Orbweaver: Succinct Linear Functional Commitments from Lattices

Ben Fisch
benjamin.fisch@yale.edu
Yale University

Zeyu Liu
zeyu.liu@yale.edu
Yale University

Psi Vesely[*]
psi.vesely@yale.edu
Yale University

December 13, 2024

Abstract

We present Orbweaver, a plausibly post-quantum functional commitment for linear relations that achieves quasilinear prover time together with $O(\log n)$ proof size and polylogarithmic verifier time. Orbweaver enables evaluation of linear functions on committed vectors over cyclotomic rings and the integers. It is extractable, preprocessing, non-interactive, structure-preserving, and supports compact public proof aggregation. The security of our scheme is based on the k-R-ISIS assumption (and its knowledge counterpart), whereby we require a trusted setup to generate a universal structured reference string. We use Orbweaver to construct succinct univariate and multilinear polynomial commitments.

Concretely, our scheme has smaller proofs than most other succinct post-quantum arguments for large statements. For binary vectors of length 2^{30} we achieve 302KiB linear map evaluation proofs with evaluation binding, and 1MiB proofs when extractability is required; for 32-bit integers these sizes are 494KiB and 1.6MiB, respectively.

This is an extended version of the work that appeared at CRYPTO 2023.

Contents

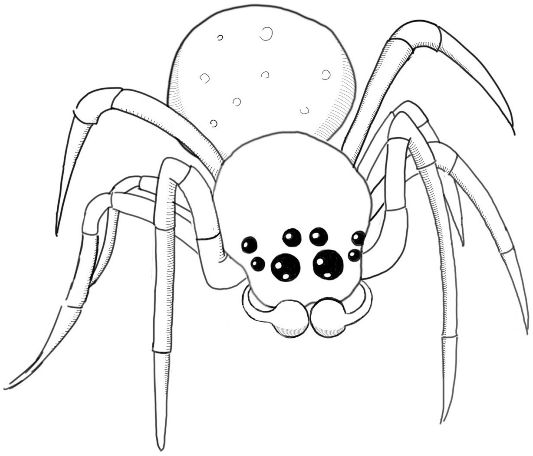
[*] Corresponding author.

1 Introduction

Over the last decade there has been tremendous progress in the development succinct non-interactive argument of knowledge protocols, called SNARKs [BCCT12]. For a public arithmetic circuit C and a public input x, these SNARKs enable a prover to convince a verifier that the prover knows a witness w such that $C(\mathsf{x}, \mathsf{w}) = 1$, where:

- the proof is short, its length is at most $\mathrm{poly}(\log|C|,\ |\mathsf{x}|,\ \lambda)$;
- the proof can be verified in time $\mathrm{poly}(\log|C|,\ |\mathsf{x}|,\ \lambda)$; and
- generating the proofs takes quasilinear time in $|C|$.

Here $|C|$ is the number of gates in C, and λ is a security parameter. The logarithmic dependence on $|C|$ makes these proofs remarkably short and fast to verify. Since the verifier needs to at least read C, there is also a one-time pre-processing phase applied to the circuit C. Once pre-processed, the prover can provide proofs for many x.



Lattice Orbweaver (Araneus thaddeus) by Jackie P (CC BY 4.0)

Beyond knowledge of a witness, many of these proof systems can be efficiently extended to also provide zero-knowledge. This progress generated considerable real-world interest, and there are now implemented SNARKs and zkSNARKs capable of handling statements involving many millions of arithmetic gates that are now deployed in real-world applications (e.g., [Ben+14]).

There are multiple techniques for constructing a pre-processing SNARK. Some require a trusted setup [Gro10; Lip12; BCCT13; GGPR13; PHGR13; BCIOP13; Gro16; GM17; MBKM19; GWC19; CHMMVW20] where a trusted party must honestly generate public parameters. This setup is called *universal* if it is not specific to the circuit C, i.e., it can be done once to generate parameters that can be used to preprocess any circuit C in a publicly verifiable way. Other systems, called transparent SNARKs, require no trusted setup [BBHR19; BFS19; COS19; BGH19]. Many of the existing SNARKs make use of a commitment scheme called a *polynomial commitment* [KZG10], or PC. A polynomial commitment enables a prover to commit to a polynomial $f \in \mathbb{F}[x]$ of degree w using a short commitment. Later, given two public values $\alpha, \beta \in \mathbb{F}$, the prover can convince a verifier that the committed polynomial f satisfies $\beta = f(\alpha)$ and that f has degree at most w. Ideally, the prover can verifiably open the polynomial at any point $\alpha \in \mathbb{F}$ using a short proof that can be efficiently checked. In fact, it has been shown that given polynomial commitment scheme with proof size $S(w)$ and verification time $T(w)$ it is possible to construct SNARKs with the same complexity characteristics in w (the length of the witness), and additional complexity dependent only on a security parameter λ [BFS20; CHMMVW20]. This compilation is in the random oracle model and relies on the Fiat-Shamir transform.

Polynomial commitment schemes are a special case of *linear functional vector commitments*, where the prover has a commitment $C = \mathsf{com}(\mathbf{x})$ to a vector $\mathbf{x} \in \mathbb{Z}_p^w$ and is able to open any linear form $f(\mathbf{x}) = \sum_{i=1}^{w} \mathbf{x}_i f_i \bmod p$. Polynomial commitments, and the more general linear vector commitments, have been built from bilinear pairings [KZG10], groups of unknown order [BFS19], proofs of proximity for Reed-Solomon codes [BBHR18], and to some degree from lattice-based assumptions [ACLMT22]. However, thus far all lattice-based constructions have had significant drawbacks, and none have achieved asymptotically (in the polynomial's degree) a quasilinear prover time with logarithmic proof size and logarithmic verification time. There are lattice-based generalizations of Bulletproofs [BCCGP16; BBBPWM18] based on the Ring-SIS assumption, which achieve quasilinear prover time and a polylogarithmic proof size, but have linear verification time [BLNS20; ACK21; AL21]. Recently, [ACLMT22] constructed general degree-d polynomial map vector commitments (which include linear functions $d = 1$ as a special case), which achieve logarithmic proof size and verification time, but have a CRS size and prover runtime that is at least $\Omega(w^{2d})$, and thus quadratic in the vector length for linear functions. This construction requires a trusted setup using lattice trapdoor sampling [MP12], and is based on k-R-ISIS, a new family of lattice-based knowledge assumptions related to Ring-SIS. Another recent work, LaBRADOR [BS22], uses recursion to achieve very compact proof sizes, but has verification time linear in w.

Polynomial commitment schemes based on codes and lattice assumptions are of particular importance due to their plausible post-quantum security. Constructions based on Reed-Solomon codes have quasilinear prover time and both polylogarithmic proof size and verifier time. So far they outperform any lattice-based construction not only asympotically, but also concretely by orders of magnitude in overall size and verification time, even when compared with recent lattice-based constructions that sacrifice prover performance for shorter proofs. Moreover, in the random oracle model, code-based constructions use weaker assumptions than lattice-based constructions. Nonetheless, we are optimistic that the additional structure lattices provide vs. generic (i.e., hash and code-based) constructions can be exploited such that lattice SNARKs eventually surpass these results. As a point of reference, hash-based signatures were originally more efficient, but after over a decade of development lattice-based signatures are an order of magnitude smaller and two faster.

A primary motivation of recent lattice-based constructions [ACLMT22; WW22] of vector commitments supporting higher degree polynomial map openings is that, unlike linear function commitments, they can be used to build SNARKs more directly (e.g., by opening an R1CS form) without invoking the Fiat-Shamir transform. Unfortunately, current approaches to supporting higher degree polynomial maps seem to fundamentally require a quadratic prover time. Given the additional structure of lattices compared with code-based proof systems that rely only on hash functions, one might expect that it would be possible to obtain smaller proof sizes and faster verification times. The recent work LaBRADOR [BS22] was the first lattice-based system to achieve proof sizes smaller (both concretely and asymptotically) than code-based systems, but this has not yet been done for combined proof size and verification time. The results of this paper make progress in this direction.

1.1 Our results

Building off of the techniques in [ACLMT22], we present the first functional commitment for linear relations from lattices that asymptotically has quasilinear prover time, logarithmic proof size, and polylogarithmic verifier time. The scheme supports commitments to vectors $\mathbf{x} \in \mathcal{R}^w$ over a cyclotomic ring $\mathcal{R}$ of degree n and openings of the form $\sum_{i=1}^{w} x_i \cdot f_i$ for any $\mathbf{f} \in \mathcal{R}^w$ mod q for a prime $q \gg \alpha$, where both $\mathbf{x}$ and $\mathbf{f}$ have norm bounded by α. In particular, proofs are $O(\log w\alpha)$ and verification time is $O(\log w\alpha \cdot \log\log w\alpha \cdot \log\log\log w\alpha)$.

We present several extensions to $\mathsf{Orbweaver}$ and show how it can be used to build other primitives including:

- A public proof aggregation protocol achieving $O(\log t)$ size aggregate proofs for t linear maps evaluations where each linear map being evaluated and committed input may be distinct.
- An inner product argument.
- A linear functional commitment for the non-cyclotomic ring $\mathcal{R} = \mathbb{Z}$ that combines ternary decomposition and proof aggregation to achieve sublogarithmic proofs in the input norm of size $O(\log w + \log\log \alpha)$.
- Univariate and multilinear polynomial commitments. For degree d univariate polynomials we achieve $O(\log d)$ openings and polylogarithmic verifier time.

We also provide concrete proof size comparisons to previous lattice-based proof systems. While succinct hash-based arguments provide better proof sizes for smaller statements, the most efficient of these systems still have $O(\log^2 n)$ proofs. Especially in settings where the input norm is small, Orbweaver provides better proof sizes for larger-but-practical circuits than state-of-the-art hash-based arguments.

1.2 Related work

Efficient lattice-based proof systems have been constructed in the designated verifier setting [GMNO18; ISW21]. In the publicly verifiable setting, there are many results that achieve succinct proof sizes but require linear verifier time [BBCPGL19; BLNS20; ACK21; BCS21; BS22]. There are also works focusing on practical lattice-based ZK proofs, which have proof size linear in the witness size, but are concretely efficient for small statements [ESLL19; LNS20; ENS20; ESLR22; LNP22].

Recently, Albrecht et al. [ACLMT22] constructed extractable lattice-based vector commitments supporting arbitrary degree polynomial maps. For degree d maps their prover time and CRS length are $\tilde{O}(w^{2d})$, while proof size and verifier time are $O(d\log(w))$. Wee and Wu [WW22] construct non-extractable lattice-based vector commitments for linear functions including polynomial commitments with $\mathrm{polylog}(w)$ openings and $\tilde{O}(w^2)$ CRS size and prover time. Castro and Peikert [CP22] construct non-extractable vector commitments for functions of bounded complexity based on the standard SIS assumption. Their polynomial commitments achieve proof size $O(\log^4 w)$. Balbás, Catalano, Fiore, and Lai construct extractable lattice-based vector commitments supporting arithmetic circuits of width w and depth d that achieve proof size $O(d\log^2 w)$, but have a CRS of size $O(w^5)$ [BCFL22]. LaBRADOR [BS22] designs a SNARK for quadratic relations with quasilinear prover and verifier time, but via a complex recursive argument achieve $O(\log^2 w)$ proofs that concretely surpass the best results from hash-based proof systems.

There is a much longer history of succinct arguments constructed from Merkle hashes that begins with [Kil92] and more recently includes FRI [BBHR19], Ligero [AHIV17], Aurora [BCRSVW19], Brakedown [GLSTW21], and Orion [XZS22] among others.

1.3 Technical overview

At a high level, our construction uses an assumption from the k-R-ISIS assumption family introduced in [ACLMT22] to translate the knowledge-of-exponent (KEA) based linear vector commitment scheme sketched in [AC20, section 9] (which is in turn based on [Gro10], and also independently similar to ideas in [LRY16]) to the lattice setting. The construction should enable a prover to commit to a vector $\boldsymbol{x}$ and later open any linear function $f(\boldsymbol{x}) = \langle [f_i]_{i=1}^{w},\ [x_i]_{i=1}^{w}\rangle = \sum_{i=1}^{w} f_i x_i$.

As observed in [AC20], given a function $x(v) = \sum_{i=1}^{w} x_i v^i$, and a function $f(v) = \sum_{i=0}^{w-1} f_i v^{-i}$, then $h(v) = x(v)f(v) = \sum_{i=-w+1}^{w-1} a_i v^i$, where $a_0 = \sum_{i=1}^{w} f_i x_i$. The idea is that the prover sends

$c = x(v)$, $y = f(\boldsymbol{x})$, and $\pi = \sum_{i=-w+1, i \neq 0}^{w-1} a_i v^i$, and then the verifier computes $f(v) = \mathsf{ck}_f$ and checks $c \cdot \mathsf{ck}_f - y = \pi$.

In order to achieve both binding and succinctness, we situate this abstract protocol in lattice setting using the techniques of [ACLMT22]. A trusted setup generates a universal (i.e., the same setup works for any linear function up to a given size) structured reference string (SRS) $(\boldsymbol{a}, v, \boldsymbol{u}_{-w+1}, \ldots, \boldsymbol{u}_{-1}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{w-1}) \in \mathcal{R}_q^\ell \times \mathcal{R}_q^\times \times (\mathcal{R}_q^\ell)^{2w-2}$, where for each $\boldsymbol{u}_i$ it holds that $\langle \boldsymbol{a},\ \boldsymbol{u}_i \rangle \equiv v^i \bmod q$ and $\boldsymbol{u}_i$ is "short" relative to q. Note that no preimage $\boldsymbol{u}_0$ is given out such that $\langle \boldsymbol{a},\ \boldsymbol{u}_0 \rangle \equiv 1 \bmod q$. Our k-R-ISIS assumption states that it should be hard to find short $(s, \boldsymbol{u}) \neq (0, \boldsymbol{0}) \in \mathcal{R} \times \mathcal{R}^\ell$ such that $\langle \boldsymbol{a},\ \boldsymbol{u} \rangle \equiv s \bmod q$ even given these preimages for other powers of v. In general, it should be hard to find the preimage of any target not in the linear span of $[v_i]_{i=-w+1, i \neq 0}^{w-1}$.

Let α be a bound on the ℓ-infinity norm of both the witness and function. The computation of commitment c, commitment key ck_f, and y are exactly as in the abstract protocol, but taken mod q for prime $q \gg \alpha$. The prover now computes

$$\pi_0 := \sum_{i=-w+1, i \neq 0}^{w-1} a_i \boldsymbol{u}_i \bmod q \ \ ,$$

and the verifier checks

$$\langle \boldsymbol{a},\ \pi_0 \rangle \equiv c \cdot \mathsf{ck}_f - y \bmod q \ \ , \tag{1}$$

and that π_0 and y have small norm. When correctly executing the protocol it follows from the fact that $\alpha \ll q$ and $\beta \ll q$, where β is a norm bound on the $\boldsymbol{u}_i$, that the proof and output will have small norm. Fixing $\mathcal{R}_q$ with respect to the security parameter, one can see that the size of π_0 is of size $O(\log w + \log \alpha)$. We also achieve (almost) the same verifier time [1] in the preprocessing setting: while computing the commitment key $\mathsf{ck}_f = \sum_{i=1}^{w} f_i v^{-i}$ takes linear time, it may be computed once in advance then subsequently used to verify openings with respect to any c and y.

Our scheme thus far achieves evaluation binding: an adversary who can open a commitment c to two different outputs for the same function can be used to break k-R-ISIS. To see this, we subtract one verifying Eq. (1) from the other to obtain

$$\langle \boldsymbol{a},\ \pi_0 - \pi_0' \rangle \equiv y' - y \bmod q$$

Since both proofs and outputs are small, this gives k-R-ISIS solution $s = y' - y, \boldsymbol{u} = \pi_0 - \pi_0'$.

Achieving extractability. While this construction is evaluation binding, this is insufficient to guarantee that the prover knows some witness consistent with the commitment. For example, the prover can randomly sample a short proof π_0, and compute the commitment $c = \langle \boldsymbol{a},\ \pi_0 \rangle / (\sum_{i=1}^{w} f_i v^{-i})$. In this case, the prover can pass the verifier checks without knowledge of the input.

To achieve extractability we require an additional knowledge assumption from [ACLMT22], one from the knowledge k-R-ISIS assumption family. Our knowledge k-R-ISIS assumption states that there is an extractor that extracts short $\boldsymbol{x}^*$ s.t. $c = \sum_{i=1}^{w} x_i^* v^i$ from any prover who, given $[\boldsymbol{u}_i']_{i \in [w]}$ s.t. $\langle \boldsymbol{a},\ \boldsymbol{u}_i' \rangle \equiv v^i \cdot t \bmod q$ (for some t drawn from a special subset of $\mathcal{R}$), outputs commitment c and a short knowledge proof π_1 s.t. $\langle \boldsymbol{a},\ \pi_1 \rangle \equiv c \cdot t \bmod q$.

