Lattice-based Succinct Arguments from Vanishing Polynomials

(Full Version)

Valerio Cini[1*], Russell W. F. Lai[2], and Giulio Malavolta[3†]

[1] AIT Austrian Institute of Technology
[2] Aalto University
[3] Bocconi University & Max Planck Institute for Security and Privacy

Abstract. Succinct arguments allow a prover to convince a verifier of the validity of any statement in a language, with minimal communication and verifier's work. Among other approaches, lattice-based protocols offer solid theoretical foundations, post-quantum security, and a rich algebraic structure. In this work, we present some new approaches to constructing efficient lattice-based succinct arguments. Our main technical ingredient is a new commitment scheme based on *vanishing polynomials*, a notion borrowed from algebraic geometry. We analyse the security of such a commitment scheme, and show how to take advantage of the additional algebraic structure to build new lattice-based succinct arguments. A few highlights amongst our results are:

(i) The first recursive folding (i.e. Bulletproofs-like) protocol for linear relations with *polylogarithmic* verifier runtime. Traditionally, the verifier runtime has been the efficiency bottleneck for such protocols (regardless of the underlying assumptions).
(ii) The first verifiable delay function (VDF) based on lattices, building on a recently introduced sequential relation.
(iii) The first lattice-based *linear-time prover* succinct argument for NP, in the preprocessing model. The soundness of the scheme is based on (knowledge)-k-R-ISIS assumption [Albrecht et al., CRYPTO'22].

1 Introduction

A succinct non-interactive argument of knowledge (SNARK) [Kil92, Mic94] allows a prover to convince a verifier of the validity of an NP relation. The argument is said to be *succinct* if the size of the proof and the runtime of the verifier are sublinear in (or ideally independent of) the time needed to check the validity of the witness. Due to these strong efficiency requirements, SNARKs for NP have become a cornerstone of modern cryptography: They count a large array of applications [BCG$^+$14, GM17, KMS$^+$16, BGH19, BDFG21, BMRS20] and have recently found their way into real-world systems in the context of blockchain-based cryptocurrencies [BCG$^+$14, GM17, KMS$^+$16, BGH19, BDFG21, BMRS20].

A promising approach for constructing efficient SNARKs is to leverage the algebraic structure offered by computational problems in lattice-based cryptography [BISW17, BISW18, GMNO18, BLNS20, AL21, ACK21, ACL$^+$22]. Compared to other approaches (see Section 1.2 for a detailed discussion), lattice-based SNARKs offer many desirable properties: (i) They are conjectured to be secure against quantum attacks, (ii) are based on computational problems with solid theoretical foundations, and (iii) have a rich algebraic structure, allowing to prove many interesting statements "natively", i.e. without needing to run the relation through an expensive Karp reduction.

In spite of these promising properties, lattice-based SNARKs are still somewhat limited compared to competing approaches. In particular, known lattice-based schemes suffer from (at least) one of the following limitations:

- They require the verifier to hold some information secret from the prover, i.e. they are in the designated-verifier settings [BISW17, BISW18, GMNO18].
- They have a non-succinct verifier, whose runtime is at least linear in the size of the relation [BLNS20, AL21, ACK21].

[*] V.C. was funded by the Austrian Science Fund (FWF) and netidee SCIENCE grant P31621-N38 (PROFET).

[†] G.M. was partially funded by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038 and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972.

- They have a slow prover runtime, i.e. quartic [ACL+22] in the size of the relation.

In this work, we propose new techniques for lattice-based SNARKs that allow us to overcome these barriers, making lattice-based SNARKs qualitatively closer (and, in some aspects, superior) to other approaches.

1.1 Our Results

We present new algebraic techniques that allow us to overcome traditional limitations of lattice-based SNARKs. Our central technical ingredient is a new lattice-based commitment scheme based on *vanishing polynomials*, an object borrowed from algebraic geometry. The security of our commitment is based on the vanishing Short Integer Solution (vSIS) problem, a variant of the well-known SIS problem that we introduce in this work. We then show how to exploit the additional algebraic structure of vSIS to obtain new results for lattice-based succinct arguments. In more details, our contributions can be summarized as follows.

(1) The Vanishing-SIS Problem. We introduce the vSIS problem, a variant of the standard SIS over rings, which asks to find a polynomial with short coefficients which vanishes at the given point(s). We show that vSIS is no easier than the k-R-ISIS problem, a recently introduced family of problems [ACL+22]. We also show that vSIS can be explained as a natural generalisation of the search NTRU problem. We propose a worst-case to average-case reduction and a reduction from search NTRU, both conditioning on the hardness of decision NTRU.

(2) New Commitments Based on vSIS. We show that the vSIS problem immediately implies the existence of a commitment scheme with useful algebraic properties which are key to our new results in succinct arguments:

- Succinct: The size of the commitment key and the commitment are logarithmic in the size of the input. In particular, this implies that the commitment is also a collision-resistant hash function with very short key.
- Homomorphic: The commitment is (bounded) linearly homomorphic and multiplicatively homomorphic for a constant number of multiplications.
- Foldable: We show that the commitment can be "folded" (in the sense of folding arguments, e.g. Bulletproofs [BLNS20]) in such a way that the folded commitment key retains a succinct representation. Loosely speaking, this allows us to combine the two halves of the committed value and simultaneously half the size of the input *and* the size of the commitment key.

(3) Simple Method for Proving Quadratic Relations. Exploiting the multiplicatively homomorphic property of vSIS commitments, we show a simple method for reducing the task of proving quadratic relations to that of proving linear relations, with only additive quasi-linear overhead in prover time. As an example, to prove that $\langle \mathbf{x}_0, \mathbf{x}_1 \rangle = y$, the prover commits to the polynomials $\bar{p}_{\mathbf{x}_0}(V) = \sum_i x_{0,i} \cdot V^{-i}$ and $p_{\mathbf{x}_1}(V) = \sum_j x_{1,j} \cdot V^j$ as $\bar{c}_{\mathbf{x}_0}$ and $c_{\mathbf{x}_1}$ respectively, and proves the linear relations that the commitments are well-formed. Then, the prover proves that the product $\bar{c}_{\mathbf{x}_0} \cdot c_{\mathbf{x}_1}$, which the verifier can compute themself, is a commitment to a polynomial whose constant term is y, which is again a linear relation. Instantiating with succinct arguments for linear relations with quasi-linear-time prover, we obtain succinct arguments for quadratic relations also with quasi-linear-time prover.

(4a) New Folding Protocols for Structured SIS. The first kind of linear relations that we consider are *structured SIS relations* (roughly) of the form

$$\begin{pmatrix} \mathbf{A} & & & \\ \mathbf{B} & \mathbf{A} & & \\ & \mathbf{B} & \ddots & \\ & & \ddots & \mathbf{A} \\ & & & \mathbf{B} \\ \mathbf{C}_1 & \mathbf{C}_2 & \dots & \mathbf{C}_n \end{pmatrix} \cdot \mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0$$

where $\mathbf{C}_1, \ldots, \mathbf{C}_n$ conform to certain foldable structure. For such relations, we obtain SNARKs with transparent setup, quasi-linear time prover, and polylogarithmic time verifier (*without* preprocessing), in the random oracle model.[4] The main technical ingredient that enables this result is a new Bulletproof-like folding protocol for *foldable* linear relations, where the verifier runtime is *polylogarithmic* in the length of the relation. Prior folding protocols had a *linear-time verifier* [BLNS20, AL21, ACK21], including those based on the discrete logarithm problem [BCC+16, BBB+18], with the exception of [BMM+21] where the verifier computation is proportional to the square root of the length of the relation.

(4b) Optimised Knowledge-based Protocols for SIS. Next, we consider *unstructured SIS relations* of the form "$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\|\mathbf{x}\| \approx 0$". For these relations, we obtain SNARKs with quasi-linear time prover and polylogarithmic time verifier after preprocessing, based on the recently introduced (knowledge-)k-R-ISIS assumption [ACL+22]. This improves upon previous schemes which do not natively support proving modular arithmetic relations [ACL+22] and require at least a quadratic-time prover [ACL+22, BCFL22].

(5) Applications. Putting everything together, we obtain SNARKs for quadratic relations with quasi-linear-time prover and polylogarithmic-time verifier (after preprocessing for the unstructured case). We highlight two particular instances.

First, we obtain SNARKs for proving "$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\mathbf{x}$ is *exactly binary*". In particular, applying the structured instantiation on the recently introduced SIS-based sequential relations [LM23], we obtain the first lattice-based verifiable delay functions (VDF). Prior lattice-based schemes [YAZ+19, BLS19, ENS20, LNP22] for exact SIS relations[5] are not succinct.

Second, we obtain SNARKs for rank-1 constraint satisfiability (R1CS). Prior lattice-based schemes [ACL+22, BCFL22] have at least quadratic-time prover.

1.2 Related Work

There is a vast amount of literature on SNARKs for different classes of relations. We do not attempt to survey all existing works here, but rather provide a high-level overview of various approaches and discuss in details those that are closely related to our work.

Pairing-based. To date, the most efficient and feature-rich SNARKs are constructed over *bilinear pairing groups* (e.g. [Gro16]) with a trusted setup. Typically, they are publicly verifiable and have simple verification algorithms consisting of a constant amount of pairing-product equations. Moreover, pairing-based SNARKs offer a rich algebraic structures that is known to enable proof batching [LMR19, BMM+21] and efficient recursive composition [BCTV14].

Hash-based. Another approach to build SNARKs is to compile an information-theoretic proof system, e.g. a probabilistically checkable proof (PCP) [Kil92, Mic94] or an interactive oracle proof (IOP), via a vector commitment scheme. Since the vector commitment is usually instantiated with a Merkle-hash tree in the random oracle (RO) model, we call this the hash-based approach. A major difference between pairing-based and hash-based SNARKs, from both theoretical and practical perspectives, is the algebraic structure of the verification algorithm. The reliance of hash-based SNARKs on an RO makes recursive composition challenging, since an RO is typically instantiated with a hash function of high multiplicative degree. On the flip side, hash-based SNARKs can be shown to be post-quantum secure [CMS19].

Lattice-based. Finally, we discuss *lattice-based* approaches to build SNARKs. Until recently, lattice-based SNARKs required the verifier to keep a secret state hidden from the prover, i.e. they are in the designated verifier settings [GMNO18, ISW21]. Excitingly, recent development sees two emerging paradigms for constructing publicly verifiable SNARKs, both of which we improve upon in this work.

The first line of work [BLNS20, AL21, ACK21] studies lattice-based folding protocols which, as discussed above, give quasi-linear-time prover SNARKs in the random oracle model. However, due to lack of preprocessing support, the verifier complexity in folding protocols has always been *linear* in the size of the relation. In this work, we work around this barrier by considering structured relations which retain

[4] The interactive variant can be proven secure without random oracles.

[5] Not counting those for more general relations.

their foldable structures after folding, and obtain the first folding protocols with a polylogarithmic-time verifier.

Another line of work [ACL$^+$22, BCFL22] constructs publicly verifiable SNARKs in the preprocessing model. At the core of these constructions are functional commitment schemes which allow to succinctly prove that a committed vector $\mathbf{x}$ satisfies $f(\mathbf{x}) = \mathbf{y}$ for low-degree polynomials [ACL$^+$22] or even unbounded-depth circuits [BCFL22]. To this end, we propose a construction with *quasi-linear*-time prover using similar techniques, while in [ACL$^+$22, BCFL22] the prover has at least quadratic complexity. We remark that while the recent work of Wee and Wu [WW23] constructs functional commitments for circuits, their scheme does not support preprocessing and therefore has inefficient verifier.

1.3 Subsequent Work

We have been informed that a recent result [Ano23] shows a counterexample that morally invalidates the knowledge version of the assumption introduced in [ACL$^+$22]. Although this is a strong indication that some algorithms may not be captured by the security model that some of our schemes (e.g. those presented in Section 7) are proven against, it does not imply a direct attack against any of our schemes. Furthermore, even in light of these recent findings, we believe that our security proofs are meaningful as *sanity checks*, for the same reason as proofs in other unsound models, such as the random oracle model or the generic group model, are also meaningful.

2 Technical Overview

We provide a high-level overview of the techniques that we develop in this work. First, we present our main new technical ingredient that is at the center of our results, namely a new commitment based on vanishing-SIS. Then we show how arguments for vanishing-SIS commitments can be efficiently composed into an argument for binary-satisfiability of both structured and unstructured linear relations. Finally, we describe our new succinct arguments in both the structured and unstructured settings, and present some immediate applications.

Throughout this overview, we will work with a cyclotomic field $\mathcal{K} = \mathbb{Q}(\zeta)$ where ζ is a root of unity of some prime order ρ, its ring of integers $\mathcal{R} = \mathbb{Z}[\zeta]$, and the quotient rings $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ for different values of $q \in \mathbb{N}$. Ring elements will be represented by their coefficient embedding and the norm of a ring element is defined accordingly. Readers not familiar with these objects can treat $\mathcal{K} = \mathbb{Q}$ and $\mathcal{R} = \mathbb{Z}$, which suffices in most places.

2.1 Vanishing-SIS Commitments

The main technical ingredient behind of our results is a new family of commitment schemes for committing to short vectors $\mathbf{x} \in \mathcal{R}^d$ and companion argument systems for proving that the committed vector is in fact a bit string, i.e. $\mathbf{x} \in \{0,1\}^d$. In their simplest form, the commitment key is a single random element $v \leftarrow\$ \mathcal{R}_q^\times$, where $\mathcal{R}_q^\times$ is the set of invertible elements in $\mathcal{R}_q$. To commit to a *short* $\mathbf{x} \in \mathcal{R}^d$, we interpret $\mathbf{x}$ as the coefficients of a degree-d polynomial $p_{\mathbf{x}}(V)$ without a constant term, and compute the commitment as the evaluation of $p_{\mathbf{x}}$ at the point v modulo q, i.e.

$$p_{\mathbf{x}}(V) = \sum_{i=1}^{d} x_i \cdot V^i \qquad \text{and} \qquad c = p_{\mathbf{x}}(v) \bmod q.$$

We refer to this family of commitment schemes as the vanishing short integer solution (vSIS) commitments, for reasons that will become clear shortly. The binding property of the vSIS commitment above is based on the following vSIS assumption which we introduce in this work.

Definition 1 (vSIS, Informal). *Given a random point $v \leftarrow\$ \mathcal{R}_q^\times$, it is hard to find a degree-d polynomial $p = \sum_{i=0}^{d} p_i \cdot V^i \in \mathcal{R}[V]$ with short coefficients such that $p(v) = 0 \bmod q$. In other words, p is a short element in $\mathcal{I}(v)$, the ideal (lattice) of polynomials vanishing at the given point v.*

In general, the vSIS assumption could be parametrised by a set $\mathcal{G}$ of (multivariate) monomials[6] over $\mathcal{R}$, where the task is to find a short linear combination $(p_g)_{g\in\mathcal{G}}$ such that $\sum_{g\in\mathcal{G}} p_g \cdot g(\mathbf{v}) = 0 \bmod q$. To gain confidence in its validity, we show that the vSIS assumptions are implied by the k-R-ISIS assumptions introduced in [ACL+22]. For certain parameter regimes (although *not* the ones that we need), we show that the vSIS problem is as hard as the search NTRU problem, conditioned on the hardness of the decision NTRU problem. For more details, we refer the reader to Section 4 and Appendix A.

The vSIS commitment schemes have nice homomorphic properties. For starters, they are clearly linearly homomorphic, similarly to the standard SIS-based commitments. More importantly for us, they are also bounded *multiplicatively homomorphic*: If c_f and c_g commit to the polynomials f and g respectively, then $c_f \cdot c_g \bmod q$ commits to the polynomial $f \cdot g$. An elementary fact that will be particularly useful later, is that if $g(V) = f(V^{-1})$, then the constant term of $f \cdot g$ is given by the inner-product of the coefficients of f and g.

Proof of Binary-Satisfiability of Linear Relations. As a warm-up, we outline the construction of a succinct argument system for a prover to convince a verifier that a vector $\mathbf{x} \in \mathcal{R}^d$ satisfies

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \mathbf{x} \in \{0,1\}^d.$$

As building blocks, we will use succinct argument systems for SIS relations with soundness gaps, i.e. they are complete and sound for relations of the form

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0$$

but the constraints on the shortness of $\mathbf{x}$ differ. That is, we will turn succinct arguments for showing that $\mathbf{x}$ satisfying a linear relation is short, into an argument for showing the $\mathbf{x}$ is *exactly binary*. While this may seem like a technicality, this *proof of binariness* will be crucial for our later applications, and can be generalised to prove arbitrary quadratic relations. Later in this overview, we will also show how to instantiate the required building blocks.

The common reference string of our argument system contains a random vector $\mathbf{h} \in \mathcal{R}_{q_1}^d$ and a vSIS commitment key $v \in \mathcal{R}_{q_2}^\times$, where $q_0 \ll q_1 \ll q_2$ and the purpose of $\mathbf{h}$ will become clear later. For $\mathbf{x} \in \mathcal{R}^d$ and $\mathbf{w} = (\mathbf{w}_-, \mathbf{w}_+) \in \mathcal{R}^{2d}$, define the (Laurent) polynomials

$$\bar{p}_{\mathbf{x}}(V) := p_{\mathbf{h}\circ\mathbf{x}}(V^{-1}) \qquad \text{and} \qquad \tilde{p}_{\mathbf{w}}(V) := p_{\mathbf{w}_-}(V^{-1}) + p_{\mathbf{w}_+}(V)$$

where $\mathbf{h} \circ \mathbf{x}$ denotes the Hadamard (component-wise) product of the two vectors. The argument proceeds as follows:

(i) The prover reveals the following "complementary" vSIS commitments to $\mathbf{x}$:

$$c_{\mathbf{x}} := p_{\mathbf{x}}(v) \bmod q_2 \qquad \text{and} \qquad \bar{c}_{\mathbf{x}} := \bar{p}_{\mathbf{x}}(v) \bmod q_2.$$

(ii) The prover then proves the following relations:

$$\exists\ \mathbf{x} \in \mathcal{R}^d, \quad \begin{matrix} \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0, \\ p_{\mathbf{x}}(v) = c_{\mathbf{x}} \bmod q_2, \\ \bar{p}_{\mathbf{x}}(v) = \bar{c}_{\mathbf{x}} \bmod q_2, \end{matrix} \quad \text{and} \quad \|\mathbf{x}\| \approx 0. \tag{1}$$

$$\exists\ \mathbf{w} \in \mathcal{R}^{2d}, \quad \tilde{p}_{\mathbf{w}}(v) = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{p}_{\mathbf{1}}(v)) \bmod q_2 \quad \text{and} \quad \|\mathbf{w}\| \approx 0. \tag{2}$$

Since $p_{\mathbf{x}}(v)$, $\bar{p}_{\mathbf{x}}(v)$, and $\tilde{p}_{\mathbf{w}}(v)$ can be computed as linear functions evaluated at the monomials expansion of v, Eqs. (2) and (6) can be proven by using argument systems for SIS relations, as required above.

The interesting bit of our protocols is that, even though the the underlying arguments for the SIS relation have soundness gaps, the verifier of our protocol will be convinced that $\mathbf{x}$ is *exactly* binary. First, from the knowledge soundness of the argument for Eq. (1), the verifier is convinced that there exists a candidate short vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{w}}$ satisfying Eq. (1) and Eq. (2) respectively. From $\hat{\mathbf{x}}$, one could derive a short vector $\hat{\mathbf{u}} = (\hat{\mathbf{u}}_-, \hat{u}_0, \hat{\mathbf{u}}_+) \in \mathcal{R}^{2d+1}$ encoding

$$\hat{p}_{\hat{\mathbf{u}}}(V) := p_{\mathbf{x}}(V) \cdot \bar{p}_{\mathbf{x}-\mathbf{1}}(V) = p_{\mathbf{x}}(V) \cdot p_{\mathbf{h}\circ(\mathbf{x}-\mathbf{1})}(V^{-1}).$$

[6] Or rational functions in general.

