Sublinear Proofs over Polynomial Rings

Mi-Ying (Miryam) Huang Xinyu Mao Jiapeng Zhang ⋆

Abstract. Designing non-interactive proof systems, such as NIZK or SNARKs, for lattice-based protocols is a well-motivated problem with numerous applications including aggregate signatures, verifiable random functions, group signatures, verifiable homomorphic encryption schemes, and others. Although many works have made efforts toward efficient lattice-based protocols, several challenges remain. In particular, a major challenge, as highlighted by Ganesh et al. (Journal of Cryptology 2023) and Zhang et al. (CCS 2025), is the algebraic gap between polynomial rings, which are typically used in lattice-based constructions, and fields, which are commonly used in the constructions of non-interactive proofs. In this paper, we introduce a novel technique called *ring switching*, which bridges this gap by transforming proofs over polynomial-ring arithmetic into proofs over finite fields or Galois rings with prime power modulus. Using ring switching, we construct an efficient non-interactive argument of knowledge for Ring-R1CS over $\mathbb{Z}_Q[X]/(X^N+1)$ for arbitrary prime-power moduli Q. Our construction achieves sublinear proof size and enables significantly faster verification by eliminating costly ring multiplications. The ring switching framework is compatible with parameter regimes used in practical lattice-based cryptosystems, including NTT-friendly and power-of-two moduli. Subsequent works building on this technique further demonstrate its efficiency, achieving substantial verification time improvements in lattice-based polynomial commitments and verifiable homomorphic encryption schemes.

1 Introduction

Lattice-based cryptography has emerged as an important research direction since it is one of the most promising approaches for building post-quantum secure cryptosystems. Although it has been extensively studied for two decades, we still do not know how to make lattice-based versions of certain primitives as efficient as their group-based counterparts. In this direction, a major emerging topic is the construction of efficient proof systems for lattices. Here, we first provide some motivation for designing efficient proof systems for lattices.

Motivations of designing proof systems for lattices. Since lattice assumptions could serve as foundations for future post-quantum secure cryptographic systems, we assume that many future practically deployed protocols, such as

⋆ Thomas Lord Department of Computer Science, University of Southern California. Research supported by NSF CAREER award 2141536. Email: {miyinghu, xinyumao, jiapengz}@usc.edu

digital signatures, encryption schemes, pseudorandom generators, and pseudorandom functions, will be lattice-based. Once we have efficient proof systems for lattice, a lot of applications follow [BGRV09, NZZ15, DPLS18, BCOS20, EKS+21, GNSV23, BLNS23, ÅAB+24, ABVPS24, HLMZ24, NKS25]. For example, by combing NIZK with pseudorandom functions, we could obtain verifiable random functions [BGRV09]; by combining SNARKs with digital signatures, we could get aggregate signatures [ÅAB+24, NKS25]; by comining SNARGs with fully homomorphic encrpytion [ABVPS24], we could construction verifiable fully homomorphic encryption. In addition, we could also use it to construct blind signatures [BLNS23], group signatures [NZZ15], and more. Therefore, once we have efficient proof systems for lattices, we can construct these primitives efficiently.

Main obstacle: Algebraic structure gap. Although proof systems for lattices have been extensively studied [GNSV23, ÅAB+24], there remain several major obstacles to achieving efficient lattice-based proof systems.

One of the most significant obstacles, as highlighted in existing works [GNSV23, ZWL+25], is the following, which we call *algebraic structure gap*. Lattice-based cryptosystems usually use ring of the form $\mathcal{R} = \mathbb{Z}_q[X]/f(X)$ where q is a relatively small prime (e.g., 32-bit), where $f(X)$ is a cyclotomic polynomial with relatively high degree (e.g., $f(X) = X^{1024} + 1$). In contrast, SNARK operations are typically performed on much larger finite fields (e.g., fields with more than 2^{128} elements). This prevents existing SNARKs from being natively applied to lattice-based cryptosystems. For example,

- Ganesh et.al [GNSV23] discusses this barrier in the context of achieving fast verifiable FHE schemes, but it actually applies to many other lattice-based constructions.
- Sumcheck protocol is a crucial component of many proof systems; however, when naively applied to lattice-based cryptosystems, it cannot achieve efficient prover time or verifier time due to the algebraic structure gap. Looking ahead, we adapt the sumcheck protocol in a non-trivial way, which resolves the issue when combined with our new technique.

Existing approaches. Proof systems for lattices have been widely studied. Recent representative works include LaBRADOR [BS23] and LatticeFold+[BC25]. Though these approaches achieve very efficient prover time and proof size. There are still some challenges to fully address the algebraic structure gap.

- These constructions are usually efficient only for small rings, such as polynomial rings of dimension 64. This gap makes them less applicable for generating proofs for lattice-based protocols that perform on large rings.
- Both LaBRADOR and LatticeFold+ only natively support non-NTT-friendly prime moduli. These are not typical choices for the majority of lattice-based protocols, as they make the arithmetic operations much slower.
- Their verification time is not optimal. For example, it is still unknown how to make the verification time of LaBRADOR sublinear.

Ring switching. Our work aims to address the algebraic structure gap and enable much more efficient proof systems for lattice-based schemes. Our key novelty is the *ring switching* technique, which converts proofs over ring arithmetic into proofs over finite-field arithmetic (or over Galois rings when the modulus is not prime, such as a power of 2) with very small overhead.

Compared to existing proof systems for lattices, ring switching in particular enables extremely fast verification. As mentioned earlier, in some application scenarios, the cyclotomic rings used in lattice-based constructions can have dimension larger than 2^{10}, making ring multiplications highly inefficient. Surprisingly, under our ring switching framework, the verifier does not need to perform any ring multiplications at all. By contrast, in several recent sumcheck-based proof systems for lattices, such as LatticeFold+ [BC25] and SALSAA [KLOT25], ring multiplication remains a major efficiency bottleneck for the verifier, and they have to limit the ring size to keep concrete efficiency. This is the crux that ring switching achieves a fast verification time.

We note that the term ring switching also appears in another work by Diamond and Posen's [DP24]. Despite coincidentally using the same terminology, our motivations, problem formulation, and techniques are completely different from theirs.

Impact and subsequent applications. Before explaining our main results, we first elaborate on several follow-up works that have adopted our ring switching framework. These works further demonstrated that ring switching's main advantages compared to the existing approaches: a much faster verification time.

The first is Hachi [NOZ26], a new lattice-based polynomial commitment scheme, which achieves a much faster verification time than Greyhound while only slightly increasing the proof size. Similar to Greyhound, Hachi first reduces the proof of polynomial evaluation to a proof of lattice equations. The only difference is that Greyhound uses LaBRADOR [BS23] to prove the lattice equations, whereas Hachi instead uses our ring switching approach. Built on ring switching, Hachi achieves a $\Theta(\lambda)$ asymptotic improvement in verification time compared to Greyhound. Experimental results further demonstrate that the ring-switching approach achieves significantly faster verification than LaBRADOR, as expected. Although we do not provide an implementation in our paper to demonstrate the concrete efficiency of ring switching, we believe that the implementation of Hachi can offer an alternative concrete efficiency comparison with LaBRADOR.

The second application is a new verifiable homomorphic encryption scheme for certain matrix multiplications [TWZ26]. Their experiments also demonstrate the high efficiency of ring switching, demonstrating that ring switching is a good solution to resolve the algebraic structure gap barrier in verifiable fully homomorphic encryption highlighted in prior work [ZWL+25].

Beyond these two applications, we expect ring switching to have further use cases. For example, prior work has used LaBRADOR to prove Falcon, obtaining a lattice-based aggregate signature scheme [ÅAB+24]. The Ethereum Foundation

benchmarked this approach and commented that "*the verification time stands out as the biggest obstacle for adoption*" [NKS25]. Similar to the case of Hachi, we believe that ring switching can replace LaBRADOR in this setting as well, leading to much faster verification.

1.1 Our Results

We elaborate our ring switching idea through a specific ring arithmetic by ***Ring-R1CS***. Rank-one constraint system (R1CS) is a popular algebraic presentation used in many SNARK constructions. The only difference in Ring-R1CS is that the coefficients of each constraint are elements in the ring $\mathbb{Z}_Q[X]/(X^N+1)$. We formally define Ring-R1CS in Section 2. In addition to Ring-R1CS, the ring switching can also be applied to other ring arithmetic representations.

To demonstrate the generality of ring switching, we present a non-interactive argument of knowledge (NIAoK) with sublinear proof size for ring-R1CS over the ring $\mathbb{Z}_Q[X]/(X^N+1)$ for a wide choice of Q: we only require Q is a prime power. We allow Q to be a prime power because some lattice-based protocols, such as TFHE and FHEW, use powers of 2 as the modulus.

The prover time and verifier time of our NIAoK are almost linear in the input size, while the proof size is sublinear. Here, we emphasize that the main purpose of this paper is to explain the ring switching idea; we do not aim to optimize the asymptotic order of our proof and verification time. In some cases, such as when Q is a prime number, we could use a much more efficient polynomial commitment scheme, and thereby further improve the performance of our NIAoK. For example, follow-up work has already optimized ring switching for certain prime moduli.

To construct such an NIAoK, we first design a polynomial commitment scheme over Galois rings.

Theorem 1 (Polynomial Commitment over Ring, informal). *Let R be a Galois ring and $\mathcal{F}_{Comb}[R,\mu,N]$ be the class of polynomials over $(\mu+1)$ variables $X, X_1, \ldots, X_\mu$ where the degree of X is less than N and $X_1, \ldots, X_\mu$ are multilinear. Assuming the existence of collision-resistant hash functions, there exists a polynomial commitment scheme for $\mathcal{F}_{Comb}[R,\mu,N]$. The commitment has size $O(\lambda)$, and the evaluation protocol runs in $O_\lambda(2^\mu N)$ Galois ring operations for the prover and $O_\lambda(\sqrt{2^\mu N})$ Galois ring operations for the verifier, with communication complexity $O_\lambda(\sqrt{2^\mu N})$ Galois ring elements.*

Based on the polynomial commitment and ring switching, we obtain a public-coin interactive argument of knowledge for Ring-R1CS with sublinear proof size, which can be transformed into an NIAoK using Fiat-Shamir transform.

Theorem 2 (Sublinear Proof for Ring-R1CS, informal). *Let λ be the security parameter and $\mathcal{R} = \mathbb{Z}_Q[X]/(X^N+1)$ be a ring that depends on the security parameter, where $Q = p^s$ is a power of prime. Let m be the dimension of the the ring-R1CS (the statement is of size m^2). Let $d = \log_p\left(2\sqrt{mN}\right) + \omega(\log\lambda)$. Assuming the existence of collision-resistant hash functions, there exists a public-coin*

interactive argument of knowledge for R1CS over the ring $\mathcal{R}$ with the following efficiency characteristics.

- *Communication complexity:* $O(\sqrt{mN \log Q})$ *elements in* $\mathsf{Gal}(p^s, d)$.
- *Prover time: The prover's running time is* $O_\lambda(m^2 + mN \log Q)$ *operations in* $\mathsf{Gal}(p^s, d)$.
- *Verifier time: The verifier's running time is* $O(m^2 + \sqrt{mN \log Q} + N \log Q)$ *operations in* $\mathsf{Gal}(p^s, d)$.

Remark 1. The PCS in theorem 1 is hash-based, but we only need PCS over Galois rings when $s > 1$, i.e., Q is a prime power. If Q is a prime, $\mathsf{Gal}(Q, d)$ is the finite field with Q^d elements. In such a case, one can employ PCS constructions over finite fields, for which many lattice-based instantiations are known (see, e.g., [NS24, CMNW24]).

Remark 2. Recently, Jia et al. proposed a more efficient PCS over Galois rings [JLX+25]. Their PCS construction can be directly plugged into our proof system, bringing the communication complexity down to $O_\lambda(\log^2(m(N+Q))$ ring elements.

We believe that our NIAoK construction can be further improved in the following two aspects. Since the main purpose of this paper is to explain the ring switching idea, we left these optimizations for future work.

- Sublinear verification time with preprocessing. If Q is prime, $\mathsf{Gal}(Q, d)$ is a finite field, then with preprocessing, one can employ the SPARK compiler in [Set20] to reduce the verification time to sublinear in m^2 (i.e., size of the statement). If Q is a prime power, it is an interesting open question to generalize SPARK compiler to Galois rings.
- Zero-knowledge. It is well known that zero knowledge is an important property in many applications. However, we do not cover it in this paper due to two reasons. The first reason is that sublinear-size proof systems already have interesting applications, such as aggregate signatures [ÅAB+24]; Secondly, there are several standard techniques that can convert such interactive proofs with zero-knowledge [Set20, BS23]. In order to adopt these techniques in our protocol, we have to check whether these results can be applied to the Galois ring. In the paper, we would like to focus on the ring switching.

1.2 Proof Overview

Now we explain the ring switching idea. It consists of two steps; each step addresses one of the challenges.

1. Ring embedding. Let $\mathcal{R} = \mathbb{Z}_Q[X]/(X^N + 1)$ be a polynomial ring and let $\mathbb{x}$ be a ring-R1CS instance over $\mathcal{R}$. Assuming that $Q = p^s$ for some prime number p and $s \geq 1$. In ring embedding, we first transfer the instance $\mathbb{x}$ into another instance $\mathbb{x}'$ over the ring $\mathsf{Gal}(p^s, d)[X]$, where $\mathsf{Gal}(p^s, d)$ is a Galois ring with

an exceptional set of size p^d, hence we can keep the soundness of proofs by choosing a suitable d. Note that we have also removed the quotient (X^N+1) during the ring embedding. This ring embedding step eliminates the concern that there is no large exceptional set.

2. Ring reduction. Though the ring $\mathsf{Gal}(p^s,d)[X]$ has a large exceptional set, however, the ring size could be very large, affecting the efficiency. Our solution to this issue is very simple. Roughly speaking, the idea is that we first ask the prover to commit the witness via a polynomial commitment over $\mathsf{Gal}(p^s,d)$, and then the verifier sends a random α in $\mathsf{Gal}(p^s,d)$ to replace X. In such a step, we reduce the instance $\mathbb{x}'$ over $\mathsf{Gal}(p^s,d)[X]$ into a instance $\mathbb{x}''$ over $\mathsf{Gal}(p^s,d)$ without increasing the number of constraints in $\mathbb{x}'$. This replacement has a perfect complementness and an $O(N/p^d)$ soundness error.

