

We give the complete permutation-check based lookup argument over $\mathbb{Z}_Q$. Let $n = 2^\mu$ be the number of queried witness values and let $m = 2^\iota$ be the table size. The committed multilinear polynomials are

$$g : B_\mu \to \mathbb{Z}_Q, \qquad T : B_\iota \to \mathbb{Z}_Q,$$

where g encodes the queried values and T encodes the table. The lookup relation asks the prover to show that there exists a map

$$\rho : B_\mu \to B_\iota$$

such that $g(x) = T(\rho(x))$ for every $x \in B_\mu$.

The prover commits to the coordinate functions $\rho_0, \ldots, \rho_{\iota-1} : B_\mu \to \mathbb{Z}_Q$. Over a field, one would certify well-formedness of ρ by checking $\rho_i(\rho_i - 1) = 0$. When the modulus admits a nontrivial CRT decomposition, this check is vulnerable to the CRT alignment ambiguity, so each coordinate is certified with the Booleanity PIOP of Fig. 7. Once these checks pass, the extracted coordinates define a genuine map into B_ι.

For $x \in B_\mu$ and $y \in B_\iota$, define the selector indicator

$$\mathbf{1}_\rho(x, y) := \prod_{i=0}^{\iota-1} \mathrm{eq}(\rho_i(x), y_i).$$

Then the lookup relation is equivalent to

$$g(x) = \sum_{y \in B_\iota} \mathbf{1}_\rho(x, y) \cdot T(y) \qquad \forall x \in B_\mu.$$

The verifier checks a random linear combination of these identities. It samples $s \leftarrow S^\mu$, where $S \subseteq \mathbb{Z}_Q$ is the exceptional set used for sumcheck, and reduces the claim to

$$\sum_{x \in B_\mu} \mathrm{eq}(x, s) g(x) = \sum_{x \in B_\mu} \sum_{y \in B_\iota} \mathrm{eq}(x, s) \mathbf{1}_\rho(x, y) T(y).$$

Let

$$\phi(y) := \sum_{x \in B_\mu} \mathbf{1}_\rho(x, y) \, \mathrm{eq}(x, s).$$

Since ρ is a function, the table of ϕ on B_ι can be computed in $O(n)$ time by scanning all $x \in B_\mu$ and accumulating the weight $\mathrm{eq}(x, s)$ at the address $\rho(x)$. The prover then uses sumcheck to prove $\sum_{y \in B_\iota} \phi(y) T(y) = A$, where A is the value also claimed for $\sum_x \mathrm{eq}(x, s) g(x)$. At the end of this sumcheck, the verifier opens $T(r_y)$ and receives a claim for $\phi(r_y)$.

It remains to verify the claimed value $\phi(r_y)$. Directly expanding the indicator gives

$$\phi(r_y) = \sum_{x \in B_\mu} \mathrm{eq}(x, s) \prod_{i=0}^{\iota - 1} \mathrm{eq}(\rho_i(x), (r_y)_i),$$

which has degree $\iota + 1$. Following Bünz et al. [9], we partition the ι coordinates into ℓ intervals $I_0, \ldots, I_{\ell-1}, |I_j| = \kappa, \ell\kappa = \iota$. For simplicity, we assume that ℓ and κ are powers of two and that $\ell\kappa = \iota$. Otherwise, we pad the coordinate list with dummy coordinates, continue to denote its length by ι, and choose $\ell, \kappa = \Theta(\sqrt{\iota})$. For each interval, define $p_j(x) := \prod_{i \in I_j} \mathrm{eq}(\rho_i(x), (r_y)_i)$.

The verifier first checks $\phi(r_y) = \sum_{x \in B_\mu} \mathrm{eq}(x, s) \prod_{j=0}^{\ell-1} p_j(x)$ by a degree-$(\ell+1)$ sumcheck. At the end, the verifier obtains a random point r_x and receives the claimed values $p_j(r_x)$ for all $j \in [\ell]$.