Clearly, $\hat{p}_{\hat{\mathbf{u}}}(v) = c_{\mathbf{x}} \cdot (\bar{c}_{\mathbf{x}} - \bar{p}_{\mathbf{1}}(v)) \bmod q_2$. This means that $\tilde{p}_{\hat{\mathbf{w}}}(V) - \hat{p}_{\hat{\mathbf{u}}}(V)$ is a polynomial with short coefficients which vanishes at v. Furthermore, notice that $\tilde{p}_{\hat{\mathbf{w}}}$ does not have a constant term, while the constant term $\hat{u}_0$ of $\hat{p}_{\hat{\mathbf{u}}}$ is given by the inner-product

$$\hat{u}_0 = \langle \mathbf{x}, \mathbf{h} \circ (\mathbf{x} - \mathbf{1}) \rangle = \sum_{i=1}^{d} h_i \cdot \underbrace{x_i \cdot (x_i - 1)}_{=0 \text{ iff } x_i \in \{0,1\}} .$$

Let us first establish that $\hat{u}_0$ must indeed be 0. This is an easy reduction to the vSIS, since it would otherwise yield a non-zero short solution to a vSIS problem, which we assume to be hard to find. However, we are not yet done, since the fact that $\hat{u}_0 = 0$ *does not* imply that all of its summands are also zero (which is what we need to ensure that $\mathbf{x}$ is binary). This is where the vector $\mathbf{h}$ comes into play, using a technique first introduced in [ACL+22]: Suppose $\hat{u}_0 = 0$, then we also have $\hat{u}_0 = \sum_{i=1}^{d} h_i \cdot x_i \cdot (x_i - 1) = 0 \bmod q_1$. If $\mathbf{x}$ is not binary, the vector $\mathbf{x} \circ (\mathbf{x} - \mathbf{1})$ would be a short non-zero solution to the RingSIS instance given by $\mathbf{h}$ over $\mathcal{R}_{q_1}$.

2.2 Efficient Proofs for SIS Relations

In the above proof of binary-satisfiability of linear relations, the prover and verifier computation costs are dominated by the costs of the succinct arguments for SIS relations with soundness gaps. Here we discuss two approaches in the literature, and how we can improve on both fronts using the algebraic properties of our vSIS-based commitment scheme.

Approach I: Folding Protocols. (Lattice-based) Bulletproofs [BLNS20, AL21, ACK21] are interactive arguments with quasi-linear time prover, and can be made non-interactive using the Fiat-Shamir transform in the random oracle model. It is based on the technique of iteratively "folding" the relation into a smaller one until a trivial relation is derived. Recall that in Bulletproofs the prover wants to convince the verifier that they know a short vector $\mathbf{x}$ satisfying

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0.$$

Let $(\mathbf{M}, \mathbf{x}, \mathbf{y}) = (\mathbf{M}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)})$. The protocol consists of $\ell + 1$ rounds, where in the i-th round the two parties "fold" the relation represented by $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into another represented by $(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)})$ where the dimension of $\mathbf{M}^{(i+1)}$ is half that of $\mathbf{M}^{(i)}$. Correspondingly, the prover folds its witness $\mathbf{x}^{(i)}$ into $\mathbf{x}^{(i+1)}$. After ℓ such folding steps, a constant-size relation $(\mathbf{M}^{(\ell)}, \mathbf{y}^{(\ell)})$ is reached and the prover simply sends the satisfying witness $\mathbf{x}^{(\ell)}$ over to the verifier.

In more detail, for $0 \le i < \ell$, the i-th of the first ℓ rounds of the protocol goes as follows. The parties split $\mathbf{M}^{(i)}$ into two halves as $\mathbf{M}^{(i)} = (\mathbf{M}_L^{(i)}, \mathbf{M}_R^{(i)})$ and the prover splits $\mathbf{x}^{(i)} = (\mathbf{x}_L^{(i)}, \mathbf{x}_R^{(i)})$. The prover sends the cross terms

$$\mathbf{y}_{LR}^{(i)} = \left\langle \mathbf{M}_L^{(i)}, \mathbf{x}_R^{(i)} \right\rangle \bmod q \qquad \text{and} \qquad \mathbf{y}_{RL}^{(i)} = \left\langle \mathbf{M}_R^{(i)}, \mathbf{x}_L^{(i)} \right\rangle \bmod q.$$

The verifier sends a random challenge $r_i \leftarrow\$ S$ sampled from some challenge set $S \subseteq \mathcal{R}^\times$. Both parties fold $(\mathbf{M}^{(i)}, \mathbf{y}^{(i)})$ into

$$(\mathbf{M}^{(i+1)}, \mathbf{y}^{(i+1)}) \coloneqq (\mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1}, \mathbf{y}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{y}_{LR}^{(i)}) \bmod q,$$

and the prover folds $\mathbf{x}$ into $\mathbf{x}^{(i+1)} = \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$. At the ℓ-th (i.e. last) round, the prover simply sends $\mathbf{x}^{(\ell)}$ and the verifier checks that $\mathbf{x}^{(\ell)}$ is short and satisfies $\left\langle \mathbf{M}^{(\ell)}, \mathbf{x}^{(\ell)} \right\rangle = \mathbf{y}^{(\ell)} \bmod q$.

It can been shown [BLNS20, AL21, ACK21, AF22] that the protocol satisfies knowledge soundness, and furthermore it is easy to see that the prover runs in time quasi-linear in the length of the witness. However, a major drawback of this approach is that the verifier computation is also quasi-linear for general linear relations $\mathbf{M}$, and it cannot be preprocessed due to the interactive nature of the scheme.

Polylogarithmic Verifier for Structured Relations. In this work, we observe that, while we cannot hope to reduce the verifier complexity for general matrices $\mathbf{M}$, for suitably structured $\mathbf{M}$ the verification can be sped up to run in time polylogarithmic in the witness length. As an example, the simplest $\mathbf{M}$ with the required structure is a vector consisting of powers of an element $v \in \mathcal{R}_q^\times$, i.e.

$$\mathbf{M} = \left(v\ v^2\ \dots\ v^d\right) \bmod q.$$

Importantly, $\mathbf{M} \cdot \mathbf{x} = p_{\mathbf{x}}(v) \bmod q$ is the vSIS commitment of $\mathbf{x}$ with commitment key v. This observation allows us to prove the knowledge of a preimage of a vSIS commitment via the above protocol with polylogarithmic verifier complexity.

To see why this is the case, it suffices to observe that the verifier complexity is dominated by the computation of the matrix $\mathbf{M}^{(\ell)}$, which is obtained by successive foldings of the starting matrix $\mathbf{M}^{(0)}$. Plugging in the structured relation, we can see that at each iteration the matrix evolves into

$$\begin{aligned}\mathbf{M}^{(i+1)} = \mathbf{M}_L^{(i)} + \mathbf{M}_R^{(i)} \cdot r_i^{-1} &= \left(v\ v^2\ \dots\ v^{d_i/2}\right) + \left(v^{d_i/2+1}\ v^{d_i/2+2}\ \dots\ v^{d_i}\right) \cdot r_i^{-1} \\ &= \left(v\ v^2\ \dots\ v^{d_i/2}\right) \cdot \left(1 + v^{d_i/2} \cdot r_i^{-1}\right) \bmod q\end{aligned}$$

where d_i is the input length at the i-th iteration. Recursing over all iterations, we obtain that the final matrix $\mathbf{M}^{(\ell)}$ is defined as

$$\mathbf{M}^{(\ell)} = \prod_{i=0}^{\ell-1} \left(1 + v^{2^{\ell-i-1}} \cdot r_i^{-1}\right) \bmod q,$$

which can be computed in time polynomial in ℓ, i.e. polylogarithmic in d. In Sections 5 and 6, we extend the above structured folding technique in three ways:

(i) We identify a general class of “foldable” (block-)matrices for which the verifier computation can be made polylogarithmic in the number of columns.

(ii) By modifying the Bulletproofs protocol with techniques borrowed from another folding protocol of Pietrzak [Pie19], we are able to support foldable matrices with an arbitrary (i.e non-power-of-2) number of columns, without breaking the foldable structure.[7]

(iii) Borrowing techniques from [Pie19] again, we can make the verifier computation *also* polylogarithmic in the number of rows of $\mathbf{M}$, for $\mathbf{M}$ with repeating block-bidiagonals, if $\mathbf{y}$ is also foldable.

Approach II: Pre-Processing (Knowledge-Based) Protocols. The second approach for lattice-based arguments for SIS relation is the recent work of [ACL+22], which is based on a recently introduced (knowledge-)k-R-ISIS assumption. In this protocol, the verifier computation can be preprocessed such that the online verification time is polylogarithmic in the relation size. However, a major drawback of this approach is that the public parameters size and the prover complexity are at least quadratic in the relation size. Let us recall (a somewhat simplified version of) the commit-and-prove protocol of [ACL+22] specialised to the case of SIS (i.e. linear) relations. The public parameters consists of

$$\mathbf{A}, \mathbf{t}, \mathbf{v}, \mathbf{h}, \left(\mathbf{A}^{-1}(\mathbf{t} \cdot (g \cdot \bar{g}')(\mathbf{v}))\right)_{g,g' \in \mathcal{G}, g \neq g'}$$

for some set of monomials $\mathcal{G}$, where $\mathbf{A}, \mathbf{t}, \mathbf{v}$ are random over $\mathcal{R}_{q_2}$, $\mathbf{h}$ is a random vector over $\mathcal{R}_{q_1}$, $\bar{g} := 1/g$ denotes the complement of g, and $\mathbf{A}^{-1}(\mathbf{t} \cdot g(\mathbf{v}))$ denotes a short preimage $\mathbf{u}_g$ satisfying $\mathbf{A} \cdot \mathbf{u}_g = \mathbf{t} \cdot g(\mathbf{v}) \bmod q_2$. To prove that

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \text{ (without mod)} \qquad \text{and} \qquad \|\mathbf{x}\| \approx 0,$$

commit to $\mathbf{x}$ as $c_{\mathbf{x}} := \sum_{g \in \mathcal{G}} x_g \cdot g(\mathbf{v}) \bmod q_2$ and derive a short vector $\mathbf{u}$ satisfying

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{t} \cdot \mathbf{h}^{\mathtt{T}} \cdot (\mathbf{M} \cdot \bar{\mathcal{G}}(\mathbf{v}) \cdot c_{\mathbf{x}} - \mathbf{y}) \bmod q_2,$$

where $\bar{\mathcal{G}}(\mathbf{v}) = (\bar{g}(\mathbf{v}))_{g \in \mathcal{G}}$. To compute such a short vector $\mathbf{u}$, the prover needs to perform a linear combination of $|\{ g \cdot \bar{g}' : g, g' \in \mathcal{G}, g \neq g' \}|$ short vectors given in the public parameters. For $\mathcal{G} = \{ V_1, \dots, V_d \}$ chosen in [ACL+22], we have $|\{ g \cdot \bar{g}' : g \neq g' \in \mathcal{G} \}| = O(d^2)$, hence the quasi-quadratic prover complexity.

[7] The usual technique of padding zero columns breaks the foldable structure.

Achieving Quasi-Linear Time Prover. A natural idea is to choose $\mathcal{G} = \{ V, V^2, \ldots, V^d \}$ so $\mathbf{v}$ becomes a single element v. This makes

$$|\{ g \cdot \bar{g}' : g, g' \in \mathcal{G}, g \neq g' \}| = |\{ V^{-i}, V^i \}_{i=1}^{d-1}| = 2d - 2 = O(d).$$

Further exploiting fast multiplication algorithms for Toeplitz matrices allows us to achieve quasi-linear prover time. Notably, with this choice of $\mathcal{G}$ we have

$$c_{\mathbf{x}} = p_{\mathbf{x}}(v) \bmod q_2 \qquad \text{and} \qquad \mathbf{h}^{\mathtt{T}} \cdot \mathbf{M} \cdot (\bar{g}(v))_{g \in \mathcal{G}} = \bar{p}_{\mathbf{M}^{\mathtt{T}} \cdot \mathbf{h}}(v) \bmod q_2,$$

and $\mathbf{h}^{\mathtt{T}} \cdot \mathbf{M} \cdot (\bar{g}(V))_{g \in \mathcal{G}} \cdot c_{\mathbf{x}} - \mathbf{h}^{\mathtt{T}} \cdot \mathbf{y}$ being a polynomial with constant term 0. In the main body, we also show how to support natively modular arithmetic, by borrowing techniques from chainable functional commitments [BCFL22]. We refer the interested reader to Section 7 for more details.

2.3 Applications

To summarise, we have constructed succinct arguments for relations of the form

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \qquad \text{and} \qquad \mathbf{x} \in \{0, 1\}^d$$

with quasi-linear time provers (in both the folding and the preprocessing settings). This gives a efficient and powerful building block for constructing advanced lattice-based cryptographic primitives which require proving relations of the above form. We provide a few examples below.

Lattice-based Verifiable Delay Functions. For the instantiation based on folding protocols, the verifier computation is polylogarithmic if the relation $(\mathbf{M}, \mathbf{y})$ conforms to a certain foldable structure. One example is the sequential-SIS relation proposed in a recent work [LM23], which was used to construct proofs of sequential work (PoSW). In more details, the sequential-SIS relation proposed in their work induces the following linear relation

$$\underbrace{\begin{pmatrix} \mathbf{G} & & & \\ \mathbf{A} & \mathbf{G} & & \\ & \mathbf{A} & \ddots & \\ & & \ddots & \mathbf{G} \\ & & & \mathbf{A} \end{pmatrix}}_{\mathbf{M}} \cdot \mathbf{x} = \underbrace{\begin{pmatrix} \mathbf{z}_0 \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{z}_T \end{pmatrix}}_{\mathbf{y}} \bmod q \qquad \text{and} \qquad \mathbf{x} \in \mathcal{R}_2^{mT}$$

for a uniformly sampled $\mathbf{A}$ and $\mathbf{z}_0$. The PoSW construction in [LM23] falls short of giving verifiable delay functions (VDF) due to the soundness gap in lattice-based folding protocols. By embedding the $\mathbb{Z}_2$ coefficients of $\mathbf{x} \in \mathcal{R}_2^{mT}$ into $\mathbf{x}' \in \{0, 1\}^{mT\varphi(\rho)}$, and plugging in the structured folding protocol constructed in this work, we immediately get the first construction of lattice-based VDFs.

Efficient Lattice-based SNARKs for NP. Recall that our results ultimately rely on the observation that the inner-product of $\mathbf{x}$ and $\mathbf{y}$ is encoded as the constant term of the polynomial $p_{\mathbf{x}} \cdot \bar{p}_{\mathbf{y}}$. In the above, we used this to encode the vectors $\mathbf{x}$ and $\mathbf{y} := \mathbf{h} \circ (\mathbf{x} - \mathbf{1})$ for proving binariness. The same idea can be used to prove general quadratic relations.

Consider the NP-complete rank-1 constraint satisfiability (R1CS) relation which is of the form

$$\exists\, \mathbf{x}, \;\; (\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = \mathbf{C} \cdot \mathbf{x} \bmod q$$

where some entries of $\mathbf{x}$ are publicly known. To prove knowledge of $\mathbf{x}$, the prover computation roughly goes as follows. First, they compute

$$\mathbf{a} := \mathbf{A} \cdot \mathbf{x}, \qquad \mathbf{b} := \mathbf{B} \cdot \mathbf{x}, \qquad \text{and} \qquad \mathbf{c} := \mathbf{C} \cdot \mathbf{x}.$$

They then commit to $(\mathbf{x}, \mathbf{h} \circ \mathbf{a}, \mathbf{b}, \mathbf{c})$ as $(c_{\mathbf{x}}, \bar{c}_{\mathbf{a}}, c_{\mathbf{b}}, c_{\mathbf{c}})$, and prove that the commitments are consistent. Finally, they prove that the constant term in (the polynomial underlying) $\bar{c}_{\mathbf{a}} \cdot c_{\mathbf{b}}$ is identical to $\langle \mathbf{h}, \mathbf{c} \rangle$ for $\mathbf{c}$ committed in $c_{\mathbf{c}}$.

While the above yields a succinct argument for R1CS based on (knowledge-)k-R-ISIS, in Appendix I, we further sketch a folding-based succinct argument for a structured variant of R1CS called succinct-R1CS [BCG+19].

3 Preliminaries

Let $\lambda \in \mathbb{N}$ denote the security parameter, and $\mathsf{poly}(\lambda)$ and $\mathsf{negl}(\lambda)$ the set of all polynomials and negligible functions in λ respectively. Denote the empty string by ϵ. For a function f which may depend on λ and other parameters, we write $O_\lambda(f) \coloneqq f \cdot \mathsf{poly}(\lambda)$ to hide fixed polynomial factors in λ. For matrices $\mathbf{A}$ and $\mathbf{B}$ with the same dimensions, write $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow 3} \coloneqq \begin{pmatrix}\mathbf{A} & & \\ \mathbf{B} & \mathbf{A} & \\ & \mathbf{B} & \mathbf{A} \\ & & \mathbf{B}\end{pmatrix}$ and define $\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n}$ analogously for $n \in \mathbb{N}$. If S is a set and $\mathcal{D}$ is a distribution over S, write $\mathcal{D} \sim S$.

3.1 Cyclotomic Rings

Let $\mathcal{K} = \mathbb{Q}(\zeta)$ be a cyclotomic field, where ζ is a root of unity of order $\rho = \mathsf{poly}(\lambda)$, and $\mathcal{R} = \mathbb{Z}[\zeta]$ be its ring of integers. If ρ is a power of 2 (resp. prime power), $\mathcal{R}$ is called a power-of-2 (resp. prime power) cyclotomic ring. For $q \in \mathbb{N}$, define the quotient ring $\mathcal{R}_q \coloneqq \mathcal{R}/q\mathcal{R}$. We denote by $\mathcal{R}^\times$ and $\mathcal{R}_q^\times$ the sets of units in $\mathcal{R}$ and $\mathcal{R}_q$ respectively. An element $a = \sum_{i=0}^{\rho-1} a_i \cdot \zeta^i \in \mathcal{R}$ (or $\mathcal{R}_q$) is represented by its coefficients $(a_0, \ldots, a_{\rho-1}) \in \mathbb{Z}^\rho$ (or $\mathbb{Z}_q^\rho$). The (infinity) norm of $a \in \mathcal{R}$ (or $\mathcal{R}_q$) is taken as $\|a\| \coloneqq \max_{i=0}^{\rho-1}(|a_i|)$, where in the case of $a_i \in \mathbb{Z}_q$ the balanced representation is taken, i.e. $a_i \in \{ -\lceil q/2 \rceil + 1, \ldots, \lfloor q/2 \rfloor \}$. For a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{R}^n$, $\|\mathbf{a}\| \coloneqq \max_{i=1}^n \|a_i\|$. For a matrix $\mathbf{A} = (A_{i,j})_{i,j}$, the max-norm is taken, i.e. $\|\mathbf{A}\| = \max_{i,j} \|A_{i,j}\|$. The ring expansion factor of $\mathcal{R}$ is defined as $\gamma_\mathcal{R} \coloneqq \max_{a,b \in \mathcal{R}} \|a \cdot b\|/(\|a\| \cdot \|b\|)$. For power-of-2 and prime-power $\mathcal{R}$, it is known that $\gamma_\mathcal{R} \leq 2\varphi(\rho)$, where φ is Euler's totient function. A set $S \subseteq \mathcal{R}$ is said to be subtractive if $a - b \in \mathcal{R}^\times$ for any distinct $a, b \in S$. For a prime-power $\mathcal{R}$, it is known that $S \coloneqq \{ (\zeta^i - 1)/(\zeta - 1) : i \in [\mathsf{rad}(\rho) - 1] \} \subset \mathcal{R}^\times$ is subtractive, where $\mathsf{rad}(\rho)$ denotes the radical. Note that $\|r\| = 1$ for all $r \in S$.