The ring reduction step is more straightforward, so we skip the discussion here and direct the details to Section 4.2. Now, we briefly illustrate the ring embedding step. In a ring-R1CS instance, both the prover and the verifier know matrices $A,B,C \in \mathcal{R}^{m\times m}$, and the goal of the prover is to convince the verifier that they know a witness $z \in \mathcal{R}^m$ such that $(A\cdot z)\circ(B\cdot z) = (C\cdot z)$, where $(\cdot)$ is matrix-vector multiplication. $(\circ)$ is the entry-wise (Hadamard) product. Then we have the following observation, for $A,B,C \in \mathcal{R}^{m\times m}$:

$$\begin{aligned} &\exists z \in \mathbb{Z}_Q[X]/(X^N+1),\ \text{s.t. } (A\cdot z)\circ(B\cdot z) = (C\cdot z) \ \text{ over } \ \mathbb{Z}_Q[X]/(X^N+1) \\ \Longleftrightarrow\ &\exists z, D \in \mathbb{Z}_Q^{<3N}[X],\ \text{s.t. } (A\cdot z)\circ(B\cdot z) = (C\cdot z) + (X^N+1)\circ D \ \text{ over } \ \mathbb{Z}_Q[X] \end{aligned}$$

Here $\exists z, D \in \mathbb{Z}_Q^{<3N}[X]$ means that we restrict the degree of each entry of z, D is a polynomial in $\mathbb{Z}_Q[X]$ with degree smaller than $3N$. This is sufficient because the degree (of X) of each entry of A,B,C is smaller than N. This is the first step of ring embedding, and in the next step, we shall replace $\mathbb{Z}_Q$ by $\mathsf{Gal} := \mathsf{Gal}(p^s,d)$. We note that there exists $z, D \in \mathbb{Z}_Q^{<3N}[X]$, s.t. $(A\cdot z)\circ(B\cdot z) = (C\cdot z)+(X^N+1)\circ D$ if and only if,

$$\exists z, D \in \mathsf{Gal}^{<3N}[X],\ \text{s.t. } (A\cdot z)\circ(B\cdot z) = (C\cdot z) + (X^N+1)\circ D \text{ and } z \in \mathbb{Z}_Q^{<3N}[X]$$

The point here is that if we transfer the proof system into $\mathsf{Gal}[X]$, the verifier has to make sure that the witness is actually from $\mathbb{Z}_Q^{<3N}[X]$. In other words, by adding more constraints, we can then embed the R1CS instances into the ring $\mathsf{Gal}[X]$, which has a large exceptional set. To prove that $z \in \mathbb{Z}_Q^{<3N}[X]$, we adopt the bit decomposition approach used by [BCOS20, HLMZ24]. That is, we require the prover to commit each bit b of z, and then add the constraint $b(1-b)=0$ to ensure b is indeed a bit. To this end, we prove that the polynomial $g(W) = W(1-W)$ has only two roots (0 and 1) in $\mathsf{Gal}(p^s,d)[X]$.

Polynomial commitment scheme over Galois ring. We generalize the polynomial commitment scheme in [GLS+23] to polynomials over Galois rings.

We largely depend on the fact that Galois rings have a large exceptional set and prove *proximity gap* results for linear code over Galois rings. Indeed, the proximity gap for codes is central to many hash-based polynomial commitment schemes. Readers familiar with polynomial commitment schemes could view it as a natural generalization to Galois rings and skip this part.

Proving the witness has a small norm. With a slight modification to our construction, besides proving the witness satisfies the ring-R1CS, we can prove it also has a small norm, i.e., each entry of the witness is a short polynomial in $\mathcal{R}$ with coefficients smaller than β for some threshold $\beta \leq Q$. Proving the witness has a small norm is very useful for basing security on lattice assumptions.

Paper organization. Section 2 recalls notations and definitions. Section 3 presents a polynomial commitment scheme over Galois ring. Section 4 presents our NIAoK construction featuring the ring switching technique.

2 Preliminary

Notations. We denote by $\mathbb{N}$ the set of natural numbers and by $\mathbb{Z}$ the set of integers. For a prime power $Q = p^s$, we write $\mathbb{Z}_Q$ for the ring of integers modulo Q. The set of polynomials over $\mathbb{Z}_Q$ is denoted by $\mathbb{Z}_Q[X]$, and we consider quotient rings of the form $\mathcal{R} = \mathbb{Z}_Q[X]/(X^N + 1)$. For any set S, the notation $\leftarrow$ is used to indicate sampling from a probability distribution or selecting an element uniformly at random. A function $\nu : \mathbb{N} \to [0, 1]$ is considered negligible, denoted as $\nu = \mathsf{negl}(\lambda)$, if for every $c \in \mathbb{N}$, there exists a sufficiently large λ such that $\nu(\lambda) \leq \frac{1}{c\lambda^c}$. PPT stands for probabilistic polynomial time.

Definition 1 (Ring-R1CS). *An instance of a Ring-R1CS is a tuple* $\mathbb{x} = (R, A, B, C, io, m, n, \beta)$ *consisting of:*

- *A commutative ring with unity* $\mathcal{R}$.
- *Two integer parameters* $m, n \in \mathbb{N}$ *and a norm parameter* β.
- *Matrices* $A, B, C \in \mathcal{R}^{m \times m}$, *each containing at most* n *nonzero entries.*
- *A public input-output vector* $io \in \mathcal{R}^{|io|}$, *where* $m \geq |io| + 1$.

The instance $\mathbb{x}$ *is satisfiable if there exists a witness vector* $w \in \mathcal{R}^{m-|io|-1}$ *such that:*

- $(A \cdot z) \circ (B \cdot z) = (C \cdot z)$;
- *let* $w = (w_1, \ldots, w_{m-|io|-1})$, *then* $\|w_i\|_\infty \leq \beta$ *for each* i.

Here $z \stackrel{\mathsf{def}}{=} (io, 1, w)$ *is the full assignment vector.* $(\cdot)$ *denotes matrix-vector multiplication.* $(\circ)$ *is the entry-wise (Hadamard) product. If* w *is such an witness, we write* $(\mathbb{x}, w) \in \mathsf{SatR1CS}$.

Compared to the standard definition of R1CS, we have an additional norm parameter β. We add this because in many latticed-based primitives, we have to prove the norm of witness is small.

2.1 Ring Theory Background

All rings studied in this paper are assumed to be commutative and have identity.

Definition 2 (Exceptional set). *Let R be a ring and let $E = \{e_1, \ldots, e_n\} \subset R$. We say that E is an **exceptional set** if $\forall i \neq j$, $e_i - e_j \in R^*$, where R^* denotes the group of units of R. The **Lenstra's constant** of R, denoted by $\mathsf{L}(R)$, is defined to be the size of the largest exceptional set in R.*

Definition 3. *The order of the identity element 1 in the additive group of the ring is called the **characteristic** of the ring.*

Galois ring. Let p be a prime and let s, d be positive integers. Let $f(X) \in \mathbb{Z}_{p^s}[X]$ be a monic polynomial of degree d such that $\overline{f}$ is irreducible in $\mathbb{Z}_p[X]$, where $\overline{(\cdot)} : \mathbb{Z}_{p^s}[X] \to \mathbb{Z}_p[X]$ is the natural homomorphism defined via 'modulo p'. The structure of the quotient ring $\mathbb{Z}_{p^s}[X]/(f)$ is independent of the choice of f and is determined up to isomorphic solely by p, s, d. This ring is referred to as ***Galois ring of characteristic p^s and degree d***; we denote this ring by $\mathsf{Gal}(p^s, d)$. Galois rings have many nice properties due to their special algebraic structure; here, we list some useful properties.

Proposition 1 (Properties Galois rings [Wan11]). *Let p be a prime and $s, d, d' \in \mathbb{N}$. Then the following statements hold true.*

- *If $d | d'$, then $\mathsf{Gal}(p^s, d')$ contains (a unique) subring isomorphic to $\mathsf{Gal}(p^s, d)$.*
- *The Lenstra's constant of $\mathsf{Gal}(p^s, d)$ is p^d.*
- *$\mathsf{Gal}(p^s, 1)$ is isomorphic to $\mathbb{Z}_{p^s}$.*
- *$\mathsf{Gal}(p, d)$ is isomorphic to $\mathbb{F}_{p^d}$, the finite field with p^d elements.*

Proposition 2 (Generalized Schwartz-Zippel Lemma [BCPS18]). *Let R be a ring and let $E \subseteq R$ be a finite exceptional set. Let $f \in R[X_1, \ldots, X_n]$ be a non-zero polynomial in n variables and denote by $\deg(f)$ the degree of f. Then $\Pr_{\boldsymbol{e} \leftarrow E^n}[f(\boldsymbol{e}) = 0] \leq \frac{\deg(f)}{|E|}$.*

2.2 Linear Codes over Ring

Let R be a commutative ring with identity. For $u, v \in R^n$, its Hamming weight of u is defined as $\mathtt{HW}(u) \stackrel{\mathsf{def}}{=} |\{i \in [n] : u[i] \neq 0\}|$, and the Hamming distance is defined as $\mathsf{HD}(u, v) \stackrel{\mathsf{def}}{=} \mathtt{HW}(u - v)$.

Definition 4. *A non-empty subset $C \subseteq R^n$ is called a **linear code** (over R) if it is an R-submodule of R^n. The minimum distance of C is defined as $\mathsf{d}(C) = \min_{c \in C \setminus \{\mathbf{0}\}} \mathtt{HW}(c)$. For $v \in R^n$, the minimum distance from v to C is defined as*

$$\mathsf{d}(v, C) \stackrel{\mathsf{def}}{=} \min_{c \in C} \mathtt{HW}(v - c) = \min_{c \in C} |\{i \in [n] : v[i] \neq c[i]\}|.$$

Definition 5. *Let $C \subseteq R^n$ be a linear code. The* ***m-fold interleaved code of*** *C, denoted by C^m, is defined as $C^m \stackrel{\mathsf{def}}{=} \{U \in R^{m\times n} : \forall i \in [m], u_i \in C\}$, where u_i denotes the i-th row of U.*

Remark 3. C^m is also an R-module.

Definition 6 (Generalized Reed-Solomon (RS) code). *Let R be a ring and n, k two positive integers such that $k < n$. Let $E = \{x_1, \ldots, x_n\} \subseteq R$ be an exceptional set of R. Consider the submodule of R^n defined as*

$$\mathsf{GRS}[R, n, k] \stackrel{\mathsf{def}}{=} \{(f(x_1), \ldots, f(x_n)) : f \in R[X] \wedge \deg(f) < k\}.$$

$\mathsf{GRS}[R, n, k]$ *is called the* ***generalized Reed-Solomon code over*** *R.*

Proposition 3 ([QBC13]). *Let $C = \mathsf{GRS}[\mathsf{Gal}(p^s, d), n, k]$ be a generalized RS code over Galois ring $\mathsf{Gal}(p^s, d)$. Then*

- *C has minimum distance $\mathsf{d}(C) = n - k + 1$; and*
- *there exists a unique decoding algorithm that corrects up to $\lfloor \frac{n-k}{2} \rfloor$ errors and uses $\widetilde{O}(nk \cdot ds \log p)$ bit operations.*

2.3 Commitment Scheme

Definition 7. *A* ***commitment scheme*** *over message space $\mathcal{M}$ is a tuple of PPT algorithms* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *with the following syntax and properties.*

- $\mathsf{Setup}(1^\lambda, d) \mapsto \mathsf{pp}$: *Sample public parameters given a security parameter λ and size parameter d.* [1]
- $\mathsf{Commit}(\mathsf{pp}, \mathfrak{m}) \mapsto (\mathsf{com}, \mathsf{st})$*: It takes as input public parameter* pp *and a message $\mathfrak{m} \in \mathcal{M}$ and outputs a commitment* com *and an auxiliary state* st.
- $\mathsf{Open}(\mathsf{pp}, \mathsf{com}, \mathfrak{m}, \mathsf{st}) \mapsto b \in \{0, 1\}$*: It takes as input public parameter* pp*, a commitment* com*, a message $\mathfrak{m} \in \mathcal{M}$, and an auxiliary state* st*; it outputs a bit b indicating whether* com *is a valid commitment to $\mathfrak{m}$ under* pp.

We require commitment schemes to satisfy the following completeness and binding properties.

- *Completeness. For all $\lambda, d \in \mathbb{N}, \mathfrak{m} \in \mathcal{M}$, we have*

$$\Pr_{\mathsf{pp}\leftarrow\mathsf{Setup}(1^\lambda,d),(\mathsf{com},\mathsf{st})\leftarrow\mathsf{Commit}(\mathsf{pp},\mathfrak{m})}[\mathsf{Open}(\mathsf{pp}, \mathsf{com}, \mathfrak{m}, \mathsf{st}) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

- *Binding. For every PPT adversary $\mathcal{A}$, it holds that*

$$\Pr_{\substack{\mathsf{pp}\leftarrow\mathsf{Setup}(1^\lambda,d)\\(\mathsf{com},(\mathfrak{m},\mathsf{st}),(\mathfrak{m}',\mathsf{st}'))\leftarrow\mathcal{A}(\mathsf{pp})}}\begin{bmatrix}\mathfrak{m} \neq \mathfrak{m}' \wedge \mathfrak{m} \in \mathcal{M} \wedge \mathfrak{m}' \in \mathcal{M}\\ \wedge\mathsf{Open}(\mathsf{pp}, \mathsf{com}, \mathfrak{m}', \mathsf{st}') = 1\\ \wedge\mathsf{Open}(\mathsf{pp}, \mathsf{com}, \mathfrak{m}, \mathsf{st}) = 1\end{bmatrix} = \mathsf{negl}(\lambda).$$

[1] For example, if messages are encoded as binary bits, d is typically the length of the message.