[1] Our verifier time is $O(\log(w \cdot \alpha) \log\log(w \cdot \alpha))$ in the preprocessing setting.

Using this assumption, it suffices to include as part of the commitment an additional value $\pi_1 = \sum_{i=1}^{w} x_i \boldsymbol{u}_i$ in order to achieve extractability. Observe that if for a verifying (c, y, π_0, π_1) it held that $f(\boldsymbol{x}^*) \neq y$, then $\boldsymbol{x}^*$ could be used to create verifying $(c, y' = f(\boldsymbol{x}^*), \pi_0', \pi_1')$, which violates binding and could be used to break plain k-R-ISIS as explained above. Therefore, the extracted witness must be consistent with the evaluation.

Lastly, we note our extracted witness has "stretch", i.e., the best norm bound we can hope to enforce on $\boldsymbol{x}^*$ is the norm bound on π_1, which in order to achieve correctness is necessarily bigger than the norm bound we impose on $\boldsymbol{x}$.

Public proof aggregation. An untrusted prover, input t verifying tuples $(c_i, \pi_{0,i}, \pi_{1,i}, y_i)$ tuples, can combine the evaluation proofs into a single proof $\pi_0 := \sum_{i=1}^{t} h_i \pi 0_i$ using some short challenges h_i. The verifier then checks

$$\langle \boldsymbol{a},\ \pi_0 \rangle \equiv \mathsf{ck} \cdot \sum_{i=1}^{t} (h_i c_i) - \sum_{i=1}^{t} (h_i y_i) \bmod q \ \ .$$

Since the h_i and $\pi_{0,i}$ are short, π_0 will be as well. If we consider the norm bound on the h_i to be dependent on the security parameter, then we can see the bit length $|\pi_0|$ is $\log t$ times the bit length of the original proofs $|\pi_{0,i}|$.

Without modifying our knowledge assumption, we cannot take linear combinations of the knowledge proofs. So we only get partial aggregation.

Polynomial commitments for integers. Furthermore, for the special case of polynomial commitments, where $f_i = z^{i-1} \bmod p$ for $i \in [1, w]$ and w is a power of 2 and $q > p^{\log w}$, $\mathsf{ck}_f = \prod_{i=0}^{\log w - 1} (z_i + v^{-2^i}) v^{-(w+1)} - z_{\log w} \bmod q$ where $z_i = z^{2^i} \bmod p$, which can be computed in time $O(\log w)$ operation over $\mathcal{R}_q$. Given this value the verifier can compute $c \cdot \mathsf{ck}_f = (\sum_{i=1}^{w} x_i v^i)(\sum_{i=1}^{w} f_i v^{-i}) = \sum_{i=1}^{w} f_i x_i + \sum_{i=-w+1}^{w-1} a_i v^i = f(\boldsymbol{x}) + \sum_{i=-w+1}^{w-1} a_i v^i$, for some ring elements a_i. The opening proof will include information that allows the verifier to cancel out the term dependent on the a_i values. To prevent the prover from cheating, we invoke the k-R-ISIS assumption by involving $\boldsymbol{a}$ in the opening. More specifically, the opening is computed as $\pi_0 = \sum_{i \in \mathbb{Z}(w) \setminus \{0\}} a_i \boldsymbol{u}_i$, where a_i's can be computed by the prover. The verifier of the opening checks $\langle \boldsymbol{a},\ \pi_0 \rangle = c \cdot \mathsf{ck}_f - y$, and that f, y, π_0 are short (i.e., their norm is bounded). The k-R-ISIS hardness assumption guarantees that the commitment is evaluation binding when f and x are both short.

2 Preliminaries

Let $\mathbb{Z}(b) := (-b, b) \cap \mathbb{Z}$ and $[a] := [1, a] \cap \mathbb{Z}$. For a ring $\mathcal{R}$ of degree n, let $\mathsf{vec}(r) \in \mathbb{Z}^n$ denote the coefficient vector of $r \in \mathcal{R}$ in the integral basis.

For $m \in \mathbb{N}$, let $\zeta_m \in \mathbb{C}$ be any fixed primitive m-th root of unity. Let $\mathcal{R} = \mathbb{Z}[\zeta_m]$ denote its ring of integers, called a cyclotomic ring. We have $\mathcal{R} \cong \mathbb{Z}[x]/\langle \Phi_m(x) \rangle$, where $\Phi_m(x)$ is the m-th cyclotomic polynomial. If m is a power of 2, we call $\mathcal{R}$ a power-of-2 cyclotomic ring. In this paper, we exclusively use power-of-2 cylotomic rings. Let $q \in \mathbb{N}$ be a prime number, we let $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ and let $\mathcal{R}_q^\times$ denote all invertible elements in $\mathcal{R}_q$. For any $f \in \mathcal{R}$, let $\mathsf{ct}(f)$ denote the constant term of f (i.e., $\mathsf{ct}(f) = \mathsf{vec}(f)_0$).

For $x \in \mathcal{R}$, let $\|x\|$ denote the ℓ-infinity norm of its coefficient vector, i.e., $\|x\| = \max_{i \in [n]} \mathsf{vec}(x)$. We use $\|\cdot\|_p$ for the ℓ_p-norm (e.g., $\|\cdot\|_2$ for the ℓ_2 norm).

Definition 2.1 (Ring expansion factor)**.** Let $\mathcal{R}$ be a ring. The *expansion factor* of $\mathcal{R}$ is defined as $\gamma_{\mathcal{R}} := \max_{a,b\in\mathcal{R}} \frac{\|a\cdot b\|}{\|a\|\cdot\|b\|}$.

Theorem 2.2 ([AL21])**.** *If $\mathcal{R} = \mathbb{Z}[\zeta_m]$ is a power-of-2 cyclotomic ring, then $\gamma_{\mathcal{R}} \leq n$.*

Theorem 2.3 ([ACLMT22])**.** *Let $q = \omega((w\cdot f)^{f/\phi(m)})$ be a rational prime such that $\mathcal{R}_q$ splits into f fields each of size $q^{\phi(m)/f}$. For $v \leftarrow\$ \mathcal{R}_q^w$, we have $v \in (\mathcal{R}_q^\times)^w$ with probability $(1 - 1/q^{\phi(m)/f})^{w\cdot f}$.*

Subsequently in this work, we set q large enough so that uniformly random $v \leftarrow\$ \mathcal{R}_q$ satisfies $v \in \mathcal{R}_q^\times$ with non-negligible probability.

The conjugation automorphism. The cyclotomic ring $\mathcal{R}$ has a group of automorphisms $\sigma_{-1}(\mathcal{R})$ that is isomorphic to $\mathbb{Z}_{2n}^\times$,

$$i \to \sigma_i : \mathbb{Z}_{2n}^\times \to \sigma_{-1}(\mathcal{R})$$

We make use of the following property: given two vectors $\mathsf{vec}(a) = (a_0, \ldots, a_{kn-1}) \in \mathbb{Z}^{kn}$, $\mathsf{vec}(b) = (b_0, \ldots, b_{kn-1}) \in \mathbb{Z}^{kn}$ where $k \geq 1$, let $c \leftarrow \sum_{i=0}^{k-1} \sigma_{-1}(\sum_{j=0}^{n-1} a_{i\cdot n+j} X^j) \cdot (\sum_{j=0}^{n-1} b_{i\cdot n+j} X^j) \in \mathcal{R}$; then the constant coefficient denoted as $\mathsf{ct}(c)$ has the property that $\mathsf{ct}(c) = \langle \mathsf{vec}(a),\ \mathsf{vec}(b)\rangle$ (adapted from [LNP22, Lemma 2.4], see also [ENS20]).

2.1 Exceptional challenge sets

For a cyclotomic ring $\mathcal{R}$ of degree n, let $\mathcal{H} \subset \mathcal{R}$ be the set of ring elements with c plus or minus one coefficients and $n - c$ zero coefficients. Then $|\mathcal{H}| = 2^c\binom{n}{c}$, and if $\mathcal{R}_q$ is a power-of-two cyclotomic, the operator norm of $\mathcal{H}$ with respect to the ℓ-infinity norm is c, i.e., $\mathsf{op}(\mathcal{H}) = c$ (Lemma 2.5). By [LS20, Corollary 1.2] for n a power of two and $q \equiv 2n + 1 \bmod 4n$ prime the $2n$-th cyclotomic polynomial $X^n + 1$ splits completely into linear factors modulo q.[2] Further, infinitely many such primes q exist and for any $h \in \mathcal{R}$ that satisfies $0 < \|h\|_2 < q^{1/n}$ has an inverse in $\mathcal{R}_q$. We wish to ensure that the difference between any two distinct elements in the challenge set is not a zero divisor, i.e., $\mathcal{H}$ is *exceptional*, in order to invoke the following theorem.

Theorem 2.4 (Generalized Alon–Füredi Theorem [BCPS18])**.** *Let $\mathcal{R}$ be a ring and let $\mathcal{H}$ be an exceptional, non-empty, finite subset of $\mathcal{R}$. Then for any non-zero polynomial $f \in \mathcal{R}[Y_1, \ldots, Y_t]$ such that $\deg(f) < |H|$ it holds that*

$$\Pr[f(h_1, \ldots, h_t) = 0 \mid h_1, \ldots, h_t \leftarrow\$ \mathcal{H}] \leq \frac{\deg(f)}{|\mathcal{H}|} \ .$$

Concretely, for $\deg(f) = 1$ and $c = 19$ the right-hand of side of the inequality above is $< 2^{-132}$. We know that differences $c_i - c_j$ for $c_i \neq c_j \in \mathcal{H}$ with 19 ± 2 coefficients can be reduced to the invertibility of 2 and polynomials with 18 ± 1 coefficients. Differences with 18 ± 2 coefficients and 2 ± 1 coefficients have ℓ_2 norm $\sqrt{74}$. We must then have $q \geq 2^{1590} > \sqrt{74}^{512}$. It turns out even larger q are optimal for proof size for concrete instantiations of our scheme capable of handling large circuits.

[2] A close reading of the proof reveals for the fully splitting case that the simpler condition $q \equiv 1 \bmod n$ suffices.

Lemma 2.5 (Operator norm of sparse ternary sets)**.** *For power-of-two n, let $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1)$ and $1 \le c \le n$. The set*

$$\mathcal{H} = \{h \in \mathcal{R}_q \mid \|h\|_\infty = 1 \wedge \|h\|_1 = c\}.$$

has operator norm

$$\mathsf{op}_\infty(\mathcal{H}) = \max_{h \in \mathcal{H}, r \in \mathcal{R}} \frac{\|hr\|_\infty}{\|h\|_\infty \|r\|_\infty} = c$$

Proof. Let $\boldsymbol{r}, \boldsymbol{c}$ be the coefficients of r and c. Let $g = rc$ with coefficient vector $\boldsymbol{g}$. Then

$$g_k = \sum_{i+j=k} r_i c_j - \sum_{i+j=n+k} r_i c_j$$

Fixing any $j, k \in \{0, \ldots, n-1\}$, there is exactly one $i \in \{0, \ldots, n-1\}$ such that either $i + j = n$ or $i + j = n + k$. Therefore, each coefficient g_k can be bounded by $c\|r\|_\infty$. □

2.2 Functional commitments

Definition 2.6 (Functional commitment)**.** A (pre-processing non-interactive) functional commitment (FC) scheme is parameterized by a function family

$$\mathcal{F} \;\; := \{\mathcal{F}_w \subseteq \{f : \mathcal{X}^w \to \mathcal{Y}\}\}_{w \in \mathbb{N}}$$

over a ring $\mathcal{R}$ for input alphabet $\mathcal{X} \subseteq \mathcal{R}$ and image space $\mathcal{Y} \subseteq \mathcal{R}$, where w is the dimension of the secret (committed) input. The FC scheme is defined by a 5-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Com}, \mathsf{Open}, \mathsf{PreVerify}, \mathsf{Verify})$, working as follows:

- $\mathsf{Setup}(1^\lambda, 1^w) \to (\mathsf{ck}, \mathsf{vk})$: Input (in unary) security parameter λ and secret input dimension w, samples commitment key ck and verification key vk.
- $\mathsf{Com}(\mathsf{ck}, \boldsymbol{x}) \to (c, \pi_1)$: Input commitment key ck and secret input $\boldsymbol{x} \in \mathcal{X}^w$, computes commitment c and a proof of knowledge π_1 of vector $\boldsymbol{x}$ such that $c = \mathsf{Com}(\mathsf{ck}, \boldsymbol{x})$.
- $\mathsf{Open}(\mathsf{ck}, f, \boldsymbol{x}) \to \pi_0$: Input commitment key ck, function $f \in \mathcal{F}_w$, and secret input $\boldsymbol{x} \in \mathcal{R}^w$, computes opening proof π_0 for the evaluation $f(\boldsymbol{x})$.
- $\mathsf{PreVerify}(\mathsf{ck}, f) \to \mathsf{vk}_f$: Input verification key vk and function $f \in \mathcal{F}_w$, computes preprocessed commitment key vk_f. Preprocessing only needs to be performed once per function and allows Verify to run in time sublinear in f.
- $\mathsf{Verify}(\mathsf{vk}_f, c, y, \pi_1, \pi_0) \to \{0, 1\}$: Input preprocessed verification key vk_f, commitment c, output $y \in \mathcal{Y}$, and proofs π_1 and π_0, the verifier returns 1 if the proofs convince them the verifier knows some $\boldsymbol{x} \in \mathcal{X}^w$ such that $\mathsf{Com}(\mathsf{ck}, \boldsymbol{x}) = c$ and $f(\boldsymbol{x}) = y$ (else 0).

We require that functional commitments satisfy correctness, extractability, and succinctness as defined below.

Definition 2.7 (Correctness)**.** An FC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is *correct* if for any $\lambda, w \in \mathbb{N}$, any $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda, 1^w)$, and for any $(f, \boldsymbol{x}, y) \in \mathcal{F} \times \mathcal{X}^w \times \mathcal{Y}$ satisfying $f(\boldsymbol{x}) = y$, any $(c, \pi_1) \leftarrow \mathsf{Com}(\mathsf{ck}, \boldsymbol{x})$, any $\pi_0 \leftarrow \mathsf{Open}(\mathsf{ck}, f, \boldsymbol{x})$, and any $\mathsf{vk}_f \leftarrow \mathsf{PreVerify}(\mathsf{ck}, f)$, it holds that $\Pr[\mathsf{Verify}(\mathsf{vk}_f, c, y, \pi_1, \pi_0) = 1] = 1$.

At a high level, extractability of an FC scheme requires that if an adversary can produce a commitment c and a valid opening (π_1, π_0) for some function f and some evaluation y, it must know $\boldsymbol{x} \in \mathcal{X}^w$ satisfying $\mathsf{Com}(\mathsf{ck}, \boldsymbol{x}) = c$ and $f(\boldsymbol{x}) = y$. Note that extractability implies *weak binding* (meaning that it is not possible to open a commitment to two different evaluations for the same function).

Definition 2.8 (Extractability)**.** A FC scheme for $(\mathcal{F}, \mathcal{X}, \mathcal{Y})$ is $(\kappa, \mathcal{X}^*)$-*extractable* if for any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{E}_{\mathcal{A}}$ that, input $(\mathsf{ck}, \mathsf{vk})$ and given black-box access to $\mathcal{A}$ and any randomness it uses, returns $\boldsymbol{x}^* \in (\mathcal{X}^*)^w$ such that

$$\Pr\left[\begin{array}{c} \mathsf{Verify}(\mathsf{vk}_f, c, y, \pi_1, \pi_0) = 1 \\ \wedge \left(\begin{array}{rl} & \boldsymbol{x} \notin (\mathcal{X}^*)^w \\ \vee & c \neq \mathsf{Com}(\mathsf{ck}, \boldsymbol{x},) \\ \vee & f(\boldsymbol{x}) \neq y \end{array}\right) \end{array} \middle| \begin{array}{r} (\mathsf{ck}, \mathsf{vk}) \leftarrow \mathsf{Setup}(1^\lambda, 1^w) \\ (f, c, y, \pi_1, \pi_0) \leftarrow \mathcal{A}(\mathsf{ck}, \mathsf{vk}) \\ \boldsymbol{x} \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{ck}, \mathsf{vk}) \\ \mathsf{vk}_f \leftarrow \mathsf{PreVerify}(\mathsf{vk}, f) \end{array}\right] \leq \kappa(\lambda, w) \ .$$

We say the scheme is $\mathcal{X}^*$-extractable if its *knowledge error* $\kappa(\lambda, w)$ is negligible in λ for $w = \mathrm{poly}(\lambda)$.