3.2 Lattice Trapdoors

In our constructions based on the (knowledge-)k-R-ISIS assumption, we will make use of lattice trapdoor algorithms. Let η, m, q, β be functions of λ. Let $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$ be PPT algorithms parametrised by (η, m, q, β) with the following syntax and properties [GPV08, MP12, GM18]:

- $(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$ generates a matrix $\mathbf{D} \in \mathcal{R}_q^{\eta \times m}$ and a trapdoor td. The distribution of $\mathbf{D}$ is statistically close to the uniform distribution over $\mathcal{R}_q^{\eta \times m}$.
- $\mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda)$ samples a vector $\mathbf{u} \in \mathcal{R}^m$. For any $(\mathbf{D}, \mathbf{v}) \in \mathcal{R}_q^{\eta \times m} \times \mathcal{R}_q^\eta$ and $\mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda)$ subject to $\mathbf{D}\mathbf{u} = \mathbf{v} \bmod q$, it is guaranteed that $\|\mathbf{u}\| \leq \beta$ with overwhelming probability. Furthermore, the following distributions are statistically close:

$$\left\{ \begin{aligned} &(\mathbf{D}, \mathbf{u}, \mathbf{v}) : \\ &\quad \mathbf{D} \leftarrow\$ \mathcal{R}_q^{\eta \times m} \\ &\quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) \\ &\quad \mathbf{v} = \mathbf{D}\mathbf{u} \bmod q \end{aligned} \right\} \quad \text{and} \quad \left\{ \begin{aligned} &(\mathbf{D}, \mathbf{u}, \mathbf{v}) : \\ &\quad \mathbf{D} \leftarrow\$ \mathcal{R}_q^{\eta \times m} \\ &\quad \mathbf{v} \leftarrow\$ \mathcal{R}_q^{\eta} \\ &\quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) : \mathbf{D}\mathbf{u} = \mathbf{v} \bmod q \end{aligned} \right\}$$

- $\mathbf{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{v})$ inputs a target vector $\mathbf{v} \in \mathcal{R}_q^\eta$ and samples a vector $\mathbf{u} \in \mathcal{R}^m$. For $(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$, it is guaranteed that $\mathbf{D} \cdot \mathbf{u} = \mathbf{v} \bmod q$ and $\|\mathbf{u}\| \leq \beta$ with overwhelming probability. Furthermore, for any $\mathbf{v} \in \mathcal{R}_q^\eta$, the following distributions are statistically close:

$$\left\{ \begin{aligned} &(\mathbf{D}, \mathbf{u}) : \\ &\quad (\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda) \\ &\quad \mathbf{u} \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{v}) \end{aligned} \right\} \quad \text{and} \quad \left\{ \begin{aligned} &(\mathbf{D}, \mathbf{u}) : \\ &\quad (\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda) \\ &\quad \mathbf{u} \leftarrow \mathsf{SampD}(1^\lambda) : \mathbf{D}\mathbf{u} = \mathbf{v} \bmod q \end{aligned} \right\}$$

3.3 Presumed Hard Problems

The Short Integer Solution (SIS) problem was introduced in the seminal work of Ajtai [Ajt96]. It asks to find a short vector in the kernel of a given random matrix modulo q. In this work, we consider the generalisation of SIS over $\mathcal{R}$ and the k-R-ISIS problem introduced in [ACL+22].

Definition 2 **(R-SIS Assumption).** *Let $m, q, \beta^* \in \mathbb{N}$ depend on λ. The Ring-SIS (or R-SIS) problem, denoted R-$\mathsf{SIS}_{\mathcal{R},m,q,\beta^*}$, is: Given $\mathbf{h} \leftarrow\$ \mathcal{R}_q^m$, find $\mathbf{u} \in \mathcal{R}^m$ such that $0 < \|\mathbf{u}\| \leq \beta^*$ and $\mathbf{h}^\mathsf{T}\mathbf{u} \equiv \mathbf{0} \bmod q$. We write $\mathsf{Adv}^{\text{r-sis}}_{\mathcal{R},m,q,\beta^*,\mathcal{A}}(\lambda)$ for the advantage of any algorithm $\mathcal{A}$ in solving R-$\mathsf{SIS}_{\mathcal{R},\eta,m,q,\beta^*}$. The R-$\mathsf{SIS}_{\mathcal{R},\eta,m,q,\beta^*}$ assumption states that, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\text{r-sis}}_{\mathcal{R},m,q,\beta^*,\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$.*

We state a streamlined version of the (knowledge) k-R-ISIS[8] assumptions defined in [ACL+22] with two main changes: (i) To improve readability, our definitions of the assumptions do not impose admissibility constraints on parameters. Instead, we mention these admissibility parameters separately outside of the definitions. (ii) We assume that all preimages $\mathbf{u}_g$ given to the adversary are sampled from the same distribution conditioned on different constraints. The original definitions [ACL+22] are more general in that they allow a different distribution per constraint.

Definition 3 **(k-R-ISIS Assumptions).** *Let $\eta, m, q, \beta, \beta^* \in \mathbb{N}$, $\mathcal{G} \cup \{ g^* \}$ be a set of w-variate Laurent monomials, $\mathcal{T} \sim \mathcal{R}_q^\eta$, and $\mathcal{D} \sim \mathcal{R}^m$, all dependent on λ. Write $\mathsf{pp} := (\mathcal{R}, \eta, m, w, q, \beta, \beta^*, \mathcal{G}, g^*, \mathcal{D}, \mathcal{T})$. The k-R-$\mathsf{ISIS}_\mathsf{pp}$ assumption states that, for any PPT adversary $\mathcal{A}$, $\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$, where $\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}(\lambda) :=$*

$$\mathsf{Pr}\left[\begin{array}{l} \mathbf{D} \cdot \mathbf{u}_{g^*} \equiv \mathbf{t} \cdot s^* \cdot g^*(\mathbf{v}) \bmod q \\ \wedge\ 0 < \|(\mathbf{u}_{g^*}, s^*)\| \leq \beta^* \end{array}\left|\begin{array}{l} \mathbf{D} \leftarrow\$ \mathcal{R}_q^{\eta \times m};\ \mathbf{t} \leftarrow\$ \mathcal{T};\ \mathbf{v} \leftarrow\$ (\mathcal{R}_q^\times)^w \\ \mathbf{u}_g \leftarrow\$ \mathcal{D} : \mathbf{D} \cdot \mathbf{u}_g = \mathbf{t} \cdot g(\mathbf{v}) \bmod q,\ \forall\ g \in \mathcal{G} \\ (s^*, \mathbf{u}_{g^*}) \leftarrow \mathcal{A}\left(\mathbf{D}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}}\right) \end{array}\right.\right].$$

Individual parameters are omitted when they are clear from the context.

Definition 4 **(Knowledge k-R-ISIS Assumptions).** *Let $\eta, m, q, \alpha^*, \beta, \beta^* \in \mathbb{N}$, $\mathcal{G}$ be a set of w-variate Laurent monomials, $\mathcal{T} \sim \mathcal{R}_q^\eta$, and $\mathcal{D} \sim \mathcal{R}^m$, all dependent on λ. Let $\mathcal{Z}$ be a PPT auxiliary input generator. Write $\mathsf{pp} := (\mathcal{R}, \eta, m, w, q, \alpha^*, \beta, \beta^*, \mathcal{G}, \mathcal{D}, \mathcal{T}, \mathcal{Z})$. The knowledge k-R-$\mathsf{ISIS}_\mathsf{pp}$ assumption states that for any PPT adversary $\mathcal{A}$ there exists a PPT extractor $\mathcal{E}_\mathcal{A}$ such that $\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda)$, where $\mathsf{Adv}^{\text{k-r-isis}}_{\mathsf{pp},\mathcal{A}}(\lambda) :=$*

$$\mathsf{Pr}\left[\begin{array}{l} \quad \mathbf{D} \cdot \mathbf{u} \equiv \mathbf{t} \cdot c \bmod q \\ \wedge\ 0 < \|\mathbf{u}\| \leq \beta^* \\ \wedge\ \neg \left(\begin{array}{l} c \equiv \sum_{g \in \mathcal{G}} x_g \cdot g(\mathbf{v}) \bmod q \\ \wedge\ \left\|(x_g)_{g \in \mathcal{G}}\right\| \leq \alpha^* \end{array}\right) \end{array}\left|\begin{array}{l} \mathbf{D} \leftarrow\$ \mathcal{R}_q^{\eta \times m};\ \mathbf{t} \leftarrow\$ \mathcal{T};\ \mathbf{v} \leftarrow\$ (\mathcal{R}_q^\times)^w \\ \mathbf{u}_g \leftarrow\$ \mathcal{D} : \mathbf{D} \cdot \mathbf{u}_g = \mathbf{t} \cdot g(\mathbf{v}) \bmod q,\ \forall\ g \in \mathcal{G} \\ \mathsf{pp} := (\mathbf{D}, \mathbf{t}, \mathbf{v}, \{ \mathbf{u}_g \}_{g \in \mathcal{G}});\ \mathsf{aux} \leftarrow \mathcal{Z}(\mathsf{pp}) \\ \left((c, \mathbf{u}), (x_g)_{g \in \mathcal{G}}\right) \leftarrow (\mathcal{A} \| \mathcal{E}_\mathcal{A})\,(\mathsf{pp}, \mathsf{aux}) \end{array}\right.\right]$$

where $(\mathcal{A} \| \mathcal{E}_\mathcal{A})$ means that $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ are run on the same input including the randomness, and $(c, \mathbf{u})$ and $(x_g)_{g \in \mathcal{G}}$ are the outputs of $\mathcal{A}$ and $\mathcal{E}_\mathcal{A}$ respectively. Individual parameters are omitted when they are clear from the context.

For both assumptions to be meaningful, we always consider $m > \eta$.[9] For non-triviality, we want $g^* \notin \mathcal{G}$ and $\mathbf{t} \neq \mathbf{0}$ with overwhelming probability. To avoid complications of giving the adversary short vectors in the kernel of $\mathbf{D}$, we do not consider the case where $\mathcal{G}$ is a multiset – all monomials in $\mathcal{G}$ are distinct.[10] To avoid SIS attacks in the image space, we want $1/|\mathcal{R}_q^\times| = \mathsf{negl}(\lambda)$.

For the knowledge assumption to be plausible, we would like that $\alpha^* \geq \beta^*$, and for $\mathbf{t} \leftarrow\$ \mathcal{T}$, $1/|\langle \mathbf{t} \rangle| = \mathsf{negl}(\lambda)$ and $|\langle \mathbf{t} \rangle| / |\mathcal{R}_q^\eta| = \mathsf{negl}(\lambda)$ with overwhelming probability. Furthermore, to avoid easy instances of ideal-SVP (relevant when $\eta = 1$), we would like the problem of finding short elements in $\{ s \in \mathcal{R} : \mathbf{t} \cdot s = \mathbf{0} \bmod q \}$ to be hard.

3.4 Argument Systems

We recall the definition of argument systems which allow a prover to convince a verifier that a relation is satisfiable. Formally, we define a (family of) relation(s) $\mathit{\Psi} (= (\mathit{\Psi}_\lambda)_{\lambda \in \mathbb{N}})$ to be polynomial-time-decidable

[8] In [ACL+22], the assumptions over modules were separately called (knowledge-)k-M-ISIS.

[9] In [ACL+22], m is considered to be large enough so that the leftover hash lemma holds. However, smaller m only makes the problems harder.

[10] In [ACL+22, Definition 22], monomials in $\mathcal{G}$ and g^* are further required to be independent of $\mathcal{R}$. We discuss in Section 4.2 why we believe that this restriction can be lifted.

triples of the form $(\mathsf{pp}, \mathsf{stmt}, \mathsf{wit})$, corresponding to the public parameters of the argument system, the statement, and the witness respectively. We consider a statement $\mathsf{stmt} = (\mathsf{stmt}_{\mathtt{off}}, \mathsf{stmt}_{\mathsf{on}})$ to consist an offline part $\mathsf{stmt}_{\mathtt{off}}$ which is potentially preprocessable and an online part $\mathsf{stmt}_{\mathsf{on}}$. For any fixed public parameters pp, we define the (sub-)relation $\Psi_{\mathsf{pp}} := \{ (\mathsf{stmt}, \mathsf{wit}) : (\mathsf{pp}, \mathsf{stmt}, \mathsf{wit}) \in \Psi \}$ and the corresponding language $\mathcal{L}_{\mathsf{pp}} := \{ \mathsf{stmt} : \exists\ \mathsf{wit},\ (\mathsf{stmt}, \mathsf{wit}) \in \Psi_{\mathsf{pp}} \}$. We focus on relations where the public parameters pp can be efficiently generated, and denote such a generator by Gen_{Ψ}. We suppress pp when it is the empty string.

Definition 5 (Arguments). *A (preprocessing) argument system consists of PPT algorithms* $(\mathsf{Setup}, \mathsf{PreVerify})$ *and PPT interactive algorithms* $(\mathsf{Prove}, \mathsf{Verify})$ *with the following syntax:*

- $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$*: Input some public parameters* pp *and generate a common reference string* crs*.*
- $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$*: Preprocess the statement* $\mathsf{stmt}_{\mathtt{off}}$*. Systems not supporting preprocessing are captured by having a trivial preverification, i.e.* $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} = (\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$*.*
- $(\mathsf{tx}, b) \leftarrow \langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle$*: An interactive protocol where the prover tries to convince the verifier about the statement* stmt*. The protocol produces a transcript* tx *and ends with the verifier outputting a bit* $b \in \{0, 1\}$*. The transcript* tx *is suppressed from the output when it is not needed. In the case where the protocol is non-interactive, i.e. the prover sends a single message, then we split the protocol into two PPT algorithms* $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$ *and* $b \leftarrow \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}, \pi)$*, where* π *is referred to as a proof.*

Definition 6 (Completeness). *An argument system* Π *is said to be complete for* Ψ *if for all adversaries* $\mathcal{A}$

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathsf{Gen}_{\Psi}(1^\lambda);\ \ \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp}) \\ (\mathsf{stmt}, \mathsf{wit}) \in \Psi_{\mathsf{pp}} & (\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs}) \\ \wedge\ \ b = 0 & \mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}}) \\ & b \leftarrow \langle \mathcal{P}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 7 (Special Soundness). *An argument system* Π *is said to be public-coin if each message sent by* $\mathcal{V}$ *is sampled from a public distribution independent of the messages sent by* Prove*. A transcript* tx *is said to be accepting for* $(\mathsf{pp}, \mathsf{stmt})$ *if* $(\mathsf{tx}, 1)$ *is in the output space of* $\langle \mathcal{P}, \mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle$ *where* $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \in \mathsf{PreVerify}(\mathsf{Setup}(1^\lambda, \mathsf{pp}), \mathsf{stmt})$*. Suppose* $\mathcal{V}$ *sends* ℓ *messages throughout the execution of* $\langle \mathcal{P}, \mathcal{V} \rangle$*. A tree* T *is said to be a* $(k_1, \ldots, k_\ell)$*-tree of accepting transcripts for* $(\mathsf{pp}, \mathsf{stmt})$ *if it is of (node-)depth* $(\ell + 1)$*, each node is labelled by a prover message, each depth-*i *node has exactly* k_i *children each connected by an edge labelled by a distinct verifier message, and the labels on each root-to-leaf path give an accepting transcript for* $(\mathsf{pp}, \mathsf{stmt})$*. The argument system* Π *is said to be* $(k_1, \ldots, k_\ell)$*-special-sound for* Ψ *if there exists a polynomial-time extractor* $\mathcal{E}$ *which on input a* $(k_1, \ldots, k_\ell)$*-tree of accepting transcripts for* $(\mathsf{pp}, \mathsf{stmt})$ *outputs* wit^* *such that* $(\mathsf{stmt}, \mathsf{wit}^*) \in \Psi_{\mathsf{pp}}$*.*

Definition 8 (Knowledge Soundness). *Let* $\kappa = \kappa(\lambda)$ *denote the knowledge error. An argument system* Π *is said to be* κ*-knowledge-sound for* Ψ *if for all PPT* $\mathcal{P}^*$ *there exists an expected polynomial-time extractor* $\mathcal{E}_{\mathcal{P}^*}$ *such that for all PPT adversaries* $\mathcal{A}$ *the following is at most* κ*:*

$$\Pr\left[\begin{array}{l|l} & \mathsf{pp} \leftarrow \mathsf{Gen}_{\Psi}(1^\lambda);\ \ \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp}) \\ (\mathsf{stmt}, \mathsf{wit}^*) \notin \Psi_{\mathsf{pp}} & (\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs}) \\ \wedge\ \ b = 1 & \mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}}) \\ & (\mathsf{wit}^*, b) \leftarrow \langle (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathcal{V}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle \end{array}\right]$$

The argument system Π *is said to be knowledge-sound for* Ψ *if it is* κ*-knowledge-sound for* Ψ *for some* $\kappa = \mathsf{negl}(\lambda)$*.*

It is known that a parallel-repetition of a $(k_1, \ldots, k_\ell)$-special-sound protocol yields a knowledge-sound protocol [AF22].

Note that it is common for lattice-based argument systems to have a "soundness gap": They are complete for a relation Ψ, but special- or knowledge-sound for a relaxed relation $\Psi' \supseteq \Psi$, i.e. the extracted witness wit^* for $(\mathsf{pp}, \mathsf{stmt})$ may not satisfy $(\mathsf{stmt}, \mathsf{wit}^*) \in \Psi_{\mathsf{pp}}$ but only $(\mathsf{stmt}, \mathsf{wit}^*) \in \Psi'_{\mathsf{pp}}$.

Definition 9 (Succinctness). *An argument system Π is said to have succinct proofs (resp. succinct verifier) for Ψ if for any $\mathsf{pp} \in \mathsf{Gen}_\Psi(1^\lambda)$, $\mathsf{crs} \in \mathsf{Setup}(1^\lambda, \mathsf{pp})$, $(\mathsf{stmt}, \mathsf{wit}) \in \Psi_{\mathsf{pp}}$, $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \in \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$, the communication complexity of $\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle$ (resp. computation complexity of $\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}})$) is $\mathsf{polylog}(|\mathsf{stmt}| + |\mathsf{wit}|) \cdot \mathsf{poly}(\lambda)$ where the $\mathsf{poly}(\lambda)$ factor is independent of $|\mathsf{stmt}|$ and $|\mathsf{wit}|$.*

Argument systems which are succinct, non-interactive, and knowledge-sound are known as succinct non-interactive arguments of knowledge (SNARK). Arguments whose soundness holds even against adversaries given the randomness of Setup are said to have transparent setups.

4 Vanishing Short Integer Solutions

In this section, we formalise the vanishing-SIS problems and assumptions, and discuss their relations with existing problems and assumptions. We also discuss the properties of the collision-resistant hash functions obtained immediately from the vanishing-SIS assumptions.

4.1 Definition

Definition 10 (Vanishing-SIS). *Let $n, d, w, q, \beta \in \mathbb{N}$ and $\mathcal{G}$, a set of w-variate (Laurent) monomials of individual degree at most d, be functions of λ. The $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ problem is the following: Given a set $V = \{ \mathbf{v}_i \}_{i=1}^n \in (\mathcal{R}_q^\times)^w$ of n uniformly random points in $(\mathcal{R}_q^\times)^w$, find a non-zero polynomial $p \in \mathcal{R}[X_1, \ldots, X_w]$ with monomial support[11] over $\mathcal{G}$ such that*

$$\forall i \in [n], \qquad p(\mathbf{v}_i) = 0 \bmod q \qquad \textit{and} \qquad \|p\| \leq \beta$$

where $\|p\|$ is the maximum of the norm of the coefficients of p. The $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ assumption states that, for any PPT adversary $\mathcal{A}$, the probability of $\mathcal{A}$ solving a uniformly random instance of $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},n,q,\beta}$ is negligible in λ. Individual parameters are omitted from the subscript when they are clear from the context. If $\mathcal{G}$ is the set of all w-variate (Laurent) monomials of individual degree at most d, we denote the problem by $\mathsf{vSIS}_{\mathcal{R},d,w,n,q,\beta}$. To emphasise certain parameters, e.g. $n = n^*$ and $w = w^*$, we sometimes write $\mathsf{vSIS}_{(n,w)=(n^*,w^*)}$.