Definition 8. *Given a (ternary) relation* $\mathfrak{R} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$*, if* $(\mathsf{pp}, \mathbb{x}, w) \in \mathfrak{R}$*, we call* pp *the public parameter,* $\mathbb{x}$ *the statement or instance,* w *the witness.*

Definition 9 (Argument of knowledge). *An* ***argument of knowledge (AoK)*** $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for relation* $\mathfrak{R}$ *consists of a PPT algorithm* Setup *and an interactive protocol between a PPT prover* $\mathcal{P}$ *and verifier* $\mathcal{V}$*, and it satisfies the following properties.*

- *Completeness. For all adversary* $\mathcal{A}$*,*

$$\Pr_{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), (\mathbb{x}, w) \leftarrow \mathcal{A}(\mathsf{pp})} [\langle \mathcal{P}(\mathsf{pp}, \mathbb{x}, w), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle = 0 \wedge (\mathsf{pp}, \mathbb{x}, w) \in \mathfrak{R}] = \mathsf{negl}(\lambda).$$

- *Knowledge soundness. There exists an expected PPT extactor* $\mathcal{E}$ *such that for any stateful PPT adversary* $\mathcal{P}^*$*,*

$$\Pr_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbb{x}, \mathsf{st}) \leftarrow \mathcal{P}^*(\mathsf{pp}) \\ w \leftarrow \mathcal{E}^{\mathcal{P}^*}(\mathsf{pp}, \mathbb{x}, st)}} [\langle \mathcal{P}^*(\mathsf{pp}, \mathbb{x}, \mathsf{st}), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle = 1 \wedge (\mathsf{pp}, \mathbb{x}, w) \notin \mathfrak{R}] = \mathsf{negl}(\lambda),$$

  where $\mathcal{E}$ *has black-box oracle access to the (malicious) prover* $\mathcal{P}^*$ *and can rewind it to any point in the interaction.*

Definition 10. *A* ***functional commitment scheme*** *for function class* $\mathcal{F}$ *is a tuple of algorithms* $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$ *with the following syntax and properties.*

- $(\mathsf{Setup}_{\mathsf{CM}}, \mathsf{Commit}, \mathsf{Open})$ *is a commitment scheme over message space* $\mathcal{F}$*.*
- $\mathsf{Eval} = (\mathsf{Setup}_{\mathsf{Eval}}, \mathcal{P}, \mathcal{V})$ *is an AoK for the relation* $\mathfrak{R}$ *defined as*

$$(\mathsf{pp}, (\mathsf{pp}_{\mathsf{CM}}, \mathsf{com}, \mathbb{x}, v), (f, \mathsf{st})) \in \mathfrak{R} \iff \mathsf{Open}(\mathsf{pp}_{\mathsf{CM}}, \mathsf{com}, f, \mathsf{st}) = 1 \wedge f(\mathbb{x}) = v.$$

2.4 Low-Degree Polynomials and Polynomials Extentions

Definition 11 (Multivariate and Multilinear Polynomials). *A polynomial that involves more than one variable is called a multivariate polynomial. If it only contains a single variable, we refer to it as a univariate polynomial. Furthermore, when each variable in a multivariate polynomial appears with degree at most one, we say that the polynomial is multilinear.*

Definition 12 (Low-Degree Polynomial). *Let* R *be a ring with cardinality* $|R|$*. A multivariate polynomial* $G(x_1, \ldots, x_m) \in R[x_1, \ldots, x_m]$ *is called a* low-degree polynomial *over* R *if, for each variable* x_i*, the degree of* G *in* x_i *is exponentially smaller than* $|R|$*.*

Definition 13 (Low-degree extension (LDE) over a ring). *Let* $G : \{0,1\}^m \to R$ *be a function that maps* m*-bit elements to elements of a ring* R*. A polynomial extension of* G *is a low-degree* m*-variate polynomial* $\widetilde{g}(\cdot)$ *over* R

such that $\widetilde{G}(x) = G(x)$ *for all* $x \in \{0,1\}^m$. *Furthermore, a multivariate polynomial is called a multilinear extension if the degree of each variable in* $\widetilde{G}$ *is at most* 1 *i.e.,* $Deg(\widetilde{G}(\cdot)) \leq 1$. *Given a function* $Z : \{0,1\}^m \to R$, *the multilinear extension of* $Z(\cdot)$ *is the unique multilinear polynomial* $\widetilde{Z} : R^m \to R$ *computed as*

$$\widetilde{Z}(x_1, \ldots, x_m) = \sum_{e \in \{0,1\}^m} Z(e) \cdot \prod_{i=1}^{m} (e_i \cdot x_i + (1 - e_i) \cdot (1 - x_i)) = \sum_{e \in \{0,1\}^m} Z(e) \cdot \widetilde{eq}(x, e)$$

where

$$\widetilde{eq}(x, e) = \prod_{i=1}^{m} (e_i \cdot x_i + (1 - e_i) \cdot (1 - x_i)).$$

which is the MLE of the function $eq(x, e) = 1$ *if* $e = x$; *otherwise,* $eq(x, e) = 0$. *Moreover, for all* $r \in R^m$, $\widetilde{Z}(r)$ *can be computed in* $O(2^m)$ *operations in* R.

3 Polynomial Commitment Scheme over Galois Rings

Polynomial commitment schemes are functional commitments where the function class is a class of polynomials. We consider several classes of polynomials over (Galois) rings that are useful in our SNARK construction.

- Multilinear polynomial. Let $\mathcal{F}_{Mul}[R, \mu]$ denote the class of multi-linear polynomials over R with μ variables.
- Univariate polynomial. Let $\mathcal{F}_{Uni}[R, N]$ denote univariate polynomials over R with degree less than n, namely, $\mathcal{F}_{Uni}[R, N] \stackrel{\mathsf{def}}{=} \{f \in R[X] : \deg(f) < N\}$.
- Combination. Let $\mathcal{F}_{Comb}[R, \mu, N]$ denote the class of polynomials over R with $(\mu + 1)$ variables $X, X_1, \ldots, X_\mu$, where the polynomials are multilinear in $X_1, \cdots, X_\mu$ and have degree less than N in X.

Note that $\mathcal{F}_{Mul}[R, \mu]$ and $\mathcal{F}_{Uni}[R, N]$ are subclasses of $\mathcal{F}_{Comb}[R, \mu, N]$.

The main result of this section is summarized below.

Theorem 3 (Polynomial commitment over Galois ring). *Let* μ, N, p, s, d *be integer functions of the security parameter* λ *such that* p *is a prime and* $d = \log_p(2 \cdot \sqrt{2^\mu \cdot N}) + \omega(\log \lambda)$. *Assuming the existence of collision-resistant hash function, there exists a functional commitment scheme for the class* $\mathcal{F}_{Comb}[\mathsf{Gal}(p^s, d), \mu, N]$ *with the following efficiency and security characteristics where* $\ell \stackrel{\mathsf{def}}{=} \sqrt{2^\mu \cdot N}$.

- *The commitment is the Merkle commitment of* $O(\ell^2)$ *elements of* $\mathsf{Gal}(p^s, d)$, *which is of size* $O(\lambda)$.
- *In an execution of the evaluation protocol,*
  - *the prover's running time is* $O_\lambda(\ell^2)$ *operations in* $\mathsf{Gal}(p^s, d)$;
  - *the verifier runs in time is* $O_\lambda(\ell)$ *operations in* $\mathsf{Gal}(p^s, d)$;
  - *the communication complexity is* $O_\lambda(\ell)$ *elements in* $\mathsf{Gal}(p^s, d)$;
  - *the soundness error is* $\varepsilon(\lambda) = \frac{2\ell}{p^d} + 2 \cdot 0.9^\lambda + \varepsilon_{\mathsf{Merkle}}(\lambda) = \mathsf{negl}(\lambda)$, *where* $\varepsilon_{\mathsf{Merkle}}(\lambda)$ *is the soundness error of Merkle commitment.*

Roadmap. Section 3.1 recalls a commonly used technique that reduces polynomial commitment scheme to vector commitment with special evaluation queries. Section 3.2 presents a vector commitment construction. Section 3.3 analyzes the efficiency of our scheme and instantiates the scheme using Galois ring.

3.1 From Vector Commitment to Polynomial Commitment

Tensor-query vector commitment over fields is introduced in [BCG20], and is used to construct a polynomial commitment scheme in Brakedown [GLS+23]. We naturally generalize the definition to rings.

Definition 14. *A* ***t*-fold tensor-query vector commitment scheme** *over ring R is a functional commitment scheme for the function class $\mathcal{F}_R$ defined as follows.*

$$\mathcal{F}_R \stackrel{\mathsf{def}}{=} \bigcup_{\ell \in \mathbb{N}} \left\{ f_u | u \in R^{\ell^t} \right\},$$

where $f_u : (R^\ell)^t \to R, (q_1, q_2, \ldots, q_t) \mapsto \langle q_1 \otimes q_2 \otimes \cdots \otimes q_t, u \rangle$.

Polynomial evaluation as a tensor query on coefficient vector. This is a commonly used technique (e.g., [BCG20, CMNW24]) that easily generalizes to rings.

- A polynomial $f(X_1, \ldots, X_\mu) \in \mathcal{F}_{Mul}[R, \mu]$ can be uniquely expressed as $f = \sum_{S \subseteq [\mu]} c_S \chi_S$, where $\chi_S(X_1, \ldots, X_\mu) \stackrel{\mathsf{def}}{=} \prod_{i \in S} X_i, c_S \in R$. Let $\mathbf{f}$ be the vector $(c_S)_{S \subseteq [\mu]} \in R^{2^\mu}$. Then we have for all $x_1, \ldots, x_\mu \in R$,

  $$f(x_1, \ldots, x_\mu) = \left\langle (1, x_1) \otimes \cdots \otimes (1, x_\mu), \mathbf{f} \right\rangle = \left\langle q_1 \otimes \cdots \otimes q_\mu, \mathbf{f} \right\rangle$$

  where $q_i \stackrel{\mathsf{def}}{=} (1, x_i)$ for $i \in [\mu]$.
- Let $f \in \mathcal{F}_{Uni}[R, N]$, say, $f(X) = \sum_{i=0}^{N-1} a_i X^i$. Let $\mathbf{f} = (a_0, \ldots, a_{N-1}) \in R^N$. WLOG, assume N is a power of 2. Then for all $x \in R$,

  $$f(x) = \left\langle (1, 1) \otimes (1, x) \otimes \cdots \otimes (1, x^{\log N - 1}), \mathbf{f} \right\rangle = \left\langle q'_1 \otimes \cdots \otimes q'_{\log N}, \mathbf{f} \right\rangle,$$

  where $q'_i \stackrel{\mathsf{def}}{=} (1, x^{i-1})$ for $i \in [\log N]$.
- A polynomial $f(X, X_1, \ldots, X_\mu) \in \mathcal{F}_{Comb}[R, \mu, N]$ can be uniquely expressed as $f(X, X_1, \ldots, X_\mu) = \sum_{i=0}^{N-1} \sum_{S \subseteq [\mu]} c_{i,S} \chi_S(X_1, \ldots X_\mu) \cdot X^i$. Let $\mathbf{f}$ be the vector $(c_{i,s})_{0 \le i < N, S \subseteq [\mu]} \in R^{N 2^\mu}$. Then for all $x, x_1, \ldots, x_\mu \in R$, it holds that

  $$f(x, x_1, \ldots, x_\mu) = \left\langle \bigotimes_{i=0}^{\log N - 1} (1, x^i) \bigotimes_{j=1}^{\mu} (1, x_\mu), \mathbf{f} \right\rangle.$$

3.2 Two-Fold Tensor Query Vector Commitment over Ring

We give an explicit construction for the case where $t = 2$, i.e., vector commitment that supports two-fold tensor queries. Generalization to the case where $t > 2$ is analogous to [GLS+23], assuming we have linear codes over the ring with good rates and encoding/decoding efficiency.

Definition 15. *For $w, w' \in R^{\ell \times M}$, define*

$$\mathsf{DiffCol}(w, w') \stackrel{\mathsf{def}}{=} \left\{ j \in [M] : \exists i \in [\ell]\ w_i[j] \neq w'_i[j] \right\},$$

where w_i and w'_i denote the i-th row of w and w' respectively.

Construction 1 (Tensor-query vector commitment over ring R) *Let $M = M(\ell)$, $S = S(\lambda)$ be parameters and let $C \subseteq R^M$ be a linear code that encodes elements in R^ℓ and $\mathsf{d}(C) \geq \gamma M$ for some constant γ.*

- $\mathsf{Setup}_{\mathsf{CM}}(1^\lambda, \ell) \mapsto \mathsf{pp}_{\mathsf{CM}}$ *: sample a public parameter for Merkle commitment of length ℓM and alphabet R.*
- $\mathsf{Commit}(u \in R^{\ell \times \ell}) \mapsto (\mathsf{com}, \mathsf{st})$*: Compute $\widehat{u} := (\mathsf{Enc}_C(u_1), \ldots, \mathsf{Enc}_C(u_\ell))$ and Merkle commitment of $\widehat{u}$ (using public parameter $\mathsf{pp}_{\mathsf{CM}}$), denoted by com; output com and $\mathsf{st} := \widehat{u}$.*
- $\mathsf{Open}(\mathsf{pp}, \mathsf{com}, u, \mathsf{st} \in R^{\ell \times M}) \mapsto \{0, 1\}$*: Output 1 if and only if st is an opening of the Merkle commitment com and $|\mathsf{DiffCol}(\mathsf{Enc}_C(u), \mathsf{st})| \leq \gamma M/4$.*
- Eval*: This is an AoK for the relation $\mathfrak{R}$ defined as follows:*

$$\begin{aligned} &(\mathsf{pp} = \lambda, (\mathsf{pp}_{\mathsf{CM}}, \mathsf{com}, (q_1, q_2), v), (u, \mathsf{st})) \in \mathfrak{R} \\ \iff &\mathsf{Open}(\mathsf{pp}_{\mathsf{CM}}, \mathsf{com}, u, \mathsf{st}) = 1 \wedge \langle q_1 \otimes q_2, u \rangle = v. \end{aligned}$$

Here, the setup algorithm of Eval outputs λ as public parameter. Using standard transformations [BSCS16], it suffices to construct an AoK Eval' for the relation $\mathfrak{R}'$ defined as

$$\begin{aligned} &(\mathsf{pp} = \lambda, (\widehat{u}, q_1, q_2, v), u) \in \mathfrak{R}' \\ \iff &|\mathsf{DiffCol}(\mathsf{Enc}_{C^\ell}(u), \widehat{u})| \leq \gamma M/4 \wedge \langle q_1 \otimes q_2, u \rangle = v. \end{aligned} \tag{1}$$

where the verifier $\mathcal{V}^{\widehat{u}}(\mathsf{pp}, q_1, q_2, v)$ in Eval' only has oracle access to $\widehat{u}$.

1. *$\mathcal{V} \to \mathcal{P}$: a uniformly random vector $r \leftarrow R^\ell$.*
2. *$\mathcal{P} \to \mathcal{V}$: a vector $u' \in R^\ell$. $\mathcal{P}$ claims that $u' = \sum_{i=1}^{\ell} r[i] u_i$.*
3. *$\mathcal{V}$: Sample a random set $Q \subseteq [M]$ of size S. For each $j \in Q$:*
   - *$\mathcal{V}$ queries the oracle to retrieve the j-th column $\widehat{u}_1[j], \ldots, \widehat{u}_\ell[j]$.*
   - *$\mathcal{V}$ rejects if $\mathsf{Enc}_C(u')[j] \neq \sum_{i=1}^{\ell} r[i] \widehat{u}_i[j]$.*
4. *$\mathcal{P} \to \mathcal{V}$: a vector $u'' \in R^\ell$. $\mathcal{P}$ claims that $u'' = \sum_{i=1}^{\ell} q_1[i] u_i$.*
5. *$\mathcal{V} \to \mathcal{P}$: a random set $Q' \subseteq [M]$ of size S. For each $j \in Q'$:*
   - *$\mathcal{V}$ queries the oracle to retrieve the j-th column $\widehat{u}_1[j], \ldots, \widehat{u}_\ell[j]$.*
   - *$\mathcal{V}$ rejects if $\mathsf{Enc}_C(u'')[j] \neq \sum_{i=1}^{\ell} q_1[i] \widehat{u}_i[j]$.*
6. *$\mathcal{V}$ accepts if $v = \langle u'', q_2 \rangle$; otherwise, reject.*

Completeness and binding of the commitment. Completeness of commitment holds by construction. Binding property follows from the binding property of Merkle commitment and the fact that for all $\widehat{u}$, there exists at most one u such that $|\mathsf{DiffCol}(\mathsf{Enc}_C(u), \widehat{u})| \le \gamma M/4 \le \mathsf{d}(C)/4$.

Argument of knowledge for evaluation.