Definition 2.9 (Succinctness)**.** Let Π be a VC scheme for the alphabet $\mathcal{X} = \{r \in \mathcal{R} \mid \|r\| \leq \alpha\}$. We say Π is *succinct* if it satisfies the following properties:

- *Proof succinctness*: $|c + \pi_1 + \pi_0| = \mathrm{poly}(\log w + \log \alpha)$.
- *Verifier succinctness*: Verify runs in time $\mathrm{poly}(\log w + \log \alpha)$.

2.3 Sampling Algorithm

The following relies on the Leftover Hash Lemma over rings to generate some vector $\boldsymbol{a}$ that is indistinguishable from a uniformly randomly sample vector, and some trapdoor that makes it possible to easily generate vectors in the kernel of $\boldsymbol{a}$. We let $\mathsf{lhl}(\mathcal{R}_q, \beta)$ denote an algorithm that outputs the minimum $\ell \in \mathbb{N}$, which ensures that the resulting distribution of the vector $\boldsymbol{a}$ is indistinguishable from the uniform distribution. It is formally defined as follows, adapted from [GPV08; MP12; GM18; ACLMT22]:

A sampling algorithm has the following three PPT algorithms (taking 1^λ as input implicitly):

- $(\boldsymbol{a}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\ell, q, \mathcal{R}, \beta)$: takes dimension $\ell \in \mathbb{N}$, a modulus $q \in \mathbb{N}$, a ring $\mathcal{R}$, and a norm bound $\beta \in \mathbb{R}$, and outputs a vector $\boldsymbol{a} \in \mathcal{R}_q^\ell$ and a trapdoor td. For any $n \in \mathrm{poly}(\lambda), \ell \geq \mathsf{lhl}(\mathcal{R}_q, b)$ where $b = O(\beta)$, the distribution of $\boldsymbol{a}$ is within $\mathrm{negl}(\lambda)$ statistical distance to $U(\mathcal{R}_q^\ell)$.

- $\boldsymbol{u} \leftarrow \mathsf{SampD}(1^\ell, \mathcal{R}_q, \beta)$: for $\ell \geq \mathsf{lhl}(\mathcal{R}_q, \beta)$, outputs $\boldsymbol{u}$ such that $\|\boldsymbol{u}\| \leq \beta$ and the distribution of $\langle \boldsymbol{a},\ \boldsymbol{u} \rangle \bmod q$ is withing $\mathrm{negl}(\lambda)$ statistical distance to $U(\mathcal{R}_q)$.

- $\boldsymbol{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, v, \beta)$: for $\ell \geq \mathsf{lhl}(\mathcal{R}_q, \beta)$ and $v \in \mathcal{R}_q$, outputs $\boldsymbol{u} \in \mathcal{R}_q^\ell$ satisfying $\|\boldsymbol{u}\| \leq \beta$, $\langle \boldsymbol{a},\ \boldsymbol{u} \rangle \equiv v \bmod q$, and that distribution of $\boldsymbol{u}$ is within $\mathrm{negl}(\lambda)$ statistical distance the distribution of $v' \leftarrow \mathsf{SampD}(1^\lambda, 1^\ell, \mathcal{R}_q, \beta)$ conditioned on $\langle \boldsymbol{a},\ \boldsymbol{u} \rangle \equiv v' \bmod q$.

2.4 Cryptographic assumptions

The Short Integer Solution (SIS) problem was first introduced in [Ajt96], which asks to find a short element (of ℓ_2 norm) in the kernel of a random matrix over the ring $\mathbb{Z}_q$. An inhomogeneous version, ISIS, instead asks for a short solution to some linear equation system [Mic02]. It has been shown that the SIS and ISIS problems are equivalent.

We define the ring version of SIS (R-SIS) from [Mic02] as follows.

Definition 2.10 (R-SIS [Mic02])**.** Let $\mathcal{R}, q, \ell, \beta$ be parameters depending on λ. The R-SIS problem states the following: for $\boldsymbol{a} \leftarrow\$\ \mathcal{R}_q^\ell$ sampled uniformly at random, and $t = 0$, find $\boldsymbol{u} \neq \boldsymbol{0} \in \mathcal{R}^\ell$ such that $\|\boldsymbol{u}\| \leq \beta$ and $\langle \boldsymbol{a},\ \boldsymbol{u} \rangle \equiv t \bmod q$.

When $t \leftarrow\$\ \mathcal{R}_q$ this becomes the ring inhomogeneous SIS (R-ISIS) assumption, which is known to be equivalent. For appropriate parameters, there are no known efficient algorithms for solving R-SIS for cyclotomic rings.

2.4.1 *k-R-ISIS* assumptions

We define a family of assumptions over rings, k-R-ISIS, introduced in [ACLMT22]. k-R-ISIS assumptions can be trivially broken if some basic conditions are not satisfied, so we begin by defining those via the notion of k-R-ISIS admissibility:

Definition 2.11 (k-R-ISIS admissible)**.** Let $g \in \mathcal{R}(X)$ be a Laurent monomial, i.e., $g(X) := \boldsymbol{X}^{\boldsymbol{e}} := \prod_{i \in [y]} X_i^{e_i}$ for some exponent vector $\boldsymbol{e} \in \mathbb{Z}^y$. Let $\mathcal{G} \subset \mathcal{R}(X)$ be a set of Laurent monomials with $k := |\mathcal{G}|$. Let $g^* \in \mathcal{R}(X)$ be a *target* Laurent monomial. We say a *monomial family* $(\mathcal{G}, g^*)$ is k-R-ISIS $\mathsf{admissible}$ if the following conditions are satisfied:
1. All $g \in \mathcal{G}$ and g^* have constant degree
2. All $g \in \mathcal{G}$ are distinct
3. $0 \notin \mathcal{G}$
4. $g^* \notin \mathcal{G}$

Remark 2.12. Condition 1 rules out the monomials that depend on $\mathcal{R}$. Condition 2 rules out that trivial linear combinations of preimages give a preimage for the target. Condition 3 rules out a trivially producing multiple preimages of the same target. Condition 4 rules out the trivial solution to get the preimage of the target.

We then define the k-R-ISIS assumption as follows:

Definition 2.13 (k-R-ISIS assumption)**.** Let $\ell \in \mathbb{N}$. Let q be a rational prime, $\mathcal{R}$ a cyclotomic ring, and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. Let $\mathcal{G} \subset \mathcal{R}(X)$ be a set of y-variate Laurent monomials and let $g^* \in \mathcal{R}(X)$ be a target Laurent monomial such that $(\mathcal{G}, g^*)$ is k-R-ISIS-admissible. Let $\beta, \beta^* \geq 1$ be reals. For $g \in \mathcal{G},\ \ell \geq \mathsf{lhl}(\mathcal{R}_q, \beta),\ \boldsymbol{a} \in \mathcal{R}_q^\ell$, and $\boldsymbol{v} \in (\mathcal{R}_q^\times)^y$, let $\mathcal{D}_{g,\boldsymbol{a},\boldsymbol{v}}$ be a distribution over

$$\{\boldsymbol{u}_g \in \mathcal{R}^\ell \mid \langle \boldsymbol{a},\ \boldsymbol{u}_g \rangle \equiv g(\boldsymbol{v}) \bmod q \ \wedge\ \|\boldsymbol{u}_g\| \leq \beta\}\ .$$

Let $\mathcal{D} := \left\{\mathcal{D}_{g,\boldsymbol{a},\boldsymbol{v}} : \ell \in \mathbb{N},\ g \in \mathcal{G},\ \boldsymbol{a} \in \mathcal{R}_q^\ell,\ \boldsymbol{v} \in (\mathcal{R}_q^\times)^y\right\}$ be the family of these distributions. Define $\mathsf{pp} := (\mathcal{R}_q, \ell, y, \mathcal{G}, g^*, \mathcal{D}, \beta, \beta^*)$. The k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ $\mathsf{assumption}$ states that for any PPT adversary $\mathcal{A}$, we have $\mathsf{Adv}_{\mathsf{pp},\mathcal{A}}^{k\text{-}R\text{-}\mathsf{ISIS}}(\lambda) \leq \mathrm{negl}(\lambda)$, where $\mathsf{Adv}_{\mathsf{pp},\mathcal{A}}^{k\text{-}R\text{-}\mathsf{ISIS}}(\lambda)$ is the following probability:

$$\Pr\left[\begin{array}{cl|r} & \langle \boldsymbol{a},\ \boldsymbol{u}_{g^*} \rangle \equiv s^* \cdot g^*(\boldsymbol{v}) \bmod q & \boldsymbol{a} \leftarrow\$\ \mathcal{R}_q^\ell \\ \wedge & \|s^*\| \leq \beta^* & \boldsymbol{v} \leftarrow\$\ (\mathcal{R}_q^\times)^y \\ \wedge & \|\boldsymbol{u}_{g^*}\| \leq \beta^* & \boldsymbol{u}_g \leftarrow\$\ \mathcal{D}_{g,\boldsymbol{a},\boldsymbol{v}},\ \ \forall g \in \mathcal{G} \\ \wedge & (s^* \cdot g^*, \boldsymbol{u}_{g^*}) \neq (0, \boldsymbol{0}) & (s^*, \boldsymbol{u}_{g^*}) \leftarrow \mathcal{A}(\boldsymbol{a}, t, [\boldsymbol{u}_g]_{g \in \mathcal{G}}, \boldsymbol{v}) \end{array}\right].$$

Remark 2.14. For simplicity, we set t in [ACLMT22, Def 23] to be fixed to 1 and thus have $\langle \{t\} \rangle = \mathcal{R}_q$. The k-R-ISIS assumption requires v to be in $\mathcal{R}_q^\times$. Otherwise, the scheme can be insecure. For example, for a power-of-two $\mathcal{R}$, if $q = 2^k$, the ideal $q\mathcal{R}$ splits into $\mathcal{I}^{k\cdot\phi(n)}$ for some ideal $\mathcal{I}$ with norm 2. Then, for $v \leftarrow\$ \mathcal{R}_q$, $\Pr[v = 0 \mod \mathcal{I}] = 1/2$. Thus, $v^{k\cdot\phi(n)} = 0 \bmod q$. Therefore, we have $p(X) = X^{k\cdot\phi(n)}$ being a solution to any R-V-SIS instance over $\mathcal{R}_q$. Further, we require $1/|\mathcal{R}_q^\times|$ to be negligible such that SIS attacks are not possible in the image space. Theorem 2.3 shows how to pick parameters to guarantee this.

In [ACLMT22] they also introduce the following meta assumption:

Assumption 1 (k-R-ISIS meta assumption)**.** *For any k-R-ISIS admissible $(\mathcal{G}, g^*)$, k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,y,\mathcal{G},g^*,\mathcal{D},\beta,\beta^*}$ is hard if R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\beta^*}$ is hard and if R-$\mathsf{SIS}_{\mathcal{R}_q,k,\beta^*}$ is hard.*

To achieve extractability we also require an additional knowledge assumption, because by [GW11], we need an unfalsifiable assumption to prove the extractability, as follows:

Definition 2.15 (Knowledge k-R-ISIS)**.** Let the parameters $(\mathcal{R}_q, \ell, y, \mathcal{G}, \beta, \beta^*)$ be defined as in Definition 2.13. Let $\alpha^* \geq 1$ be a real. Let $\mathcal{T} \subset \mathcal{R}_q$ be such that for all $t \in \mathcal{T}$ it holds that:

1. $|\langle t\rangle|/|\mathcal{R}_q| = \mathrm{negl}(\lambda)$, and
2. finding $s' \in \mathcal{R}_q$ satisfying $s' \cdot t \equiv 0 \bmod q$ and $0 < \|s'\| \leq \alpha^*$ is hard.

For $g \in \mathcal{G},\ \ell \geq \mathsf{lhl}(\mathcal{R}_q, \beta),\ \boldsymbol{a} \in \mathcal{R}_q^\ell,\ t \in \mathcal{T}$, and $\boldsymbol{v} \in (\mathcal{R}_q^\times)^y$, let $\mathcal{D}_{g,\boldsymbol{a},t,\boldsymbol{v}}$ be a distribution over

$$\{\boldsymbol{u}_g \in \mathcal{R}^\ell \mid \langle \boldsymbol{a},\ \boldsymbol{u}_g\rangle \equiv g(\boldsymbol{v}) \cdot t \bmod q \ \wedge\ \|\boldsymbol{u}_g\| \leq \beta\}\ .$$

Let $\mathcal{D} := \left\{\mathcal{D}_{g,\boldsymbol{a},t,\boldsymbol{v}} : \ell \in \mathbb{N},\ g \in \mathcal{G},\ \boldsymbol{a} \in \mathcal{R}_q^\ell,\ t \in \mathcal{T},\ \boldsymbol{v} \in (\mathcal{R}_q^\times)^y\right\}$ be the family of these distributions. Define $\mathsf{pp} := (\mathcal{R}_q, \ell, y, \mathcal{G}, \mathcal{D}, \mathcal{T}, \alpha^*, \beta, \beta^*)$. The knowledge k-R-$\mathsf{ISIS}_{\mathsf{pp}}$ assumption states that for any PPT adversary $\mathcal{A}$ there exists a PPT extractor $\mathcal{E}_\mathcal{A}$ s.t. $\mathsf{Adv}_{\mathsf{pp},\mathcal{A}}^{k\text{-}R\text{-}\mathsf{ISIS}} \leq \mathrm{negl}(\lambda)$, where $\mathsf{Adv}_{\mathsf{pp},\mathcal{A}}^{k\text{-}R\text{-}\mathsf{ISIS}}$ is the following probability:

$$\Pr\left[\begin{array}{ll|r} & \langle \boldsymbol{a}, \boldsymbol{u}\rangle \equiv c \cdot t \bmod q & \\ \wedge & \|\boldsymbol{u}\| \leq \beta^* \ \wedge\ (c, \boldsymbol{u}) \neq (0, \boldsymbol{0}) & \boldsymbol{a} \leftarrow\$ \mathcal{R}_q^\ell \bmod q;\ t \leftarrow\$ \mathcal{T};\ \boldsymbol{v} \leftarrow\$ (\mathcal{R}_q^\times)^w \\ \wedge & \neg\left(\begin{array}{ll} & c = \sum_{g\in\mathcal{G}} x_g \cdot g(\boldsymbol{v}) \bmod q \\ \wedge & \|x_g\| \leq \alpha^*,\ \ \forall g \in \mathcal{G}\end{array}\right) & \boldsymbol{u}_g \leftarrow\$ \mathcal{D}_{g,\boldsymbol{a},\boldsymbol{v}},\ \ \forall g \in \mathcal{G} \\ & & ((c, \boldsymbol{u}), [x_g]_{g\in\mathcal{G}}) \leftarrow (\mathcal{A}\|\mathcal{E}_\mathcal{A})(\boldsymbol{a}, t, [\boldsymbol{u}_g]_{g\in\mathcal{G}}, \boldsymbol{v})\end{array}\right]\ .$$

To understand the first restriction on $\mathcal{T}$, consider an adversary who samples a random short $\boldsymbol{u}$ and then checks if $\langle \boldsymbol{a},\ \boldsymbol{u}\rangle \in \langle t\rangle$, setting the commitment to be c s.t. $\langle \boldsymbol{a},\ \boldsymbol{u}\rangle \equiv c \bmod q$ if so. Since $\langle \boldsymbol{a},\ \boldsymbol{u}\rangle$ is close to uniformly distributed over $\mathcal{R}_q$, restriction 1 ensures such an adversary succeeds with negligible probability. To understand the second restriction, consider an adversary who finds s' as above and outputs $(c, \boldsymbol{u}) = (s', s' \cdot \boldsymbol{u}_g)$ for an arbitrary $g \in \mathcal{G}$. Observe that $\boldsymbol{u}$ is short since s' and $\boldsymbol{u}_g$ are, and since $\langle \boldsymbol{a},\ \boldsymbol{u}_g\rangle \equiv g(v) \cdot t \bmod q$ it follows $\langle \boldsymbol{a},\ \boldsymbol{u}\rangle \equiv s' \cdot g(v) \cdot t \equiv 0 \equiv c \cdot t \bmod q$. However, we have that $s' \cdot g(\boldsymbol{v}) \not\equiv c \equiv s' \bmod q$ unless $g(\boldsymbol{v}) \equiv 1 \bmod q$, which should only happen with negligible probability.