Another way to phrase the problem, borrowing terminologies from algebraic geometry, is that it asks to find an element of bounded norm and degree in the ideal $\mathcal{I}(V)$ of polynomials vanishing at the set of points V. Clearly, the subset of bounded-degree polynomials in $\mathcal{I}(V)$ forms a (module) lattice. Therefore a vanishing-SIS problem can also be seen as an average-case approximate shortest vector problem (SVP) over such lattices.[12]

The connection of the vanishing-SIS problem to the standard SIS problem stems from the following simple observation: If we interpret the coefficients of a solution p as a vector $\mathbf{p}$, and write the relation in matrix form, we obtain

$$\begin{pmatrix} 1 & v_{1,1} & \cdots & v_{1,w} & \cdots & \prod_{j=1}^w v_{1,j}^{e_j} & \cdots & \prod_{j=1}^w v_{1,j}^d \\ 1 & v_{2,1} & \cdots & v_{2,w} & \cdots & \prod_{j=1}^w v_{2,j}^{e_j} & \cdots & \prod_{j=1}^w v_{2,j}^d \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & v_{n,1} & \cdots & v_{n,w} & \cdots & \prod_{j=1}^w v_{n,j}^{e_j} & \cdots & \prod_{j=1}^w v_{n,j}^d \end{pmatrix} \cdot \mathbf{p} = \mathbf{0} \bmod q \qquad \text{and} \qquad \|\mathbf{p}\| \leq \beta,$$

a SIS relation with respect to a (Vandermonde-like) structured matrix.

Note that since $v_{i,j} \in \mathcal{R}_q^\times$ for all i and j, it is not important for p to be a polynomial with only non-negative powers. Laurent polynomials can be captured scaling the each i-th row of the matrix by $\prod_{j=1}^w v_{i,j}^{-e_j}$ for any desired powers $(e_1, \ldots, e_w) \in \mathbb{Z}^w$. In fact, using the matrix formulation, the scaling factors for each row could be different.

It is easy to observe that the vanishing-SIS assumption is implied by the k-R-ISIS assumption with related parameters. In Appendix A, we discuss this implication in more detail, show that the converse holds conditioned on a related knowledge-k-R-ISIS assumption, and explore the connections of vanishing-SIS to more established assumptions, i.e. NTRU and RingLWE.

[11] e.g. the monomial support of $3X_1X_2 + 2X_2^2 + 1$ is $\{ X_1X_2, X_2^2, 1 \}$

[12] Interestingly, after restricting to a bounded-degree subset, we no longer have an ideal. Therefore this approximate SVP problem is not over ideal-lattices.

4.2 On Choice of Parameters

On the modulus q. Note that, for some (preferable) parameters settings, it is important for $q > d$ for the vSIS assumption to be plausible. Indeed, for example, if q is prime and is such that $q\mathcal{R}$ splits completely into $\varphi(\rho)$ ideals, then we have $v^{q-1} - 1 = 0 \bmod q$ for any $v \in \mathcal{R}$. This gives rise to trivial solutions, e.g. $p(X) = X^{q-1} - 1$, to the vSIS problem.

On the space of V. It is also important for the set of points V to be chosen over $\mathcal{R}_q^\times$ instead of $\mathcal{R}_q$. For example, consider a power-of-2 $\mathcal{R}$ and $q = 2^\ell$. The ideal $q\mathcal{R}$ splits into $q\mathcal{R} = \mathcal{I}^{\ell \cdot \varphi(\rho)}$ for some ideal $\mathcal{I}$ of (algebraic) norm $\mathcal{N}(\mathcal{I}) = 2$. Therefore, with probability 1/2, a random element $v \leftarrow\$ \mathcal{R}_q$ satisfies $v = 0 \bmod \mathcal{I}$ and hence $v^{\ell \cdot \varphi(\rho)} = 0 \bmod q$. This means that $p(X) = X^{\ell \cdot \phi(\rho)}$ is a solution to any vanishing-SIS over $\mathcal{R}_q$ if instances were sampled from $\mathcal{R}_q$.[13]

On the cardinality $|\mathcal{R}_q^\times|$. It is crucial that the cardinality $|\mathcal{R}_q^\times|$ is large enough so that $1/|\mathcal{R}_q^\times| = \mathsf{negl}(\lambda)$. Suppose not, then there might exist small $e \in \mathbb{N}$ such that $\{ v, v^2, \ldots, v^e \}$ contains a short element modulo q. Note that the set of elements in $\mathcal{R}$ of norm at most β has cardinality $(2\beta + 1)^{\varphi(\rho)}$. If we heuristically model the multiplication-by-v map $a \mapsto a \cdot v \bmod q$ as a random permutation for $v \leftarrow\$ \mathcal{R}_q^\times$, and if $\mathcal{R}_q^\times$ is large enough, we have some confidence to believe that small powers of v modulo q will not be short.

In general, it appears that $|\mathcal{R}_q^\times|$ is usually quite close to $q^{\varphi(\rho)}$. We calculate this cardinality for some specific choices of q and $\mathcal{R}$. For $q = 2^\ell$ and ρ being a power of 2, we have $|\mathcal{R}_q^\times| = q^{\varphi(\rho)}/2$. For arbitrary $\mathcal{R}$ and prime $q = 1 \bmod \varphi(\rho)$, we have $|\mathcal{R}_q^\times| = (q-1)^{\varphi(\rho)}$. In either case, if $\beta \leq q/4$, we have $\Pr\left[\|x\| \leq \beta \;\middle|\; x \leftarrow\$ \mathcal{R}_q^\times\right] < 2^{-\varphi(\rho)}$ which is negligible in ρ.

4.3 A Family of Hash Functions with Short Keys

Similar to the standard SIS-based hash function, the vanishing-SIS assumption immediately implies the existence of a collision-resistant hash function, except that in this case the keys are very small, and could potentially be *logarithmic* in the message size. Furthermore, the hash function satisfies many desirable properties, such as (approximate) ring homomorphism.

In more detail, for any set of points $V = \{ \mathbf{v}_i \}_{i=1}^n \subseteq ((\mathcal{R}_q^\times)^w)^n$, define

$$\mathcal{H}_V : \mathcal{R}_\beta^{(d+1)w} \to \mathcal{R}_q^n, \;\; \mathcal{H}_V(p) = (p(\mathbf{v}_1), \ldots, p(\mathbf{v}_n)) \bmod q$$

where an input $\mathbf{p} \in \mathcal{R}_\beta^{(d+1)w}$ is interpreted, for example, as a polynomial $p \in \mathcal{R}_\beta[X_1, \ldots, X_w]$ of individual degree at most d.

It is easy to show that this function is collision resistant by observing that $\mathcal{H}_V(p) = \mathcal{H}_V(p')$ implies

$$\forall i \in [n], \;\; (p - p')(\mathbf{v}_i) = 0 \bmod q \qquad \text{and} \qquad \|p - p'\| \leq \beta,$$

i.e. $p - p'$ is a solution to the vSIS instance V.

Observe that each hash function can be described by a key of size $n \cdot w \log q$ bits, and can hash messages of length $(d+1) \cdot w \cdot \log \beta$ bits to $n \cdot \log q$ bits, where n and w could be as small as 1. As discussed in Section 4.1, for the vSIS assumption to be plausible for the case where q fully splits, which is desirable for efficiency, it is necessary that $q > d$. For $q = O(d)$ and $n, w, \beta = \mathsf{poly}(\lambda)$, the key size and the message length are $O_\lambda(\log d)$ and $O_\lambda(d)$ respectively.

Similar to the standard SIS-based hash function, $\mathcal{H}_V$ is almost linearly homomorphic in the sense that

$$\mathcal{H}_V(p) + \mathcal{H}_V(p') = \mathcal{H}_V(p + p') \bmod q \qquad \text{and} \qquad \|p + p'\| \leq \|p\| + \|p'\|.$$

Different from the standard SIS-based hash function, however, is that $\mathcal{H}_V$ is also almost multiplicatively homomorphic in the sense that

$$\mathcal{H}_V(p) \cdot \mathcal{H}_V(p') = \mathcal{H}_V(p \cdot p') \bmod q \qquad \text{and} \qquad \|p \cdot p'\| \leq (d+1)^w \cdot \|p\| \cdot \|p'\| \cdot \gamma_{\mathcal{R}},$$

[13] This is the reason why $\mathcal{G}$ was restricted to be independent of $\mathcal{R}$ in the definition of "k-R-ISIS-admissible" parameters in [ACL+22, Definition 22]. However, since [ACL+22, Definition 23] also restricts $\mathbf{v} \in (\mathcal{R}_q^\times)^w$, the restriction on $\mathcal{G}$ appears to be redundant.

with multiplications taken over $\mathcal{R}_q$ and $\mathcal{R}[\mathbf{X}]$ respectively.

For our purpose of construction linear-time succinct arguments, the univariate case (i.e. $w = 1$) is the most interesting due to the exponential dependency of various parameters on w. Moreover, we notice that if $p_0(X)$ and $p_1(X)$ encode the vectors $\mathbf{p}_0$ and $\mathbf{p}_1$ respectively as their coefficients, then the product polynomial $p(X) \cdot p(X^{-1})$ has norm at most $\|\mathbf{p}_0\| \cdot \|\mathbf{p}_1\| \cdot \gamma_{\mathcal{R}}$, and its constant term encodes the inner product $\langle \mathbf{p}_0, \mathbf{p}_1 \rangle$.

5 Foldable Structures

We define a family of monomials, polynomials, vectors, and matrices that exhibit "foldable" structures.

Definition 11 (Foldable Polynomials). *Let $\ell \geq 0$, $k_\ell > 0$, and $k_{\ell-1}, \ldots, k_0 \geq 0$ be integers. A sequence of (monic multivariate Laurent) monomials $\mathbf{m}$*[14] *of length $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ (where k_i are not necessarily binary) is said to be $(k_0, k_1, \ldots, k_\ell)$-foldable if the following properties are satisfied:*

- *$\mathbf{m} = \mathbf{m}_0$ can be generated from a "seed" $\mathbf{m}_\ell$ and a "generator" $(\ell_i, \mathbf{c}_i, r_i)_{i=0}^{\ell-1}$, where $\mathbf{m}_\ell$ is a sequence of monomials of length k_ℓ, $\mathbf{c}_i$ is a sequence of monomials of length k_i, and ℓ_i, r_i are monomials, in a recursive fashion:*[15]
$$\forall i \in [\ell],\ \mathbf{m}_{i-1}^{\mathtt{T}} := \left(\ell_{i-1} \cdot \mathbf{m}_i^{\mathtt{T}} \,\|\, \mathbf{c}_{i-1}^{\mathtt{T}} \,\|\, r_{i-1} \cdot \mathbf{m}_i^{\mathtt{T}}\right).$$
- *For all $i \in \{0, \ldots, \ell\}$, $\mathbf{m}_i$ consists of distinct monomials.*

We say that $\mathbf{m}$ is foldable if it is $(k_0, k_1, \ldots, k_\ell)$-foldable for some $(k_0, k_1, \ldots, k_\ell)$. A foldable polynomial is a polynomial whose supporting monomials can be arranged into a foldable sequence of monomials.

Note that any sequence of monomials $\mathbf{m}$ of length n is trivially $(0, \ldots, 0, n)$-foldable. However, we are most interested in sequences which are $(k_0, k_1, \ldots, k_\ell)$-foldable for small constants k_i, e.g. $k_i \in \{0, 1, 2\}$, for all $i \in \{0, \ldots, \ell\}$. Below, we state some elementary properties satisfied by foldable monomials.

Lemma 1. *Let $\mathbf{m}$ of length n be $(k_0, \ldots, k_\ell)$-foldable. Let $k^* := \max_{i=0}^{\ell} k_i$. It holds that $\ell \leq \log n < \ell + \log 2 \cdot k^*$.*

The proof of Lemma 1 is deferred to Appendix B. The following properties follow immediately from the definition and are stated without proof.

Lemma 2 (Chaining/Decomposition). *If $\mathbf{m}$ is foldable with seed and generator $(\mathbf{m}', \mathbf{g}')$ and $\mathbf{m}'$ is foldable with seed and generator $(\mathbf{m}'', \mathbf{g}'')$, then $\mathbf{m}$ is foldable with seed and generator $(\mathbf{m}'', \mathbf{g}'' \| \mathbf{g}')$.*

Lemma 3 (Closure under Hadamard Product). *If $\mathbf{m}$ and $\mathbf{m}'$ are both $(k_0, k_1, \ldots, k_\ell)$-foldable with, where $\mathbf{m}$ and $\mathbf{m}'$ are supported by disjoint sets of variables and have seeds and generators*
$$(\mathbf{s}, (\ell_i, \mathbf{c}_i, r_i)_{i=0}^{\ell-1}) \qquad \text{and} \qquad (\mathbf{s}', (\ell_i', \mathbf{c}_i', r_i')_{i=0}^{\ell-1})$$
respectively, then the Hadamard product $\mathbf{m} \circ \mathbf{m}'$ is also $(k_0, k_1, \ldots, k_\ell)$-foldable with seed and generator
$$(\mathbf{s} \circ \mathbf{s}', (\ell_i \cdot \ell_i', \mathbf{c}_i \circ \mathbf{c}_i', r_i \cdot r_i')_{i=0}^{\ell-1}).$$

Next, we extend the definition of foldable monomials and polynomials to that of (block-)foldable vectors and matrices. We then give examples of such objects. The proofs are elementary and are deferred to Appendix B.

Definition 12 (Foldable Vectors and Matrices). *A vector $\mathbf{a} = (a_1, \ldots, a_n)$ is said to be $(k_0, k_1, \ldots, k_\ell)$-foldable if there exists a $(k_0, k_1, \ldots, k_\ell)$-foldable sequence of monomials $\mathbf{m} = (m_1, \ldots, m_n)$ and a point $\mathbf{v} \in (\mathcal{R}^\times)^k$ such that $a_i = m_i(\mathbf{v})$ for all $i \in [n]$, i.e. the i-th entry of $\mathbf{a}$ is obtained by evaluating the i-th monomial in $\mathbf{m}$ at the point $\mathbf{v}$. The point $\mathbf{v}$ is said to be the evaluation point of $\mathbf{a}$. A matrix is said to be foldable if every row of it is foldable with a common evaluation point $\mathbf{v}$. A block-matrix $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_n)$ where $\mathsf{ncol}(\mathbf{A}_i) = w$ for all $i \in [n]$ is said to be block-foldable with block-size w if, for all (i, j), the vector formed by taking the (i, j)-th entry of each of $(\mathbf{A}_1, \ldots, \mathbf{A}_n)$ is foldable.*

[14] That is, each entry of $\mathbf{m}$ is a monic multivariate Laurent monomial.

[15] In the recursive expression, "$\cdot$" denotes the symbolic multiplication of monomials. For example, $X \cdot (X^2, X^3) = (X^3, X^4)$.

Lemma 4 (Power Sequence). *For any $n \in \mathbb{N}$, express n uniquely*[16] *as $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ with $k_i \in \{1,2\}$ for $i \in \{0,\dots,\ell\}$. Then for any $v \in \mathcal{R}$, the vector $\mathbf{v}^{\mathrm{T}} = (v, v^2, \dots, v^n)$. is $(k_0, k_1, \dots, k_\ell)$-foldable. Generalising, for $w \in \mathbb{N}$, the vector $\mathbf{v}^{\mathrm{T}} = (v, v^2, \dots, v^{wn})$ is $(k_0, k_1, \dots, k_\ell)$-block-foldable with block-size w.*

Lemma 5 (Balanced Power Sequence). *For any $n \in \mathbb{N}$, express n uniquely as $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$ with $k_\ell = 1$ and $k_i \in \{0,1\}$ for all $i \in \{0,\dots,\ell-1\}$. Then for any $v \in \mathcal{R}$, the following vector is $(0, k_0, k_1, \dots, k_\ell)$-foldable:*

$$\mathbf{v}^{\mathrm{T}} = (v^{-n}, \dots, v^{-2}, v^{-1}, v, v^2, \dots, v^n).$$

Lemma 6 (Compression Vector). *For any integers $\ell \geq 0$, $k_\ell > 0$ and $k_0, \dots, k_{\ell-1} \geq 0$, let X_{i,j_i} be independent variables for $i \in \{0,\dots,\ell\}$ and $j_i \in \{0,\dots,k_i\}$. The seed and generator*

$$((X_{\ell,1}, \dots, X_{\ell,k_\ell}), (1, (X_{i,1}, \dots, X_{i,k_i}), X_{i,0})_{i=0}^{\ell-1})$$

generate a $(k_0, k_1, \dots, k_\ell)$-foldable sequence of monomials $\mathbf{m}$. Furthermore, let $\mathbf{x} = (x_{i,j})_{i=0,j=1}^{\ell,k_i}$ be a vector over $\mathcal{R}$ with $\|\mathbf{x}\| \leq \alpha$. Let $\mathbf{h} := \mathbf{m}(\mathbf{x})$ be the foldable vector obtained by evaluating $\mathbf{m}$ at $\mathbf{x}$. It holds that $\|\mathbf{h}\| \leq \alpha^{\ell+1} \cdot \gamma_{\mathcal{R}}^{\ell}$.

6 Folding Protocols

We state two folding protocols $\Pi_0^{\mathtt{fold}}$ and $\Pi_1^{\mathtt{fold}}$ for bounded-norm satisfiability of (structured) linear relations which respect the foldable structures (Section 5) of the matrices and vectors defining the relations. Both protocols have trivial (hence transparent) setup and trivial pre-verification, i.e. $\mathsf{crs} = \Pi_b^{\mathtt{fold}}.\mathsf{Setup}(1^\lambda, \mathsf{pp}) = (1^\lambda, \mathsf{pp})$ and $\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} = \Pi_b^{\mathtt{fold}}.\mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}}) = (\mathsf{crs}, \mathsf{stmt})$. We detail below the prove-verify protocols

$$\Pi_b^{\mathtt{fold}}.\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}}) \rangle.$$

6.1 Type-0 Linear Relations

Define the relation $\Psi_0^{\mathtt{fold}} = \Psi_0^{\mathtt{fold}}[\mathcal{R}, h_0, h_1, w, n, q_0, q_1, \alpha]$:

$$\Psi_0^{\mathtt{fold}} := \left\{ (\mathsf{pp}, ((\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{y}), \mathbf{z}), \mathbf{x}) : \begin{array}{c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n} \cdot \mathbf{x} = \mathbf{y} \bmod q_0, \\ \mathbf{C} \cdot \mathbf{x} = \mathbf{z} \bmod q_1, \end{array} \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha, \right\},$$

$\mathcal{R}$ is a prime-power ring for a prime ≥ 5, $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h_0 \times w}$, $\mathbf{C} = (\mathbf{C}_1, \dots, \mathbf{C}_n) \in \mathcal{R}_{q_1}^{h_1 \times wn}$, $\mathbf{y} \in \mathcal{R}_{q_0}^{h_0 \cdot (n+1)}$, $\mathbf{z} \in \mathcal{R}_{q_1}^{h_1}$, and $\mathbf{x} \in \mathcal{R}^{wn}$. Note that the linear constraints consist of a sparse structured part represented by a block-bidiagonal matrix and a dense part. By default, we suppress all parameters of $\Psi_0^{\mathtt{fold}}$ except those that we highlight. Note that the above constraints are independent of pp, therefore $\Psi_0^{\mathtt{fold}}$ is compatible with any parameter generator Gen. We describe $\Pi_0^{\mathtt{fold}}$ which is complete for $\Psi_0^{\mathtt{fold}}[\alpha]$ and knowledge sound for $\Psi_0^{\mathtt{fold}}[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. The protocol $\Pi_0^{\mathtt{fold}}$ is essentially a merge between (the lattice analogue of) Pietrzak's folding protocol [Pie19] and the lattice-based Bulletproofs protocol [BLNS20]. Consider $n > 2$ and let $n' = \lfloor (n-1)/2 \rfloor$. Our protocol hinges on the following observation: Depending on whether n is odd or even, we have

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n} = \left(\begin{array}{c|c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} & \mathbf{A} & \\ \hline & \mathbf{B} & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} \end{array} \right) \quad \text{or} \quad \left(\begin{array}{c|c|c} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} & \mathbf{A} & \\ \hline & \mathbf{B}\ \mathbf{A} & \\ \hline & \mathbf{B} & \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n'} \end{array} \right).$$

[16] Suppose the expression is not unique, let $n = \sum_{i=0}^{\ell} 2^i \cdot k_i = \sum_{i=0}^{\ell} 2^i \cdot k_i'$ with $k_i, k_i' \in \{1,2\}$. Let $d_i = k_i - k_i' \in \{-1,0,1\}$. We have $\sum_{i=0}^{\ell} 2^i \cdot d_i = 0$, which means that $d_0 = 0$ or else the LHS is odd while the RHS is even. Dividing both sides by 2, we get $\sum_{i=0}^{\ell-1} 2^i \cdot d_{i+1} = 0$. By the same argument, we have $d_1 = 0$. Repeating this for all i yields $d_i = 0$ for all $i \in \{0,\dots,\ell\}$, a contradiction.