Lemma 1. *Suppose that there exists an efficient decoding algorithm Dec_C for C that corrects up to $\gamma M/4$ errors. Then, the interactive protocol Eval' defined in construction 1 is an AoK for the relation $\mathfrak{R}'$ defined in eq. (1) with soundness error*

$$\varepsilon = \frac{M}{\mathsf{L}(R)} + (1-\gamma/4)^S + (1-(3/4)\gamma)^S.$$

In particular, ε is negligible in λ if $\mathsf{L}(R) = M \cdot \lambda^{\omega(1)}, S = \omega(\log \lambda)$.

Following [GLS+23], it is convenient to divide Eval' into two phases.

- Codeword testing phase. Via Step 1 –Step 3 of Eval', $\mathcal{V}$ verifies that $\widehat{u}$ is close to some codeword of C^ℓ.
- Evaluation phase. Via Step 4 – Step 6, $\mathcal{V}$ evaluates and verifies $u'' \stackrel{?}{=} \sum_{i=1}^{\ell} q_1[i] u_i$ and $v \stackrel{?}{=} \langle u'', q_2 \rangle = \langle u, q_1 \otimes q_2 \rangle$.

We first analyze the codeword testing phase.

Lemma 2 (Soundness of the codeword testing phase). *Consider the predicate $\mathsf{isClose} : R^{\ell \times M} \to \{0,1\}$ defined as*

$$\mathsf{isClose}(\widehat{u}) = 1 \iff \exists c_1, \ldots, c_m \in C \ \ |\{j \in [M] : \exists i \in [m], \widehat{u}_i[j] \neq c_i[j]\}| \le \gamma M/4. \tag{2}$$

For every PPT $\mathcal{P}^*$ and $\widehat{u} \in R^{\ell \times M}$, if $\mathsf{isClose}(\widehat{u}) = 0$, then

$$\mathbf{Pr}\left[\mathcal{V}^{\widehat{u}}(\cdots) \text{ rejects in Steps 1–3 when interacting with } \mathcal{P}^*\right] \ge 1 - \left(\frac{M}{\mathsf{L}(R)} + (1-\gamma/4)^S\right).$$

(Recall that $\mathsf{L}(R)$ is the Lenstra's constant of R, i.e., the size of the largest exceptional set in R.)

The technical tool for proving lemma 2 is a new *proximity gap* result in the ring setting. The proximity gap for linear codes is a crucial property in IOP-based polynomial commitment schemes. We generalize proximity gap results [AHIV17, BSCI+20] for interleaved linear codes to the ring setting:

Lemma 3 (Proximity gap for interleaved linear code). *Let $C \subseteq R^M$ be linear code over ring R with minimum distance d, and let $C^m \subseteq R^{m \times M}$ be the m-fold interleaved code of C. Then the following holds for all $u \in R^{m \times M}$. Define*

$$H \stackrel{\mathsf{def}}{=} \{j \in [M] : \exists i \in [m] \ \text{s.t.}\ u_i[j] \neq c_i[j]\},$$

where u_i denotes the i-th row of u and $c_i \in C$ is the codeword closest to u_i. For every $h \in \mathbb{N}$ such that $d \ge 4h$ and $|H| \ge h$, it holds that

$$\mathbf{Pr}_{\alpha_1, \ldots, \alpha_m \leftarrow R}[\mathsf{d}(\alpha_1 u_1 + \cdots \alpha_m u_m, C) < h] \le \frac{h}{\mathsf{L}(R)}.$$

We defer the proof of lemma 3 to appendix A and first use it to prove lemma 2.

Proof (Proof of lemma 2). Let $w \stackrel{\mathsf{def}}{=} \sum_{i=1}^{\ell} r_i \widehat{u}_i$ where $r \leftarrow R^\ell$ is the first message of $\mathcal{V}$. Applying lemma 3 with $h = \gamma M/4 \le \mathsf{d}(C)/4$, we have

$$\mathop{\mathbf{Pr}}_{r \leftarrow R^\ell}\left[\mathsf{d}(w, C) < \gamma M/4\right] \le \frac{M}{\mathsf{L}(R)}.$$

In Step 3, $\mathcal{V}$ chooses S coordinates uniformly at random and verifies that for every chosen coordinates j, $\mathsf{Enc}_C(u')[j] = w[j]$. Note that $\mathsf{d}(w, C) \ge \gamma M/4$ implies $\mathsf{HD}(w, \mathsf{Enc}_C(u')) \ge \gamma M/4$. Therefore, for every fixed r such that $\mathsf{d}(w, C) \ge \gamma M/4$, $\mathcal{V}$ rejects in Step 3 with probability at least $1 - (1 - \gamma/4)^S$. This finishes the proof.

Now we prove lemma 1, showing that Eval' is indeed an AoK.

Proof (Proof of lemma 1). Completeness of Eval' readily follows from the linearity of Enc_C. To see this, suppose $((\widehat{u}, q_1, q_2), u) \in \mathfrak{R}'$ and $\mathcal{P}$ acts honestly. Then, in Step 3,

$$\mathsf{Enc}_C(u')[j] = \sum_{i=1}^{\ell} r[i] \cdot \mathsf{Enc}_C(u_i)[j] = \sum_{i=1}^{\ell} r[i] \cdot \widehat{u}_i[j].$$

Hence, $\mathcal{V}$ would not reject in Step 3. Step 4 – Step 5 is identical to Step 2 – Step 3 except that r is replaced by q_1, and thus the same argument goes. Finally, when computing $\langle q_1 \otimes q_2, u\rangle$, u is interpreted as a vector of length ℓ^2. Hence,

$$\langle q_1 \otimes q_2, u\rangle = \sum_{i,j \in [\ell]} q_1[i] \cdot q_2[j] \cdot u_i[j] = \left\langle q_2, \sum_{i=1}^{\ell} q_1[i] \cdot u_i \right\rangle = \langle q_2, u''\rangle,$$

meaning that $\mathcal{V}$ accepts.

Next, we prove the knowledge soundness. Consider the extractor $\mathcal{E}(\mathsf{pp}, \widehat{u})$ that runs Dec_C on every row of $\widehat{u}$. That is, $\mathcal{E}(\mathsf{pp}, \widehat{u})$ outputs $u \in R^{\ell \times \ell}$ where the i-th row $u_i := \mathsf{Dec}_C(\widehat{u}_i)$ for each $i \in [\ell]$. We shall prove that for all (malicious) prover $\mathcal{P}^*$, it holds that

$$\begin{aligned}\mathsf{adv}_{\mathcal{P}^*}(\lambda) &\stackrel{\mathsf{def}}{=} \mathop{\mathbf{Pr}}_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x=(\widehat{u}, q_1, q_2, v), \mathsf{st}) \leftarrow \mathcal{P}^*(\mathsf{pp}) \\ u \leftarrow \mathcal{E}(\mathsf{pp}, \widehat{u})}} \left[\langle \mathcal{P}^*(x, \mathsf{st}), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge (\mathsf{pp}, x, u) \notin \mathfrak{R}'\right] \\ &\le \mathsf{negl}(\lambda),\end{aligned}$$

where the setup algorithm simply outputs $\mathsf{pp} := \lambda$. Note that

$$\begin{aligned}\mathsf{adv}_{\mathcal{P}^*}(\lambda) \le \overbrace{\mathop{\mathbf{Pr}}_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x=(\widehat{u}, q_1, q_2, v), \mathsf{st}) \leftarrow \mathcal{P}^*(\mathsf{pp}) \\ u \leftarrow \mathcal{E}(\mathsf{pp}, \widehat{u})}} \left[\mathsf{isClose}(\widehat{u}) = 1 \wedge \langle \mathcal{P}^*(\mathsf{pp}, x, \mathsf{st}), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge (\mathsf{pp}, x, u) \notin \mathfrak{R}'\right]}^{\stackrel{\mathsf{def}}{=} p(\lambda)} \\ + \mathop{\mathbf{Pr}}_{\substack{\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda) \\ (x=(\widehat{u}, q_1, q_2, v), \mathsf{st}) \leftarrow \mathcal{P}^*(\mathsf{pp}) \\ u \leftarrow \mathcal{E}(\mathsf{pp}, \widehat{u})}} \left[\mathsf{isClose}(\widehat{u}) = 0 \wedge \langle \mathcal{P}^*(\mathsf{pp}, x, \mathsf{st}), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1\right],\end{aligned}$$

where the second term is at most $\left(\frac{M}{\mathsf{L}(R)} + (1-\gamma/4)^S\right)$ according to lemma 2. Therefore, it suffices to bound the second term from above:

Claim. $p(\lambda) \leq (1-3/4\gamma)^S$.

Proof. We shall prove that for every fixed $\mathsf{pp} = \lambda, x = (\widehat{u}, q_1, q_2, v)$ with $\mathsf{isClose}(\widehat{u}) = 1$, and malicious prover $\mathcal{P}^*$, it holds that

$$\Pr_{u \leftarrow \mathcal{E}(\mathsf{pp},\widehat{u})}\left[\langle \mathcal{P}^*(\mathsf{pp}, x), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge (\mathsf{pp}, x, u) \notin \mathfrak{R}'\right] \leq (1-3/4\gamma)^S. \quad (3)$$

This would imply the claim. To this end, we show that

$$\Pr_{u \leftarrow \mathcal{E}(\mathsf{pp},\widehat{u})}\left[\langle \mathcal{P}^*(\mathsf{pp}, x), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge u'' \neq t\right] \leq (1-3/4\gamma)^S. \quad (4)$$

where $t \stackrel{\mathsf{def}}{=} \sum_{i\in[\ell]} q_1[i] \cdot u_i$ and u'' is the message sent by $\mathcal{P}^*$ in Step 4.

We first show that eq. (4) indeed implies eq. (3). Recall that $u_i = \mathsf{Dec}_C(\widehat{u}_i)$. Whenever $\mathsf{isClose}(\widehat{u}) = 1$, by the correctness of Dec_C, we have $|\mathsf{DiffCol}(\mathsf{Enc}_{C^\ell}(u), \widehat{u})| \leq \gamma M/3$. Consequently, if $(\mathsf{pp}, x, u) \notin \mathfrak{R}'$, it must be that $v \neq \langle q_1 \otimes q_2, u\rangle = \langle q_2, t\rangle = \sum_{i\in[\ell]} q_2[i] \cdot t[i]$. Moreover, if $\mathcal{V}$ accepts in the last step, we have $v = \langle q_2, u''\rangle = \sum_{i\in\ell} q_2[i]u''[i]$. That is,

$$\begin{aligned}
&\langle \mathcal{P}^*(\mathsf{pp}, x), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge (\mathsf{pp}, x, u) \notin \mathfrak{R}' \\
\implies &\langle \mathcal{P}^*(\mathsf{pp}, x), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge \langle q_2, u''\rangle \neq \langle q_2, t\rangle \\
\implies &\langle \mathcal{P}^*(\mathsf{pp}, x), \mathcal{V}^{\widehat{u}}(\mathsf{pp}, (q_1, q_2, v))\rangle = 1 \wedge u'' \neq t.
\end{aligned}$$

Now we prove eq. (4) by showing that if $u'' \neq t$, then $\mathcal{V}$ rejects with overwhelming probability in Step 5. Suppose that $u'' \neq t$. Since $\mathsf{Enc}_C(u'')$ and $w \stackrel{\mathsf{def}}{=} \mathsf{Enc}_C(t)$ are two distinct codewords, they agree on at most $(1-\gamma)M$ coordinates, i.e., letting $A \stackrel{\mathsf{def}}{=} \{i \in [M] : \mathsf{Enc}_C(u'')[i] = w[i]\}$, we have $|A| \leq (1-\gamma)M$. Observe that $\mathsf{isClose}(\widehat{u}) = 1$ implies that $w' \stackrel{\mathsf{def}}{=} \sum_{i\in[\ell]} q_1[i]\widehat{u}_i$ and $w = \sum_{i\in[\ell]} q_1[i]\mathsf{Enc}_C(u_i)$ differ on at most $\gamma M/4$ coordinates. That is, letting $B \stackrel{\mathsf{def}}{=} \{i \in [M] : w'[i] \neq w[i]\}$, we have $|B| \leq \gamma M/4$. Hence, $\mathsf{Enc}_C(u'')$ and w' agree on at most

$$|A \cup B| \leq |A| + |B| \leq (1 - (3/4)\gamma)\, M$$

coordinates. In Step 5, $\mathcal{V}$ chooses S coordinates uniformly at random and verifies that for every chosen coordinates j, $\mathsf{Enc}_C(u'')[j] = w'[j]$. Therefore, if $u'' \neq t$, $\mathcal{V}$ rejects in Step 5 with probability at least $1 - (1-(3/4)\gamma)^S$, and thus eq. (4) holds.

3.3 Efficiency and Instantiation by Galois Ring

By inspection of construction 1, we have the following theorem describes its efficiency.

Theorem 4. *Construction 1 has the following efficiency and security characteristics when running on security parameter λ and size parameter ℓ (i.e., the committed vector is in R^{ℓ^2}):*

- *Commitment. The commitment is the Merkle hash of $O(\ell M)$ elements of R, which is of size $O(\lambda)$.*
- *Evaluation. In an execution of* Eval*,*
  - *the prove's running time is $O_\lambda(\ell^2)$ operations over R;*
  - *the verifier's running time is $O_\lambda(\ell)$ operations over R;*
  - *the communication complexity is $O_\lambda(\ell)$ elements over R.*

Instantiation by Galois ring. Let $d = d(\lambda)$ be a parameter and $Q = p^s$ be a modulus, where $p = p(\lambda)$ is a prime and $s = s(\lambda) \in \mathbb{N}$. The ring R is set to be $\mathsf{Gal}(Q, d)$. We require the size parameter $\ell = \ell(\lambda)$ to satisfy

$$d \geq \log_p(2\ell) + \omega(\log \lambda).$$

For linear code, we use generalized Reed-Solomon code $C = \mathsf{GRS}[\mathsf{Gal}(Q, d), 2\ell, \ell]$ described in definition 6. Consequently, letting $M := 2\ell, \gamma := \frac{1}{2}$, we have (1) $\mathsf{d}(C) = \ell + 1 \geq \gamma M$; (2) the decoding algorithm given by proposition 3 can correct up to $\lfloor \ell/2 \rfloor \geq \gamma M/4$ errors; and (3) by proposition 1, $M/\mathsf{L}(R) = 2\ell/p^d = \mathsf{negl}(\lambda)$. Finally, by lemma 1, we set $S = S(\lambda) = \lambda$ so that the soundness error is

$$\frac{2\ell}{p^d} + 0.875^\lambda + 0.125^\lambda + \varepsilon_{\mathsf{Merkle}}(\lambda) < \frac{2\ell}{p^d} + 2 \cdot 0.9^\lambda + \varepsilon_{\mathsf{Merkle}}(\lambda) = \mathsf{negl}(\lambda),$$

Here, $\varepsilon_{\mathsf{Merkle}}(\lambda)$ is the soundness error of Merkle commitment, which is incurred due to the standard transform from Eval' to Eval. Combined with the reduction from polynomial commitment to vector commitment with tensor queries (in section 3.1), we get a polynomial commitment scheme over the Galois ring, as stated in theorem 3.