In addition to the two contraints on $\mathcal{T}$ in our knowledge k-R-ISIS definition, in [ACLMT22] they give a third. Namely, that $1/|\langle t\rangle| = \mathrm{negl}(\lambda)$. We have omitted this constraint given it is implied by constraint 2. Let $\phi : \mathcal{R}_q \mapsto \langle t\rangle$. Then $|\mathsf{ker}(\phi)|/|\mathcal{R}_q| = 1/|\langle t\rangle|$. If $1/|\langle t\rangle|$ is not negligible, then $s' \leftarrow\$ \mathcal{R}_q$ satisfies $s' \cdot t \equiv 0 \bmod q$ with non-negligible probability.

We believe this assumption is suitable as it can be used to prove the extractability of our scheme as in the proof of Theorem 4.2, and is not trivially broken due to all the restrictions above. As in [ACLMT22], we can set $\mathcal{T}$ to contain all the elements t such that exactly half of the elements in the NTT representation of t are zero. Note this is only defined when $\langle q\rangle$ is not a prime ideal in $\mathcal{R}$; in this case $\mathcal{R}_q \simeq \mathbb{Z}_{q^n}$, which only has one NTT coefficient. Our restriction that $q \equiv 2n + 1 \bmod\ 4n$ ensures that $\langle q\rangle$ is not a prime ideal, and in particular that $\mathcal{R}_q$ is isomorphic to the direct product of n copies of $\mathbb{Z}_q$.

2.4.2 The *k-P-R-ISIS* assumption

Our construction relies on a particular assumption in the k-R-ISIS assumption family and its knowledge counterpart, parameterized by the monomial sets $[X^i]_{i\in\mathbb{Z}(w)\setminus\{0\}}$ and $[X^i]_{i\in[w]}$, respectively. This assumption could be seen as a lattice analogue of the k-Strong-Diffie-Hellman (k-SDH) problem, and we refer to it as the k-Powers R-ISIS (k-P-R-ISIS) assumption.

Assumption 2 (k-P-R-ISIS assumption)**.** *Define* k-P-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\mathcal{D},\beta,\beta^*} := k$-$R$-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,y,\mathcal{G},g^*,\mathcal{D},\beta,\beta^*}$ *with fixed* $\mathcal{G} = [X^i]_{i\in\mathbb{Z}(k)\setminus\{0\}}$*,* $g^*(X) = 1$*, and consequently* $y = 1$*. The* k-P-R-ISIS *assumption is hard if* R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\beta^*}$ *is.*

In the discrete log setting, this is akin to asking an adversary to compute g given $(g^{x^{w-1}}, \ldots, g^{x^{-1}}, g^x, \ldots, g^{x^w})$. Since $(\mathcal{G}, g^*)$ is k-M-ISIS admissible (Definition 2.11), our assumption follows from the much broader meta assumption of Lai et al. (Assumption 1). We also require the following knowledge assumption:

Assumption 3 (Knowledge k-P-R-ISIS assumption)**.** *Define knowledge* k-P-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\mathcal{D},\mathcal{T},\alpha^*,\beta,\beta^*}$ *as knowledge* k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,y,\mathcal{G},\mathcal{D},\mathcal{T},\alpha^*,\beta,\beta^*}$ *with* $\alpha^* \geq \beta^* \geq 1$*, fixed* $\mathcal{G} = [X^i]_{i\in[k]}$*, and consequently* $y = 1$*. The knowledge* k-P-R-ISIS *assumption holds if* R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell,\beta^*}$ *does.*

3 Cryptanalysis of k-P-R-ISIS

In this section, we consider how the k-P-R-ISIS assumption may be attacked. We begin by formulating a special case of R-SIS assumption when $\boldsymbol{a}$ is generated by the powers of a single ring element, which we call ring Vandermonde SIS (R-V-SIS). Then we show how an attacker who efficiently solves R-V-SIS can break the security of Orbweaver.

Definition 3.1. (R-V-SIS) Let $\mathcal{R}, q, \beta_V$ be parameters depending on λ and w. The R-V-SIS problem asks the following: given $\boldsymbol{v} := [v^i]_{i\in\mathbb{Z}(w)\setminus\{0\}} \in \mathcal{R}_q^{2w-3}$ for $v \leftarrow\$ \mathcal{R}_q^\times$, find $(s, \boldsymbol{z}) \neq (0, \boldsymbol{0}) \in \mathcal{R} \times \mathcal{R}^{2w-3}$ such that $0 < \|s\| \leq \beta_V$, $\|\boldsymbol{z}\| \leq \beta_V$, and $\langle \boldsymbol{v},\ \boldsymbol{z}\rangle \equiv s \bmod q$.

We note that similar assumptions have been proposed by prior works (see [BSS22] for a summary). Thus far, there are no known algorithms which solve these variants faster than their non-Vandermonde counterparts.

We now show how an adversary who breaks R-V-SIS can break k-P-R-ISIS. Since we are given $\boldsymbol{a} \cdot \boldsymbol{u}_i \equiv v^i \bmod q$ for all $i \in \mathbb{Z}(w) \setminus \{0\}$, and our goal is to find short $(s^*, \boldsymbol{u}_{g^*}) \neq (0, \boldsymbol{0})$ such that $\boldsymbol{a} \cdot \boldsymbol{u}_{g^*} \equiv s^* \cdot v^0 \bmod q$. We first solve R-V-SIS, obtaining $s, \boldsymbol{z}$ satisfying the constraints in Definition 3.1. Then, we let $\boldsymbol{u}_{g^*} = \sum_{i\in\mathbb{Z}(w)\setminus\{0\}} z_i \cdot \boldsymbol{u}_i$ and $s^* = s$. We observe that $\langle \boldsymbol{a},\ \boldsymbol{u}_{g^*}\rangle \equiv \sum_{i\in\mathbb{Z}(w)\setminus\{0\}} z_i \cdot \langle \boldsymbol{a},\ \boldsymbol{u}_i\rangle \equiv s^* v^0 \bmod q$.

Notice, however, we do get some blowup in the norm, where now $\|\boldsymbol{u}_{g^*}\| \leq (2w-3) \cdot \beta_V \cdot \beta \cdot \gamma_{\mathcal{R}}$. So we must solve R-V-SIS for a β_V such that $(2w-3) \cdot \beta_V \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \beta^*$.

In [ACLMT22], they generalize this attack, requiring the attacker find (not necessarily short) $\boldsymbol{z}$ and short s^* satisfying $\langle \boldsymbol{a},\ \boldsymbol{u}_{g^*}\rangle \equiv s^* v^0 \bmod q$ and $\|\boldsymbol{u}_{g^*}\| \leq \beta^*$ for $\boldsymbol{u}_{g^*} = \sum_{i\in\mathbb{Z}(w)\setminus\{0\}} z_i \cdot \boldsymbol{u}_i$. This attack applies to our scheme as well. See Section 4.1 of their paper for details.

Setting parameters according to the direct SIS attack. Given there are no specific algorithms for the above problems, when picking parameters we consider the cost of a direct SIS attack that ignores the preimages and their algebraic dependencies. The following analysis is standard. We first reduce R-ISIS to R-SIS: recall that to break the assumption, the attacker needs to find some $\boldsymbol{u}_g$ such that $\langle \boldsymbol{a},\ \boldsymbol{u}_g\rangle \equiv s^* \bmod q$; instead, we ask the attacker to find $\boldsymbol{u}'_g$ such that $\langle \boldsymbol{a}',\ \boldsymbol{u}'_g\rangle \equiv 0 \bmod q$, where $\boldsymbol{a}' := \boldsymbol{a}\|-1$. Then, here the last entry of $\boldsymbol{u}'_g$ becomes s^*.

We then view this as a SIS instance: for $\boldsymbol{A} \in \mathbb{Z}_q^{n\times n(\ell+1)}$ find $\boldsymbol{u} \in \mathbb{Z}_q^{n(\ell+1)}$ such that $\boldsymbol{A}\cdot\boldsymbol{u} \equiv 0 \bmod q$. Let $\boldsymbol{L} \in \mathbb{Z}_q^{n(\ell+1)\times n\ell}$ be a basis for the right kernel of $\boldsymbol{A}$. Then solving this SIS instance is equivalent to finding a short vector in $\Lambda(\boldsymbol{L})$. Observe that

$$\boldsymbol{L}' = \left[\begin{array}{c|c} L & \begin{array}{c} 0 \\ qI_n \end{array} \end{array}\right]$$

is an equivalent basis for this lattice. If we compute the Gram–Schmidt format of $\boldsymbol{L}'$, we get the form $\left[\begin{array}{c|c} I & 0 \\ \tilde{\boldsymbol{L}} & qI_n \end{array}\right]$ for some $\tilde{\boldsymbol{L}} \in \mathbb{Z}_q^{n(\ell+1)\times n\ell}$.

Thus, it is easy to compute the volume $\mathsf{Vol}(\Lambda) = \prod_{i\in[n(\ell+1)]} \|b_i\|_\infty = q^n$, where $[b_1,\ldots,b_{n(\ell+1)}] = \tilde{\boldsymbol{L}}$. Thus, the goal is to find a vector in $\Lambda(\boldsymbol{L})^\intercal$ in a $d \le n\cdot(\ell+1)$ dimensional lattice with volume q^n. We thus consider the adversary wins if it finds a vector with ℓ_2 norm of $\sqrt{d}\cdot\beta^*$, where β^* is the ℓ_∞ norm we allow in our hardness assumption.

Recall that lattice reduction with lattice parameter n (i.e., our ring degree) returns a vector with ℓ_2 norm $\approx \delta^d \cdot \mathsf{Vol}(\Lambda)^{1/d}$, where δ is the root Hermite factor. With the state-of-the-art lattice reduction algorithm [BDGL16], to achieve $\delta \approx (\frac{n}{2\pi e})^{1/(2n)}$, it takes roughly $2^{0.292n+o(n)}$ time. When $d \approx \sqrt{n\cdot log(q)/\log(\delta)}$, the norm of the vector found (i.e., $\delta^d\cdot\mathsf{Vol}(\Lambda)^{1/d}$) is minimized [MR09].

Thus, we can simply set $\delta^d\cdot\mathsf{Vol}(\Lambda)^{1/d} \ge \sqrt{d}\cdot\beta^*$ for the d that minimizes the left hand side, and $2^{0.292n+o(n)} \ge 2^\lambda$. Thus, when we set parameters for R-SIS problem with parameter $\mathcal{R}, q, \ell, \beta^*$ (where $\mathcal{R}$ has ring degree n), essentially, we are requiring that the adversary cannot find a vector with ℓ_2 norm $\le \sqrt{n\cdot log(q)/\log(\delta)}\cdot\beta^*$ in $\le 2^\lambda$ time using the attack above, except with neglgible probability. [3]

4 Orbweaver: linear functional commitments for rings

We present $\mathsf{Orbweaver}$, a vector commitment (VC) for linear functions that is non-interactive, publicly verifiable, preprocessing, and structure-preserving. Together, these features enable efficient recursive composition of our construction.

$\mathsf{Orbweaver}$ supports opening committed vectors with respect to the function family $\mathcal{F} = \{\mathcal{F}_w\}_{w\in\mathbb{N}}$, where $\mathcal{F}_w = \mathcal{X}^w$ and $f \in \mathcal{F}_w$ is computed as $f(\boldsymbol{x}) \equiv \langle [f_i]_{i=1}^w,\ \boldsymbol{x}\rangle \bmod q$. The input alphabet $\mathcal{X} = \{x \in \mathcal{R} \mid \|x\| \le \alpha\}$ is specified by the ring $\mathcal{R}$ and norm bound α. Let $\delta_M = w\cdot\alpha^2\cdot\gamma_{\mathcal{R}}$, where $\gamma_{\mathcal{R}}$ is the expansion factor of $\mathcal{R}$ (see Definition 2.1). The image space $\mathcal{Y}$ is then $\mathcal{Y} = \{y \in \mathcal{R} \mid \|y\| \le \delta_M\}$.

Let $P := \mathbb{Z}(w) \setminus \{0\}$. We use the monomial set $\mathcal{G}_0 := [g_i(X) = X^i]_{i\in P}$ to generate our opening proof $[\boldsymbol{u}_{0,i}]_{i\in P}$, the monomial set $\mathcal{G}_1 := [g_i(X) = X^i]_{i\in[w]}$ to generate our knowledge proof SRS $[\boldsymbol{u}_{1,i}]_{i\in[w]}$, the target monomial $g^*(X) := 1$, and the distribution $\mathcal{T}$ satisfying the requirements of Assumption 3.

$\underline{\mathsf{Setup}(1^\lambda, 1^w) \to \mathsf{ck}}$

$v \leftarrow\$\ \mathcal{R}_q^\times;\ \ t \leftarrow\$\ \mathcal{T}$

[3] Note that there are known poly-time attacks against some parameter selection for R-SIS, e.g., [PXWC21]. Thus, we also need to avoid those parameter selections. In more detail, we follow what is suggested in [ACLMT22] and pick q such that $\mathcal{R}_q$ fully splits and pick t as specified in Definition 2.15, which makes the attack from [PXWC21] not work.

Table 1: A list of parameters used in Orbweaver.

$w \in \mathbb{N}$		Dimension of secret input $\boldsymbol{x}$
$n \in \mathbb{N}$		Degree of $\mathcal{R}$
$\alpha \in \mathbb{R}$	$\mathrm{poly}(\lambda)$	Norm bound for $\boldsymbol{x}$ and f
$\beta_0 \in \mathbb{R}$	$\mathrm{poly}(\lambda)$	Norm bound for public preimages $\boldsymbol{u}_{0,\cdot}$ for π_0
$\beta_1 \in \mathbb{R}$	$\mathrm{poly}(\lambda)$	Norm bound for public preimages $\boldsymbol{u}_{1,\cdot}$ for π_1
$\delta_0 \in \mathbb{R}$	$w^2 \cdot \alpha^2 \cdot \beta_0 \cdot \gamma_{\mathcal{R}}^2$	Norm bound for opening proof π_0
$\delta_1 \in \mathbb{R}$	$w \cdot \alpha \cdot \beta_1 \cdot \gamma_{\mathcal{R}}$	Norm bound for commitment knowledge proof π_1
$\delta_M \in \mathbb{R}$	$w \cdot \alpha^2 \cdot \gamma_{\mathcal{R}}$	Norm bound for evaluation of a linear function with coefficients of norm bounded by α at a point of norm bounded by α
$q \in \mathbb{R}$		Modulus for $\mathcal{R}_q$
$\ell_i \in \mathbb{N}$	$\geq \mathsf{lhl}(\mathcal{R}_q, \beta_i)$	Number of ring elements in $\boldsymbol{a}$
$\gamma_{\mathcal{R}}$		Ring expansion factor of $\mathcal{R}$
$\mathcal{X}$	$\{x \in \mathcal{R} : \|x\| \leq \alpha\}$	$\mathcal{R}$ elements with norm bound α
$\mathcal{F}$		family of w-variate linear functions over $\mathcal{X}$

$(\boldsymbol{a}_0, \mathsf{td}_0) \leftarrow \mathsf{TrapGen}(1, 1^\ell, \mathcal{R}_q, \beta_0)$; $(\boldsymbol{a}_1, \mathsf{td}_1) \leftarrow \mathsf{TrapGen}(1, 1^\ell, \mathcal{R}_q, \beta_1)$
$\boldsymbol{u}_{0,i} \leftarrow \mathsf{SampPre}(\mathsf{td}_0, v^i, \beta_0), \ \forall i \in P$; $\boldsymbol{u}_{1,i} \leftarrow \mathsf{SampPre}(\mathsf{td}_1, v^i \cdot t, \beta_1), \ \forall i \in [w]$
Return $\mathsf{ck} := (\boldsymbol{a}_0, [\boldsymbol{u}_{0,i}]_{i \in P}, \boldsymbol{a}_1, t, [\boldsymbol{u}_{1,i}]_{i \in [w]}, v)$

$\underline{\mathsf{Com}(\mathsf{ck}, \boldsymbol{x}) \to (c, \pi_1)}$
$c := \sum_{i=1}^{w} x_i v^i \bmod q$
$\pi_1 := \sum_{i=1}^{w} x_i \cdot \boldsymbol{u}_{1,i} \bmod q$
Return (c, π_1)

$\underline{\mathsf{Open}(\mathsf{ck}, f, \boldsymbol{x}) \to \pi_0}$
Let $a_{-w+1}, ..., a_{w-1}$ denote the $2w-1$ coefficients of $\boldsymbol{x}(v)f(v) := \left(\sum_{i=1}^{w} x_i v^i\right)\left(\sum_{i=1}^{w} f_i v^{-i}\right)$
Return $\pi_0 := \sum_{i \in P} a_i \cdot \boldsymbol{u}_{0,i} \bmod q$

$\underline{\mathsf{PreVerify}(\mathsf{ck}, f) \to \mathsf{vk}_f}$
If $\|f\| > \alpha$, abort
Return $\mathsf{vk}_f := \sum_{i=1}^{w} f_i \cdot v^{-i} \bmod q$

$\underline{\mathsf{Verify}(\mathsf{vk}_f, c, \pi_1, y, \pi_0) \to \{0, 1\}}$
Output 1 if the following conditions all hold (else 0):
$\|y\|_\infty \leq \delta_M$; $\|\pi_1\|_\infty \leq \delta_1$; $\|\pi_0\|_\infty \leq \delta_0$
$\langle \boldsymbol{a}_1, \ \pi_1 \rangle \equiv c \cdot t \bmod q$
$\langle \boldsymbol{a}_0, \ \pi_0 \rangle \equiv \mathsf{vk}_f \cdot c - y \bmod q$

We next prove that Orbweaver satisfies correctness, extractability, and succinctness.