The protocol $\Pi_0^{\mathtt{fold}}.\langle\mathsf{Prove}(\mathsf{crs},\mathsf{stmt},\mathsf{wit}),\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathsf{off}}},\mathsf{stmt}_{\mathsf{on}})\rangle$ consists of $\ell+1$ rounds and makes use of the subtractive set $S \subset \mathcal{R}^\times$ mentioned in Section 3.1. Denote $(\mathbf{C}^{(0)},\mathbf{x}^{(0)},\mathbf{y}^{(0)},\mathbf{z}^{(0)},\alpha^{(0)}) := (\mathbf{C},\mathbf{x},\mathbf{y},\mathbf{z},\alpha)$. Express n uniquely as $n = \sum_{j=0}^{\ell} 2^j \cdot k_j$ where $k_j \in \{1,2\}$. Note that $\mathbf{y}$ consists of $n' := n+1 = \sum_{j=0}^{\ell-1} 2^j \cdot (k_j - 1) + 2^\ell \cdot (k_\ell + 1)$ blocks. For $i \in \{0,\ldots,\ell\}$, define $n_i := \sum_{j=i}^{\ell} 2^{j-i} \cdot k_j$ and $n_i' := \sum_{j=i}^{\ell-1} 2^{j-i} \cdot (k_j - 1) + 2^{\ell-i} \cdot (k_\ell + 1)$. Then, for $i < \ell$, the i-th round of the protocol is as follows:

- Parse $(\mathbf{C}^{(i)},\mathbf{x}^{(i)},\mathbf{y}^{(i)})$ as
$$(\mathbf{C}_L^{(i)},\mathbf{C}_c^{(i)},\mathbf{C}_R^{(i)}), \qquad (\mathbf{x}_L^{(i)},\mathbf{x}_c^{(i)},\mathbf{x}_R^{(i)}), \qquad \text{and} \qquad (\mathbf{y}_L^{(i)},\mathbf{y}_c^{(i)},\mathbf{y}_R^{(i)})$$
respectively where $\mathsf{ncol}(\mathbf{C}_L^{(i)}) = \mathsf{ncol}(\mathbf{C}_R^{(i)}) = \mathsf{nrow}(\mathbf{x}_L^{(i)}) = \mathsf{nrow}(\mathbf{x}_R^{(i)}) = n_i \cdot w$ and $\mathsf{nrow}(\mathbf{y}_L^{(i)}) = \mathsf{nrow}(\mathbf{y}_R^{(i)}) = n_i' \cdot h_0$. Note that $\mathsf{nrow}(\mathbf{x}_c^{(i)}) = k_i$ and $\mathsf{nrow}(\mathbf{y}_c^{(i)}) = k_i - 1$, meaning that $\mathbf{y}_c^{(i)}$ is empty when $k_i = 1$.
- $\mathcal{P}$ sends
$$\mathbf{x}_c^{(i)}, \qquad \mathbf{z}_{LR}^{(i)} := \mathbf{C}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \bmod q_1, \qquad \text{and} \qquad \mathbf{z}_{RL}^{(i)} := \mathbf{C}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \bmod q_1.$$
- $\mathcal{V}$ checks that $\left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)}$. If $k_i = 2$, $\mathcal{V}$ further checks that $\big(\mathbf{B}\ \mathbf{A}\big) \cdot \mathbf{x}_c^{(i)} = \mathbf{y}_c^{(i)} \bmod q_0$. If any of these checks fails, $\mathcal{V}$ aborts.
- $\mathcal{V}$ samples $r_i \leftarrow\!\!\$\, S$ and sends r_i to $\mathcal{P}$.
- $\mathcal{P}$ computes the compressed witness $\mathbf{x}^{(i+1)} := \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$.
- $\mathcal{P}$ and $\mathcal{V}$ compute the compressed statement
$$\begin{aligned}
\mathbf{C}^{(i+1)} &:= \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1} \bmod q_1 \\
\mathbf{y}^{(i+1)} &:= \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix} \mathbf{B} \cdot r_i \\ \mathbf{0} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0 \\
\mathbf{z}^{(i+1)} &:= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i \bmod q_1 \\
\alpha^{(i+1)} &:= 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}}
\end{aligned}$$

In the ℓ-th (i.e. final) round, $\mathcal{P}$ sends $\mathbf{x}^{(\ell)}$ and $\mathcal{V}$ checks that
$$\begin{aligned} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow k_\ell} \cdot \mathbf{x}^{(\ell)} &= \mathbf{y}^{(\ell)} \bmod q_0, \\ \mathbf{C}^{(\ell)} \cdot \mathbf{x}^{(\ell)} &= \mathbf{z}^{(\ell)} \bmod q_1, \end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(\ell)}\right\| \leq \alpha^{(\ell)} = (2\gamma_{\mathcal{R}})^\ell \cdot \alpha.$$

Analysis. We show that $\Pi_0^{\mathtt{fold}}$ is complete and (unconditionally) special-sound. We further show that $\Pi_0^{\mathtt{fold}}$ has short proofs, quasi-linear-time prover, and polylogarithmic-time verifier.

For readability, we defer the proofs of the above claims to Appendix D.

Theorem 1. *$\Pi_0^{\mathtt{fold}}$ is complete for $\Psi_0^{\mathtt{fold}}[\alpha]$.*

Theorem 2. *If $\alpha^* \geq (8\gamma_{\mathcal{R}}^4)^{\log n}\alpha$, $\Pi_0^{\mathtt{fold}}$ is $(3,\ldots,3)$-special sound for $\Psi_0^{\mathtt{fold}}[\alpha^*]$.*

For the purpose of estimating the complexities of $\Pi_0^{\mathtt{fold}}$, let $h_0, h_1, w, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ. Pick α^* to be tight in Theorem 2 and set $q_0, q_1 = O_\lambda(\alpha^*) = \lambda^{O(\log n)}$. The following theorem states the complexities of $\Pi_0^{\mathtt{fold}}$ with the above parameter choices.

Theorem 3. *Let $h_0, h_1, w, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ, and $q_0, q_1 = \lambda^{O(\log n)}$. $\Pi_0^{\mathtt{fold}}$ has (i) prover time $O_\lambda(n \cdot \log^2 n)$, and (ii) proof size $O_\lambda(\log^2 n)$. If $\mathbf{C}$ is $(k_0,\ldots,k_\ell)$-block-foldable with block-size w and $\mathbf{y}$ is $(k_0 - 1,\ldots,k_{\ell-1} - 1, k_\ell + 1)$-block-foldable with block-size h_0, then the verifier time is $O_\lambda(\log^3 n)$.*

6.2 Type-1 Linear Relations

Define the relation $\Psi_1^{\mathtt{fold}} = \Psi_1^{\mathtt{fold}}[\mathcal{R}, h, w, n, q, \alpha]$:

$$\Psi_1^{\mathtt{fold}} := \left\{(\mathsf{pp}, (\mathbf{A}, \mathbf{y}), \mathbf{x}) : \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha\right\},$$

$\mathcal{R}$ is a prime-power ring for a prime ≥ 5, $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_n)$, $\mathbf{A}_i \in \mathcal{R}_q^{h \times w}$, $\mathbf{y} \in \mathcal{R}_q^h$, and $\mathbf{x} \in \mathcal{R}^{wn}$. By default, we suppress parameters of $\Psi_1^{\mathtt{fold}}$ except those that we highlight. Note that the above constraints are independent of pp, therefore $\Psi_1^{\mathtt{fold}}$ is compatible with any parameter generator Gen. We describe $\Pi_1^{\mathtt{fold}}$ which is complete for $\Psi_1^{\mathtt{fold}}[\alpha]$ and knowledge sound for $\Psi_1^{\mathtt{fold}}[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. We construct in Appendix C a protocol $\Pi_1^{\mathtt{fold}}$ which can be seen as a simplification of $\Pi_0^{\mathtt{fold}}$ by removing components responsible for the structured part of the relation.

Analysis. We state the formal claims about the completeness, special-soundness, and efficiency of $\Pi_1^{\mathtt{fold}}$. The proofs of these claims are almost identical to those of Theorems 1 to 3 and are therefore omitted.

Theorem 4. *$\Pi_1^{\mathtt{fold}}$ is complete for $\Psi_1^{\mathtt{fold}}[\alpha]$.*

Theorem 5. *For $\alpha^* \geq (8\gamma_{\mathcal{R}}^4)^{\log n} \cdot \alpha$, $\Pi_1^{\mathtt{fold}}$ is $(3, \ldots, 3)$-special sound for $\Psi_1^{\mathtt{fold}}[\alpha^*]$.*

Theorem 6. *Let $h, w = \mathsf{poly}(\lambda)$ and $q = \lambda^{O(\log n)}$. $\Pi_1^{\mathtt{fold}}$ has (i) prover time $O_\lambda(n \cdot \log^2 n)$, and (ii) proof size $O_\lambda(\log^2 n)$. If $\mathbf{A}$ is $(k_0, \ldots, k_\ell)$-block-foldable with block-size w, then the verifier time is $O_\lambda(\log^3 n)$.*

7 Knowledge-based Protocols

Mirroring the folding protocols constructed in Section 6, we present below two argument systems $\Pi_0^{\mathtt{know}}$ and $\Pi_1^{\mathtt{know}}$ for unstructured linear relations based on the (knowledge-)k-R-ISIS assumptions. Different from existing protocols based on the same family of assumptions and construction template, the constructions below feature quasi-linear-time provers.

7.1 Linear Relations

Define the relation $\Psi_0 = \Psi_0[\mathcal{R}, s, t, q_0, q_1, q_2, \alpha]$:

$$\Psi_0 := \left\{ ((\mathbf{v}, \mathbf{h}), ((\mathbf{M}, \mathbf{y}), (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})), \mathbf{x}) : \begin{array}{r} \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0, \\ \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} = c_{\mathbf{x}} \bmod q_3, \\ (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{x} = \bar{c}_{\mathbf{x}} \bmod q_3, \end{array} \quad \|\mathbf{x}\| \leq \alpha \right\}$$

where $\mathbf{M} \in \mathcal{R}_{q_3}^{t \times s}$, $\mathbf{y} \in \mathcal{R}_{q_3}^t$, $c_{\mathbf{x}}, \bar{c}_{\mathbf{x}} \in \mathcal{R}_{q_3}$, $\mathbf{x} \in \mathcal{R}^s$, $\mathbf{v} = (v, v^2, \ldots, v^s)$, and $\bar{\mathbf{v}} = (v^{-1}, v^{-2}, \ldots, v^{-s})$. Accompanying the relation, we define a parameter generator $\mathsf{Gen}^{\mathtt{unstr}}$ which samples $v \leftarrow\!\!\$\; \mathcal{R}_{q_3}^\times$ and $\mathbf{h} \leftarrow\!\!\$\; \mathcal{R}_{q_1}^s$ and outputs $(\mathbf{v}, \mathbf{h})$. Note that the compression vector $\mathbf{h}$ is unstructured. By default, we suppress all parameters of Ψ_0 except those that we highlight. We describe a protocol $\Pi_0^{\mathtt{know}}$ which is complete for $\Psi_0[\alpha]$ and knowledge sound for $\Psi_0[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. Let $\mathcal{R}, s, t, \eta, m, (q_i)_{i=0}^3, \beta, (\delta_i)_{i=0}^3, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (Section 3.2) parametrised by (η, m, q_3, β), in Fig. 1 we give a formal description of $\Pi_0^{\mathtt{know}}$, which is based on the construction template of functional commitments in [ACL+22]. In particular, in $\Pi_0^{\mathtt{know}}$ the prover proves to the verifier that they know witnesses to the following relations

$$\begin{pmatrix} \mathbf{v}^{\mathtt{T}} \\ (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathtt{T}} \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} c_{\mathbf{x}} \\ \bar{c}_{\mathbf{x}} \end{pmatrix} \bmod q_3, \quad \text{and} \quad \|\mathbf{x}\| \leq \alpha, \tag{3}$$

$$\mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} = c_{\mathbf{r}}, \quad \text{with} \quad \mathbf{r} \in \mathcal{R}^t, \tag{4}$$

and

$$\mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \quad \|\mathbf{x}\| \leq \alpha. \tag{5}$$

```
$\mathsf{Setup}(1^\lambda, \mathsf{pp})$
    $(\mathbf{v}, \mathbf{h}) \leftarrow \mathsf{pp}$
    $\mathbf{f}_0 \leftarrow\!\!\$\; \mathcal{R}_{q_2}^t,\ \mathbf{f}_1 \leftarrow\!\!\$\; \mathcal{R}_{q_2}^s$
    $I_0 \coloneqq \pm[\max\{s, t\}],\ I_1 \coloneqq [s]$
    $I_2 \coloneqq -[s],\ I_3 \coloneqq [t]$
    for $i \in \{0, 1, 2, 3\}$ do
        $(\mathbf{D}_i, \mathsf{td}_i) \leftarrow \mathsf{TrapGen}(1^\lambda)$
        $\mathbf{t}_i \leftarrow\!\!\$\; \mathcal{T}$
        $\mathbf{u}_{i,j} \leftarrow \mathsf{SampPre}(\mathsf{td}_i, \mathbf{t}_i \cdot v^j),\ \forall j \in I_i$
    $\mathsf{crs} \coloneqq \begin{pmatrix} (\mathbf{D}_i,\ \mathbf{t}_i,\ (\mathbf{u}_{i,j})_{j \in I_i})_{i=0}^3, \\ v \quad \mathbf{h} \quad \mathbf{f}_0, \quad \mathbf{f}_1 \end{pmatrix}$
    return $\mathsf{crs}$
```

```
$\mathsf{Prove}(\mathsf{crs}, ((\mathbf{M}, \mathbf{y}), (c_\mathbf{x}, \bar{c}_\mathbf{x})), \mathbf{x})$
    $\mathbf{v}_t \coloneqq (v, v^2, \ldots, v^t)$
    $c_\mathbf{r} \coloneqq \mathbf{v}_t^\mathtt{T} \cdot \mathbf{r} \bmod q_3$
    $\mathbf{u}_{0,0} \coloneqq \sum_{i \in [s], k \in [t]} f_{0,k} M_{k,i} \sum_{j \in [s]: j \neq i} \mathbf{u}_{0,j-i} x_j$
        $+ \sum_{i,k \in [t]} f_{0,k} q_0 \sum_{j \in [t]: j \neq i} \mathbf{u}_{0,j-i} r_j$
    $\mathbf{u}_{0,1} \coloneqq \sum_{j \in [s]} h_j f_{1,j} \sum_{i \in [s]: i \neq j} \mathbf{u}_{0,i-j} x_i$
        $- \sum_{i \in [s]} f_{1,i} \sum_{j \in [s]: j \neq i} \mathbf{u}_{0,i-j} h_j \cdot x_j$
    $\mathbf{u}_0 \coloneqq \mathbf{u}_{0,0} + \mathbf{u}_{0,1}$
    $\mathbf{u}_1 \coloneqq \sum_{j \in [s]} \mathbf{u}_{1,j} \cdot x_j$
    $\mathbf{u}_2 \coloneqq \sum_{j \in [s]} \mathbf{u}_{2,-j} \cdot h_j \cdot x_j$
    $\mathbf{u}_3 \coloneqq \sum_{j \in [t]} \mathbf{u}_{3,j} \cdot r_j$
    return $\pi \coloneqq (c_\mathbf{x}, \bar{c}_\mathbf{x}, c_\mathbf{r}, \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$
```

```
$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{M}, \mathbf{y}))$
    $\mathbf{v} \coloneqq (v, v^2, \ldots, v^s)$
    $\bar{\mathbf{v}} \coloneqq (v^{-1}, v^{-2}, \ldots, v^{-s})$
    $\bar{\mathbf{v}}_t \coloneqq (v^{-1}, v^{-2}, \ldots, v^{-t})$
    $\bar{c}_\mathbf{M} \coloneqq \mathbf{f}_0^\mathtt{T} \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \bmod q_3$
    $\bar{c}_{q_0} \coloneqq \mathbf{f}_0^\mathtt{T} \cdot q_0 \cdot \bar{\mathbf{v}}_t \bmod q_3$
    $\bar{c}_\mathbf{I} \coloneqq \mathbf{f}_1^\mathtt{T} \cdot \mathbf{I} \cdot (\bar{\mathbf{v}} \circ \mathbf{h})$
        $= \mathbf{f}_1^\mathtt{T} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \bmod q_3$
    $c_\mathbf{I} \coloneqq \mathbf{v}^\mathtt{T} \cdot \mathbf{I} \cdot \mathbf{f}_1 = \mathbf{v}^\mathtt{T} \cdot \mathbf{f}_1 \bmod q_3$
    $\hat{c}_\mathbf{y} \coloneqq \mathbf{f}_0^\mathtt{T} \cdot \mathbf{y} \bmod q_3$
    $\mathsf{pp}_{\mathbf{M},\mathbf{y},c_\mathbf{x},\bar{c}_\mathbf{x}} \coloneqq \begin{pmatrix} (\mathbf{D}_i, \mathbf{t}_i)_{i=0}^3, \\ \bar{c}_\mathbf{M}, \bar{c}_{q_0}, \bar{c}_\mathbf{I}, c_\mathbf{I}, \hat{c}_\mathbf{y} \end{pmatrix}$
    return $\mathsf{pp}_{\mathbf{M},\mathbf{y}}$
```

```
$\mathsf{Verify}(\mathsf{crs}_{\mathbf{M},\mathbf{y}}, (c_\mathbf{x}, \bar{c}_\mathbf{x}), \pi)$
    $c_{0,0} \coloneqq \bar{c}_\mathbf{M} \cdot c_\mathbf{x} + \bar{c}_{q_0} \cdot c_\mathbf{r} - \hat{c}_\mathbf{y} \bmod q_3$
    $c_{0,1} \coloneqq \bar{c}_\mathbf{I} \cdot c_\mathbf{x} - \bar{c}_\mathbf{x} \cdot c_\mathbf{I} \bmod q_3$
    $c_0 \coloneqq c_{0,0} + c_{0,1} \bmod q_3$
    $c_1 \coloneqq c_\mathbf{x}$
    $c_2 \coloneqq \bar{c}_\mathbf{x}$
    $c_3 \coloneqq c_\mathbf{r}$
    for $i \in \{0, 1, 2, 3\}$ do
        $b_i \coloneqq \begin{pmatrix} \mathbf{D}_i \cdot \mathbf{u}_i \overset{?}{\equiv} \mathbf{t}_i \cdot c_i \bmod q_3 \\ \wedge \quad \|\mathbf{u}_i\| \overset{?}{\leq} \delta_i \end{pmatrix}$
    return $b_0 \wedge b_1 \wedge b_2 \wedge b_3$
```

Fig. 1. Our argument system Π_0^{know}.