The last task is to check these ℓ group-product claims. We batch them using another random point $t \leftarrow S^{\log \ell}$. Let $\langle j \rangle \in B_{\log \ell}$ denote the binary encoding of j, and define

$$p(\langle j \rangle, x) := \prod_{i \in I_j} \mathrm{eq}(\rho_i(x), (r_y)_i).$$

The parties run one sumcheck for

$$\sum_{j \in B_{\log \ell}} \sum_{x \in B_\mu} \mathrm{eq}(\langle j \rangle, t) \, \mathrm{eq}(r_x, x) p(\langle j \rangle, x) = \sum_{j \in B_{\log \ell}} \mathrm{eq}(\langle j \rangle, t) p_j(r_x).$$

This sumcheck has degree $\kappa + 1$ and domain size ℓn. We use the bucketing algorithm of Bünz et al. [9, Algorithm 6] to compute the prover messages for

this batched sumcheck in $O(\kappa^2 n)$ time, rather than the direct $O(\kappa^2 \ell n)$ time. At the terminal point (r'_t, r'_x), the verifier opens $\rho_0(r'_x), \ldots, \rho_{\iota-1}(r'_x)$ and computes

$$p(r'_t, r'_x) = \sum_{b \in B_{\log \ell}} \mathrm{eq}(b, r'_t) \prod_{i \in I_b} \mathrm{eq}(\rho_i(r'_x), (r_y)_i).$$

The full protocol is summarized in Fig. 8.

Theorem 9 (RingLUT-Perm). *Let $Q > 2$, $n = 2^\mu$, and $m = 2^\iota$. Let $T : B_\iota \to \mathbb{Z}_Q$ be the lookup table and $g : B_\mu \to \mathbb{Z}_Q$ be the witness function. Assume:*

- *the Booleanity PIOP of Fig. 7 for size n has completeness probability at least $1 - \varepsilon^n_{\mathbb{Z}_Q,\mathsf{Bool}}$ and knowledge-soundness error $\delta^n_{\mathbb{Z}_Q,\mathsf{Bool}}$;*
- *the sumcheck argument over $\mathbb{Z}_Q$ for a size-N, individual-degree-d instance has perfect completeness and knowledge-soundness error $\delta^{N,d}_{\mathbb{Z}_Q,\mathsf{Sum}}$, and its challenges are sampled from an exceptional set S.*

Then the protocol in Fig. 8:

1. *has completeness probability at least $1 - \iota\varepsilon^n_{\mathbb{Z}_Q,\mathsf{Bool}}$;*
2. *has knowledge-soundness error at most*

$$\iota\delta^n_{\mathbb{Z}_Q,\mathsf{Bool}} + \frac{\mu}{|S|} + \delta^{n,2}_{\mathbb{Z}_Q,\mathsf{Sum}} + \delta^{m,2}_{\mathbb{Z}_Q,\mathsf{Sum}} + \delta^{n,\ell+1}_{\mathbb{Z}_Q,\mathsf{Sum}} + \delta^{\ell n,\kappa+1}_{\mathbb{Z}_Q,\mathsf{Sum}},$$

   where $\ell\kappa = \iota$ after padding if necessary;
3. *has PIOP complexity, excluding oracle commitment costs:*
   - *prover complexity $O(\iota n + m + \ell^2 n + \kappa^2 n)$;*
   - *verifier complexity $O(\mathrm{poly}(\lambda, \log n, \log m))$;*
   - *proof size $O(\mathrm{poly}(\lambda, \log n, \log m))$ ring elements.*

In particular, choosing $\ell, \kappa = \Theta(\sqrt{\iota})$ gives prover complexity $O(\iota n + m)$.

RingLUT-Perm: permutation-check based lookup over rings.

- **Common input:** oracle commitments $[[g]]_Q, [[T]]_Q$ for multilinear polynomials $g : B_\mu \to \mathbb{Z}_Q$ and $T : B_\iota \to \mathbb{Z}_Q$.
- **Statement:** there exists $\rho : B_\mu \to B_\iota$ such that $g(x) = T(\rho(x))$ for all $x \in B_\mu$.
- **Auxiliary oracles:** P commits to multilinear polynomials $\rho_i : B_\mu \to \mathbb{Z}_Q$ for $i \in [\iota]$, intended to be the coordinates of ρ.

Booleanity. P and V run the Booleanity protocol of Fig. 7 for each ρ_i.

Outer sumchecks. V samples $s \leftarrow S^\mu$. P sends A. The parties run sumcheck to prove $A = \sum_{x \in B_\mu} \mathrm{eq}(x, s) g(x)$, with a final opening of g. The prover computes $\phi(y) = \sum_x \mathbf{1}_\rho(x, y) \, \mathrm{eq}(x, s)$, and the parties run a second sumcheck to prove $A = \sum_{y \in B_\iota} \phi(y) T(y)$. At the end, the verifier opens $T(r_y)$ and receives the claimed value $\phi(r_y)$.