Theorem 4.1. *Orbweaver is correct (Definition 2.7) for $\delta_1 \geq w \cdot \alpha \cdot \beta_0 \cdot \gamma_{\mathcal{R}}$ and $\delta_0 \geq w^2 \cdot \alpha^2 \cdot \beta_1 \cdot \gamma_{\mathcal{R}}^2$.*

Proof. We begin by proving the norm bound checks hold. $y = f(\boldsymbol{x}) = \sum_{i=1}^{w} f_i \cdot x_i \bmod q$, where $\|f_i \cdot x_i\| \leq \alpha^2 \cdot \gamma_{\mathcal{R}}$ for all i. Therefore, $\|y\| \leq w \cdot \alpha^2 \cdot \gamma_{\mathcal{R}} = \delta_M$. Next, $\pi_1 = \sum_{i=1}^{w} x_i \cdot \boldsymbol{u}_{1,i} \bmod q$, where $\|x_i \cdot \boldsymbol{u}_{1,i}\| \leq \alpha \cdot \beta_1 \cdot \gamma_{\mathcal{R}}$ for all i. It follows $\|\pi_1\| \leq w \cdot \alpha \cdot \beta_1 \cdot \gamma_{\mathcal{R}} = \delta_1$. Last, $\pi_0 = \sum_{i \in P} a_i \cdot \boldsymbol{u}_{0,i} \bmod q$, where $a_{-w+1}, ..., a_{w-1}$ denote the $2w-2$ coefficients of $\boldsymbol{x}(v)f(v) = \left(\sum_{i=1}^{w} x_i v^i\right)\left(\sum_{i=1}^{w} f_i v^{-i}\right)$. Since

$\|\boldsymbol{u}_{0,i}\| \leq \beta_0$ for all i, $\|\pi_0\| \leq \beta_0 \cdot \gamma_{\mathcal{R}} \cdot \sum_{i \in P} a_i$. Then $\left\|a_{|i|}\right\| \leq (w-i) \cdot \alpha^2 \cdot \gamma_{\mathcal{R}}$ for $i \in [w-1]$. Thus, $\|\pi_0\| \leq w^2 \cdot \alpha^2 \cdot \beta_0 \cdot \gamma_{\mathcal{R}}^2 = \delta_0$.
Next, we show the verification equations hold:

$$\begin{aligned} \langle \boldsymbol{a}_1,\ \pi_1 \rangle &\equiv \langle \boldsymbol{a}_1,\ \textstyle\sum_{i=1}^{w} x_i \cdot \boldsymbol{u}_{1,i} \rangle \bmod q \\ &\equiv \textstyle\sum_{i=1}^{w} x_i \cdot \langle \boldsymbol{a}_0,\ \boldsymbol{u}_{1,i} \rangle \bmod q \\ &\equiv \textstyle\sum_{i=1}^{w} x_i \cdot v^i \cdot t \bmod q \\ &\equiv c \cdot t \bmod q \end{aligned} \qquad \begin{aligned} \langle \boldsymbol{a}_0,\ \pi_0 \rangle &\equiv \langle \boldsymbol{a}_0,\ \textstyle\sum_{i \in P} a_i \cdot \boldsymbol{u}_i \rangle \bmod q \\ &\equiv \textstyle\sum_{i \in P} a_i \cdot v^i \bmod q \\ &\equiv \left(\textstyle\sum_{i=1}^{w} x_i v^i\right)\left(\textstyle\sum_{i=1}^{w} f_i v^{-i}\right) - a_0 \cdot v^0 \bmod q \\ &\equiv \mathsf{vk}_f \cdot c - \textstyle\sum_{i=1}^{w} f_i \cdot x_i \equiv \mathsf{vk}_f \cdot c - y \bmod q \ . \end{aligned}$$

□

Theorem 4.2. *Let $\mathcal{X}^* := \{x \in \mathcal{R} \mid \|x\| \leq \alpha^*\}$. Orbweaver is $\mathcal{X}^*$-extractable if*

$$\begin{aligned} &\ell_0 \geq \mathsf{lhl}(\mathcal{R}_q, b_0),\ \ \ell_1 \geq \mathsf{lhl}(\mathcal{R}_q, b_1) \\ &\alpha^* \geq \beta_1^* \geq \delta_1 \\ &\beta_0^* \geq 2 \cdot w^2 \cdot \alpha \cdot \alpha^* \cdot \beta_0 \cdot \gamma_{\mathcal{R}}^2 \end{aligned}$$

where $b_0 = O(\beta_0), b_1 = O(\beta_1)$, and the k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_0,1,\mathcal{G}_0,g_0^*,\mathcal{D}_0,\beta_0,\beta_0^*}$ (i.e., k-P-R-ISIS) and knowledge k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_1,1,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T},\alpha^*,\beta_1,\beta_1^*}$ (i.e., knowledge k-P-R-ISIS) assumptions hold, where $\mathcal{D}_0$ and $\mathcal{D}_1$ are such that

$$\begin{aligned} &\left\{ (\boldsymbol{a}_0, [\boldsymbol{u}_{0,g}]_{g \in \mathcal{G}_0}, v) \;\middle|\; \begin{array}{l} \boldsymbol{a}_0 \leftarrow\$\ \mathcal{R}_q^{\ell_0};\ \ v \leftarrow\$\ \mathcal{R}_q^{\times} \\ \boldsymbol{u}_{0,g} \leftarrow\$\ \mathcal{D}_{g,\boldsymbol{a}_0,v},\ \ \forall i \in P \end{array} \right\} \\ \approx\ &\left\{ (\boldsymbol{a}_0, [\boldsymbol{u}_{0,g}]_{g \in \mathcal{G}_0}, v) \;\middle|\; \begin{array}{l} \boldsymbol{a}_0 \leftarrow\$\ \mathcal{R}_q^{\ell_0};\ \ v \leftarrow\$\ \mathcal{R}_q^{\times} \\ \boldsymbol{u}_{0,g} \leftarrow\$\ \mathsf{SampD}(1^{\ell_0}, \mathcal{R}_q, \beta_0) : \langle \boldsymbol{a}_0,\ \boldsymbol{u}_{0,g} \rangle \equiv v^i \bmod q,\ \ \forall i \in P \end{array} \right\} \end{aligned}$$

and

$$\begin{aligned} &\left\{ (\boldsymbol{a}_1, t_i, [\boldsymbol{u}_{1,g}]_{g \in \mathcal{G}_1}, v) \;\middle|\; \begin{array}{l} \boldsymbol{a}_1 \leftarrow\$\ \mathcal{R}_q^{\ell_1};\ \ t \leftarrow\$\ \mathcal{T};\ \ v \leftarrow\$\ \mathcal{R}_q^{\times} \\ \boldsymbol{u}_{1,g} \leftarrow\$\ \mathcal{D}_{g,\boldsymbol{a}_1,t,v},\ \ \forall i \in [w] \end{array} \right\} \\ \approx\ &\left\{ (\boldsymbol{a}_1, t_i, [\boldsymbol{u}_{1,g}]_{g \in \mathcal{G}_1}, v) \;\middle|\; \begin{array}{l} \boldsymbol{a}_1 \leftarrow\$\ \mathcal{R}_q^{\ell_1};\ \ t \leftarrow\$\ \mathcal{T};\ \ v \leftarrow\$\ \mathcal{R}_q^{\times} \\ \boldsymbol{u}_{1,g} \leftarrow\$\ \mathsf{SampD}(1^{\ell_1}, \mathcal{R}_q, \beta_1) : \langle \boldsymbol{a}_1,\ \boldsymbol{u}_{1,g} \rangle \equiv v^i \cdot t \bmod q,\ \ \forall i \in [w] \end{array} \right\} . \end{aligned}$$

Proof. Suppose there exists a PPT adversary $\mathcal{A}$ that, on input $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda, 1^w)$, outputs (f, c, y, π_1, π_0) such that $\mathsf{PreVerify}(\mathsf{ck}, f)$ does not abort and $\mathsf{Verify}(\mathsf{vk}_f, c, y, \pi_1, \pi_0) = 1$ with non-negligible probability in λ over any randomness it uses and the choice of ck. We construct an extractor $\mathcal{E}_{\mathcal{A}}$ that, input ck and given black-box access to $\mathcal{A}$ and any randomness it uses, returns $\boldsymbol{x}^* \in (\mathcal{X}^*)^w$ such that $c = \mathsf{Com}(\mathsf{ck}, \boldsymbol{x}^*)$ and $f(\boldsymbol{x}^*) = y$ with all but neglibible probability in λ when $\mathcal{A}$ succeeds.

For commitment key $\mathsf{ck} = (\boldsymbol{a}_0, [\boldsymbol{u}_{0,i}]_{i \in P}, \boldsymbol{a}_1, t, [\boldsymbol{u}_{1,i}]_{i \in [w]}, v)$, define $\mathsf{ck}_0 := (\boldsymbol{a}_0, [\boldsymbol{u}_{1,i}]_{i \in [w]}, v)$ and $\mathsf{ck}_1 := (\boldsymbol{a}_1, t, [\boldsymbol{u}_{1,i}]_{i \in [w]}, v)$. Let $\mathcal{B}_{\mathcal{A}}$ be an adversary with black-box access to $\mathcal{A}$ that runs on input ck_1' sampled according to the knowledge k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_1,1,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T},\alpha^*,\beta_1,\beta_1^*}$ definition (Definition 2.15). First, $\mathcal{B}_{\mathcal{A}}$ samples ck_0' according to the k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_0,1,\mathcal{G}_0,g_0^*,\mathcal{D}_0,\beta_0,\beta_0^*}$ definition (Definition 2.13), except fixing $v' = v$. Then $\mathcal{B}_{\mathcal{A}}$ runs $\mathcal{A}$ on $(\mathsf{ck}_0', \mathsf{ck}_1)$ (passing its own random input tape) to obtain (f, c, y, π_1, π_0), which it parses to output (c, π_1).

Let $\mathcal{E}_{\mathcal{B}_{\mathcal{A}}}$ be the extractor promised by our knowledge assumption for $\mathcal{B}_{\mathcal{A}}$. Extractor $\mathcal{E}_{\mathcal{A}}$, on input $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1)$, runs $\mathcal{E}_{\mathcal{B}_{\mathcal{A}}}(\mathsf{ck}_1)$ to obtain $\boldsymbol{x}^*$. Next consider the following hybrid experiments:

1. Hyb_0: the real experiment, where $\mathsf{ck} \leftarrow \mathsf{Setup}(1^\lambda, 1^w)$, $(f, c, y, \pi_1, \pi_0) \leftarrow \mathcal{A}(\mathsf{ck})$, and $\boldsymbol{x}^* \leftarrow \mathcal{E}_{\mathcal{A}}(\mathsf{ck})$.

2. Hyb_1: the same as Hyb_0, but $\mathsf{ck} = (\mathsf{ck}_0, \mathsf{ck}_1')$, where ck_1' is sampled according to the knowledge k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_1,1,\mathcal{G}_1,\mathcal{D}_1,\mathcal{T},\alpha^*,\beta_1,\beta_1^*}$ definition, conditioned on $v' = v$.
3. Hyb_2: the same as Hyb_1, but $\mathsf{ck} = (\mathsf{ck}_0'', \mathsf{ck}_1')$, where ck_1' is sampled as in Hyb_1, but ck_0'' is sampled according to the k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_0,1,\mathcal{G}_0,g_0^*,\mathcal{D}_0,\beta_0,\beta_0^*}$ definition, again conditioned on $v'' = v'$.

By our assumption on $\mathcal{D}_1$ (resp., $\mathcal{D}_0$) hybrids 0 and 1 (resp., 1 and 2) are statistically close. Therefore, they are all statistically close. This implies if a property holds with respect to the output of one hybrid with a certain probability, it will hold with respect to the output of the others with statistically close probability.

From our knowledge assumption, we know that in Hyb_1 the extractor outputs $\boldsymbol{x}^*$ satisfying $c \equiv \sum_{i=1}^{w} x_i \cdot v^i \bmod q$ and $\|x_i\| \leq \alpha^*$ for all $i \in [w]$ with probability all but negligible in λ. Now consider the output in Hyb_2 satisfies $c \equiv \sum_{i=1}^{w} x_i \cdot v^i \bmod q$ for $\boldsymbol{x}^* \in \mathcal{X}^*$, but $f(\boldsymbol{x}^*) \neq y$. When this happens we can break k-R-$\mathsf{ISIS}_{\mathcal{R}_q,\ell_0,1,\mathcal{G}_0,g_0^*,\mathcal{D}_0,\beta_0,\beta_0^*}$ as follows. Let $y^* := f(\boldsymbol{x}^*)$ and $\pi_0^* \leftarrow \mathsf{Open}(\mathsf{ck}, f, \boldsymbol{x}^*)$. Then we obtain the solution $(y^* - y, \pi_0 - \pi_0^*)$ for k-R-ISIS, where

$$\langle \boldsymbol{a}_0,\ \pi_0 - \pi_0^* \rangle \equiv \mathsf{vk}_f \cdot c - y - (\mathsf{vk}_f \cdot c - y^*) \equiv (y^* - y) \cdot 1 \equiv (y^* - y) \cdot g^*(v) \bmod q\ \ .$$

To see that the norm bounds hold, we can bound $\delta_M^* := \|y^*\| \leq w\alpha^*\alpha\gamma_{\mathcal{R}}$ and $\delta_0^* := \|\pi_0^*\| \leq w^2\alpha^*\alpha\beta_0\gamma_{\mathcal{R}}^2 \leq \beta_0^*/2$ by modifying the calculations of δ_M and δ_0 in the correctness proof. Therefore, this can only happen with negligible probability in λ. Together, these two properties about the results of Hyb_1 and Hyb_2 show that $\mathsf{Orbweaver}$ is $\mathcal{X}^*$-extractable. □

Theorem 4.3. *Orbweaver is succinct (Definition 2.9).*

Proof. We first show $\mathsf{Orbweaver}$ has $O(\log w + \log \alpha)$ size commitments and proofs. From our construction we have that $|c| = n\lceil \log(q) \rceil$, $|\pi_0| = n\ell_0\lceil \log \delta_0 \rceil$, and $|\pi_1| = n\ell_1\lceil \log \delta_1 \rceil$. We set a base $b_0 = O(\beta_0), b_1 = O(\beta_1)$ for trapdoor generation and let $\ell_0 := \mathsf{lhl}(\mathcal{R}_q, b_0) = O(\log(q)/\log(\beta_0)))$, $\ell_1 := \mathsf{lhl}(\mathcal{R}_q, b_1) = O(\log(q)/\log(\beta_1))$. We consider the ring degree n to be a function of the security parameter and treat it as a constant for the following analysis.

As implicated by our extractability theorem (Theorem 4.2) together with our assumptions relating the hardness of plain and knowledge k-P-R-ISIS to R-SIS (Assumptions 2 and 3), we must have that R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\beta_0^*}$ and R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\delta_1}$ are hard. Recall $\beta_0^* = 2w^2\alpha\alpha^*\beta_0\gamma_{\mathcal{R}}^2$. Using $\alpha^* = \delta_1 = w\alpha\beta_1\gamma_{\mathcal{R}}$, we can write $\beta_0^* = 2w^3\alpha^2\beta_0\beta_1\gamma_{\mathcal{R}}^3$. Since $\delta_1 < \beta_0^*$, we can just pick q s.t. R-$\mathsf{SIS}_{\mathcal{R}_q,\ell,\beta_0^*}$ is hard.