The prover will prove that $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$ are well-formed by proving knowledge of a short opening of the commitments $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$ with respect to the commitment key $(v^i)_{i \in [s]}$, $(v^{-i})_{i \in [s]}$, and $(v_i)_{i \in [t]}$ respectively. To prove consistency between $c_{\mathbf{x}}$ and $\bar{c}_{\mathbf{x}}$, the prover proves knowledge of a short opening of the commitment $\bar{c}_{\mathbf{I}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{x}} \cdot c_{\mathbf{I}}$, where the values $\bar{c}_{\mathbf{I}}$ and $c_{\mathbf{I}}$ can be precomputed by the verifier. This is with respect to the commitment key $(v^k)_{k \in \pm[\max\{s,t\}]}$. Finally, to prove Eq. (5), the prover proves knowledge of a short opening of the commitment $\bar{c}_{\mathbf{M}} \cdot c_{\mathbf{x}} + \bar{c}_{q_0} \cdot c_{\mathbf{r}} - \hat{c}_{\mathbf{y}}$, where the values $\bar{c}_{\mathbf{M}}, \bar{c}_{q_0}$, and $\hat{c}_{\mathbf{y}}$ can be precomputed by the verifier. This is again with respect to the commitment key $(v^k)_{k \in \pm[\max\{s,t\}]}$.

We highlight a few crucial differences with [ACL+22]:

(i) The witness $\mathbf{x}$ is committed using a univariate vSIS commitment, i.e. the commitment key is $\mathbf{v} = (v, v^2, \ldots, v^s)$, while in [ACL+22] the commitment is an s-variate vSIS commitment. The fact that $|\{ v^{i-j} : i, j \in [s] \}|$ has cardinality $O(s)$ and that the prover computation consists of mainly Toeplitz-vector multiplications are crucial for obtaining a quasi-linear-time prover.

(ii) We support proving relations modulo q_0 natively[17] by introducing the auxiliary witness $\mathbf{r}$ satisfying $\mathbf{M} \cdot \mathbf{x} + q_0 \cdot \mathbf{r} = \mathbf{y}$. In [ACL+22], modular arithmetic is handled via generic and expensive bit-decomposition techniques.

(iii) To prove that values committed in multiple commitments, i.e. $c_{\mathbf{x}}$, $\bar{c}_{\mathbf{x}}$, and $c_{\mathbf{r}}$, satisfy some relation, we adapt techniques developed for the recent construction of chainable functional commitments [BCFL22].

Analysis. We show that Π_0^{know} is correct and knowledge-sound under (knowledge-)k-R-ISIS and R-SIS assumptions. We further show that Π_0^{know} has short CRS and proofs, quasi-linear-time prover and preprocessing, and polylogarithmic-time verifier after preprocessing. The proofs are deferred to Appendix F.

Theorem 7 (Completeness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold. For*

$$\delta_0 \geq (s+t)^4 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3, \qquad \delta_1 \geq s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}},$$
$$\delta_2 \geq s \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}, \qquad \text{and} \qquad \delta_3 \geq s^2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2,$$

Π_0^{know} in Fig. 1 is complete for $\Psi_0[\alpha]$.

Theorem 8 (Knowledge Soundness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold. Let $w = 1$, $\mathcal{G}_0 = \{ X^i : i \in \pm[\max\{ s, t \}] \}$, $\mathcal{G}_1 = \{ X^i : i \in [s] \}$, $\mathcal{G}_2 = \{ X^i : i \in -[s] \}$, and $\mathcal{G}_3 = \{ X^i : i \in [t] \}$ be sets of monomials in X. Let $\mathcal{D} = \mathsf{SampD}(1^\lambda)$. For $i \in \{ 1, 2, 3 \}$, let $\mathcal{Z}_i(1^\lambda)$ be almost identical to $\mathsf{Setup}(1^\lambda, \mathsf{Gen}^{\mathsf{unstr}}(1^\lambda))$, except that it inputs $(\mathbf{D}_i, \mathbf{t}_i, v, \{ \mathbf{u}_{i,j} \}_{j \in I_i})$ and generates the rest of crs. Let*

$$\alpha_i^* \geq \delta_i, \ \forall i \in [3], \qquad \alpha^* := \max\{ \alpha_1^*, \alpha_2^*, \alpha_3^* \}, \qquad q_2 \geq \beta_{q_2}^* \geq s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}},$$
$$q_3 \geq \beta_{q_3}^* \geq \max\{ 2\delta_0, (s+t)^3 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \}.$$

Π_0^{know} in Fig. 1 is knowledge-sound for $\Psi_0[\alpha_1^*]$ if the following assumptions hold:

Assumption 0. *k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\beta,\beta_{q_3}^*,\mathcal{G}_0,g^*=1,\mathcal{D},\mathcal{T}}$,*
Assumption 1. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_1^*,\beta,\delta_1,\mathcal{G}_1,\mathcal{D},\mathcal{T},\mathcal{Z}_1}$,*
Assumption 2. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_2^*,\beta,\delta_2,\mathcal{G}_2,\mathcal{D},\mathcal{T},\mathcal{Z}_2}$,*
Assumption 3. *knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_3^*,\beta,\delta_3,\mathcal{G}_3,\mathcal{D},\mathcal{T},\mathcal{Z}_3}$, and*
Assumption 4. *R-$\mathsf{SIS}_{\mathcal{R},s+t,q_2,\beta_{q_2}^*}$.*

For the purpose of estimating complexities, we assume that the assumptions in Theorem 8 hold for moduli which are a fixed polynomial factor larger than their norm bounds, e.g. $q_2 \geq \beta_{q_2}^* \cdot \mathsf{poly}(\lambda)$ for the R-$\mathsf{SIS}_{\mathcal{R},s+t,q_2,\beta_{q_2}^*}$ assumption. For the k-R-ISIS assumptions, we assume that they hold for $m = O(\eta \cdot \log q)$.

Let $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ. For our application in Section 8, we want $q_1 = O(s^2 \cdot \alpha^2) = O_\lambda(s^2)$. Pick $\delta_1, \delta_2, \delta_3, \alpha_1^*, \alpha_2^*, \alpha_3^*$ so that they match their lower bounds given in Theorem 7 and Theorem 8 respectively. Substituting q_1, we have $\alpha_1^* = \delta_1 = O_\lambda(s)$, $\alpha_2^* = \delta_2 = O_\lambda(s^3)$, and $\alpha_3^* = \delta_3 = O_\lambda(s^2)$. We therefore have $\alpha^* = O_\lambda(s^3)$. Pick $q_0 = O_\lambda(\alpha_1^*) = O_\lambda(s)$. Pick $\beta_{q_2}^*$ so that it matches its lower bound in Theorem 8, and set $q_2 = O_\lambda(\beta_{q_2}^*)$. Substituting (q_0, q_1, α^*), we have

[17] Relations without modular reduction are captured by setting $q_0 = 0$.

$q_2 = O_\lambda(s^7)$. Pick δ_0 so that it matches its lower bound given in Theorem 7. Substituting (q_0, q_1, q_2), we have $\delta_0 = O_\lambda((s+t)^{14})$. Pick $\beta^*_{q_3}$ so that it matches its lower bound in Theorem 8, and set $q_3 = O_\lambda(\beta^*_{q_3})$. Substituting $(q_0, q_1, q_2, \alpha^*)$, we have $q_3 = O_\lambda((s+t)^{16})$. Let $n = \max\{|\mathbf{M}|, s+t\}$, where $|\mathbf{M}|$ denote the number of non-zero entries in $\mathbf{M}$. Pick $m = O(\eta \cdot \log q) = O_\lambda(\log n)$.

Theorem 9 states the complexities of $\Pi_0^{\mathtt{know}}$ with the above parameter choices.

Theorem 9 (Efficiency). *Let $n = \max\{|\mathbf{M}|, s+t\}$, where $|\mathbf{M}|$ denote the number of non-zero entries in $\mathbf{M}$, $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ be fixed polynomials in λ, and $(m, q_0, q_1, q_2, q_3) = (\log n, s, s^2, s^7, (s+t)^{16}) \cdot \mathsf{poly}(\lambda)$. Then $\Pi_0^{\mathtt{fold}}$ has (i) common reference string size $O_\lambda(n \cdot \log n)$, (ii) proof size $O_\lambda(\log^2 n)$, (iii) prover time $O_\lambda(n \cdot \log^3 n)$, (iv) preprocessing time $O_\lambda(n \cdot \log^2 n)$, and (v) verifier time $O_\lambda(\log^3 n)$ after preprocessing.*

7.2 Well-formedness of vSIS Commitments

Define the relation $\Psi_1 = \Psi_1[\mathcal{R}, s, q_1, q_3, \alpha]$ equipped with the same parameter generator $\mathsf{Gen}^{\mathtt{unstr}}$ as Ψ_0:

$$\Psi_1 := \left\{ ((\mathbf{v}, \mathbf{h}), (\epsilon, c_{\mathbf{z}}), \mathbf{z}) : \left(\bar{\mathbf{v}}^{\mathtt{T}} \ \mathbf{v}^{\mathtt{T}}\right) \cdot \mathbf{z} = c_{\mathbf{z}} \bmod q_3 \quad \wedge \quad \|\mathbf{z}\| \leq \alpha \right\}$$

where $c_{\mathbf{z}} \in \mathcal{R}_{q_3}$, $\mathbf{z} \in \mathcal{R}^{2s}$, $\mathbf{v} = (v, v^2, \ldots, v^s)$, and $\bar{\mathbf{v}} = (v^{-1}, v^{-2}, \ldots, v^{-s})$. By default, we suppress all parameters of Ψ_1 except those that we highlight. We describe a protocol $\Pi_1^{\mathtt{know}}$ which is complete for $\Psi_1[\alpha]$ and knowledge sound for $\Psi_1[\alpha^*]$ for some $\alpha^* > \alpha$.

Construction. We construct in Appendix E a protocol $\Pi_1^{\mathtt{know}}$ for the relation Ψ_1. The proof for $\mathbf{z}$ is simply $\mathbf{D}^{-1}(\mathbf{t} \cdot c_{\mathbf{z}})$ with $(\mathbf{D}, \mathbf{t})$ given in crs.

Analysis. $\Pi_1^{\mathtt{know}}$ is correct and knowledge-sound under the knowledge-k-R-ISIS assumption. It has short CRS and proofs, quasi-linear-time prover, and polylogarithmic-time verifier. Below, we state these claims formally and omit the (trivial) proofs.

Theorem 10 (Completeness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold. For $\delta \geq 2s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}$ $\Pi_1^{\mathtt{know}}$ in Fig. 1 is complete for $\Psi_1[\alpha]$.*

Theorem 11 (Knowledge Soundness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold, $w = 1$, $\alpha^* \geq \delta$, $\mathcal{G} = \{\, X^i : i \in \pm[s] \,\}$ be a set of monomials in X, $\mathcal{D}$ denote the distribution $\mathsf{SampD}(1^\lambda)$, and $\mathcal{Z}$ be trivial (i.e. it outputs $\perp$). $\Pi_1^{\mathtt{know}}$ in Fig. 1 is knowledge-sound for $\Psi_0[\alpha^*]$ if the knowledge-k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha^*,\beta,\delta,\mathcal{G},\mathcal{D},\mathcal{T},\mathcal{Z}}$ assumption holds.*

Theorem 12 (Efficiency). *Let parameters be as in Theorem 9. $\Pi_1^{\mathtt{fold}}$ has (i) common reference string size $O_\lambda(n \cdot \log n)$, (ii) proof size $O_\lambda(\log^2 n)$, (iii) prover time $O_\lambda(n \cdot \log^2 n)$, (iv) trivial preprocessing, and (v) verifier time $O_\lambda(\log^3 n)$.*

8 Applications

We show how to compose arguments obtain in Sections 6 and 7 to build efficient arguments for more complex relations. In particular, we show how to construct arguments for the binary-satisfiability of (structured) linear equations and rank-1 constraint satisfiability (R1CS).

8.1 Proving Binary-Satisfiability of (Structured) Linear Equations

Recall that in Section 6 we built succinct arguments $\Pi_0^{\mathtt{fold}}$ and $\Pi_1^{\mathtt{fold}}$ for the relations $\Psi_0^{\mathtt{fold}}$ and $\Psi_1^{\mathtt{fold}}$ respectively, while in Section 7 we constructed $\Pi_0^{\mathtt{know}}$ and $\Pi_1^{\mathtt{know}}$ for the relations Ψ_0 and Ψ_1 respectively. By inspection, we see that Ψ_1 is a special case of $\Psi_1^{\mathtt{fold}}$, and thus $\Pi_1^{\mathtt{fold}}$ can be specialised to give a succinct argument for Ψ_1. Similarly, $\Pi_0^{\mathtt{fold}}$ can be specialised as to give a succinct argument for the following special case of Ψ_0 which we denote by $\Psi_0^{\mathtt{str}} = \Psi_0^{\mathtt{str}}[\mathcal{R}, h, w, n, q_0, q_1, q_3, \alpha]$, where $\mathbf{M}$ is restricted to be of the form $\mathbf{M} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n}$ succinctly represented by some $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h \times w}$.

Accompanying $\Psi_0^{\mathtt{str}}$, we define the parameter generator $\mathsf{Gen}^{\mathtt{str}}$ which samples $(\mathbf{v}, \mathbf{h})$ which are $(k_0, \ldots, k_\ell)$-block-foldable with block-size w where $n = \sum_{i=0}^{\ell} k_i$ for $k_i \in \{1, 2\}$. More concretely,

```
$\mathsf{Setup}(1^\lambda)$
    $\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda)$
    $\mathsf{crs}' \leftarrow \Pi'.\mathsf{Setup}(1^\lambda, \mathsf{pp})$
    $\mathsf{crs}'' \leftarrow \Pi''.\mathsf{Setup}(1^\lambda, \mathsf{pp})$
    return $\mathsf{crs} := (\mathsf{crs}', \mathsf{crs}'')$
```

```
$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{M}, \mathbf{y}))$
    $\mathsf{crs}'_{(\mathbf{M},\mathbf{y})} \leftarrow \Pi'.\mathsf{PreVerify}(\mathsf{crs}', (\mathbf{M}, \mathbf{y}))$
    $\mathsf{crs}''_\epsilon \leftarrow \Pi''.\mathsf{PreVerify}(\mathsf{crs}'', \epsilon)$
    return $\mathsf{crs}_{(\mathbf{M},\mathbf{y})} := (\mathsf{crs}'_{(\mathbf{M},\mathbf{y})}, \mathsf{crs}''_\epsilon)$
```

Fig. 2. Setup and $\mathsf{PreVerify}$ algorithms of the argument system $\Pi^{\texttt{bin-sat}}$.

$\mathsf{Gen}^{\mathtt{str}}$ does the following: (i) Sample $v \leftarrow\$ \mathcal{R}_q^\times$ and $\tilde{\mathbf{h}} \leftarrow\$ \mathcal{R}_{q_1}^{\tilde{n}}$. (ii) Set $\mathbf{v} := (v, \ldots, v^s) \bmod q_3$. (iii) Let $\tilde{n} := \sum_{i=0}^{\ell-1}(k_i + 1) + k_\ell$. (iv) Generate w copies of $\tilde{n}$-variate monomial sequences $\mathbf{m}_1, \ldots, \mathbf{m}_w$ according to Lemma 6, and concatenate them in an interleaved manner into a monomial sequence $\mathbf{m} = (m_{1,1}, m_{2,1}, \ldots, m_{w,1}, m_{1,2}, \ldots, m_{w,n})$. (v) Evaluate $\mathbf{m}$ at $\tilde{\mathbf{h}}$ to produce $\mathbf{h} = \mathbf{m}(\tilde{\mathbf{h}})$.

Equipped with succinct arguments for Ψ_0 (or $\Psi_0^{\mathtt{str}}$) and Ψ_1, we construct a succinct argument $\Pi^{\texttt{bin-sat}}$ for the binary-satisfiability of system of (structured) linear equations mod p. Formally, define the relation $\Psi^{\texttt{bin-sat}} = \Psi^{\texttt{bin-sat}}[\mathcal{R}, s, t, p]$:

$$\Psi^{\texttt{bin-sat}} := \left\{(((\mathbf{M}, \mathbf{y}), \epsilon), \mathbf{x}) : \mathbf{M} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \quad \wedge \quad \mathbf{x} \in \{0,1\}^s\right\},$$

with offline statement $(\mathbf{M}, \mathbf{y}) \in \mathcal{R}_{q_0}^{t \times s} \times \mathcal{R}_{q_0}^t$ and witness $\mathbf{x} \in \mathcal{R}^s$, and the corresponding structured variant $\Psi^{\texttt{str-bin-sat}} = \Psi^{\texttt{str-bin-sat}}[\mathcal{R}, h, w, n, p]$ where $\mathbf{M}$ is of the form $\mathbf{M} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n}$ succinctly represented by some $\mathbf{A}, \mathbf{B} \in \mathcal{R}_{q_0}^{h \times w}$.

Let q_1, q_3 depend on λ. Let Π' and Π'' be argument systems for Ψ_0 (or $\Psi_0^{\mathtt{str}}$) and Ψ_1 respectively, and let $\mathsf{Gen} = \mathsf{Gen}^{\mathtt{unstr}}$ (or $\mathsf{Gen}^{\mathtt{str}}$) be the accompanying parameter generator. The algorithms $\Pi^{\texttt{bin-sat}}.(\mathsf{Setup}, \mathsf{PreVerify})$ are described in Fig. 2. The protocol $\Pi^{\texttt{bin-sat}}.\left\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{(\mathbf{M},\mathbf{y})}, \epsilon)\right\rangle$ is below:

- Prove computes
  (i) $c_{\mathbf{x}} := \langle \mathbf{v}, \mathbf{x} \rangle \bmod q_3$,
  (ii) $\bar{c}_{\mathbf{x}} := \langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x} \rangle \bmod q_3$, and
  (iii) $\mathbf{z} := \left(\sum_{0 \leq i,j \leq s : i-j=k} h_j \cdot x_j \cdot (x_i - 1)\right)_{-s \leq k \leq s}$.
- Prove sends $(c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})$ to Verify.
- Prove and Verify compute:
  - $c_{\mathbf{z}} := \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1} \rangle) \bmod q_3$.
  - $\mathsf{stmt}' := ((\mathbf{M}, \mathbf{y}), (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))$, $\mathsf{stmt}'' := (\epsilon, c_{\mathbf{z}})$.
  - $(\mathsf{tx}', b') \leftarrow \Pi'.\left\langle \mathsf{Prove}(\mathsf{crs}', \mathsf{stmt}', \mathbf{x}), \mathsf{Verify}(\mathsf{crs}'_{(\mathbf{M},\mathbf{y})}, (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))\right\rangle$.
  - $(\mathsf{tx}'', b'') \leftarrow \Pi''.\langle \mathsf{Prove}(\mathsf{crs}'', \mathsf{stmt}'', \mathbf{z}), \mathsf{Verify}(\mathsf{crs}''_\epsilon, c_{\mathbf{z}})\rangle$.
- Output (tx, b), where $\mathsf{tx} = (\mathsf{tx}', \mathsf{tx}'')$ and $b = b' \wedge b''$.

We show that $\Pi^{\texttt{bin-sat}}$ is complete and knowledge-sound, and that it has short proofs, quasi-linear-time prover, and polylogarithmic-time verifier (after preprocessing in the unstructured case). All proofs are deferred to Appendix G.