Grouped indicator sumcheck. After padding with dummy coordinates if necessary, partition $[\iota]$ into ℓ intervals $I_0, \ldots, I_{\ell-1}$ of size $\kappa = \iota/\ell$. With

$$p_j(x) = \prod_{i \in I_j} \mathrm{eq}(\rho_i(x), (r_y)_i),$$

the parties run sumcheck to prove

$$\phi(r_y) = \sum_{x \in B_\mu} \mathrm{eq}(x, s) \prod_{j=0}^{\ell-1} p_j(x).$$

At the end, P sends $p_j(r_x)$ for every $j \in [\ell]$.

Batched group-product check. V samples $t \leftarrow S^{\log \ell}$. The parties run one batched sumcheck to prove

$$\sum_{j \in B_{\log \ell}} \sum_{x \in B_\mu} \mathrm{eq}(\langle j \rangle, t) \, \mathrm{eq}(r_x, x) p(\langle j \rangle, x) = \sum_{j \in B_{\log \ell}} \mathrm{eq}(\langle j \rangle, t) p_j(r_x),$$

where $p(\langle j \rangle, x) = \prod_{i \in I_j} \mathrm{eq}(\rho_i(x), (r_y)_i)$. The prover computes the sumcheck messages using the bucketing algorithm of Bünz et al. [9, Algorithm 6]. At the terminal point, V opens the required ρ_i's and evaluates the claimed terminal value locally.

Fig. 8: RingLUT-Perm over $\mathbb{Z}_Q$.

Proof. Each sumcheck is perfectly complete. The protocol invokes ι Booleanity instances, each of which accepts an honest Boolean oracle with probability at least $1 - \varepsilon^n_{\mathbb{Z}_Q, \mathsf{Bool}}$. A union bound therefore gives completeness probability at least $1 - \iota \varepsilon^n_{\mathbb{Z}_Q, \mathsf{Bool}}$.

The knowledge-soundness error is the sum of the errors of the subprotocols. We prove knowledge soundness by a union bound over the subprotocols. First, except with probability at most $\iota \delta^n_{\mathbb{Z}_Q, \mathsf{Bool}}$, all extracted coordinates ρ_i are Boolean on B_μ, and therefore define a map $\rho : B_\mu \to B_\iota$. Conditioned on this event, the two outer sumchecks establish the same claimed value A for

$$\sum_{x \in B_\mu} \mathrm{eq}(x, s) g(x) \quad \text{and} \quad \sum_{y \in B_\iota} \phi(y) T(y),$$

except with errors $\delta^{n,2}_{\mathbb{Z}_Q,\mathsf{Sum}}$ and $\delta^{m,2}_{\mathbb{Z}_Q,\mathsf{Sum}}$. The grouped indicator sumcheck then establishes the claimed value of $\phi(r_y)$, except with error $\delta^{n,\ell+1}_{\mathbb{Z}_Q,\mathsf{Sum}}$. Finally, the batched group-product check establishes that the values $p_j(r_x)$ are consistent with the committed ρ_i's, except with error $\delta^{\ell n,\kappa+1}_{\mathbb{Z}_Q,\mathsf{Sum}}$.

When all these checks are sound, the verifier has certified

$$\sum_{x\in B_\mu} \mathrm{eq}(x,s)\left(g(x)-\sum_{y\in B_\iota}\mathbf{1}_\rho(x,y)T(y)\right)=0.$$

Since the ρ_i's are Boolean, the inner sum is exactly $T(\rho(x))$. Define the multilinear polynomial

$$F(Z_0,\dots,Z_{\mu-1}) := \sum_{x\in B_\mu} \mathrm{eq}(x,Z)\big(g(x)-T(\rho(x))\big).$$

If the lookup relation is false, its evaluation table contains a nonzero entry, so F is a nonzero multilinear polynomial of total degree at most μ. The generalized Schwartz–Zippel lemma therefore gives

$$\Pr_{s\leftarrow S^\mu}[F(s)=0]\le\frac{\mu}{|S|}.$$

Adding this random-compression error to the errors of the extracted Booleanity and sumcheck subprotocols gives the stated knowledge-soundness bound.

For complexity, the ι Booleanity instances cost $O(\iota n)$ prover time. The prover computes the table of ϕ by one scan over B_μ, and the two outer sumchecks cost $O(n+m)$. The grouped indicator sumcheck has degree $\ell+1$ and size n, which gives $O(\ell^2 n)$ prover time under the standard dense sumcheck accounting. The batched group-product check has degree $\kappa+1$. Using the bucketing algorithm of Bünz et al. [9, Algorithm 6], its prover time is $O(\kappa^2 n)$. This gives $O(\iota n+m+\ell^2 n+\kappa^2 n)$.