In particular, we let $q \approx \beta_0^* n \log(n)$ such that $\log(q) = O(\log \beta_0^*) = O(\log(w) + \log(\alpha) + \log(\beta_0) + \log(\beta_1))$ since for power-of-two cyclotomic rings $\gamma_{\mathcal{R}} \leq n$ (Theorem 2.2). Setting $\beta_0 := \max\{w, \alpha\}$ and $\beta_1 := \max\{w, \alpha\}$, we obtain logarithmic commitment and proof sizes:

$$\begin{aligned} |c| &= n\lceil \log(q) \rceil \\ &\approx n \log(n \log(n) \beta_0^*) \\ &= O(\log \beta_0^*) \\ &= O(\log w + \log \alpha) \end{aligned} \qquad \begin{aligned} |\pi_0| &= n\ell_0\lceil \log \delta_0 \rceil \\ &= 2\lceil \log(q)/\log(\beta_0) \rceil n \lceil \log \delta_0 \rceil \\ &\approx 2n \log(\beta_0^* n \log n) \log \delta_0 / \log \beta_0 \\ &= O(\log(\beta_0^*) \cdot \log(\delta_0)/\log \beta_0) \\ &= O(\log w + \log \alpha) \end{aligned}$$

$$\begin{aligned} |\pi_1| &= n\ell_0\lceil \log \delta_1 \rceil \\ &= 2\lceil \log(q)/\log(\beta_1) \rceil n \lceil \log \delta_1 \rceil \\ &\approx 2n \log(\beta_0^* n \log n) \log \delta_1 / \log \beta_1 \\ &= O(\log \beta_0^* \cdot \log \delta_1 / \log \beta_1) \\ &= O(\log w + \log \alpha) \end{aligned}$$

Next, we show that the Orbweaver verifier runs in time $O(\log(w\alpha)\log\log(w\alpha))$ time. The verifier work is dominated by computing the inner products $\langle \boldsymbol{a}_1,\ \pi_1\rangle$ and $\langle \boldsymbol{a}_0,\ \pi_0\rangle$, where $|\boldsymbol{a}_i| = \ell_i$. This work is dominated by $\ell_0 + \ell_1$ multiplications in the ring $\mathcal{R}_q$ of degree n. Using the NTT, this can be computed with $(\ell_0 + \ell_1)(n \log n)$ multiplications in $\mathbb{Z}_q$. Though since we've defined n in terms of λ, which itself is treated as a constant in our asymptotic analysis, we can simplify this to $O(\ell_0 + \ell_1) = O(\log q/\log\beta_0 + \log q/\log\beta_1) = O(\log\beta_0^*/\log\beta_0 + \log\beta_0^*/\log\beta_1) = O((\log w + \log\alpha)/\log\beta_0 + (\log w + \log\alpha)/\log\beta_1) = O(1)$ operations over the ring $\mathbb{Z}_q$, where elements have bit length $\log q = O(\log w + \log\alpha)$. Asymptotically, $\mathbb{Z}_q$ multiplications can be performed in $\log(q)\log\log(q)$ time [HH21], giving verifier complexity is $O(\log(w\alpha)\log\log(w\alpha))$. [4] □

4.1 Extensions

This section discuss extensions to Orbweaver including support for proof aggregation, an inner product argument, and recursive proof composition. Even if the reader is only interested in using Orbweaver just for ring relations, we suggest looking at the section on linear functional commitments for $\mathbb{Z}_M$ (Section 4.2) anyway as many of the optimizations introduced there including ternary decomposition can be applied in the ring setting as well.

Universal SRS. In order to support all instances of length $w \leq W$, observe it suffices to run the protocol without modification using only powers in $\mathbb{Z}(w)$. Since parameters are set such that R-SIS is hard for β_0^* based on W, it is not an issue that the extractor outputs an input of length W.

Public proof aggregation. Orbweaver can be extended to support public proof aggregation (i.e., the aggregator can be an untrusted third party) by taking advantage of the linearly homomorphic property of the opening proofs. Using short challenges $[[h_{i,j}]_{i=1}^{t}]_{j=1}^{T[i]}$ we can check that commitments $[c_i]_{i=1}^{t}$, evaluated at function sets $[f_j]_{j=1}^{T[i]}$, output values $[[y_{i,j}]_{i=1}^{t}]_{j=1}^{T[i]}$, respectively. The verifier checks the opening proof equation

$$\langle \boldsymbol{a},\ \pi_0\rangle \equiv \sum_{i=1}^{t}\sum_{j=1}^{T[i]} h_{i,j}\cdot\left(c_i\cdot \mathsf{ck}_{f_{i,j}} - y_{i,j}\right) \bmod q \ \ ,\ \text{where}\quad \pi_0 = \sum_{i=1}^{t}\sum_{j=1}^{T[i]} h_{i,j}\pi_{0,i,j} \quad \bmod q$$

We give an instantiation of $\mathcal{H}$ from exceptional sets over $\mathcal{R}_q$ in Section 2.1. With this challenge set, the protocol becomes interactive: the aggregator sends $[(c_i, \pi_{1,i}, [y_{i,j}]_{j=1}^{T[i]})]_{i=1}^{t}$, the verifier samples $[h_{i,j}]_{i=1,j=1}^{t\ ,T[i]} \leftarrow\$\ \mathcal{H}$, and the aggregator responds with π_0. Using an R-SIS instance as a fixed challenge string as introduced in [ACLMT22] provides a way to maintain non-interactivity, while using an exceptional set provides better performance.

To see how security holds, after using our knowledge assumption extractor to obtain the $[\boldsymbol{x}_i^*]_{i=1}^{t}$, we have by k-P-R-ISIS that

$$\sum_{i=1}^{t}\sum_{j=1}^{T[i]} h_{i,j}\cdot(\langle f_{i,j},\ \boldsymbol{x}_i^*\rangle - y_{i,j}) \equiv 0 \bmod q$$

[4] We note [HH21] is a galactic algorithm, i.e., it is only efficient at instance sizes so large it is never used in practice. For the integer sizes we care about (several thousand bits) recent works have shown NTT-based approaches, with complexity $O(\log w \log\log w \log\log\log w)$, concretely most efficient [BHKPY22].

When interactively sampling challenges from an exceptional set $\mathcal{H}$ over $\mathcal{R}_q$, we obtain that $\langle f_{i,j},\ \boldsymbol{x}_i^*\rangle = y_{i,j}$ for all i, j except with probability $1/|\mathcal{H}|$ by Theorem 2.4. If instead we use a fixed R-SIS instance $\boldsymbol{h} \leftarrow\$\ \mathcal{R}_p$ for $\alpha \ll p \ll q$, then if $\langle f_{i,j},\ \boldsymbol{x}_i^*\rangle \neq y_{i,j}$ for some i, j, we obtain a non-trivial R-SIS solution.

Observe that because we extract the $\boldsymbol{x}_i^*$ directly from the knowledge proofs, we do not need to extract the hypothetical $\pi_{0,i,j}$ inside π_0 for our security reduction. This lets use an exceptional set over $\mathcal{R}_q$, which can be exponential in size (see Section 2.1 for an explicit construction). We can then invoke the Alon–Füredi Theorem Theorem 2.4 to say the adversary succeeds with. On the other hand, when extraction is required current techniques rely on using exceptional sets over $\mathcal{R}$, and there's evidence such sets are all polynomial-sized for rings of interest [AL21].

When using exceptional sets our interactive protocol is public-coin and we may therefore apply the Fiat–Shamir transform to once again obtain a non-interactive protocol. The interactive protocol is round-by-sound since Σ-protocols are, and since its relation (finding a non-zero polynomial such that the hash of its commitment is one of its zeros) is sparse, the non-interactive version is secure in the random oracle model [CCHLRR18]. Security in the standard model is also possible if there is a correlation intractable hash function family for the relation.

As introduced in [ACLMT22], it's also possible to use a public R-SIS vector $\boldsymbol{h}$ over a $\mathcal{R}_{c'}$ for $c' \ll q$ as a fixed challenge vector rather than sampling a fresh $\boldsymbol{h}$ each time from an exceptional set as above. This requires no interaction, but comes at the cost of a much larger operator norm $c = c' \cdot \gamma_{\mathcal{R}}$ versus the exceptional sets approach.

Private aggregation. If the aggregator is the prover itself, they may compute the linear combination of the $a_{i,j,k}$ values first to obtain a vector $\boldsymbol{a}'$ and then compute the aggregate proof $\pi_0 = \sum_i a_i' \cdot \boldsymbol{u}_{0,i}$ thereby reducing the amount of costly ring operations.

Inner product argument. We can easily construct an inner product argument (IPA) using Orbweaver, allowing a prover to prove the inner product $y := \langle \boldsymbol{x},\ \boldsymbol{x}'\rangle \bmod q$ between two committed vectors. The prover runs commit twice to produce c, c', π_1, and π_1', and runs $\mathsf{Open}(\mathsf{ck}, \boldsymbol{x}', \boldsymbol{x})$ to obtain π_0. The verifier checks the norm bounds of all three proofs, verifies both knowledge proofs as in Verify, and then checks the opening proof satisfies $\langle \boldsymbol{a},\ \pi_0\rangle \equiv c' \cdot c - y \bmod q$.

It should be noted that this requires a modification of the norm bound β_0^* for which R-SIS must be hard. In the modified proof of Theorem 4.2 the extractor would generate π_0^* using extracted values $\boldsymbol{x}^*$ and f^*, both of norm α^* (whereas in the linear function case they use f of norm α). Then we must set $\beta_0^* = w^2(\alpha^*)^2\beta_0\gamma_{\mathcal{R}}^2 = w^4\alpha^2\beta_1^2\beta_0\gamma_{\mathcal{R}}^4$.

Recursive proof composition. Observe that $\mathsf{vk}_f \cdot \mathsf{ck} - y \bmod q$ and $c \cdot t \bmod q$ are Ajtai R-SIS commitments to π_0 and π_1 for respective public vectors $\boldsymbol{a}_0$ and $\boldsymbol{a}_1$. Using any proof-succinct argument of knowledge for R-SIS commitment preimages (e.g., [BS22]), we can achieve better proof sizes by instead sending an AoK of π_0 and π_1. The composed scheme will still have log-time verification even if the outer argument has a linear-time verifier.

One may also recursively compose Orbweaver with itself by sending commitments to the proofs and proving they satisfy the (linear) verification equations. To deal with the norm bound checks, the only non-structure-preserving part of verification, one could send a random Johnson-Lindenstrauss projection, a linear operation which requires only constant communication, then prove it was computed correctly with respect to the committed proofs. At the 128-bit security level, an optimized version of Orbweaver achieves witness compression starting at sizes a little under 668KiB, or 165KiB if we forego extractability and only require evaluation binding; achieving smaller proof sizes will require switching to recursion using another (possibly even linear verifier) scheme at this point, as described above.

Linear functional commitments for integers. We can also extended Orbweaver to support the input alphabets $\mathcal{X} = \{z \in \mathbb{Z} \mid |z| < \alpha\}$ and, using the centered representation, $\mathcal{X} = \{z \in \mathbb{Z} \mid 0 \leq z \leq 2\alpha\}$. One way to do this is to use the coefficient embedding to encode $\hat{f}, \hat{\boldsymbol{x}} \in \mathbb{Z}^N$ as $\boldsymbol{x}, f' \in \mathcal{R}^w$ where $N = wn$. We further set $f = [\sigma_{-1}(f'_i)]_{i \in [w]}$, where $\sigma_{-1} \in \mathsf{aut}(\mathcal{R}_q)$ is the Galois automorphism corresponding to $\sigma_{-1}(X) = X^{-1}$ (see Section 2). We then have that $\langle \hat{f},\ \hat{\boldsymbol{x}} \rangle \equiv \mathsf{ct}(\langle f,\ \boldsymbol{x} \rangle) \bmod q$, where ct returns the constant coefficient. We run Orbweaver with commitments to $f, \boldsymbol{x}$ and with $y = \langle f,\ \boldsymbol{x} \rangle$. While y is part of the instance over $\mathcal{R}$, only $\mathsf{ct}(y)$ is in the instance over $\mathbb{Z}$ in this shim protocol, while the non-constant coefficients of y are instead part of the proof for the relation over $\mathbb{Z}$.

4.2 Sublogarithmic proofs in the input norm

We present a functional commitment based on Orbweaver that achieves $O(\log w + \log \log \alpha)$ commitment and proof size by combining ternary decomposition and proof aggregation. The construction below is restricted to the input alphabets consisting of bounded-size balanced integers $\mathcal{X} = \{z \in \mathbb{Z} \mid |z| < \alpha\}$ and (using the centered representation) unbalanced integers $\mathcal{X} = \{z \in \mathbb{Z} \mid 0 \leq z \leq 2\alpha\}$. While our techniques should extend to subsets of cyclotomic rings more generally, we leave this to future work.

Let $t = \log_3(\alpha)$. Ternary decomposition allows us express a linear map over length-N (where $N = wn$, a notation switch that will soon be helpful) vectors of norm α as $2t - 1$ linear maps of length-(Nt) vectors of norm 1. Using the aggregation techniques in Section 4.1, we can create a single compact aggregate proof for k separate linear maps of size $O(\log(N\alpha k))$. For $k = 2t - 1$ and $\alpha = 1$ this gives $O(\log N + \log t)$ proofs. In terms of proof norms, we trade a power of α for a higher power of $\log_3(\alpha)$. Similarly, committing to the ternary decomposition of the witness results in a $O(\log(Nt) + \log \alpha) = O(\log N + \log t)$ sized commitment. Experimentally, at the 128-bit security level decomposition and aggregation performed better starting around $\alpha = 2^{16}$ for all input lengths of interest for this scheme, i.e., for input vectors of length at least 2^{16} where we begin to achieve witness compression.

We preface our result below with a few remarks. The requirement $\alpha = 3^t$ is not strict and the proceeding construction works if we set $t = \lceil \log_3(\alpha) \rceil$. This is more than just with respect to correctness: since Orbweaver proves $\langle f,\ \boldsymbol{x} \rangle = y$ over the integers, it will also hold mod α. It is always possible to include in a commitment some elements of norm larger than correctness is guaranteed for. When taking the result mod α by committing to these values the prover is effectively committing to their coset representative.

Theorem 4.4. *Let $\hat{\mathcal{X}} = \left\{z \in \mathbb{Z} \mid |z| \leq \alpha = 3^t\right\}$, and for $N = wn$ let $\hat{f}, \hat{\boldsymbol{x}} \in \hat{\mathcal{X}}^N$. We obtain a succinct functional commitment for $\langle \hat{f},\ \hat{\boldsymbol{x}} \rangle = y$ where the size of c, π_1, and π_0 are all $O(\log N + \log \log \alpha)$.*

Proof (informal). We can write

$$\langle \hat{f},\ \hat{\boldsymbol{x}} \rangle = \sum_{i=0}^{2t-2} 3^i \langle f'_i,\ \boldsymbol{x}' \rangle$$

where $[f'_i]_{i=1}^t, \boldsymbol{x}' \in \{-1, 0, 1\}^{Nt}$. We then combine the techniques from "private aggregation" and "linear functional commitments for bounded integers" above to compute this using Orbweaver.

If we view each f'_i as a column vector and, abusing notation, let $f'_{i,j}$ refer to j-th (for $j \in \{0, \dots, N-1\}$) sequential t-length subvector of f'_i we see the following form

$$[f'_{0,j} \cdots f'_{2t-1,j}] = \begin{bmatrix} b_0 & b_1 & b_2 & \cdots & b_{t-1} & 0 & 0 & \cdots & 0 \\ 0 & b_0 & b_1 & \cdots & b_{t-2} & b_{t-1} & 0 & \cdots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \cdots & 0 & b_0 & b_1 & b_2 & \cdots & b_{t-1} \end{bmatrix} \in \{-1, 0, 1\}^{t \times (2t-1)}$$

where $[b_k]_{k=0}^{t-1}$ is the ternary decomposition of $\hat{f}_j$. For simplicity, let $n = rt$ (otherwise, padding may be applied) for $r \in Z^+$, thus supporting decomposition for norm bounds up to 3^{512} (where, again, modifications enable more generality). Let $f_i \in \mathcal{R}^{wt}$ denote the coefficient embedding of f'_i. Then we have for

$$\tilde{f} = \sum_{i=0}^{2t-2} h_i f_i$$

that $\left\|\tilde{f}_j\right\| \leq \mathsf{min}\{c, r\} \cdot t^2 \leq ct^2$ for all $j \in [wt]$, where c is the operator norm of the challenge set $\mathcal{H}$ used. This follows from the fact we are taking the sum over $2t-1$ vectors $h_i f_i$, where when averaging over the i the norm bound of each element is $\mathsf{min}\{c, r\} \cdot t^2/(2t-1)$. The norm of $\tilde{f}$ is now larger than $\boldsymbol{x} \in \mathcal{R}^{wt}$, and for a more tight result we recompute the norm bounds with this in mind.