Theorem 13. *If* $\mathsf{Gen} = \mathsf{Gen}^{\mathtt{str}}$ *(resp.* $\mathsf{Gen}^{\mathtt{unstr}}$*),* Π' *is complete for* $\Psi_0^{\mathtt{str}}[\alpha = 1]$*, and* Π'' *is complete for* $\Psi_1[\alpha = s \cdot (q_1/2)^{\ell+1} \cdot \gamma_{\mathcal{R}}^\ell]$ *(resp.* $\Psi_1[\alpha = s \cdot q_1/2]$*) then* $\Pi^{\texttt{bin-sat}}$ *is complete for* $\Psi^{\texttt{str-bin-sat}}$ *(resp.* $\Psi^{\texttt{-bin-sat}}$*).*

Theorem 14. *Let* $\mathsf{Gen} = \mathsf{Gen}^{\mathtt{str}}$ *(resp.* $\mathsf{Gen}^{\mathtt{unstr}}$*). Let* $\mathcal{G} := \{X^j : -s \leq j \leq s\}$ *and* $\mathcal{G}_{\mathbf{h}}$ *be the set of monomials generated as in* $\mathsf{Gen}^{\mathtt{str}}$*. Let* $q_1, q_3, \alpha', \alpha'', \beta_{q_1}, \beta_{q_3}$ *be such that (i)* $\beta_{q_1} \geq (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}$*, (ii)* $\beta_{q_3} \geq \alpha'' + s \cdot (q_1/2)^{\ell+1} \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^{\ell+2}$ *(resp.* $\alpha'' + s \cdot q_1/2 \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^2$*), (iii)* Π' *is knowledge-sound for* $\Psi_0^{\mathtt{str}}[\alpha']$ *(resp.* $\Psi_0^{\mathtt{unstr}}[\alpha']$*), and (iv)* Π'' *is knowledge-sound for* $\Psi_1[\alpha'']$*.* $\Pi^{\texttt{bin-sat}}$ *is knowledge-sound for* $\Psi^{\texttt{str-bin-sat}}$ *(resp.* $\Psi^{\texttt{bin-sat}}$*), if the following assumptions hold:*

Assumption 0. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G}_{\mathbf{h}},1,q_1,\beta_{q_1}}$ *(resp.* R-$\mathsf{SIS}_{\mathcal{R},s,q_1,\beta_{q_1}}$*), and*
Assumption 1. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},1,q_3,\beta_{q_3}}$.

Below, we estimate the complexities of $\Pi^{\texttt{bin-sat}}$ for parameters chosen in such a way that completeness and knowledge-soundness (are believed to) hold.

Theorem 15. *In the structured setting, let* $\mathsf{Gen} = \mathsf{Gen}^{\texttt{str}}$*,* $\Pi' = \Pi_0^{\texttt{fold}}$ *(specialised for* $\Psi_0^{\texttt{str}}$*),* $\Pi'' = \Pi_1^{\texttt{fold}}$ *(specialised for* Ψ_1*),* $\gamma_{\mathcal{R}}, \alpha', \alpha'', h, w = \mathsf{poly}(\lambda)$ *be fixed polynomials in* λ*, and* $q_0, q_1, q_3 = \lambda^{O(\log n)}$*.* $\Pi^{\texttt{bin-sat}}$ *has (i) common reference string size* $O_\lambda(\log^2 n)$*, (ii) prover time* $O_\lambda(n \cdot \log^3 n)$*, and (iii) proof size* $O_\lambda(\log^2 n)$*. If* $\mathbf{y}$ *is* $(k_0 - 1, \ldots, k_{\ell-1} - 1, k_\ell + 1)$*-block-foldable with block-size* h*, then the verifier time is* $O_\lambda(\log^3 n)$*.*

In the unstructured setting, let $\mathsf{Gen} = \mathsf{Gen}^{\texttt{unstr}}$*,* $\Pi' = \Pi_0^{\texttt{know}}$*,* $\Pi'' = \Pi_1^{\texttt{know}}$*,* $\gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ *be a fixed polynomial in* λ*,* $n = \max\{|\mathbf{M}|, s + t\}$ *where* $|\mathbf{M}|$ *denote the number of non-zero entries in* $\mathbf{M}$*,* $(q_0, q_1, q_3) = (s, s^2, (s + t)^{16}) \cdot \mathsf{poly}(\lambda)$ *and other internal parameters of* $\Pi_0^{\texttt{know}}$ *and* $\Pi_1^{\texttt{know}}$ *be chosen as in Theorems 9 and 12.* $\Pi^{\texttt{bin-sat}}$ *has (i) common reference string size* $O_\lambda(n \cdot \log n)$*, (ii) proof size* $O_\lambda(\log^2 n)$*, (iii) prover time* $O_\lambda(n \cdot \log^3 n)$*, (iv) preprocessing time* $O_\lambda(n \cdot \log^2 n)$*, and (v) verifier time* $O_\lambda(\log^3 n)$ *after preprocessing.*

8.2 Rank-1 Constraint Systems

We show how to use the same ideas to construct an argument of knowledge, $\Pi_{\texttt{R1CS}}$, for the satisfiability of Rank-1 Constraint Systems. Formally, define the relation $\Psi^{\texttt{R1CS}} = \Psi^{\texttt{R1CS}}[\mathcal{R}, t, s_1, s_2, q_0, \alpha]$:

$$\Psi^{\texttt{R1CS}} := \{((\mathbf{x}_1, \mathbf{E}, \mathbf{F}, \mathbf{G}), \mathbf{x}_2) : (\mathbf{E} \cdot \mathbf{x}) \circ (\mathbf{F} \cdot \mathbf{x}) = \mathbf{G} \cdot \mathbf{x} \bmod q_0 \ \wedge \ \|\mathbf{x}\| \leq \alpha\},$$

where $\mathbf{x} := (\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{R}^{s_1} \times \mathcal{R}^{s_2}$, $\mathbf{E}, \mathbf{F}, \mathbf{G} \in \mathcal{R}_{q_0}^{t \times s}$, and $s = s_1 + s_2$. If we let $\mathbf{e} := \mathbf{E} \cdot \mathbf{x}$, $\mathbf{f} := \mathbf{F} \cdot \mathbf{x}$, and $\mathbf{g} := \mathbf{G} \cdot \mathbf{x}$, the above equation can be rewritten as $\mathbf{e} \circ \mathbf{f} + q_0 \cdot \mathbf{r} = \mathbf{g}$, for some $\mathbf{r} \in \mathcal{R}^t$. For readability, we informally describe here how the argument system works. A formal description of $\Pi^{\texttt{R1CS}}$ can be found in Fig. 4 in Appendix H.

In $\Pi_{\texttt{R1CS}}$, the prover proves that they know witnesses to the following relations

$$\mathbf{v}_2^{\mathsf{T}} \cdot \mathbf{x}_2 = c_{\mathbf{x}_2} \bmod q_3, \quad \text{and} \quad \|\mathbf{x}_2\| \leq \alpha, \tag{6}$$

$$\begin{pmatrix} (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathsf{T}} \cdot \mathbf{E} \\ \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{F} \\ \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{G} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \bar{c}_{\mathbf{e}} \\ c_{\mathbf{f}} \\ c_{\mathbf{g}} \end{pmatrix} \bmod q_3, \quad \text{and} \quad \left\|\left(\mathbf{x}_2\right)\right\| \leq \alpha, \tag{7}$$

$$\left(\bar{\mathbf{v}}^{\mathsf{T}} \| \mathbf{v}^{\mathsf{T}}\right) \cdot \mathbf{z} = c_{\mathbf{z}} \bmod q_3, \quad \text{and} \quad \|\mathbf{z}\| \leq \alpha', \tag{8}$$

where $\mathbf{v}_2 = (v^{s_1+1}, \ldots, v^s)$, $\mathbf{h} \in \mathcal{R}_{q_1}^t$, and $\mathbf{z} = (z_k)_{k \in \pm[s]}$, $z_k = \sum_{i,j,i-j=k} h_j \cdot e_j \cdot f_i + q_0 \cdot h_j \cdot r_i - g_i \cdot h_j$, $c_{\mathbf{z}} = \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot c_{\mathbf{r}} \cdot \bar{c}_{\mathbf{I}} - c_{\mathbf{g}} \cdot \bar{c}_{\mathbf{I}}$, and $c_{\mathbf{r}} = \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{r}$.

The prover will prove that $c_{\mathbf{x}_2}$ is well-formed, i.e. relation in Eq. (6), by proving knowledge of a short opening of the commitment $c_{\mathbf{x}_2}$ with respect to the commitment key $(v^i)_{i \in [s_1+1;s]}$. To prove consistency between $c_{\mathbf{x}_2}$ and $\bar{c}_{\mathbf{e}}$, the prover proves knowledge of a short opening of the commitment $\bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I}} \cdot \bar{c}_{\mathbf{e}}$ where $c_{\mathbf{x}} := c_{\mathbf{x}_1} + c_{\mathbf{x}_2}$, and the values $c_{\mathbf{x}_1} := \mathbf{v}_1^{\mathsf{T}} \cdot \mathbf{x}_1$, $\bar{c}_{\mathbf{E}}$, and $c_{\mathbf{I}}$ can be precomputed by the verifier. This with respect to the commitment key $(v^{i-j})_{i-j=k, k \in \pm[s]}$. Proofs of consistency between $c_{\mathbf{x}_2}$ and $c_{\mathbf{f}}$, $c_{\mathbf{x}_2}$ and $c_{\mathbf{g}}$ are obtained similarly. This suffices to prove the relation in Eq. (7).

Finally, to prove that $\mathbf{e} \circ \mathbf{f} = \mathbf{g} \bmod q_0$, i.e. relation in Eq. (8), the prover will prove knowledge of a short opening of the commitment

$$c_{\mathbf{z}} = \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I}} \cdot c_{\mathbf{r}} - c_{\mathbf{g}} \cdot \bar{c}_{\mathbf{I}}$$

again with respect to the commitment key $(v^{i-j})_{i-j=k, k \in \pm[s]}$.

Analysis. In Appendix H we show that $\Pi^{\texttt{R1CS}}$ is correct and knowledge-sound under (knowledge-)k-R-ISIS and R-SIS assumptions. We further show that $\Pi^{\texttt{R1CS}}$ has short CRS and proofs, quasi-linear-time prover and preprocessing, and polylogarithmic-time verifier after preprocessing. For readability, we defer formal claims and relative proofs to Appendix H.2, and Appendix H.3.

References

ACK21. Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed Σ-protocol theory for lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 549–579, Virtual Event, August 2021. Springer, Heidelberg. 1, 3, 6

ACL$^+$22. Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, and Sri Aravinda Krishnan Thyagarajan. Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 102–132. Springer, Heidelberg, August 2022. 1, 2, 3, 4, 5, 6, 7, 9, 10, 13, 17, 19, 25

AF22. Thomas Attema and Serge Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 415–443. Springer, Heidelberg, August 2022. 6, 11

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996. 9

AL21. Martin R. Albrecht and Russell W. F. Lai. Subtractive sets over cyclotomic rings - limits of Schnorr-like arguments over lattices. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 519–548, Virtual Event, August 2021. Springer, Heidelberg. 1, 3, 6, 31

Ano23. Anonymous. Lattice-based functional commitments: Fast verification and cryptanalysis. private communication, May 2023. 4

BBB$^+$18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018. 3

BCC$^+$16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016. 3

BCFL22. David Balbás, Dario Catalano, Dario Fiore, and Russell W. F. Lai. Functional commitments for circuits from falsifiable assumptions. Cryptology ePrint Archive, Report 2022/1365, 2022. https://eprint.iacr.org/2022/1365. 3, 4, 8, 19

BCG$^+$14. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014. 1

BCG$^+$19. Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, and Nicholas Spooner. Linear-size constant-query IOPs for delegating computation. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 494–521. Springer, Heidelberg, December 2019. 8, 46

BCTV14. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 276–294. Springer, Heidelberg, August 2014. 3

BDFG21. Dan Boneh, Justin Drake, Ben Fisch, and Ariel Gabizon. Halo infinite: Proof-carrying data from additive polynomial commitments. In *Annual International Cryptology Conference*, pages 649–680. Springer, 2021. 1

BGH19. Sean Bowe, Jack Grigg, and Daira Hopwood. Halo: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Report 2019/1021, 2019. https://eprint.iacr.org/2019/1021. 1

BISW17. Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Lattice-based SNARGs and their application to more efficient obfuscation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *EUROCRYPT 2017, Part III*, volume 10212 of *LNCS*, pages 247–277. Springer, Heidelberg, April / May 2017. 1

BISW18. Dan Boneh, Yuval Ishai, Amit Sahai, and David J. Wu. Quasi-optimal SNARGs via linear multi-prover interactive proofs. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 222–255. Springer, Heidelberg, April / May 2018. 1

BLNS20. Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 441–469. Springer, Heidelberg, August 2020. 1, 2, 3, 6, 15

BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 176–202. Springer, Heidelberg, August 2019. 3

BMM$^+$21. Benedikt Bünz, Mary Maller, Pratyush Mishra, Nirvan Tyagi, and Psi Vesely. Proofs for inner pairing products and applications. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 65–97. Springer, Heidelberg, December 2021. 3

BMRS20. Joseph Bonneau, Izaak Meckler, Vanishree Rao, and Evan Shapiro. Coda: Decentralized cryptocurrency at scale. *Cryptology ePrint Archive*, 2020. 1

CMS19. Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 1–29. Springer, Heidelberg, December 2019. 3

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, December 2020. 3

GL96. Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, USA, 1996. 37

GM17. Matthew Green and Ian Miers. Bolt: Anonymous payment channels for decentralized currencies. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 473–489. ACM Press, October / November 2017. 1

GM18. Nicholas Genise and Daniele Micciancio. Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 174–203. Springer, Heidelberg, April / May 2018. 9

GMNO18. Rosario Gennaro, Michele Minelli, Anca Nitulescu, and Michele Orrù. Lattice-based zk-SNARKs from square span programs. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 556–573. ACM Press, October 2018. 1, 3

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008. 9

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. 3

ISW21. Yuval Ishai, Hang Su, and David J. Wu. Shorter and faster post-quantum designated-verifier zkSNARKs from lattices. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 212–234. ACM Press, November 2021. 3

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *24th ACM STOC*, pages 723–732. ACM Press, May 1992. 1, 3

KMS$^+$16. Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy*, pages 839–858. IEEE Computer Society Press, May 2016. 1

LM23. Russell W. F. Lai and Giulio Malavolta. Lattice-based timed-cryptography. In *CRYPTO 2023*, 2023. 3, 8

LMR19. Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. Succinct arguments for bilinear group arithmetic: Practical structure-preserving cryptography. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2057–2074. ACM Press, November 2019. 3

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Heidelberg, August 2022. 3

Mic94. Silvio Micali. CS proofs (extended abstracts). In *35th FOCS*, pages 436–453. IEEE Computer Society Press, November 1994. 1, 3

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, Heidelberg, April 2012. 9

Pie19. Krzysztof Pietrzak. Simple verifiable delay functions. In Avrim Blum, editor, *ITCS 2019*, volume 124, pages 60:1–60:15. LIPIcs, January 2019. 7, 15

PS21. Alice Pellet-Mary and Damien Stehlé. On the hardness of the NTRU problem. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part I*, volume 13090 of *LNCS*, pages 3–35. Springer, Heidelberg, December 2021. 26

WW23. Hoeteck Wee and David J. Wu. Succinct vector, polynomial, and functional commitments from lattices. In *EUROCRYPT 2023*, 2023. To appear. 4

YAZ$^+$19. Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 147–175. Springer, Heidelberg, August 2019. 3

A Relating Vanishing-SIS and other Assumptions

In the following, we show that the vanishing-SIS assumption is implied by, and tightly related to, the k-R-ISIS assumption. We also explore the connections between the vanishing-SIS, NTRU, and RingLWE assumptions.

A.1 Relations with k-R-ISIS

We discuss how the vSIS assumption relates to the k-R-ISIS family of assumptions defined in [ACL+22]. We show implications in both directions that hold in different parameter regimes.

k-R-ISIS $\implies$ vSIS. We show that $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ is no easier than k-R-$\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ with $\beta^* = |\mathcal{G}|\cdot\alpha\cdot\beta\cdot\gamma_{\mathcal{R}}$. Assuming that we have a solver for $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ that outputs a polynomial p with

$$p(\mathbf{v}) = \sum_{g\in\mathcal{G}} p_g \cdot g(\mathbf{v}) + p_{g^*}\cdot g^*(\mathbf{v}) = 0 \bmod q,$$

we can construct an algorithm solving k-R-$\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ as follows. The algorithm is given as input $\left(\mathbf{D},\mathbf{t},\mathbf{v},\{\mathbf{u}_g\}_{g\in\mathcal{G}}\right)$. It runs the $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha}$ solver on $\mathbf{v}$ to obtain p, and returns

$$\mathbf{u}_{g^*} = \sum_{g\in\mathcal{G}} p_g\cdot\mathbf{u}_g \text{ and } s^* = -p_{g^*}.$$

Note that this is a valid solution for k-R-$\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ since

$$\mathbf{D}\cdot\mathbf{u}_{g^*} = \mathbf{D}\cdot\sum_{g\in\mathcal{G}} p_g\cdot\mathbf{u}_g = \sum_{g\in\mathcal{G}} p_g\cdot g(\mathbf{v}) = -p_{g^*}\cdot g^*(\mathbf{v}) \bmod q$$

and furthermore, by assumption, we have that $\|p\|\le\alpha$ and thus

$$\|\mathbf{u}_{g^*}\| \le |\mathcal{G}|\cdot\alpha\cdot\beta\cdot\gamma_{\mathcal{R}} = \beta^* \qquad \text{and} \qquad \|p_{g^*}\|\le\alpha<\beta^*.$$

Knowledge k-R-ISIS and vSIS $\implies$ k-R-ISIS. We show that, conditioned on knowledge k-R-$\mathsf{ISIS}_{\mathcal{G},\alpha^*,\beta,\beta^*}$, the hardness of $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\},\alpha^*}$ implies that of k-R-$\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$, for $\alpha^*\ge\beta^*$. At first glance, it may appear strange that we are using the knowledge version of k-R-ISIS to prove k-R-ISIS itself. Nevertheless the statement is meaningful, since knowledge k-R-ISIS is not a computational problem, but rather an assumption about the attacker itself. In some sense, this statement shows that vSIS is the underlying computational assumption that connects k-R-ISIS and knowledge k-R-ISIS. We sketch the reduction in the following. Assume that we are given a k-R-$\mathsf{ISIS}_{\mathcal{G},g^*,\beta,\beta^*}$ solver that, on input $\left(\mathbf{D},\mathbf{t},\mathbf{v},\{\mathbf{u}_g\}_{g\in\mathcal{G}}\right)$, outputs $(\mathbf{u}_{g^*},s^*)$ such that

$$\mathbf{D}\cdot\mathbf{u}_{g^*} = \mathbf{t}\cdot s^*\cdot g^*(\mathbf{v}) \qquad \text{and} \qquad \|(\mathbf{u}_{g^*},s^*)\|\le\beta^*\le\alpha^*.$$

By knowledge k-R-$\mathsf{ISIS}_{\mathcal{G},\alpha^*,\beta,\beta^*}$, there exists an extractor that returns $\{x_g\}_{g\in\mathcal{G}}$ such that

$$s^*\cdot g^*(\mathbf{v}) = \sum_{g\in\mathcal{G}} x_g\cdot g(\mathbf{v}) \qquad \text{and} \qquad \|x_g\|\le\alpha^*.$$

It follows that $p=\sum_{g\in\mathcal{G}} x_g\cdot g - s^*\cdot g^*$ is a valid solution for $\mathsf{vSIS}_{\mathcal{G}\cup\{g^*\}}$ since

$$\sum_{g\in\mathcal{G}} x_g\cdot\mathbf{u}_g - s^*\cdot g^*(\mathbf{v}) = p(\mathbf{v}) = 0 \bmod q \qquad \text{and} \qquad \|p\|\le\alpha^*.$$

A.2 Relations with NTRU

In the following, we study the relation between vSIS and the search NTRU assumption, conditioned on the decision NTRU assumption. Recall that the decision NTRU assumption states that the "NTRU distribution", i.e. that of $h=f/g \bmod q\in\mathcal{R}_q^\times$ where $f,g \leftarrow\$ \mathcal{R}$ are random short elements, is indistinguishable from the uniform distribution over $\mathcal{R}_q^\times$. From the decision NTRU assumption, we immediately have that the distribution of vSIS instances is indistinguishable from a modified version where each entry of each point $\mathbf{v}\in V$ is sampled from the NTRU distribution instead of uniformly from $\mathcal{R}_q^\times$. In the following, we refer to this modified version of vSIS as NTRU-vSIS.