4 Sublinear proofs for Ring-R1CS via Ring Switching

In this section, we first recap the sumcheck protocol for ring polynomials. Then, we design an interactive argument with sublinear communication cost via the aforementioned sumcheck protocol and our ring switching technique and then compile it into a family of NIAoKs in the Random Oracle Model (ROM). In what follows, we assume the norm parameter $\beta = Q$; then, in section 4.3, we generalize this to the case where β is an arbitrary power of two.

4.1 Sumcheck Protocol for Ring Polynomials

We first review the sumcheck protocol for rings. Consider a polynomial $G \in R[X_1, \ldots, X_n]$ be a degree-d polynomial whose coefficients in a ring R. Let E be

an exceptional set of R. A (potentially malicious) prover P_{SC} want to convince the a verifier V_{SC} the following claim:

$$T = \sum_{b_1 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} G(b_1, \ldots, b_n). \tag{5}$$

We denote the sumcheck protocol as $e \leftarrow \langle P_{SC}(G), V_{SC}(r)\rangle(n, \ell, T)$, where $r = (r_1, \ldots, r_n)$ is the randomness of the verifier used throughout the interaction (it can sample different random elements in each round) and the prover P_{SC} the polynomial G as the input.

$\langle P_{SC}(G), V_{SC}(r)\rangle(n, \ell, T)$

1. $\mathcal{P}$'s round. Send some degree-ℓ polynomial p_n. The honest prover shall send $h_n \in R[X]$ defined as follows.

$$h_n(X_n) \stackrel{\mathsf{def}}{=} \sum_{b_1 \in \{0,1\}} \cdots \sum_{b_{n-1} \in \{0,1\}} G(b_1, \ldots, b_{n-1}, X_n).$$

2. $\mathcal{V}$'s round.
   (a) On receiving a degree-ℓ polynomial p_n, output $\perp$ (reject symbol) if $p_n(0) + p_n(1) \neq T$.
   (b) Parse $r = (r_{[1:n-1]}, r_n)$ where $r_{[1:n-1]} \in E^{n-1}$ and $r_n \in E$.
   (c) If $n = 1$, output $e := p_1(r_1)$. Otherwise, send r_n to $\mathcal{P}$, and then recursively run the protocol $\langle P_{SC}(G|_{X_n = r_n}), V_{SC}(r_{[1:n-1]})\rangle(n-1, \ell, p_n(r_n))$ to check

$$p_n(r_n) = \sum_{b_2 \in \{0,1\}} \cdots \sum_{b_n \in \{0,1\}} G(b_1, \ldots, b_{n-1}, r_n).$$

Fig. 1. Sumcheck protocol for ring.

[HLMZ24] proved the completeness and soundness of the above sumcheck protocol for rings with a large exceptional set.

Lemma 4 ([HLMZ24], Theorem 5.1). *The protocol in fig. 1 for the statement in eq. (5) satisfies the following properties.*

- *Completeness. If G has degree d and $T = \sum_{b_1,\ldots,b_n \in \{0,1\}} G(b_1, \ldots, b_n)$, then*

$$\mathbf{Pr}\left[e = G(r) \mid r \leftarrow E^n,\ e \leftarrow \langle P_{SC}(G), V_{SC}(r)\rangle(n, \ell, T)\right] = 1.$$

- *Soundness. If G has degree d and $T \neq \sum_{b_1,\ldots,b_n \in \{0,1\}} G(b_1, \ldots, b_n)$, then for every malicious prover P^*_{SC},*

$$\mathbf{Pr}\left[e = G(r) \mid r \leftarrow E^n,\ e \leftarrow \langle P^*_{SC}, V_{SC}(r)\rangle(n, \ell, T)\right] \leq \frac{nd}{|E|}.$$

4.2 Sublinear Proofs over Ring

Let $\mathcal{R} \stackrel{\mathsf{def}}{=} \mathbb{Z}_Q[X]/(X^N+1)$. Let $s = \log m$. We regard A, B, C as functions from $\{0,1\}^s \times \{0,1\}^s$ to $\mathcal{R}$. Given a witness $w \in \mathcal{R}^{m/2}$. Define $F_w : \{0,1\}^s \to \mathcal{R}$ by,

$$F_w(x) \stackrel{\mathsf{def}}{=} \left(\sum_{y\in\{0,1\}^s} A(x,y)Z(y)\right)\left(\sum_{y\in\{0,1\}^s} B(x,y)Z(y)\right) - \left(\sum_{y\in\{0,1\}^s} C(x,y)Z(y)\right)$$

here we also use a function $Z : \{0,1\}^s \to \mathcal{R}$ to encode $(w, 1, io) \in \mathcal{R}^m$.

Let $\phi : \mathcal{R} \to \mathbb{Z}_Q[X]^{<N}$ be a natural bijection from $\mathcal{R}$ to $\mathbb{Z}_Q[X]^{<N}$ where the addition and multiplication in $\mathbb{Z}_Q[X]^{<N}$ are carried out in $\mathbb{Z}_Q[X]$.[2] Let $A' : \{0,1\}^{2s} \to \mathbb{Z}_Q[X]^{<N}$ be the composition of ϕ and A, i.e., $A'(x,y) \stackrel{\mathsf{def}}{=} \phi(A(x,y))$; and B', C', w', Z' are defined in the same way.

Consider the function $\widehat{F}_{w'} : \{0,1\}^s \to \mathbb{Z}_Q[X]$ that computes $F_w(x)$ on $\mathbb{Z}_Q[X]$ instead of $\mathcal{R}$, i.e.,

$$\widehat{F}_{w'}(x) \stackrel{\mathsf{def}}{=} \left(\sum_{y\in\{0,1\}^s} A'(x,y)Z'(y)\right)\left(\sum_{y\in\{0,1\}^s} B'(x,y)Z'(y)\right) - \left(\sum_{y\in\{0,1\}^s} C'(x,y)Z'(y)\right)$$

where the addition and multiplication are carried out in $\mathbb{Z}_Q[X]$ *without* reducing modulo (X^N+1). Observe that,

$$\begin{aligned}
&(\mathbb{x}, w) \in \mathsf{SatR1CS} \\
\iff &\forall x \in \{0,1\}^s, F_w(x) = 0_{\mathcal{R}} \\
\iff &\forall x \in \{0,1\}^s, \exists D_x \in \mathbb{Z}_Q[X]^{<3N}, \widehat{F}_{w'}(x) = D_x \cdot (X^N+1) \\
\iff &\exists D \in (\mathbb{Z}_Q[X]^{<3N})^{\{0,1\}^s}, \forall x \in \{0,1\}^s, \widehat{F}_{w'}(x) = D(x) \cdot (X^N+1).
\end{aligned}$$

Given a polynomial $w' \in (\mathbb{Z}_Q[X]^{<N})^{\frac{m}{2}}$ and a function $D : \{0,1\}^s \to \mathbb{Z}_Q[X]^{<3N}$, define a function $G_{w',D} : \{0,1\}^s \to \mathbb{Z}_Q[X]^{<3N}$ via

$$G_{w',D}(x) \stackrel{\mathsf{def}}{=} \widehat{F}_{w'}(x) - D(x) \cdot (X^N+1).$$

We summarize the above observation as the following lemma.

Lemma 5. *Let $w' \in (\mathbb{Z}_Q[X]^{<N})^{\frac{m}{2}}$. Then $\phi^{-1}(w')$ is a witness of $\mathbb{x}$ if and only if there exists a function $D : \{0,1\}^s \to \mathbb{Z}_Q[X]^{<3N}$ such that*

$$\forall x \in \{0,1\}^s, G_{w',D}(x) = 0.$$

Note that $G_{w',D}$ is a function, not a polynomial. Let $\widetilde{A'}, \widetilde{B'}, \widetilde{C'}, \widetilde{Z'}, \widetilde{D}, \widetilde{\mathsf{eq}}$ be multilinear extensions of $A', B', C', Z', D, \mathsf{eq}$ respectively. We define the polynomial extensions of $F_{w'}$ and $G_{w',D}$ as follows.

$$\widetilde{F_{w'}}(x) \stackrel{\mathsf{def}}{=} \left(\sum_{y\in\{0,1\}^s} \widetilde{A'}(x,y)\widetilde{Z'}(y)\right)\left(\sum_{y\in\{0,1\}^s} \widetilde{B'}(x,y)\widetilde{Z'}(y)\right) - \left(\sum_{y\in\{0,1\}^s} \widetilde{C'}(x,y)\widetilde{Z'}(x,y)\right)$$

[2] We sometimes abuse the notation of ϕ to act on a vector $v \in \mathcal{R}^n$, which applies ϕ to v entry-wise.

$$\widetilde{G_{w',D}}(x) \stackrel{\mathsf{def}}{=} \widetilde{F_{w'}}(x) - \widetilde{D}(x) \cdot (X^N + 1).$$

Similar to Spartan [Set20], we consider the polynomial $Q_{w',D} \in \mathbb{Z}_Q[X][X_1, \ldots, X_s]$

$$Q_{w',D}(t) \stackrel{\mathsf{def}}{=} \sum_{x \in \{0,1\}^s} \widetilde{\mathsf{eq}}(t, x) \cdot \widetilde{G_{w',D}}(x).$$

Lemma 6. *Let $w' \in (\mathbb{Z}_Q[X]^{<N})^{\frac{m}{2}}$. Then $\phi^{-1}(w')$ is a witness of $\mathbb{x}$ if and only if there exists a function $D : \{0,1\}^s \to \mathbb{Z}_Q[X]^{<3N}$ such that $Q_{w',D}$ is identical to zero.*

Proof. By lemma 5, it suffices to prove the equivalence between the following two statements:

- $\forall x \in \{0,1\}^s, G_{w',D}(x) = 0$.
- $Q_{w',D}$ is zero polynomial.

The first statement implies $\forall x \in \{0,1\}^s, \widetilde{G_{w',D}}(x) = G_{w',D}(x) = 0$. Then by definition, $Q_{w',D}(t) \equiv 0$. Now suppose the second statement holds. Recall that $\widetilde{\mathsf{eq}}(x, x') = \mathsf{eq}(x, x') = 1_{x=x'}$ when $x, x' \in \{0,1\}^s$. Hence, for every $x \in \{0,1\}^s$,

$$0 = Q_{w',D}(x) = \widetilde{G_{w',D}}(x) = G_{w',D}(x).$$

Building upon lemma 6, one would hope to perform a probabilistic check on the polynomial $Q_{w',D} \in \mathbb{Z}_Q[X]$. However, probabilistically checking over $\mathbb{Z}_Q[X]$ is less favorable as $\mathbb{Z}_Q$ does not have a large exceptional set. To get around this problem, we develop the following ring embedding technique.

Ring Embedding. Our idea is to embed $\mathbb{Z}_Q[X]$ into $\mathsf{Gal}(Q, d)[X]$ using a Galois ring $\mathsf{Gal}(Q, d)$ that admits a large exceptional set E. After the embedding, our goal is to check the relation $Q_{w',D} \equiv 0$ in $\mathbb{Z}_Q[X]$ probabilistically over $\mathsf{Gal}(Q, d)[X]$. Like Spartan, the prover first commits the multilinear extension of the witness w'. However, when we move into the ring $\mathsf{Gal}(Q, d)[X]$, the verifier has to additionally check whether the committed polynomial $\widetilde{w'}$ is from $\mathbb{Z}_Q[X]$, but not from $\mathsf{Gal}(Q, d)[X]$. To check this, we require the prover to also commit the multilinear extension of the bit representation (denoted by bits) of w'. Once the prover has committed the multilinear extension of bits, the verifier checks two things via sumcheck protocols:

1. The outputs of bits are $0, 1$.
2. bits is a bit representation of w'.

Given a function $\mathsf{bits} : \{0,1\}^{s-1} \times \{0, 1 \ldots, N-1\} \times \{0, 1, \ldots, \log Q - 1\} \to \mathsf{Gal}(Q, d)$, we first explain how to check the outputs of bits are $\{0, 1\}$. We identify its domain with $\{0,1\}^{s+s'-1}$ by naive binary coding, where $s' \stackrel{\mathsf{def}}{=} \lceil \log(N \log Q) \rceil$. Let $\widetilde{\mathsf{bits}}(\cdot)$ be the multilinear extension of $\mathsf{bits}(\cdot)$. We define the polynomial $Q'_{\mathsf{bits}} \in \mathsf{Gal}(Q, d)[X_1, \ldots, X_{s+s'-1}]$

$$Q'_{\mathsf{bits}}(t) \stackrel{\mathsf{def}}{=} \sum_{\sigma \in \{0,1\}^{s+s'-1}} \widetilde{\mathsf{eq}}(t, \sigma) \cdot [\widetilde{\mathsf{bits}}(\sigma)^2 - \widetilde{\mathsf{bits}}(\sigma)].$$

Lemma 7. bits *takes value in* $\{0,1\}$ *if and only of* Q'_{bits} *is zero polynomial.*

Proof. First, if bits takes values in $\{0,1\}$, then $\widetilde{\mathsf{bits}}(\sigma)^2 - \widetilde{\mathsf{bits}}(\sigma) = \mathsf{bits}(\sigma)^2 - \mathsf{bits}(\sigma) = 0$ for every $\sigma \in \{0,1\}^{s+s'-1}$. Then by definition, $Q'_{\mathsf{bits}}(t) \equiv 0$. Second, suppose that Q'_{bits} is the zero polynomial. Recall that $\widetilde{\mathsf{eq}}(\sigma,\sigma') = \mathsf{eq}(\sigma,\sigma') = 1_{\sigma=\sigma'}$ when $\sigma,\sigma' \in \{0,1\}^{s+s'-1}$. Hence, for every $\sigma \in \{0,1\}^{s+s'-1}$,

$$0 = Q'_{\mathsf{bits}}(\sigma) = \widetilde{\mathsf{bits}}(\sigma)^2 - \widetilde{\mathsf{bits}}(\sigma) = \mathsf{bits}(\sigma)^2 - \mathsf{bits}(\sigma),$$

meaning that $\mathsf{bits}(\sigma)$ is a solution to the equation $y^2 - y = 0$ in $\mathsf{Gal}(Q,d)$.

Claim. $g(y) = y^2 - y$ has only roots $\{0,1\}$ in any Galois ring $\mathsf{Gal}(Q,d)$.

Proof. Every Galois ring is a local ring i.e., a commutative ring with a unique maximal ideal. A local ring admits no nontrivial idempotents. That is, if y is not an idempotent which is not $0,1$, then $y^2 - y = y(y-1)$ shows that both y and $y-1$ are zero divisors and, in particular, not invertible, so must be in the maximal ideal, but then $1 = y - (y-1)$ is also in the maximal ideal, which leads to a contradiction.

With the above claim, we know that $\mathsf{bits}(\sigma) \in \{0,1\}$. Therefore, bits takes values in $\{0,1\}$. We conclude the proof of lemma 7.