First, recall $y_i = \langle f_i,\ \boldsymbol{x} \rangle$. For $i \in \{0, \dots, t-1\}$ we then obtain

$$\|y_i\| \leq (i+1)rwt = (i+1)N = \delta_{M,i}$$

and for $i \in \{t, \dots, 2t-1\}$ we have

$$\|y_i\| \leq (2n - (i+1)r)wt = (2t - i - 1)N = \delta_{M,i}$$

Next, $\pi_1 = \sum_{i=1}^{w} x_i \boldsymbol{u}_{1,i} \bmod q$, where $\|x_i \boldsymbol{u}_{1,i}\| \leq \beta_1 \gamma_{\mathcal{R}}$ for all i. It follows

$$\|\pi_1\| \leq wt\beta_1\gamma_{\mathcal{R}} = Nt\beta_1 = \delta_1$$

Then, $\pi_0 = \sum_{k \in P} a_k \cdot \boldsymbol{u}_{0,k} \bmod q$, where $a_{-wt+1}, \dots, a_{wt-1}$ denote the $2wt-2$ coefficients of $\boldsymbol{x}(v)\tilde{f}(v) = \left(\sum_{j=1}^{wt} x_j v^j\right)\left(\sum_{j=1}^{wt} \tilde{f}_j v^{-j}\right)$ and $P := \{-wt+1, \dots, wt-1\} \setminus \{0\}$. Since $\|\boldsymbol{u}_{0,k}\| \leq \beta_0$ for all k, $\|\pi_0\| \leq \beta_0 \gamma_{\mathcal{R}} \sum_{k \in P} a_k$. Then $\left\|a_{|k|}\right\| \leq (wt - k)ct^2\gamma_{\mathcal{R}}$ for $k \in [w-1]$. We then have

$$\sum_{k \in P} a_k = 2ct^2\gamma_{\mathcal{R}} \sum_{\kappa=1}^{wt-1} (wt - \kappa) = ct^2\gamma_{\mathcal{R}}(w^2t^2 - wt)$$

Thus,

$$\begin{aligned} \|\pi_0\| &\leq \beta_0 ct^2 \gamma_{\mathcal{R}}^2((N/n)^2t^2 - (N/n)t) = N^2ct^4\beta_0 - Nct^3\beta_0 n \\ &= \delta_0 \\ &\leq (N^2 - N)ct^4\beta_0 \leq N^2ct^4\beta_0 \end{aligned}$$

Last, we consider δ_0^*. Replacing, $\boldsymbol{x}$ with the extracted $\boldsymbol{x}^*$ in the analysis above, where $\left\|x_j^*\right\| \leq \delta_1 = Nt\beta_1$ for all $j \in [wt]$, gives

$$\|\pi_0^*\| \leq Nct^3\beta_0\beta_1\gamma_{\mathcal{R}}^2((N/n)^2t^2 - (N/n)t) = N^3ct^5\beta_0\beta_1 - N^2ct^4\beta_0\beta_1 n$$

Finally,

$$\begin{aligned} \|\pi_0 - \pi_0^*\| &\leq N^2ct^4\beta_0 - Nct^3\beta_0 n + N^3ct^5\beta_0\beta_1 - N^2ct^4\beta_0\beta_1 n \\ &= \delta_0^* \\ &\leq N^3ct^5\beta_0\beta_1 \end{aligned}$$

□

4.3 Polynomial commitments for integers

We show how to construct $O(\log N)$ proof size and $O(\log N \log\log N)$ verifier time polynomial commitment (PC) for both degree-N univariate polynomials and N-linear polynomials over $\mathbb{Z}$ using $\mathsf{Orbweaver}$. How to do this while preserving a quasilog-verifier is not immediate. In a PC protocol based on $\mathsf{Orbweaver}$, the prover commits to the coefficients of the polynomial and the function f becomes the powers of a challenge evaluation point z. Previously we considered the preprocessing setting, where we assume the same function will be evaluated repeatedly on different commitments. In this amortized setting the cost to run $\mathsf{PreVerify}$ once is considered a constant. On the other hand, in the PC setting f is defined by z, which is usually a fresh challenge sampled uniformly at random from $\mathbb{Z}_M$—a setting in which preprocessing no longer makes sense.

We present the univariate degree-N PC. Omitting a couple details, by replacing z^{2^i} with z_i for all i below we obtain the N-linear version. Again, we choose $\alpha \geq (M-1)/2$ and assume wlog that $\alpha = 3^t$, $wn = N$, and w is a power of 2. We show how to compute, in log-time, a commitment key $\mathsf{ck}'_z \equiv \mathsf{ck}_z \bmod M$, where ck_z is the commitment key corresponding to running $\mathsf{PreVerify}$ on the function $f = (f_1, ..., f_w) \in \mathbb{Z}_M^w$, where $\sigma_{-1}(f_i) \in \mathcal{R}_q$ is equal to the coefficient embedding of the vector $z^{n(i-1)} \cdot (z^0, z^{1+n(i-1)}, \ldots, z^{n-1}) := z^{n(i-1)} \cdot f_{\heartsuit} \in \mathbb{Z}_M^n$. This choice of f follows the techniques we used in Section 4.2 such that the evaluation of the committed polynomial at z will be equal to the constant coefficient of y. The verifier computes

$$\mathsf{ck}'_z := f_{\heartsuit} \cdot v^{-w} \cdot \prod_{i=0}^{\log(w)-1} ((z^{n \cdot 2^i} \bmod M) + v^{2^i}) = \sum_{i=1}^{w} f_{\heartsuit} \cdot \hat{z}_{i-1} \cdot v^{-i} \equiv \mathsf{ck}_z \bmod M$$

In particular this holds because each $\hat{z}_i \equiv z^{in} \bmod M$. The downside of computing the key this way is that it corresponds to a commitment key for the function $\hat{f} := f_{\heartsuit} \cdot \hat{z} \in \mathcal{R}_q^w$, where $\left\|\hat{f}\right\| \leq \alpha^{\log w + 1}$ since the norm of $\hat{z}_i$ is bounded by α to the power of the Hamming weight of i. Roughly speaking, computing the commitment key this way comes at a cost of a $\log \alpha$ times larger opening proof.

5 Evaluation

We used SageMath to compute proof and CRS sizes for an optimized version $\mathsf{Orbweaver}$ for integers in Section 5.2, and in Section 5.2.1 we compare an unoptimized version for cyclotomic rings to previous work [ACLMT22]. Before getting there, we describe the additional optimizations in Section 5.1.

5.1 Implementation optimizations

Sending only $\boldsymbol{\pi_1}$ and not $\boldsymbol{c}$. We can slightly reduce communication and both prover and verifier complexity by only sending π_1 and not c. The verifier first computes $\langle \boldsymbol{a}_1,\ \pi_1 \rangle = c_t$, then checks

$c_t \in \langle t \rangle$,[5] and verifies

$$\langle \boldsymbol{a}_0,\ \pi_0 \rangle \cdot t = c_t \cdot \mathsf{ck}_f - z \cdot t$$

as well as the standard norm checks. This works because

$$t(\langle \boldsymbol{a},\ \pi_0 \rangle - c \cdot \mathsf{ck}_f + z) \equiv 0 \bmod q$$

implies

$$\langle \boldsymbol{a}_0,\ \pi_0 \rangle = c \cdot \mathsf{ck}_f - z$$

by assumption (2) Since c is a deterministic function of π_1, an extractor can still obtain $\boldsymbol{x}^*$ given π_1. To see the evaluation proof check still holds, note that $t \leftarrow\$ \mathcal{T}$ is sampled such that finding $s' = 0$ and $s't = 0$ is hard. This implies that if $lt = rt$, then $l - r = 0$.

Distinct vs. equal $\boldsymbol{a_0}$ and $\boldsymbol{a_1}$. We choose to fix a single public R-SIS vector $\boldsymbol{a}$ with respect to which preimages for v^i and $v^i t$ are generated, as this enables better verifier complexity. We note that with distinct trapdoor SIS vectors $\boldsymbol{a}_0, \boldsymbol{a}_1$, we can further optimize proof size, but only very slightly in our experiments.

The most expensive part for the verifier is computing $\langle \boldsymbol{a}_0,\ \pi_0 \rangle$ and $\langle \boldsymbol{a}_1,\ \pi_1 \rangle$, which requires $O(n \log n \cdot (\ell_0 + \ell_1))$ multiplications between the arbitrary $\mathcal{R}_q$ elements in the $\boldsymbol{a}_i$ and the bounded norm elements in the π_i. For concrete parameters, we believe these multiplications might be computed most efficiently using an NTT-based algorithm [BHKPY22]. Either way, by fixing a single $\boldsymbol{a} = \boldsymbol{a}_0 = \boldsymbol{a}_1$ the verifier runtime can be nearly halved using the verification equation

$$\langle \boldsymbol{a},\ (\mathsf{ck}_f \cdot \pi_1 - t \cdot \pi_0) \rangle = z \cdot t$$

Computationally indistinguishable trapdoor matrix. Recall that trapdoor public RSIS vectors in [MP12] are of the form $\boldsymbol{a} = [\hat{\boldsymbol{a}} | \boldsymbol{g} - \boldsymbol{r}\hat{\boldsymbol{a}}]$, where $\hat{\boldsymbol{a}} \leftarrow\$ \mathcal{R}_q^\ell$, $\boldsymbol{g}$ is the public gadget vector and $\boldsymbol{r}$ is the trapdoor/ In prior work [ACLMT22], $\ell = \lceil \log_\beta q \rceil$ was chosen, where it can be shown using the leftover hash lemma the resulting $\boldsymbol{a}$ is indistinguishable from random. We can instead use $\ell = 2$, where one can see the resulting $\boldsymbol{a}$ is an RLWE sample, $[\hat{\boldsymbol{a}}, \boldsymbol{g} - \hat{A}\boldsymbol{r}]$ where $\hat{\boldsymbol{a}} = [1 | \hat{\boldsymbol{a}}^\intercal \leftarrow \mathcal{R}_q^{\log_b q}] \in \mathcal{R}_q^{\log_b q \times 2}$, for some $b = O(\beta)$ and $\boldsymbol{r} \leftarrow\$ \mathcal{D}$ where $\mathcal{D}$ is some distribution for the RLWE secret and error. Experimentally, we find the best distribution, b, β such that the proof size is minimized and the RLWE problem has 128-bit hardness.

Other trapdoor optimizations. We only consider one optimization to the computationally indistinguishable variant of [MP12]: the heuristic from [GMPW20] for computing s_1. However, works focused on improving lattice signature sizes have introduced optimizations we estimate together could reduce our proof sizes by 5–10× [CGM; DSH21; JHT22]. We leave parameterizing such variants as future work.

5.2 Proof and CRS sizes

In the tables below we present results for our optimized bounded integer maps for input alphabets $\mathcal{X}_1 = \{0, 1, 2\}$ and $\mathcal{X}_{32} = \{0, \ldots, 2^{32} + 1\}$, both with and without extractability, at the 128 bit classical security level. Our proof sizes are competitive with other non-recursive post-quantum

[5] Recall that we pick t by setting half its NTT coefficients t_i to 0 and picking the rest at random from $\mathbb{Z}_q$. The condition $b \in \langle t \rangle$ is equivalent to $b_i \in \langle t_i \rangle$ (in the NTT basis), which can be verified by checking $b_i = 0$ for all i such that $t_i = 0$ since $b_i, t_i \in \mathbb{F}_q$.

arguments, especially for witnesses of length $> 2^{25}$. The verifier keys remain very small even up to 2^{30}, while for larger instances the prover key grows quite large.

Sparse inputs. We note that the proof sizes below are actually upper bounds. When working with sparse f and/or $\boldsymbol{x}$, as is common in practice, the proof size will be dependent on the number of non-zero entries

$\mathcal{X}_{32}$, extractable	2^{18}	2^{19}	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}
commitment (B)	147	167	154	163	173	188	185	200	211	227	243	236	252
total proof size (KiB)	845	904	977	1,028	1,081	1,137	1,197	1,257	1,315	1,376	1,438	1,505	1,571
witness compression	1.2×	2.3×	4.2×	8.0×	15×	28×	54×	104×	199×	381×	729×	1,394×	2,670×
verifier key size (MiB)	28	29	31	32	33	34	36	37	38	40	41	42	43
prover key size (GiB)	0.3	0.7	1.3	2.8	5.9	12.8	25.3	53.2	111	234	494	982	2,070

$\mathcal{X}_{32}$, binding	2^{17}	2^{18}	2^{19}	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}
commitment (B)	51	56	57	60	65	66	71	75	76	82	83	92	93	95
total proof size (KiB)	264	278	294	311	327	344	360	379	398	415	434	453	474	494
witness compression	1.9×	3.7×	7.0×	13×	25×	48×	91×	173×	329×	632×	1207×	2314×	4427×	8483×
verifier key size (MiB)	1.7	1.9	2.1	2.2	2.4	2.6	2.8	3.0	3.2	3.4	3.7	3.9	4.2	4.4
prover key size (GiB)	< 0.1	0.1	0.2	0.3	0.7	1.5	3.1	6.6	13.5	28.2	57.7	124	253	517

$\mathcal{X}_{1}$, extractable	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}
commitment (B)	106	118	129	145	162	188	166	180	178
total proof size (KiB)	668	712	757	805	862	931	986	1,038	1,096
witness compression	1.2×	2.3×	4.3×	8.1×	15×	28×	53×	100×	190×
verifier key size (MiB)	2.6	2.8	3.0	3.2	3.4	3.7	3.9	4.2	4.4
prover key size (GiB)	3.8	8.3	17.7	38.4	83.7	187	360	764	1,545

$\mathcal{X}_{1}$, binding	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}	2^{27}	2^{28}	2^{29}	2^{30}
commitment (B)	35	38	42	42	49	50	51	55	59	63	65
total proof size (KiB)	165	178	189	202	215	229	243	256	272	286	302
witness compression	1.2×	2.3×	4.3×	8.0×	15×	28×	54×	101×	191×	364×	689×
verifier key size (MiB)	1.0	1.1	1.3	1.4	1.5	1.7	1.8	1.9	2.1	2.3	2.5
prover key size (GiB)	0.2	0.5	1.0	2.1	4.5	9.3	19.1	40.5	85.5	180	371

5.2.1 Comparison to [ACLMT22]

For sake of better comparison with [ACLMT22], in Figure 2 we consider an unoptimized version of Orbweaver (as trit decomposition in Section 4.2, better trapdoor sampling, and many other types of optimizations in Section 5.1 can also be applied to [ACLMT22]). We also note that due to errors in their code, they (1) set a root Hermite factor much smaller than currently achievable (resulting in a security level higher than 128 bits); and (2) overestimate the desired preimage quality, i.e., the shortness/norm bound of the preimages. In the comparison below, we apply the same errors that "oversecure" our scheme for $\lambda = 128$.

In order to make the comparison as fair as possible, we also modified the code of [ACLMT22] to remove the overhead induced by their scheme's support for batching and higher degree polynomial maps. While our unoptimized proof sizes are about $2-3\times$ smaller, perhaps even more important practically was our reduction of the CRS size and prover time from quadratic to quasilinear in w and α. The Orbweaver CRS size is calculated as $3w\ell n \log\beta$ bits, as it consists of $3w$ vectors of ring elements, each of which with bit length bounded by by $\log(\beta)$. In contrast, [ACLMT22] requires about $w^2\ell n \log\beta$ bits for the SRS.