The verifier and communication costs are polylogarithmic in the domain sizes, apart from the polynomial dependence on λ and the logarithmic number of sumcheck rounds. Choosing $\ell,\kappa=\Theta(\sqrt{\iota})$, after padding if necessary, makes $\ell^2,\kappa^2=O(\iota)$, so the prover bound becomes $O(\iota n+m)$. □

D Instantiating GlueLUT-Fold with Brakedown

This appendix gives a Brakedown-specific instantiation of the folded gluing method from Section 6. The general construction uses two sumcheck instances to certify each folded consistency claim. When the witness polynomials are committed with Brakedown [31], its row-combination opening procedure can certify the required cross-modulus agreement directly. This removes the sumcheck layer while preserving the centered-representative argument used in the main construction.

Let $n=2^\mu$. For simplicity, assume that μ is even and reshape the evaluation tables of $f:B_\mu\to\mathbb{F}_P$ and $g:B_\mu\to\mathbb{Z}_Q$ into $\sqrt{n}\times\sqrt{n}$ matrices U_f and U_g. Let

Brakedown-specific instantiation of GlueLUT-Fold.

- **Common input:** a table $T : B_\iota \to \mathbb{Z}_Q$ and a Brakedown commitment $[[g]]_Q$ to $g : B_\mu \to \mathbb{Z}_Q$. Let $n = 2^\mu$, $m = 2^\iota$, and let q be the smallest prime divisor of Q. For simplicity, assume that μ is even.
- **Public parameters:** choose a prime

$$P > \max\left\{Q,\, 2\sqrt{n}\,\lfloor Q/2\rfloor \lfloor q/2\rfloor\right\},$$

and define $\widetilde{T}(a) := \mathrm{rep}_Q(T(a)) \bmod P$. Choose k such that $q^{-k} \le 2^{-\lambda}$.

Auxiliary-field witness. The prover commits with Brakedown over $\mathbb{F}_P$ to a multilinear polynomial $f : B_\mu \to \mathbb{F}_P$.

Brakedown gluing checks. Write U_f and U_g for the $\sqrt{n} \times \sqrt{n}$ matrices obtained by reshaping the evaluation tables of f and g, respectively. After receiving $[[f]]_P$, the verifier independently samples

$$r_0, \ldots, r_{k-1} \leftarrow \mathcal{I}_q^{\sqrt{n}}.$$

For every $i \in [k]$, the prover sends

$$v_i := r_i^\top U_f \in \mathbb{F}_P^{\sqrt{n}}, \qquad u_i := r_i^\top U_g \in \mathbb{Z}_Q^{\sqrt{n}},$$

together with Brakedown opening proofs that these are the row combinations of the committed evaluation matrices. The verifier checks all opening proofs and rejects unless, for every $i \in [k]$ and $j \in [\sqrt{n}]$,

$$\mathrm{rep}_P(v_{i,j}) \equiv u_{i,j} \pmod{Q}.$$

Auxiliary-field lookup. The parties run a field lookup argument proving

$$f(B_\mu) \subseteq \widetilde{T}(B_\iota).$$

Fig. 9: Brakedown-specific auxiliary-field gluing for lookup over $\mathbb{Z}_Q$.

q be the smallest prime divisor of Q. For a random coefficient vector $r \leftarrow \mathcal{I}_q^{\sqrt{n}}$, Brakedown allows the prover to open the row combinations

$$v = r^\top U_f \in \mathbb{F}_P^{\sqrt{n}}, \qquad u = r^\top U_g \in \mathbb{Z}_Q^{\sqrt{n}}.$$

For every coordinate j, the verifier checks $\mathrm{rep}_P(v_j) \bmod Q = u_j$. The auxiliary-field lookup ensures that every entry of U_f is the centered lift of a table entry. Consequently, choosing

$$P > \max\left\{Q,\, 2\sqrt{n}\,\lfloor Q/2\rfloor \lfloor q/2\rfloor\right\}$$

makes the centered table lift faithful and places every field-side row combination in $\mathcal{I}_P$. The comparison therefore tests the same signed integer sum modulo Q, without wraparound in either direction. Because the verifier checks the entire row-combination vector, a nonzero discrepancy gives a nonzero degree-one polynomial in r. Thus one check has error at most $1/q$, rather than the degree-two bound $2/q$ of the general rank-one folded check.