Next, we show how to prove that bits is a bit representation of w'. Given a polynomial $w' \in (\mathsf{Gal}(Q,d)[X]^{<N})^{m/2}$, we define the following function $H_{w',\mathsf{bits}} : \{0,1\}^{s-1} \to \mathsf{Gal}(Q,d)[X]^{<N}$

$$H_{w',\mathsf{bits}}(x) \stackrel{\mathsf{def}}{=} w'(x) - \sum_{i=0}^{N-1} \sum_{j=0}^{\log Q-1} \mathsf{bits}(x,i,j) \cdot 2^j \cdot X^i,$$

Let $\widetilde{H_{w',\mathsf{bits}}} \in \mathcal{F}_{Comb}[\mathsf{Gal}(Q,d), s-1, N] \subseteq \mathsf{Gal}(Q,d)[X, X_1, \ldots, X_{s-1}]$ denote the polynomial extension

$$\widetilde{H_{w',\mathsf{bits}}}(x) \stackrel{\mathsf{def}}{=} \widetilde{w'}(x) - \sum_{i=0}^{N-1} \sum_{j=0}^{\log Q-1} \widetilde{\mathsf{bits}}(x,i,j) \cdot 2^j \cdot X^i.$$

and consider the polynomial $Q''_{w',\mathsf{bit}} \in \mathcal{F}_{Comb}[\mathsf{Gal}(Q,d), s-1, N] \subseteq \mathsf{Gal}(Q,d)[X, X_1, \ldots, X_{s-1}]$

$$Q''_{w',\mathsf{bits}}(t) \stackrel{\mathsf{def}}{=} \sum_{x \in \{0,1\}^{s-1}} \widetilde{\mathsf{eq}}(t,x) \cdot \widetilde{H_{w',\mathsf{bits}}}(x).$$

Lemma 8. *Let* $w' \in (\mathsf{Gal}(Q,d)[X]^{<N})^{m/2}$. *Then* $w' \in (\mathbb{Z}_Q[X]^{<N})^{m/2}$ *if and only if there exists a function* $\mathsf{bits} : \{0,1\}^{s-1} \times \{0,1\ldots,N-1\} \times \{0,1,\ldots,\log Q-1\} \to \mathsf{Gal}(Q,d)$ *such that both* Q'_{bits} *and* $Q''_{w',\mathsf{bits}}$ *are zero polynomials.*

Proof. Suppose $w' \in (\mathbb{Z}_Q[X]^{<N})^{m/2}$. Then for every $x \in \{0,1\}^{s-1}$, $w'(x) \in \mathbb{Z}_Q[X]^{<N}$, so there exists $c_{x,i} \in \mathbb{Z}_Q$ such that $w'(x) = \sum_{i=0}^{N-1} c_{x,i} \cdot X^i$. Moreover, each $c_{x,i}$ has a binary representation $c_{x,i} = \sum_{j=0}^{\log Q-1} b_{x,i,j} \cdot 2^j$ where each $b_{x,i,j} \in \{0,1\}$. Consider the function $\mathsf{bits}(x,i,j) := b_{x,i,j}$. Then for every $x \in \{0,1\}^{s-1}$,

$$\widetilde{H_{w',\mathsf{bits}}}(x) = H_{w',\mathsf{bits}}(x) = w'(x) - \sum_{i=0}^{N-1} \sum_{j=0}^{\log Q-1} \mathsf{bits}(x,i,j) \cdot 2^j \cdot X^i = 0.$$

Hence $Q''_{w',\mathsf{bits}} \equiv 0$ by definition. Since bits takes values in $\{0,1\}$, we have $Q'_{\mathsf{bits}} \equiv 0$ by lemma 7. Now we prove the converse. Suppose Q'_{bits} and $Q''_{w',\mathsf{bits}}$ are both zero polynomials. By lemma 7, bits must take values in $\{0,1\}$. Recall that $\widetilde{\mathsf{eq}}(x,x') = \mathsf{eq}(x,x') = 1_{x=x'}$ when $x,x' \in \{0,1\}^s$. Thus, for every $x \in \{0,1\}^{s-1}$,

$$0 = Q''_{w',\mathsf{bits}}(x) = \widetilde{H_{w',\mathsf{bits}}}(x) = H_{w',\mathsf{bits}}(x).$$

Plugging in the definition of $H_{w',\mathsf{bits}}(x)$ and using the property that bits takes values in $\{0,1\}$, we get

$$w'(x) = \sum_{i=0}^{N-1} \sum_{j=0}^{\log Q-1} \mathsf{bits}(x,i,j) \cdot 2^j \cdot X^i \in \mathbb{Z}_Q[X]^{<N}.$$

Lemma 9. *Let $w' \in (\mathsf{Gal}(Q,d)[X]^{<N})^{\frac{m}{2}}$. Then $w' \in (\mathbb{Z}_Q[X]^{<N})^{\frac{m}{2}}$ and $\phi^{-1}(w')$ is a witness of $\mathbb{x}$ if and only if there exists a function $D : \{0,1\}^s \to \mathsf{Gal}(Q,d)[X]$ and a function $\mathsf{bits} : \{0,1\}^{s-1} \times \{0,1\ldots,N-1\} \times \{0,1,\ldots,\log Q-1\} \to \mathsf{Gal}(Q,d)$ such that $Q_{w',D}, Q'_{\mathsf{bits}}$ and $Q''_{w',\mathsf{bits}}$ are zero polynomials.*

Proof. Following lemma 6 and lemma 8, we have almost everything, except that D now takes values in $\mathsf{Gal}(Q,d)[X]$ rather than $\mathbb{Z}_Q[X]$. Fortunately, by lemma 6, we have that $G_{w',D}$ and $Q_{w',D}$ are equivalent, with $Q_{w',D} = 0$. Combining this with lemma 5, we obtain $G_{w',D}(x) = \hat{F}_{w'}(x) - D(x) \cdot (X^N + 1) = 0$, which implies $D(x) = \frac{\hat{F}_{w'}(x)}{X^N+1}$. Since w' and $\hat{F}_{w'}(x)$ take values in $\mathbb{Z}_Q[X]$, it follows that $D(x)$ also takes values in $\mathbb{Z}_Q[X]$.

For the convenience of describing the interactive argument, we additionally define the following functions and polynomials. For a polynomial f in variable X, which might have other variables, we sometimes use $f|_{X=\alpha}$ or simply $f|_\alpha$ to denote the result obtained by letting X take value α; the result $f|_\alpha$ is a polynomial in other variables or a value when X is the only variable.

– For $r \in (\mathsf{Gal}(Q,d))^s$, we define $\overline{A'}, \overline{B'}, \overline{C'}$, three s-variate polynomials in $\mathsf{Gal}(Q,d)[X]$, as

$$\overline{A'}(r) \stackrel{\mathsf{def}}{=} \sum_{y \in \{0,1\}^s} \widetilde{A'}(r,y)\widetilde{Z'}(y),$$
$$\overline{B'}(r) \stackrel{\mathsf{def}}{=} \sum_{y \in \{0,1\}^s} \widetilde{B'}(r,y)\widetilde{Z'}(y),$$
$$\overline{C'}(r) \stackrel{\mathsf{def}}{=} \sum_{y \in \{0,1\}^s} \widetilde{C'}(r,y)\widetilde{Z'}(y).$$

– For $r \in (\mathsf{Gal}(Q,d))^{s-1}$ and $\alpha \in \mathsf{Gal}(Q,d)$, define

$$\overline{\mathsf{bits}}(r) \stackrel{\mathsf{def}}{=} \sum_{i=0}^{N-1} \sum_{j=0}^{\log Q-1} \widetilde{\mathsf{bits}}(r,i,j) \cdot 2^j \cdot X^i \in \mathsf{Gal}(Q,d)[X].$$

and

$$\mathsf{bits}_r : \{0,1\}^{s'} \mapsto \mathsf{Gal}(Q,d), (i,j) \mapsto \mathsf{bits}(r,i,j),$$

where we interpret the s'-bit input string as a pair $(i,j) \in \{0,\ldots,N-1\} \times \{0,\ldots,\log Q-1\}$. We use this notation henceforth. Define

$$\widetilde{\mathsf{bits}_{r,\alpha}}(\sigma) \stackrel{\mathsf{def}}{=} \widetilde{\mathsf{bits}_r}(\sigma)\widetilde{W_\alpha}(\sigma)$$

where $\widetilde{W_\alpha}$ is the MLE of function W_α defined below

$$W_\alpha : \{0,1\}^{s'} \to \mathsf{Gal}(Q,d), (i,j) \mapsto 2^j \cdot \alpha^i.$$

– For $\tau \in (\mathsf{Gal}(Q,d))^s$ and $\alpha \in \mathsf{Gal}(Q,d)$, we define the polynomial $\mathcal{G}_{1,\tau,\alpha} \in \mathsf{Gal}(Q,d)[X_1,\ldots,X_s]$ as

$$\mathcal{G}_{1,\tau,\alpha}(x) \stackrel{\mathsf{def}}{=} \widetilde{\mathsf{eq}}(\tau,x) \cdot \widetilde{G_{w',D}}(x)|_{X=\alpha}.$$

– For $\tau \in (\mathsf{Gal}(Q,d))^{s-1}$ and $\alpha \in \mathsf{Gal}(Q,d)$, we define the polynomial $\mathcal{G}_{2,\tau,\alpha} \in \mathsf{Gal}(Q,d)[X_1,\ldots,X_{s-1}]$ as

$$\mathcal{G}_{2,\tau,\alpha}(x) \stackrel{\mathsf{def}}{=} \widetilde{\mathsf{eq}}(\tau,x) \cdot \widetilde{H_{w',\mathsf{bits}}}(x)|_{X=\alpha}.$$

– For $\tau \in (\mathsf{Gal}(Q,d))^{s-1}$, define polynomial $\mathcal{G}_{3,\tau} \in \mathsf{Gal}(Q,d)[X_1,\ldots,X_{s-1}]$ as

$$\mathcal{G}_{3,\tau}(x) \stackrel{\mathsf{def}}{=} \widetilde{\mathsf{eq}}(\tau,x) \cdot [\widetilde{\mathsf{bits}}(x)^2 - \widetilde{\mathsf{bits}}(x)].$$

– For $r \in (\mathsf{Gal}(Q,d))^s$ and $r_A, r_B, r_C, \alpha \in \mathsf{Gal}(Q,d)$, define an s-variate polynomial over $\mathsf{Gal}(Q,d)$ as

$$M_{r,r_A,r_B,r_C,\alpha}(y) \stackrel{\mathsf{def}}{=} r_A\widetilde{A'}(r,y)|_\alpha \cdot \widetilde{Z'}(y)|_\alpha + r_B\widetilde{B'}(r,y)|_\alpha \cdot \widetilde{Z'}(y)|_\alpha + r_C\widetilde{C'}(r,y)|_\alpha \cdot \widetilde{Z'}(y)|_\alpha.$$

Now, we summarize the whole interactive protocol below.

Interactive argument of knowledge for RICS over ring $\mathcal{R}$

Ingredients:

– Set $d = \omega(\log\lambda)\log N$.
– PC: an extractable polynomial commitment for $\mathcal{F}_{Comb}[\mathsf{Gal}(Q,d), s-1, N]$.
– PC': an extractable polynomial commitment for $\mathcal{F}_{Comb}[\mathsf{Gal}(Q,d), s, 3N]$.
– PC'': an extractable polynomial commitment for $\mathcal{F}_{MUL}[\mathsf{Gal}(Q,d), s+s']$.

Procedures:

- $\mathsf{Setup}(1^\lambda)$: sample $\mathsf{pp}_{\mathsf{PC}} \leftarrow \mathsf{PC.Setup}(1^\lambda), \mathsf{pp}'_{\mathsf{PC}} \leftarrow \mathsf{PC}'.\mathsf{Setup}(1^\lambda), \mathsf{pp}''_{\mathsf{PC}} \leftarrow \mathsf{PC}''.\mathsf{Setup}(1^\lambda)$, and output $\mathsf{pp} := (\mathsf{pp}_{\mathsf{PC}}, \mathsf{pp}'_{\mathsf{PC}}, \mathsf{pp}''_{\mathsf{PC}})$.
- $\langle \mathcal{P}(\mathsf{pp}, \mathbb{x}, w), \mathcal{V}(\mathsf{pp}, \mathbb{x}) \rangle$ where $\mathbb{x} = (\mathcal{R}, A, B, C, m)$.
  1. Both parties parse $\mathsf{pp} = (\mathsf{pp}_{\mathsf{PC}}, \mathsf{pp}'_{\mathsf{PC}}, \mathsf{pp}''_{\mathsf{PC}})$.
  2. $\mathcal{P}$: Compute w', D, bits from $(\mathbb{x}, w)$ and
     - $(\mathsf{com}_{w'}, \mathsf{st}_{w'}) \leftarrow \mathsf{PC.Commit}(\mathsf{pp}_{\mathsf{PC}}, \widetilde{w'})$;
     - $(\mathsf{com}_D, \mathsf{st}_D) \leftarrow \mathsf{PC}'.\mathsf{Commit}(\mathsf{pp}'_{\mathsf{PC}}, \widetilde{D})$;
     - $(\mathsf{com}_{\mathsf{bits}}, \mathsf{st}_{\mathsf{bits}}) \leftarrow \mathsf{PC}''.\mathsf{Commit}(\mathsf{pp}''_{\mathsf{PC}}, \widetilde{\mathsf{bits}})$.

     Send $(\mathsf{com}_1, \mathsf{com}_2, \mathsf{com}_3)$ to $\mathcal{V}$.
  3. $\mathcal{V}$: Sample $\tau_1 \leftarrow E^s, \tau_2 \leftarrow E^{s-1}, \tau_3 \leftarrow E^s, \alpha_1 \leftarrow E, \alpha_2 \leftarrow E$ and send them all to $\mathcal{P}$. Additionally sample $r_1 \leftarrow E^s, r_2 \leftarrow E^{s-1}, r_3 \leftarrow E^s$.
  4. **sumcheck #1.** Run sumcheck protocols.
     - $e_1 \leftarrow \mathsf{SumCheck}\langle P_{SC}(\mathcal{G}_{1,\tau_1,\alpha_1}), V_{SC}(r_1)\rangle(s, 3, 0)$.
     - $e_2 \leftarrow \mathsf{SumCheck}\langle P_{SC}(\mathcal{G}_{2,\tau_2,\alpha_2}), V_{SC}(r_2)\rangle(s-1, 3, 0)$.
     - $e_3 \leftarrow \mathsf{SumCheck}\langle P_{SC}(\mathcal{G}_{3,\tau_3}), V_{SC}(r_3)\rangle(s+s', 2, 0)$.
  5. $\mathcal{P}$: Compute
     - $v_A := \overline{A'}(r_1)|_{\alpha_1}, v_B := \overline{B'}(r_1)|_{\alpha_1}, v_C := \overline{C'}(r_1)|_{\alpha_1}, v_D := \widetilde{D}(r_1)|_{\alpha_1} \in \mathsf{Gal}(Q, d)$;
     - $v_2 := \widetilde{w'}(r_2)|_{\alpha_2}, \widehat{v}_2 := \overline{\mathsf{bits}}(r_2)|_{\alpha_2} \in \mathsf{Gal}(Q, d)$;
     - $v_3 := \widetilde{\mathsf{bits}}(r_3) \in \mathsf{Gal}(Q, d)$.