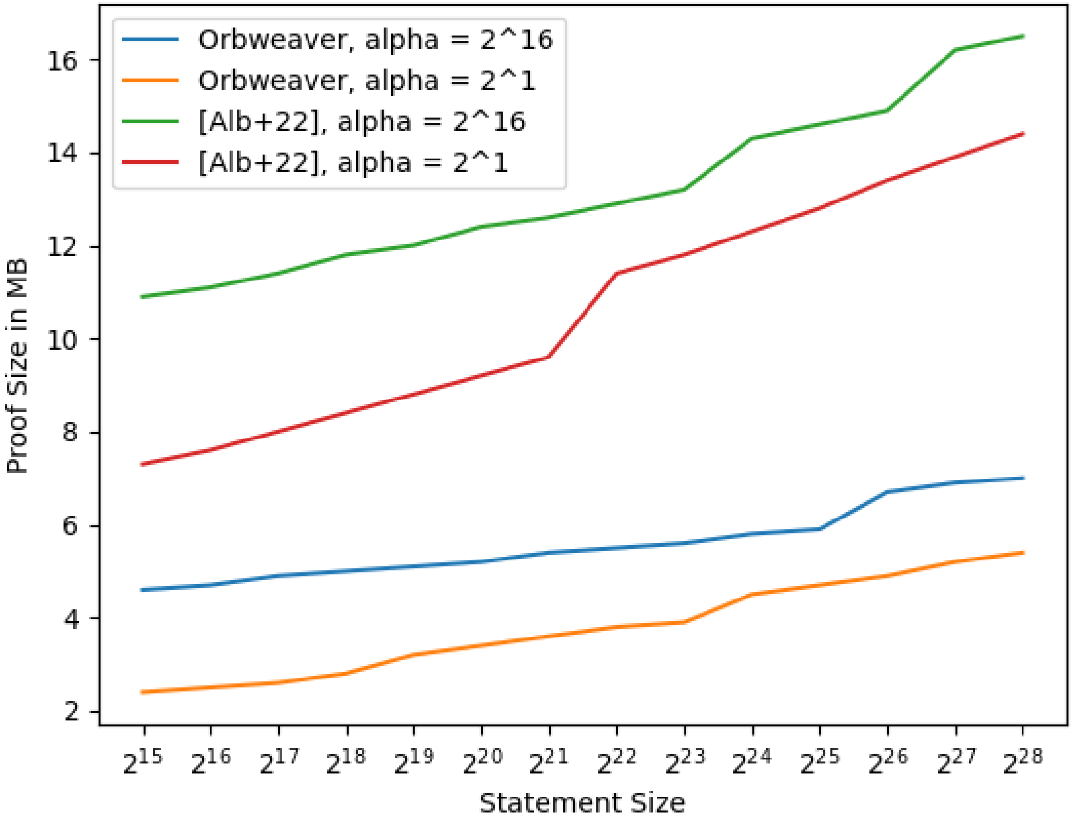


Figure 2: Comparison of combined commitment and proof size for an unoptimized version of Orbweaver and [ACLMT22] with $n \geq 2^9$. We have purposely left errors in the code when producing the data in this figure for better comparison with [ACLMT22] (see details at the start of Section 5.2.1).

Acknowledgments

We would like to thank Martin Albrecht and Russell W.F. Lai for answering questions about [ACLMT22]. We would also like to thank Nicholas Genise for answering our questions about lattice trapdoors.

References

[AC20] T. Attema and R. Cramer. “Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics”. In: *40th Annual International Cryptology Conference*. CRYPTO ’20. 2020, pp. 513–543 (cited on p. 4).

[ACK21] T. Attema, R. Cramer, and L. Kohl. “A Compressed Σ-Protocol Theory for Lattices”. In: *CRYPTO 2021, Part II*. Vol. 12826. LNCS. Springer, Aug. 2021, pp. 549–579 (cited on pp. 3, 4).

[ACLMT22] M. R. Albrecht, V. Cini, R. W. F. Lai, G. Malavolta, and S. A. K. Thyagarajan. “Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable - (Extended Abstract)”. In: *CRYPTO 2022, Part II*. Vol. 13508. LNCS. Springer, Aug. 2022, pp. 102–132 (cited on pp. 3–5, 7, 9–13, 17, 18, 21–24).

[AHIV17] S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. “Ligero: Lightweight Sublinear Arguments Without a Trusted Setup”. In: *2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017, pp. 2087–2104 (cited on p. 4).

[Ajt96] M. Ajtai. “Generating Hard Instances of Lattice Problems (Extended Abstract)”. In: *Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC ’96. 1996, pp. 99–108 (cited on p. 9).

[AL21] M. R. Albrecht and R. W. F. Lai. “Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices”. In: *CRYPTO 2021, Part II*. Vol. 12826. LNCS. Springer, Aug. 2021, pp. 519–548 (cited on pp. 3, 7, 18).

[BBBPWM18] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. “Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting”. In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 315–334 (cited on p. 3).

[BBCPGL19] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. “Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations”. In: *38th Annual International Cryptology Conference*. CRYPTO ’19. 2019, pp. 669–699 (cited on p. 4).

[BBHR18] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. “Fast Reed-Solomon Interactive Oracle Proofs of Proximity”. In: *ICALP 2018*. Vol. 107. LIPIcs. Schloss Dagstuhl, July 2018, 14:1–14:17 (cited on p. 3).

[BBHR19] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev. “Scalable Zero Knowledge with No Trusted Setup”. In: *CRYPTO 2019, Part III*. Vol. 11694. LNCS. Springer, Aug. 2019, pp. 701–732 (cited on pp. 2, 4).

[BCCGP16] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. “Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting”. In: *EUROCRYPT 2016, Part II*. Vol. 9666. LNCS. Springer, May 2016, pp. 327–357 (cited on p. 3).

[BCCT12] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. “Functional Commitments for All Functions, with Transparent Setup”. In: *ITCS 2012*. ACM, Jan. 2012, pp. 326–349 (cited on p. 2).

[BCCT13] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer. “Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments”. In: *45th ACM STOC*. ACM Press, June 2013, pp. 111–120 (cited on p. 2).

[BCFL22] D. Balbás, D. Catalano, D. Fiore, and R. W. F. Lai. *Functional Commitments for Circuits from Falsifiable Assumptions*. Cryptology ePrint Archive, Report 2022/1365. 2022 (cited on p. 4).

[BCIOP13] N. Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky, and O. Paneth. "Succinct Non-interactive Arguments via Linear Interactive Proofs". In: *TCC 2013*. Vol. 7785. LNCS. Springer, Mar. 2013, pp. 315–333 (cited on p. 2).

[BCPS18] A. Bishnoi, P. L. Clark, A. Potukuchi, and J. R. Schmitt. "Functional Commitments for All Functions, with Transparent Setup". In: *Comb. Probab. Comput.* (2018), pp. 310–333. URL: https://arxiv.org/pdf/1508.06020v2.pdf (cited on p. 7).

[BCRSVW19] E. Ben-sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. Ward. "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup". In: *Advances in Cryptology – EUROCRYPT 2019*. 2019, pp. 103–128 (cited on p. 4).

[BCS21] J. Bootle, A. Chiesa, and K. Sotiraki. "Sumcheck Arguments and Their Applications". In: *Advances in Cryptology – CRYPTO 2021*. 2021, pp. 742–773 (cited on p. 4).

[BDGL16] A. Becker, L. Ducas, N. Gama, and T. Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving". In: *27th SODA*. ACM-SIAM, Jan. 2016, pp. 10–24 (cited on p. 13).

[Ben+14] E. Ben-Sasson et al. "Functional Commitments for All Functions, with Transparent Setup". In: *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 459–474 (cited on p. 2).

[BFS19] B. Bünz, B. Fisch, and A. Szepieniec. *Transparent SNARKs from DARK Compilers*. Cryptology ePrint Archive, Report 2019/1229. https://eprint.iacr.org/2019/1229. 2019 (cited on pp. 2, 3).

[BFS20] B. Bünz, B. Fisch, and A. Szepieniec. "Transparent SNARKs from DARK Compilers". In: *EUROCRYPT 2020, Part I*. Vol. 12105. LNCS. Springer, May 2020, pp. 677–706 (cited on p. 2).

[BGH19] S. Bowe, J. Grigg, and D. Hopwood. *Halo: Recursive Proof Composition without a Trusted Setup*. Cryptology ePrint Archive, Report 2019/1021. https://eprint.iacr.org/2019/1021. 2019 (cited on p. 2).

[BHKPY22] H. Becker, V. Hwang, M. J. Kannwischer, L. Panny, and B.-Y. Yang. "Efficient Multiplication of Somewhat Small Integers Using Number-Theoretic Transforms". In: *Advances in Information and Computer Security*. Springer Nature. 2022 (cited on pp. 17, 22).

[BLNS20] J. Bootle, V. Lyubashevsky, N. K. Nguyen, and G. Seiler. "A non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *CRYPTO 2020, Part II*. Vol. 12171. LNCS. Springer, Aug. 2020, pp. 441–469 (cited on pp. 3, 4).

[BS22] W. Beullens and G. Seiler. *LaBRADOR: Compact Proofs for R1CS from Module-SIS*. Cryptology ePrint Archive, Paper 2022/1341. 2022 (cited on pp. 3, 4, 18).

[BSS22] K. Boudgoust, A. Sakzad, and R. Steinfeld. "Vandermonde Meets Regev: Public Key Encryption Schemes Based on Partial Vandermonde Problems". In: *Des. Codes Cryptography* (2022), pp. 1899–1936 (cited on p. 12).

[CCHLRR18] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, and R. D. Rothblum. *Fiat-Shamir From Simpler Assumptions*. Cryptology ePrint Archive, Report 2018/1004. 2018 (cited on p. 18).

[CGM] Y. Chen, N. Genise, and P. Mukherjee. "Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures". In: *25th International Conference on the Theory and Application of Cryptology and Information Security*. Ed. by S. D. Galbraith and S. Moriai. ASIACRYPT '19, pp. 3–32 (cited on p. 22).

[CHMMVW20] A. Chiesa, Y. Hu, M. Maller, P. Mishra, P. Vesely, and N. P. Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *EUROCRYPT 2020, Part I*. Vol. 12105. LNCS. Springer, May 2020, pp. 738–768 (cited on p. 2).

[COS19] A. Chiesa, D. Ojha, and N. Spooner. *Transparent SNARKs from DARK Compilers*. Cryptology ePrint Archive, Report 2019/1076. https://eprint.iacr.org/2019/1076. 2019 (cited on p. 2).

[CP22] L. de Castro and C. Peikert. *Functional Commitments for All Functions, with Transparent Setup*. Cryptology ePrint Archive, Paper 2022/1368. 2022 (cited on p. 4).

[DSH21] A. L. Dévéhat, H. Shizuya, and S. Hasegawa. "On the Higher-Bit Version of Approximate Inhomogeneous Short Integer Solution Problem". In: CANS. 2021, pp. 253–272 (cited on p. 22).

[ENS20] M. Esgin, N. K. Nguyen, and G. Seiler. "Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations". In: *Advances in Cryptology – ASIACRYPT 2020*. 2020, pp. 259–288 (cited on pp. 4, 7).

[ESLL19] M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. "Lattice-Based Zero-Knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications". In: *39th Annual International Cryptology Conference*. CRYPTO '19. 2019, pp. 115–146 (cited on p. 4).

[ESLR22] M. F. Esgin, R. Steinfeld, D. Liu, and S. Ruj. *Functional Commitments for All Functions, with Transparent Setup*. Cryptology ePrint Archive, Paper 2022/141. 2022 (cited on p. 4).

[GGPR13] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *EUROCRYPT 2013*. Vol. 7881. LNCS. Springer, May 2013, pp. 626–645 (cited on p. 2).

[GLSTW21] A. Golovnev, J. Lee, S. Setty, J. Thaler, and R. S. Wahby. *Brakedown: Linear-time and post-quantum SNARKs for R1CS*. Cryptology ePrint Archive, Paper 2021/1043. 2021 (cited on p. 4).

[GM17] J. Groth and M. Maller. "Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs". In: *CRYPTO 2017, Part II*. Vol. 10402. LNCS. Springer, Aug. 2017, pp. 581–612 (cited on p. 2).

[GM18] N. Genise and D. Micciancio. "Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus". In: *Advances in Cryptology – EUROCRYPT 2018*. 2018, pp. 174–203 (cited on p. 9).

[GMNO18] R. Gennaro, M. Minelli, A. Nitulescu, and M. Orrù. "Lattice-Based Zk-SNARKs from Square Span Programs". In: *2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS '18. 2018, pp. 556–573 (cited on p. 4).

[GMPW20] N. Genise, D. Micciancio, C. Peikert, and M. Walter. "Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography". In: PKC. 2020, pp. 623–651 (cited on p. 22).

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. "Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus". In: *Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. 2008, pp. 197–206 (cited on p. 9).

[Gro10] J. Groth. "Functional Commitments for All Functions, with Transparent Setup". In: *ASIACRYPT 2010*. Vol. 6477. LNCS. Springer, Dec. 2010, pp. 321–340 (cited on pp. 2, 4).

[Gro16] J. Groth. "On the Size of Pairing-based Non-interactive Arguments". In: *EUROCRYPT 2016, Part II*. Vol. 9666. LNCS. Springer, May 2016, pp. 305–326 (cited on p. 2).

[GW11] C. Gentry and D. Wichs. "Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions". In: *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*. STOC '11. San Jose, California, USA: Association for Computing Machinery, 2011 (cited on p. 11).

[GWC19] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. *PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge*. Cryptology ePrint Archive, Report 2019/953. `https://eprint.iacr.org/2019/953`. 2019 (cited on p. 2).

[HH21] D. Harvey and J. van der Hoeven. "Integer multiplication in time O(n log n)". In: *Annals of Mathematics* (2021) (cited on p. 17).

[ISW21] Y. Ishai, H. Su, and D. J. Wu. "Shorter and Faster Post-Quantum Designated-Verifier ZkSNARKs from Lattices". In: *2021 ACM SIGSAC Conference on Computer and Communications Security*. CCS '21. 2021, pp. 212–234 (cited on p. 4).

[JHT22] H. Jia, Y. Hu, and C. Tang. "Lattice-based hash-and-sign signatures using approximate trapdoor, revisited". In: *IET Inf. Secur.* 16.1 (2022), pp. 41–50 (cited on p. 22).

[Kil92] J. Kilian. "A note on efficient zero-knowledge proofs and arguments (extended abstract)". In: *Symposium on the Theory of Computing*. 1992 (cited on p. 4).

[KZG10] A. Kate, G. M. Zaverucha, and I. Goldberg. "Constant-Size Commitments to Polynomials and Their Applications". In: *ASIACRYPT 2010*. Vol. 6477. LNCS. Springer, Dec. 2010, pp. 177–194 (cited on pp. 2, 3).

[Lip12] H. Lipmaa. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *TCC 2012*. Vol. 7194. LNCS. Springer, Mar. 2012, pp. 169–189 (cited on p. 2).

[LNP22] V. Lyubashevsky, N. K. Nguyen, and M. Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *Advances in Cryptology – CRYPTO 2022*. 2022, pp. 71–101 (cited on pp. 4, 7).

[LNS20] V. Lyubashevsky, N. K. Nguyen, and G. Seiler. "Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations". In: *2020 ACM SIGSAC Conference on Computer and Communications Security*. CCS '20. 2020, pp. 1051–1070 (cited on p. 4).

[LRY16] B. Libert, S. C. Ramanna, and M. Yung. "Functional Commitment Schemes: From Polynomial Commitments to Pairing-Based Accumulators from Simple Assumptions". In: *ICALP 2016*. Ed. by I. Chatzigiannakis, M. Mitzenmacher, Y. Rabani, and D. Sangiorgi. Vol. 55. LIPIcs. Schloss Dagstuhl, July 2016, 30:1–30:14. DOI: `10.4230/LIPIcs.ICALP.2016.30` (cited on p. 4).

[LS20] V. Lyubashevsky and G. Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs". In: EUROCRYPT '20. 2020, pp. 204–224 (cited on p. 7).

[MBKM19] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn. "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings". In: *ACM CCS 2019*. ACM Press, Nov. 2019, pp. 2111–2128 (cited on p. 2).

[Mic02] D. Micciancio. "Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions". In: *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* 2002, pp. 356–365 (cited on pp. 9, 10).

[MP12] D. Micciancio and C. Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012*. Vol. 7237. LNCS. Springer, Apr. 2012, pp. 700–718 (cited on pp. 3, 9, 22).

[MR09] D. Micciancio and O. Regev. "Functional Commitments for All Functions, with Transparent Setup". In: *Post-Quantum Cryptography*. Ed. by D. J. Bernstein, J. Buchmann, and E. Dahmen. 2009, pp. 147–191 (cited on p. 13).

[PHGR13] B. Parno, J. Howell, C. Gentry, and M. Raykova. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *2013 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2013, pp. 238–252 (cited on p. 2).

[PXWC21] Y. Pan, J. Xu, N. Wadleigh, and Q. Cheng. "On the Ideal Shortest Vector Problem over Random Rational Primes". In: *EUROCRYPT 2021, Part I*. Ed. by A. Canteaut and F.-X. Standaert. Vol. 12696. LNCS. Springer, Oct. 2021, pp. 559–583. DOI: 10.1007/978-3-030-77870-5_20 (cited on p. 13).

[WW22] H. Wee and D. J. Wu. *Succinct Vector, Polynomial, and Functional Commitments from Lattices*. Cryptology ePrint Archive, Paper 2022/1515. 2022 (cited on pp. 3, 4).

[XZS22] T. Xie, Y. Zhang, and D. Song. "Orion: Zero Knowledge Proof with Linear Prover Time". In: *Advances in Cryptology – CRYPTO 2022*. Ed. by Y. Dodis and T. Shrimpton. 2022 (cited on p. 4).