Theorem 10 (Brakedown-specific auxiliary-field lookup). *Let $n = 2^\mu$, $m = 2^\iota$, and let q be the smallest prime divisor of Q. Assume that μ is even.*

Choose k such that $q^{-k} \le 2^{-\lambda}$, and choose a prime

$$P > \max\left\{Q,\, 2\sqrt{n}\,\lfloor Q/2\rfloor \lfloor q/2\rfloor\right\}.$$

Assume that:

- *a Brakedown row-combination opening over a ring $\mathfrak{R}$ has perfect completeness, knowledge-soundness error $\delta^n_{\mathfrak{R},\mathsf{BD}}$, prover complexity $O(n)$, and verifier and communication complexity $O(\sqrt{n} + \mathrm{poly}(\lambda, \log n))$;*
- *the lookup argument over $\mathbb{F}_P$, for vector length n and table size m, has perfect completeness, knowledge-soundness error $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$, prover complexity $O(n + m)$, and logarithmic verifier complexity and proof size.*

Then the protocol in Fig. 9 has perfect completeness and knowledge-soundness error at most $q^{-k} + k\delta^n_{\mathbb{F}_P,\mathsf{BD}} + k\delta^n_{\mathbb{Z}_Q,\mathsf{BD}} + \delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$. Excluding oracle commitment costs, its prover complexity is $O(kn + m)$, and its verifier complexity and proof size are $O(k\sqrt{n} + \mathrm{poly}(\lambda, \log n) + \log m)$. For fixed security parameter and fixed q, the prover complexity is $O(n+m)$, while the verifier complexity and proof size are $O(\sqrt{n} + \mathrm{poly}(\lambda, \log n) + \log m)$.

Proof. **Completeness.** For a valid lookup, let $f(b) = \mathrm{rep}_Q(g(b)) \bmod P$. The auxiliary-field lookup and all Brakedown openings are then valid. For every sampled vector r_i and column $y \in B_{\mu/2}$, the corresponding signed integer row combination satisfies

$$\left|\sum_{x \in B_{\mu/2}} r_i(x)\,\mathrm{rep}_Q(g(x,y))\right| \le \sqrt{n}\,\lfloor q/2\rfloor\lfloor Q/2\rfloor < \frac{P}{2}.$$

Its centered representative modulo P is therefore the same integer, and its reduction modulo Q equals the ring-side row combination. Every coordinate comparison accepts.

Knowledge soundness. Except with probability at most $k\delta^n_{\mathbb{F}_P,\mathsf{BD}} + k\delta^n_{\mathbb{Z}_Q,\mathsf{BD}}$, knowledge soundness of the Brakedown openings extracts the two committed evaluation matrices and establishes all claimed row combinations. Except with probability $\delta^{n,m}_{\mathbb{F}_P,\mathsf{LUT}}$, the field lookup also gives

$$\mathrm{rep}_P(f(b)) \in \mathcal{I}_Q \qquad \text{for every } b \in B_\mu.$$

Suppose the extracted ring witness does not satisfy the lookup relation. On the event that the field lookup is valid, the matrix

$$\Delta_{x,j} := \left(\mathrm{rep}_P((U_f)_{x,j}) \bmod Q\right) - (U_g)_{x,j} \in \mathbb{Z}_Q$$

has a nonzero entry in $\mathbb{Z}_Q$. Fix a column j^* containing such an entry. The polynomial

$$H_{j^*}(R) := \sum_x R_x \Delta_{x,j^*} \in \mathbb{Z}_Q[(R_x)_x]$$

is nonzero and has total degree one. Because $\mathcal{I}_q$ is an exceptional set, Lemma 1 gives

$$\Pr_{r_i \leftarrow \mathcal{I}_q^{\sqrt{n}}}[H_{j^*}(r_i) = 0] \leq \frac{1}{q}.$$

The choice of P ensures that each field-side row combination is represented by its signed integer value before reduction modulo Q. Hence the verifier's coordinate comparison is precisely the test $H_{j^*}(r_i) = 0$ in $\mathbb{Z}_Q$. The k independent repetitions miss a nonzero discrepancy with probability at most q^{-k}. Adding the Brakedown and field-lookup errors gives the stated knowledge-soundness bound.

Complexity. Each repetition computes and opens one row combination of each $\sqrt{n} \times \sqrt{n}$ evaluation matrix, costing $O(n)$ prover operations and $O(\sqrt{n} + \mathrm{poly}(\lambda, \log n))$ verifier work and communication. The field lookup contributes $O(n + m)$ prover operations and logarithmic verifier work and communication. Summing these costs proves the claimed bounds. □