     Send $(v_A, v_B, v_C, v_D, v_2, \widehat{v}_2, v_3)$ to $\mathcal{V}$.
  6. Check the following openings; reject unless they all accept.
     - $\mathsf{PC.Eval}\langle \mathcal{P}(\mathsf{pp}_{\mathsf{PC}}, (\mathsf{com}_{w'}, (r_2, \alpha_2), v_2), \mathsf{st}_{w'}), \mathcal{V}(\mathsf{pp}_{\mathsf{PC}}, (\mathsf{com}_{w'}, (r_2, \alpha_2), v_2))\rangle$
     - $\mathsf{PC}'.\mathsf{Eval}\langle \mathcal{P}(\mathsf{pp}'_{\mathsf{PC}}, (\mathsf{com}_D, (r_1, \alpha_1), v_D), \mathsf{st}_D), \mathcal{V}(\mathsf{pp}'_{\mathsf{PC}}, (\mathsf{com}_D, (r_1, \alpha_1), v_D))\rangle$
     - $\mathsf{PC}''.\mathsf{Eval}\langle \mathcal{P}(\mathsf{pp}''_{\mathsf{PC}}, (\mathsf{com}_{\mathsf{bits}}, r_3, v_3), \mathsf{st}_{\mathsf{bits}}), \mathcal{V}(\mathsf{pp}''_{\mathsf{PC}}, (\mathsf{com}_{\mathsf{bits}}, r_3, v_3))\rangle$
  7. $\mathcal{V}$: Check the following equations; reject unless they all hold true.
     - $e_1 \stackrel{?}{=} \widetilde{\mathsf{eq}}(\tau_1, r_1) \cdot (v_A \cdot v_B - v_C - v_D \cdot (\alpha_1^N + 1))$.
     - $e_2 \stackrel{?}{=} \widetilde{\mathsf{eq}}(\tau_2, r_2) \cdot (v_2 - \widehat{v}_2)$.
     - $e_3 \stackrel{?}{=} \widetilde{\mathsf{eq}}(\tau_3, r_3) \cdot (v_3^2 - v_3)$.
  8. $\mathcal{V}$: Sample $r_A, r_B, r_C \leftarrow E$ and send them to $\mathcal{P}$.
  9. **sumcheck #2.** Run sumcheck protocols.
     - $e'_1 \leftarrow \mathsf{SumCheck}\langle \mathcal{P}(\widetilde{M_{r_1,r_A,r_B,r_C,\alpha_1}}), \mathcal{V}(r'_1)\rangle(s, 2, r_A v_A + r_B v_B + r_C v_C)$
     - $e'_2 \leftarrow \mathsf{SumCheck}\langle \mathcal{P}(\widetilde{\mathsf{bits}_{r_2,\alpha_2}}), \mathcal{V}(r'_2)\rangle(s+s', 2, \widehat{v}_2)$
  10. $\mathcal{P}$: Compute $v'_1 = \widetilde{w'}(r'_1[1...])|_{\alpha_1}, v'_2 = \widetilde{\mathsf{bits}}(r_2, r'_2), \in \mathsf{Gal}(Q, d)$ and send (v'_1, v'_2) to $\mathcal{V}$.
  11. $\mathcal{V}$: Let $\sigma := (r_2, r'_2) \in \{0,1\}^{s+s'}$. Check the following openings; reject unless they all accept.
      - $\mathsf{PC.Eval}\langle \mathcal{P}(\mathsf{pp}_{\mathsf{PC}}, (\mathsf{com}_{w'}, (r'_1[1...], \alpha_1), v'_1), \mathsf{st}_{w'}), \mathcal{V}(\mathsf{pp}_{\mathsf{PC}}, (\mathsf{com}_{w'}, (r'_1[1...], \alpha_1), v'_1))\rangle$
      - $\mathsf{PC}''.\mathsf{Eval}\langle \mathcal{P}(\mathsf{pp}'_{\mathsf{PC}}, (\mathsf{com}_{\mathsf{bits}}, \sigma, v'_2), \mathsf{st}_{\mathsf{bits}}), \mathcal{V}(\mathsf{pp}'_{\mathsf{PC}}, (\mathsf{com}_{\mathsf{bits}}, \sigma, v'_2))\rangle$
  12. $\mathcal{V}$: Check the following equations; reject unless they all hold true.

- $e'_1 \stackrel{?}{=} (r_A v'_A + r_B v'_B + r_C v'_C) \cdot v_Z$, where

$$v'_A := \widetilde{A'}(r_1, r'_1)|_{\alpha_1},\ v'_B := \widetilde{B'}(r_1, r'_1)|_{\alpha_1},\ v'_C := \widetilde{C'}(r_1, r'_1)|_{\alpha_1}$$
$$v_Z := (1 - r'_1[0]) \cdot v'_1 + r_1[0] \cdot \widetilde{(1, io)}(r'_1[1...])|_{\alpha_1}.$$

- $e'_2 \stackrel{?}{=} v'_2 \cdot \widetilde{W}_{\alpha_1}(r'_2)$.

13. $\mathcal{V}$: Accept.

Theorem 5. *The protocol above is a secure public-coin interactive argument of knowledge for R1CS over the ring $\mathcal{R} = \mathbb{Z}_Q[X]/(X^N + 1)$ when $\mathsf{PC}, \mathsf{PC}', \mathsf{PC}''$ are instantiated with extractable polynomial commitments constructed in theorem 3.*

Proof. Completeness follows directly from the fact that if the Ring-R1CS instance is satisfiable, the honest prover can construct all required polynomials so that they all vanish when evaluated according to the protocol. Concretely, given a valid witness $w \in \mathcal{R}^{\frac{m}{2}}$, one derives polynomials $Q_{w',D}, Q'_{\mathsf{bits}}, Q''_{w',\mathsf{bits}}$ that are zero polynomials. In each sub-protocol (including all sumcheck protocols), the verifier's checks pass with probability 1 by completeness of the sumcheck protocol, and all polynomial openings in the commitment scheme are correct. Hence, these prove the completeness.

Then we prove the knowledge soundness. Here we construct the knowledge extractor $\mathcal{E}$ as first obtaining the commitments $\mathsf{com}_{w'}, \mathsf{com}_D, \mathsf{com}_{\mathsf{bits}}$, from P^*, and then using the extractor of the extractable polynomial commitment to obtain a polynomial $w' \in (\mathsf{Gal}(Q,d)[X]^{<N})^{m/2}$ and functions D, bits, and set $w := \phi^{-1}(w')$ as the output witness. By lemma 9, if $(\mathsf{pp}, \mathbb{x}, w) \notin \mathfrak{R}$, then at least one of $Q_{w',D}, Q'_{\mathsf{bits}}, Q''_{w',\mathsf{bits}}$ is not a zero polynomial. Hence, we have

$$\begin{aligned}
&\mathbf{Pr}\left[\langle \mathcal{P}^*(\mathsf{pp}, \mathbb{x}, \mathsf{st}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\rangle = 1 \wedge (\mathsf{pp}, \mathbb{x}, w) \notin \mathfrak{R}\right] \\
\leq &\mathbf{Pr}\left[\langle \mathcal{P}^*(\mathsf{pp}, \mathbb{x}, \mathsf{st}), \mathcal{V}(\mathsf{pp}, \mathbb{x})\rangle = 1 \wedge (Q_{w',D} \neq 0 \vee Q''_{w',\mathsf{bits}} \neq 0 \vee Q'_{\mathsf{bits}} \neq 0)\right] \\
\leq &\mathbf{Pr}\left[\mathcal{V} \text{ accepts } \wedge Q_{w',D} \neq 0\right] + \mathbf{Pr}\left[\mathcal{V} \text{ accepts } \wedge Q''_{w',\mathsf{bits}} \neq 0\right] \\
&+ \mathbf{Pr}\left[\mathcal{V} \text{ accepts } \wedge Q'_{\mathsf{bits}} \neq 0\right].
\end{aligned}$$

Instead of directly checking whether $Q_{w',D}$ and $Q''_{w',\mathsf{bits}}$ are zero polynomials, the verifier first substitutes the variable X with random elements $\alpha_1, \alpha_2 \in E$. By the generalized Schwartz-Zippel lemma, this introduces soundness error

$$\epsilon_{\mathsf{substitute}} \leq \frac{\deg_X(Q_{w',D})}{|E|} + \frac{\deg_X(Q''_{w',\mathsf{bits}})}{|E|} \leq \frac{O(N)}{|E|}.$$

The protocol runs three sumcheck sub-protocols in the first stage. By the generalized Schwartz–Zippel lemma and union bound, the soundness error of these sumcheck sub-protocols is

$$\epsilon_{\mathsf{sumcheck\#1}} \leq \frac{3s}{|E|} + \frac{3(s-1)}{|E|} + \frac{2(s+s')}{|E|} \leq \frac{O(s+s')}{|E|}.$$

Next, the protocol reduces from checking v_A, v_B, v_C to checking $r_A v_A + r_B v_B + r_C v_C$, which introduces soundness error $\epsilon_{\mathsf{combine}} \leq \frac{1}{|E|}$. The protocol then runs two sumcheck sub-protocols in the second stage. By the generalized Schwartz–Zippel lemma and union bound, the soundness error of these sumcheck sub-protocols is

$$\epsilon_{\mathsf{sumcheck\#2}} \leq \frac{2s}{|E|} + \frac{2(s+s')}{|E|} \leq \frac{O(s)}{|E|}.$$

Finally, the prover might still convince the verifier although one of e_i or e'_i derived from the protocol is not the correct evaluation of the corresponding polynomial. By the argument of knowledge and binding property of the polynomial commitment, such an event occurs only with probability at most $\varepsilon_{\mathsf{polycommit}} \leq 2\left(\frac{\sqrt{mN\log Q}}{|E|} + 0.9^{\lambda}\right) + \varepsilon_{\mathsf{Merkle}}(\lambda)$ (by theorem 3). In total, the knowledge soundness error of the protocol is at most

$$\begin{aligned}&\varepsilon_{\mathsf{substitute}} + \varepsilon_{\mathsf{sumcheck\#1}} + \varepsilon_{\mathsf{combine}} + \varepsilon_{\mathsf{sumcheck\#2}} + \varepsilon_{\mathsf{polycommit}} \\ &= O\left(\frac{N + \sqrt{mN\log Q}}{|E|} + 0.9^{\lambda} + \varepsilon_{\mathsf{Merkle}}\right).\end{aligned}$$

Efficiency. We analyze the complexity of our interactive argument for RICS over the ring $\mathcal{R}$. Recall that $s = \lceil \log m \rceil$ and $s' = \lceil \log(N \log Q) \rceil$.

- Prover time: The main bottlenecks of the prover's computation are as follows. Committing to w', D, bits requires computing $O(2^{s+s'})$ Merkle hashes. Running the sumcheck protocols as the prover costs $O(2^{s+s'})$ Galois ring operations [3]. The computation of v_A, v_B, v_C costs $O(m^2)$ Galois ring operations. The evaluation protocol of the polynomial commitment requires $O(2^{s+s'})$ Galois ring operations. The total cost is $O(m^2 + 2^{s+s'})$ Galois ring operations and Merkle hashes.
- Verifier time: The bottleneck of the verifier consists of running the evaluation protocols of the polynomial commitments as the verifier, and evaluating multi-linear extensions of $A', B', C', \widetilde{(1, io)}, \widetilde{W}'_{\alpha}$ on a single point each [4]. The former (PCS openings) costs $O(\sqrt{2^{s+s'}}) = O(\sqrt{mN\log Q})$ ring operations according to theorem 3. The latter costs $O(m^2 + N\log Q)$ ring operations by the definition of multi-linear extensions, taking into account the substitution of X with α_1. The total cost is

$$O(\underbrace{m^2}_{\text{evaluating } A',B',C'} + \underbrace{\sqrt{mN\log Q}}_{\text{PCS openings}} + \underbrace{N\log Q}_{\text{evaluating } \widetilde{W}_{\alpha}})$$

  Galois ring operations.

[3] This can be done by employing [Tha13, WJB+17, XZZ+19] to implement a linear-time prover for the sumcheck protocol

[4] We ignore $\widetilde{\mathsf{eq}}$ as it is a simple polynomial and can be easily evaluated.

– Communication cost: The bottleneck appears in the evaluation protocols of the polynomial commitments. The total communication cost is $O(\sqrt{mN \log Q})$ Galois ring elements.

We conclude this section by transforming the interactive argument of knowledge in the previous section into a non-interactive argument of knowledge with sublinear proof size via the Fiat–Shamir transformation [FS86] in the Random Oracle Model (ROM).

Theorem 6 (Sublinear Proof for Ring-R1CS via Ring Switching). *Let λ be the security parameter and $\mathcal{R} = \mathbb{Z}_Q[X]/(X^N + 1)$ be a ring that depends on the security parameter, where $Q = p^s$ is a power of prime. Let $d = \log_p\left(2\sqrt{mN}\right) + \omega(\log \lambda)$. Assuming the existence of collision-resistant hash functions, there exists a non-interactive argument of knowledge (under the Fiat–Shamir heuristic) for R1CS over the ring $\mathcal{R}$ with the following efficiency and security characteristics.*

- ***Proof Size:*** *The proof size is $O(\sqrt{mN \log Q})$ elements in $\mathsf{Gal}(p^s, d)$.*
- ***Prover Time:*** *The prover's running time is $O_\lambda(m^2 + mN \log Q)$ operations in $\mathsf{Gal}(p^s, d)$.*
- ***Verifier Time:*** *The verifier's running time is $O(m^2 + \sqrt{mN \log Q} + N \log Q)$ operations in $\mathsf{Gal}(p^s, d)$.*

4.3 Proving the Witness Has a Small Norm

Let $\beta < Q$ be a power-of-two threshold. In lemma 8, we restrict each entry of witness w to be in the subring $\mathcal{R}$ by requiring that Q'_{bits} and $Q''_{w',\mathsf{bits}}$ to be zero polynomial. Recall that bits is a function with domain $\{0,1\}^{s-1} \times \{0, 1 \ldots, N-1\} \times \{0, 1, \ldots, \log Q - 1\}$ and we interpret it as follows: For every $x \in \{0,1\}^{s-1}$, the entry of w' encoded by x has bit decomposition

$$\sum_{i=0}^{N-1} \sum_{j=0}^{\log Q - 1} \mathsf{bits}(x, i, j) \cdot 2^j \cdot X^i \in \mathbb{Z}_Q[X]^{<N},$$

where $\mathsf{bits}(x, i, j)$ takes value in $\{0, 1\}$. Consider replacing $\log Q$ by $\log \beta$. Then the above sum equals to $\sum_{i=0}^{N-1} a_i X^i \in \mathbb{Z}_Q[X]^{<N}$ for some $a_0, \ldots, a_{N-1} \in \{0, 1, \ldots, \beta - 1\}$. This way, we can restrict every entry of w' to be a polynomial in $\mathbb{Z}_Q[X]^{<N}$ with coefficients smaller than β.

References

ÅAB+24. Marius A. Årdal, Diego F. Aranha, Katharina Boudgoust, Sebastian Kolby, and Akira Takahashi. Aggregating Falcon signatures with LaBRADOR. In *Advances in Cryptology – CRYPTO 2024, Part I*, volume 14920 of *Lecture Notes in Computer Science*, pages 71–106. Springer, 2024. 2, 3, 5

ABVPS24. Shahla Atapoor, Karim Baghery, Hilder VL Pereira, and Jannik Spiessens. Verifiable fhe via lattice-based snarks. *IACR Communications in Cryptology (CiC)*, 1(1), 2024. 2

AHIV17. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017. 14

BC25. Dan Boneh and Binyi Chen. Latticefold+: faster, simpler, shorter lattice-based folding for succinct proof systems. In *Annual International Cryptology Conference*, pages 327–361. Springer, 2025. 2, 3

BCG20. Jonathan Bootle, Alessandro Chiesa, and Jens Groth. Linear-time arguments with sublinear verification from tensor codes. In *Theory of Cryptography: 18th International Conference, TCC 2020, Durham, NC, USA, November 16–19, 2020, Proceedings, Part II 18*, pages 19–46. Springer, 2020. 12

BCOS20. Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. Efficient post-quantum snarks for rsis and rlwe and their applications to privacy. In *Post-Quantum Cryptography: 11th International Conference, PQCrypto 2020, Paris, France, April 15–17, 2020, Proceedings 11*, pages 247–267. Springer, 2020. 2, 6

BCPS18. Anurag Bishnoi, Pete L Clark, Aditya Potukuchi, and John R Schmitt. On zeros of a polynomial in a finite grid. *Combinatorics, Probability and Computing*, 27(3):310–333, 2018. 8

BGRV09. Zvika Brakerski, Shafi Goldwasser, Guy N Rothblum, and Vinod Vaikuntanathan. Weak verifiable random functions. In *Theory of cryptography conference*, pages 558–576. Springer, 2009. 2

BLNS23. Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Lattice-based blind signatures: Short, efficient, and round-optimal. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 16–29, 2023. 2

BS23. Ward Beullens and Gregor Seiler. Labrador: compact proofs for r1cs from module-sis. In *Annual International Cryptology Conference*, pages 518–548. Springer, 2023. 2, 3, 5

BSCI+20. Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed–solomon codes. *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 900–909, 2020. 14

BSCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part II 14*, pages 31–60. Springer, 2016. 13

CMNW24. Valerio Cini, Giulio Malavolta, Ngoc Khanh Nguyen, and Hoeteck Wee. Polynomial commitments from lattices: post-quantum security, fast verification and transparent setup. In *Annual International Cryptology Conference*, pages 207–242. Springer, 2024. 5, 12

DP24. Benjamin E Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. *Cryptology ePrint Archive*, 2024. 3

DPLS18. Rafaël Del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 574–591, 2018. 2

EKS+21. Muhammed F Esgin, Veronika Kuchta, Amin Sakzad, Ron Steinfeld, Zhenfei Zhang, Shifeng Sun, and Shumo Chu. Practical post-quantum few-time verifiable random function with applications to algorand. In *International Conference on Financial Cryptography and Data Security*, pages 560–578. Springer, 2021. 2

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer, 1986. 27

GLS+23. Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S Wahby. Brakedown: Linear-time and field-agnostic snarks for r1cs. In *Annual International Cryptology Conference*, pages 193–226. Springer, 2023. 6, 12, 13, 14

GNSV23. Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: Snarks for ring arithmetic. *Journal of Cryptology*, 36(4):41, 2023. 2

HLMZ24. Mi-Ying Miryam Huang, Baiyu Li, Xinyu Mao, and Jiapeng Zhang. Fully homomorphic encryption with efficient public verification. *Cryptology ePrint Archive*, 2024. 2, 6, 18

JLX+25. Yuhao Jia, Songsong Li, Chaoping Xing, Yizhou Yao, and Chen Yuan. Polynomial commitments for galois rings and applications to snarks over z 2 k. In *Annual International Cryptology Conference*, pages 515–548. Springer, 2025. 5

KLOT25. Shuto Kuriyama, Russell WF Lai, Michał Osadnik, and Lorenzo Tucci. Salsaa–sumcheck-aided lattice-based succinct arguments and applications. *Cryptology ePrint Archive*, 2025. 3

NKS25. David Nevado, Dohoon Kim, and Miha Stopar. Lattice-based signature aggregation. Ethereum Research forum post, May 2025. Benchmark data for LaBRADOR-based Falcon aggregation. 2, 4

NOZ26. Ngoc Khanh Nguyen, George O'Rourke, and Jiapeng Zhang. Hachi: Efficient lattice-based multilinear polynomial commitments over extension fields. Cryptology ePrint Archive, Paper 2026/156, 2026. 3

NS24. Ngoc Khanh Nguyen and Gregor Seiler. Greyhound: fast polynomial commitments from lattices. In *Annual International Cryptology Conference*, pages 243–275. Springer, 2024. 5

NZZ15. Phong Q Nguyen, Jiang Zhang, and Zhenfeng Zhang. Simpler efficient group signatures from lattices. In *IACR international workshop on public key cryptography*, pages 401–426. Springer, 2015. 2

QBC13. Guillaume Quintin, Morgan Barbier, and Christophe Chabot. On generalized reed–solomon codes over commutative and noncommutative rings. *IEEE transactions on information theory*, 59(9):5882–5897, 2013. 9

Set20. Srinath Setty. Spartan: Efficient and general-purpose zksnarks without trusted setup. In *Annual International Cryptology Conference*, pages 704–737. Springer, 2020. 5, 20

Tha13. Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In *Annual Cryptology Conference*, pages 71–89. Springer, 2013. 26

TWZ26. Louis Tremblay Thibault, Michael Walter, and Jiapeng Zhang. Practical snargs for matrix multiplications over encrypted data. *Cryptology ePrint Archive*, 2026. 3

Wan11. Zhe-Xian Wan. *Finite fields and Galois rings*. World Scientific Publishing Company, 2011. 8

WJB+17. Riad S Wahby, Ye Ji, Andrew J Blumberg, Abhi Shelat, Justin Thaler, Michael Walfish, and Thomas Wies. Full accounting for verifiable outsourcing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2071–2086, 2017. 26

XZZ+19. Tiacheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39*, pages 733–764. Springer, 2019. 26

ZWL+25. Xinxuan Zhang, Ruida Wang, Zeyu Liu, Binwu Xiang, Yi Deng, and Xianhui Lu. Fhe-snark vs. snark-fhe: From analysis to practical verifiable computation. *Cryptology ePrint Archive*, 2025. 2, 3

A Proximity Gap of Interleaved Linear Codes over Ring

Lemma 10 (Lemma 3 restated). *Let $C \subseteq R^M$ be linear code over ring R with minimum distance d, and let $C^m \subseteq R^{m\times M}$ be the m-fold interleaved code of C. Then the following holds for all $u \in R^{m\times M}$. Define*

$$H \stackrel{\mathsf{def}}{=} \{j \in [M] : \exists i \in [m] \ \textit{s.t.}\ u_i[j] \neq c_i[j]\},$$

where u_i denotes the i-th row of u and $c_i \in C$ is the codeword closest to u_i. For every $h \in \mathbb{N}$ such that $d \geq 4h$ and $|H| \geq h$, it holds that

$$\Pr_{\alpha_1,\ldots,\alpha_m \leftarrow R}[\mathsf{d}(\alpha_1 u_1 + \cdots \alpha_m u_m, C) < h] \leq \frac{h}{\mathsf{L}(R)}.$$

Proof. Write $S \stackrel{\mathsf{def}}{=} Ru_1 + \cdots + Ru_m$. Let $\mathcal{D}$ be the distribution of $\alpha_1 u_1 + \cdots \alpha_m u_m$ where $\alpha_1, \ldots, \alpha_m \leftarrow R$.

Claim. For every $v \in S$, the following distribution is identical to $\mathcal{D}$:

- $\mathcal{D}'$: sample $\alpha_1, \ldots, \alpha_m \leftarrow R$ and output $w := v + \alpha_1 u_1 + \cdots \alpha_m u_m$.

Proof (of Claim). Write $v = \beta_1 u_1 + \cdots + \beta_m u_m$ where $\beta_1, \ldots, \beta_m \in R$. It suffices to prove that for all $w^* \in S$, $\Pr_{w\leftarrow\mathcal{D}'}[w = w^*] = \Pr_{w\leftarrow\mathcal{D}}[w = w^*]$. Fix $w^* \in S$. Clearly,

$$\Pr_{w\leftarrow\mathcal{D}}[w = w^*] = |A|/|R|^m, \Pr_{w\leftarrow\mathcal{D}'}[w = w^*] = |A'|/|R|^m$$

where

$$\begin{aligned} A &= \{(\alpha_1, \ldots, \alpha_m) \in R^m : \alpha_1 u_1 + \cdots \alpha_m u_m = w^*\}, \\ A' &= \left\{(\alpha'_1, \ldots, \alpha'_m) \in R^m : v + \alpha'_1 u_1 + \cdots \alpha'_m u_m = w^*\right\} \\ &= \left\{(\alpha'_1, \ldots, \alpha'_m) \in R^m : (\alpha'_1 + \beta_1)u_1 + \cdots (\alpha'_m + \beta_m)u_m = w^*\right\}. \end{aligned}$$

Note that $(\alpha_1, \ldots, \alpha_m) \mapsto (\alpha_1 - \beta_1, \ldots, \alpha_m - \beta_m)$ is a bijection between A and A', and the claim follows.

Let E be an exceptional set of R with $|E| = \mathsf{L}(R)$. We consider two cases.
Case 1. There exists some $v^* \in S$ such that $\mathsf{d}(v^*, C) \geq 2h$. By the claim above, if one samples $\alpha \leftarrow E, w' \leftarrow \mathcal{D}$ and set $w := \alpha v^* + w'$, then the distribution of w is still $\mathcal{D}$. Hence,

$$\Pr_{w \leftarrow \mathcal{D}}[\mathsf{d}(w, C) < h] = \Pr_{\alpha \leftarrow E, w' \leftarrow \mathcal{D}}[\mathsf{d}(\alpha v^* + w', C) < h].$$

We shall prove that for every fixed value of w', there exists at most one $\alpha \in E$ satisfying $\mathsf{d}(\alpha v^* + w', C) < h$; this would imply

$$\Pr_{\alpha \leftarrow E, w' \leftarrow \mathcal{D}}[\mathsf{d}(\alpha v^* + w', C) < h] \leq \frac{1}{|E|}.$$

Fix $w' \in S$ and assume towards contradiction that there are $\alpha, \alpha' \in E$ such that $\alpha \neq \alpha'$ and $\mathsf{d}(\alpha v^* + w', C) < h, \mathsf{d}(\alpha' v^* + w', C) < h$. Since C is a linear code, by triangle inequality we have $\mathsf{d}((\alpha - \alpha')v^*, C) < 2h$, meaning that there exists a codeword $c \in C$ such that $\mathsf{d}((\alpha - \alpha')v^*, c) \leq 2h$. Since $(\alpha - \alpha')$ is a unit, we have

$$\mathsf{d}(v^*, C) \leq \mathsf{d}(v^*, (\alpha - \alpha')^{-1} \cdot c) = \mathsf{d}((\alpha - \alpha') \cdot v^*, c) < 2h.$$

which contradicts the choice of v^*.
Case 2. For all $v \in S$, $\mathsf{d}(v, C) < 2h$. For $i \in [m]$, define $H_i \stackrel{\mathsf{def}}{=} \{j \in [M] : u_i[j] \neq c_i[j]\}$. Clearly, $H = \cup_{i \in [m]} H_i$. Since $2h \leq d/2$, each u_i can be uniquely expressed as $u_i = c_i + \delta_i$ where $c_i \in C$ and the non-zero entries of δ_i is exactly H_i. We shall prove that for every $j \in H$,

$$\Pr_{w \leftarrow \mathcal{D}}[j \notin \Delta(w, C) \wedge \mathsf{d}(w, C) < h] \leq \frac{1}{|E|}.$$

Let B_j denote the event that $j \notin \Delta(w, C) \wedge \mathsf{d}(w, C) < h$. Note that if $|\Delta(w, C)| = \mathsf{d}(w, C) < h = |H|$, there must be some $j \in H$ such that $j \notin \Delta(w, C)$, meaning B_j does happen. Hence, by union bound,

$$\Pr_{w \leftarrow \mathcal{D}}[\mathsf{d}(w, C) < h] \leq \Pr_{w \leftarrow \mathcal{D}}\left[\cup_{j \in H} B_j\right] \leq \frac{|H|}{|E|} \leq \frac{h}{|E|}.$$

It remains to prove $\mathbf{Pr}_{w \leftarrow \mathcal{D}}\left[B_j\right] \leq 1/|E|$ for all $j \in H$. Fix $j \in H$, say, $j \in H_i$. Again, by the claim above,

$$\Pr_{w \leftarrow \mathcal{D}}[B_i] = \Pr_{\alpha \leftarrow E, w' \leftarrow \mathcal{D}}[j \notin \Delta(\alpha u_i + w', C) \wedge \mathsf{d}(\alpha u_i + w', C) < h].$$

It suffices to show that, for every fixed value of w', there exists at most one $\alpha \in E$ such that $\Delta(\alpha u_i + w', C) \wedge \mathsf{d}(\alpha u_i + w', C) < h$.

Fix $w' \in S$ and assume towards contradictions that there exists $\alpha, \alpha' \in E$ such that $\alpha \neq \alpha'$ and

$$j \notin \Delta(z, C),\ \mathsf{d}(z, C) < h, \quad j \notin \Delta(z', C),\ \mathsf{d}(z', C) < h,$$

where $z \stackrel{\mathsf{def}}{=} \alpha u_i + w'$ and $z' \stackrel{\mathsf{def}}{=} \alpha' u_i + w'$. Since $\mathsf{d}(z, C) < h$ and $\mathsf{d}(z', C) < h$, z and z' can be uniquely written as $z = c_z + \delta_z, z' = c_{z'} + \delta_{z'}$ where $c_z, c_{z'} \in C$ and the non-zero entries of δ_z and $\delta_{z'}$ are exactly $\Delta(z, C)$ and $\Delta(z', C)$ respectively. Since $u_i = (\alpha - \alpha')^{-1}(z - z')$, we have $c_i = c_z + c_{z'}$ and $\Delta(u_i, C) \subseteq \Delta(z, C) \cup \Delta(z', C)$; thus, $j \notin \Delta(u_i, C)$, contradicting with $j \in H_i$.