Univariate NTRU-vSIS $\implies$ Search NTRU. First, we make a simple observation that NTRU-vSIS generalises search NTRU. The search NTRU problem is the following: Given h sampled from the NTRU distribution, find short f', g' such that that $g'h + f' = 0 \bmod q$, i.e. find a degree-1 polynomial p with short coefficients which vanishes at h modulo q. Clearly, a search NTRU solver also solves NTRU-$\mathsf{vSIS}_{n=1,\mathcal{G}}$ if $\{1, X\} \subseteq \mathcal{G}$.

Solution Space. The search NTRU problem can be viewed (see e.g. [PS21]) as the problem of finding a short vector spanned by $\begin{pmatrix} q & -h \\ & 1 \end{pmatrix}$. Similarly, the $\mathsf{vSIS}_{(n,w)=(1,1)}$ problem can be viewed as the problem of finding a short vector in the rank-$(d+1)$ module lattice spanned by $\begin{pmatrix} q & \begin{bmatrix} -v \\ 1 \end{bmatrix}_{\searrow d} \end{pmatrix}$. In [PS21], Pellet-Mary and Stehlé showed that all solutions to a search NTRU problem lie in a unique rank-1 submodule of the module-lattice spanned by $(-f, g)^{\intercal}$.[18] Similarly, we can show that, for large enough q (exponential in d), all solutions to an NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d}$ problem lie in a unique rank-d submodule. The argument roughly goes as follows.

Consider $v = f/g \bmod q$ where $\|f\|, \|g\| \leq \alpha$, and let $\mathbf{p}$ be a solution to the NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d}$ instance v. We have $\sum_{j=0}^{d} p_j \cdot v^j = 0 \bmod q$ or equivalently $\sum_{j=0}^{d} p_j \cdot f^j \cdot g^{d-j} = 0 \bmod q$. Assuming that $q > 2 \cdot (d+1) \cdot \alpha^d \cdot \beta \cdot \gamma_{\mathcal{R}}^d$, we have $\sum_{j=0}^{d} p_j \cdot f^j \cdot g^{d-j} = 0$, with arithmetic done over $\mathcal{K}$. In other words, the solution lies in the kernel of $(g^d, f \cdot g^{d-1}, \ldots, f^d)$ for which a basis is given by $\begin{bmatrix} -f \\ g \end{bmatrix}_{\searrow d}$.

Decision NTRU + Worst-Case $\implies$ Average-Case. The $\mathsf{vSIS}_{n=1}$ problem admits a worst-case to average-case reduction, conditioned on the hardness of decision NTRU. Note that this reduction produces a solution of norm exponential in d and w. In the following, we sketch the reduction.

Let $\mathbf{v}^*$ be any fixed $\mathsf{vSIS}_{n=1,\beta^*}$ instance for some β^* to be specified later. For each $j \in [w]$, sample an NTRU element $h_j = f_j/g_j \bmod q$ where $\|f_j\|, \|g_j\| \leq \alpha$ for all j. Define $\mathbf{v}$ where $v_j \coloneqq v_j^* \cdot h_j \bmod q$. Note that $\mathbf{v} \circ \mathbf{g} = \mathbf{v}^* \circ \mathbf{f}$ where $\circ$ denotes the Hadamard product. By the decision NTRU assumption, $\mathbf{v}$ is indistinguishable from a random $\mathsf{vSIS}_{n=1,\beta}$ instance. Suppose p is a solution to the vSIS_{β} instance $\mathbf{v}$, i.e. $p(\mathbf{v}) = 0 \bmod q$ and $\|p\| \leq \beta$, then

$$\prod_{j=1}^{w} g_j^d \cdot p(\mathbf{v}) = \prod_{j=1}^{w} g_j^d \cdot p\left(\mathbf{v}^* \circ \left(\frac{f_1}{g_1}, \ldots, \frac{f_w}{g_w}\right)\right) = 0 \bmod q.$$

Note that $\prod_{j=1}^{w} g_j^d \cdot p\left(\mathbf{v}^* \circ \left(\frac{f_1}{g_1}, \ldots, \frac{f_w}{g_w}\right)\right)$ can be seen as a polynomial with coefficients in $\mathcal{R}$ evaluated at $\mathbf{v}^*$ (since all denominators are cancelled out). Denote this polynomial by p^*. We have $p^*(\mathbf{v}^*) = 0 \bmod q$. Furthermore, notice that $\|p^*\| \leq \|p\| \cdot \alpha^{d \cdot w} \cdot \gamma_{\mathcal{R}}^{d \cdot w} = \alpha^{d \cdot w} \cdot \beta \cdot \gamma_{\mathcal{R}}^{d \cdot w}$. Therefore p^* is a solution to the $\mathsf{vSIS}_{n=1,\beta^*}$ instance $\mathbf{v}^*$ with $\beta^* = \alpha^{d \cdot w} \cdot \beta \cdot \gamma_{\mathcal{R}}^{d \cdot w}$.

vSIS, NTRU, and RingLWE. It is clear that the $\mathsf{vSIS}_{n=1}$ problem can be reduced to the vSIS problem (with the same parameters except that n is changed from 1 to an arbitrary polynomial). In the following, we sketch a reduction from the search NTRU problem to the $\mathsf{vSIS}_{(n,w)=(1,1),d}$ problem, conditioned on the hardness of either decision NTRU or RingLWE. We note that this reduction could only work for a certain extreme parameter regime which is not suitable for our application of succinct arguments.

Using Decision NTRU. Given a NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ solver for some β', we would like to find a solution to a random NTRU instance v of norm bounded by some β^*. Interpreting v as an NTRU-$\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance, using the decision-NTRU rerandomisation technique in the above worst-case to average-case reduction, we can rerandomise v to d $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ instances v_i for $i \in [d]$, where $\beta \coloneqq \alpha^d \cdot \beta' \cdot \gamma_{\mathcal{R}}^d$. Let p_i' be a solution of norm β' to the i-th instance v_i. Using the transformation as in the above worst-case to average-case reduction, we can obtain d solutions $(p_i)_{i=1}^{d}$ to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$

[18] Recovering this submodule (represented by a possibly long vector) was formalised as the $\mathrm{NTRU}_{\mathrm{mod}}$ problem in [PS21]. This variant of the search NTRU problem is trivially not harder than the standard variant, and [PS21] gave a reduction from the decision NTRU problem.

instance v. Note that each p_i is a degree-d polynomial of norm β. Recursively running the following algorithm produces a degree-1 polynomial of norm $(2\gamma_{\mathcal{R}})^{2^{d-1}-1} \cdot \beta^{2^{d-1}} < (2\beta\gamma_{\mathcal{R}})^{2^{d-1}} =: \beta^*$:

- Input: $L = (p_0, p_1, \ldots, p_{d-1})$, degree-$d$ polynomials each of norm β.
- Output: $L' = (p'_1, \ldots, p'_{d-1})$, degree-$(d-1)$ polynomials each of norm $2\beta^2\gamma_{\mathcal{R}}$.
- Procedure:
  - For $0 \leq i < d$, let a_i be the coefficient of the degree-d term in p_i.
  - Output $L' = (p'_1, \ldots, p'_i)$ where $p'_i = a_0 \cdot p_i - a_i \cdot p_0$.

If we could argue that $(p_1, \ldots, p_d)$ are linearly independent (as polynomials over $\mathcal{K}$) and set $q \gg \beta^*$, then the above gives a solution of norm β^* to the NTRU instance v. In particular, if $\beta^* \ll \sqrt{q}$, then a search NTRU solver would also solve decision NTRU, contradicting the initial assumption that decision NTRU holds. We therefore obtain a reduction from decision NTRU to NTRU-vSIS. Note that for this reduction to work it is necessary to have q being doubly-exponential in the number of monomials $d+1$, which forces d to be constant.

Using RingLWE. Instead of using decision NTRU for rerandomisation, we could use RingLWE by exploiting the fact that we start with a random search NTRU instance v.[19] Specifically, we can rerandomise v to $v_i := v \cdot s_i + e_i \bmod q$ for $\|s_i\|, \|e_i\| \leq \alpha$, provided that we reject those v_i which are not invertible. By the (normal-form) RingLWE assumption, each v_i is indistinguishable from a random $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance. Suppose p'_i is a solution to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta'}$ instance v_i, then

$$p'_i(v_i) = p'_i(v \cdot s_i + e_i)$$

meaning that $p_i(X) = p'_i(X \cdot s_i + e_i)$ is a solution to the $\mathsf{vSIS}_{(n,w)=(1,1),d,\beta}$ instance v, where now $\beta = d \cdot \alpha^d \cdot \beta' \cdot \gamma_{\mathcal{R}}^d$. The rest then follows similar to the reduction using decision NTRU.

B Proofs for Foldable Structures

B.1 Proof of Lemma 1

Proof. By the definition of a foldable sequence, $n = \sum_{i=0}^{\ell} 2^i \cdot k_i$. Since $k_\ell \geq 1$, $k_i \leq k^*$ for all $i \in \{0, \ldots, \ell\}$, and $\sum_{i=0}^{\ell} 2^i = 2^{\ell+1} - 1$, we can derive $2^\ell \leq n < k^* \cdot 2^{\ell+1}$. The claim then follows. □

B.2 Power Sequence - Proof of Lemma 4

Proof. For the first claim, it suffices to show that the sequence of monomials $\mathbf{m} = (X, X^2, \ldots, X^n)$ in variable X is $(k_0, k_1, \ldots, k_\ell)$-foldable, and realise that $\mathbf{v}$ can be obtained by evaluating $\mathbf{m}$ at the point v. We construct a seed and a generator of $\mathbf{m}$ recursively as follows. Define a procedure, which on input a seed $\mathbf{m} = (X, X^2, \ldots, X^k)$ of length k and a generator $\mathbf{g} = \epsilon$, does the following:

- If $k \leq 2$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 2$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1})$, and $r' = X^{k'+1}$.
- If k is even (hence $k \geq 4$), write $k = 2 \cdot k' + 2$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1}, X^{k'+2})$, and $r' = X^{k'+2}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\mathbf{m}, \epsilon)$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters.

For the generalised claim, we similarly define a procedure, which on input a seed $\mathbf{m} = (X, X^2, \ldots, X^{wk})$ of length wk and a generator $\mathbf{g} = \epsilon$, does the following:

- If $k \leq 2$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 2$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{wk'})$, $\ell' = 1$, $\mathbf{c}' = (X^{wk'+1}, \ldots, X^{wk'+w})$, and $r' = X^{wk'+w}$.

[19] We could not do this in the worst-case to average-case reduction where v was fixed.

- If k is even (hence $k \geq 4$), write $k = 2 \cdot k' + 2$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{wk'})$, $\ell' = 1$, $\mathbf{c}' = (X^{wk'+1}, \ldots, X^{wk'+2w})$, and $r' = X^{wk'+2w}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\mathbf{m}, \epsilon)$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters. □

B.3 Balanced Power Sequence - Proof of Lemma 5

Proof. It suffices to show that the sequence of monomials

$$\mathbf{m} = (X^{-n}, \ldots, X^{-2}, X^{-1}, X, X^2, \ldots, X^n)$$

in variable X is $(0, k_0, k_1, \ldots, k_\ell)$-foldable, and realise that $\mathbf{v}$ can be obtained by evaluating $\mathbf{m}$ at the point v.

Let $\hat{\mathbf{m}} = (X, X^2, \ldots, X^n)$, $\hat{\ell} = X^{-(n+1)}$ $\hat{\mathbf{c}} = \epsilon$ the empty vector, and $\hat{r} = 1$. Let $\hat{\mathbf{g}} = (\hat{l}, \hat{\mathbf{c}}, \hat{r})$. Clearly, $\hat{\mathbf{m}}$ is foldable with seed $\hat{\mathbf{m}}$ and generator $\hat{\mathbf{g}}$.

We next construct a generator of $\hat{\mathbf{m}}$ recursively as follows. Define a procedure, which on input a sequence $\mathbf{m}$ of length k and possibly partial generator $\mathbf{g}$, does the following:

- If $k = 1$, output $(\mathbf{m}, \mathbf{g})$.
- If $k > 1$ is odd (hence $k \geq 3$), write $k = 2 \cdot k' + 1$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = (X^{k'+1})$, and $r' = X^{k'+1}$.
- If k is even (hence $k \geq 2$), write $k = 2 \cdot k'$. Let $\mathbf{m}' = (X, X^2, \ldots, X^{k'})$, $\ell' = 1$, $\mathbf{c}' = \epsilon$ the empty vector, and $r' = X^{k'}$.
- Let $\mathbf{g}' = (l', \mathbf{c}', r') \| \mathbf{g}$.
- Run the procedure on $(\mathbf{m}', \mathbf{g}')$.

It is easy to observe that running the above procedure on $(\hat{\mathbf{m}}, \hat{\mathbf{g}})$ finds a seed and a generator of $\mathbf{m}$ with the desired parameters. □

B.4 Compression Vector - Proof of Lemma 6

Proof. To show that $\mathbf{m}_\ell \coloneqq \mathbf{m}$ is $(k_0, k_1, \ldots, k_\ell)$-foldable, it suffices to show that $\mathbf{m}_0, \ldots, \mathbf{m}_\ell$ induced by the given seed and generator as described in the procedure of Definition 11 each consists of distinct monomials.

By construction, $\mathbf{m}_\ell = (X_{\ell,1}, \ldots, X_{\ell,k_\ell})$ consists of distinct monomials. Suppose $\mathbf{m}_i$ consists of distinct monomials. Consider

$$\mathbf{m}_{i-1}^{\mathtt{T}} = (\mathbf{m}_i^{\mathtt{T}}, X_{i-1,1}, \ldots, X_{i-1,k_{i-1}}, X_{i-1,0} \cdot \mathbf{m}_i^{\mathtt{T}}).$$

By construction, none of the monomials in $\mathbf{m}_i^{\mathtt{T}}$ is a multiple of $X_{i-1,j}$ for any $j \in \{0, \ldots, k_{i-1}\}$. Therefore $\mathbf{m}_{i-1}$ consists of distinct monomials. The claim thus follows from induction.

Finally, the norm bound of $\mathbf{h}$ follows from the observation that each entry of $\mathbf{h}$ is a product of at most $\ell + 1$ entries of $\mathbf{x}$. □

C Folding Argument for Type-1 Linear Relations

The protocol $\mathit{\Pi}_1^{\mathtt{fold}}.\langle \mathsf{Prove}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit}), \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathsf{on}})\rangle$ consists of $\ell + 1$ rounds and makes use of the subtractive set $S \subset \mathcal{R}^\times$ mentioned in Section 3.1. Denote

$$(\mathbf{A}^{(0)}, \mathbf{x}^{(0)}, \mathbf{y}^{(0)}, \alpha^{(0)}) \coloneqq (\mathbf{A}, \mathbf{x}, \mathbf{y}, \alpha).$$

Let $n = \sum_{j=0}^{\ell} 2^j \cdot k_j$ be the binary representation of n, where $k_j \in \{0, 1\}$. For $i \in \{0, \ldots, \ell\}$, define $n_i \coloneqq \sum_{j=i}^{\ell} 2^{j-i} \cdot k_j$. Then, for $i < \ell$, the i-th round of the protocol is as follows:

- Parse
  - $\mathbf{A}^{(i)}$ as $(\mathbf{A}_L^{(i)}, \mathbf{A}_c^{(i)}, \mathbf{A}_R^{(i)})$, and

- $\mathbf{x}^{(i)}$ as $(\mathbf{x}_L^{(i)}, \mathbf{x}_c^{(i)}, \mathbf{x}_R^{(i)})$

  where $\mathsf{ncol}(\mathbf{A}_L^{(i)}) = \mathsf{ncol}(\mathbf{A}_R^{(i)}) = \mathsf{nrow}(\mathbf{x}_L^{(i)}) = \mathsf{nrow}(\mathbf{x}_R^{(i)}) = n_{i+1} \cdot w$. Note that $\mathsf{ncol}(\mathbf{A}_c^{(i)}) = k_i \cdot w$, meaning that $\mathbf{A}_c^{(i)}$ and $\mathbf{x}_c^{(i)}$ are empty when $k_i = 0$.

– $\mathcal{P}$ sends
  - $\mathbf{x}_c^{(i)}$ (if $k_i > 0$),
  - $\mathbf{y}_{LR}^{(i)} := \mathbf{A}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \bmod q$, and
  - $\mathbf{y}_{RL}^{(i)} := \mathbf{A}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \bmod q$.

– $\mathcal{V}$ samples $r_i \leftarrow\$ S$ and sends r_i to $\mathcal{P}$.

– $\mathcal{P}$ computes the compressed witness

$$\mathbf{x}^{(i+1)} := \mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i$$

– $\mathcal{P}$ and $\mathcal{V}$ compute the compressed statement

$$\begin{aligned} \mathbf{A}^{(i+1)} &:= \mathbf{A}_L^{(i)} + \mathbf{A}_R^{(i)} \cdot r_i^{-1} \bmod q, \\ \mathbf{y}^{(i+1)} &:= \mathbf{y}^{(i)} - \mathbf{A}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{y}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{y}_{LR}^{(i)} \cdot r_i \bmod q \\ \alpha^{(i+1)} &:= 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}}. \end{aligned}$$

In the ℓ-th (i.e. final) round, $\mathcal{P}$ sends $\mathbf{x}^{(\ell)}$ and $\mathcal{V}$ checks that

$$\mathbf{A}^{(\ell)} \cdot \mathbf{x}^{(\ell)} = \mathbf{y}^{(\ell)} \bmod q \qquad \text{and} \qquad \left\|\mathbf{x}^{(\ell)}\right\| \leq \alpha^{(\ell)}.$$

D Proofs for Folding Arguments

D.1 Completeness - Proof of Theorem 1

Proof. The case where $n \leq 2$ is trivial. For $n > 2$, it is clear that for each i the checks $\left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)}$ and, if $k_i = 2$, $\begin{pmatrix}\mathbf{B} & \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} = \mathbf{y}_c^{(i)} \bmod q_0$, pass. It remains to show that, for each i, if

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} &= \mathbf{y}^{(i)} \bmod q_0, \\ \mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} &= \mathbf{z}^{(i)} \bmod q_1, \end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(i)}\right\| \leq \alpha^{(i)},$$

then

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}^{(i+1)} &= \mathbf{y}^{(i+1)} \bmod q_0, \\ \mathbf{C}^{(i+1)} \cdot \mathbf{x}^{(i+1)} &= \mathbf{z}^{(i+1)} \bmod q_1, \end{aligned} \qquad \text{and} \qquad \left\|\mathbf{x}^{(i+1)}\right\| \leq \alpha^{(i+1)}.$$

First, by $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_i}\right) \cdot \mathbf{x}^{(i)} = \mathbf{y}^{(i)} \bmod q_0$ we have

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_L^{(i)} + \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} &= \mathbf{y}_L^{(i)} \bmod q_0, \\ \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} + \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_R^{(i)} &= \mathbf{y}_R^{(i)} \bmod q_0. \end{aligned}$$

It follows that

$$\begin{aligned} \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}^{(i+1)} &= \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot (\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i) \\ &= \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0}\\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix}\mathbf{B}\\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \cdot r_i \end{aligned}$$

$$= \mathbf{y}^{(i+1)} \bmod q_0.$$

Next, by $\mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} = \mathbf{z}^{(i)} \bmod q_1$ we have

$$\mathbf{C}_L^{(i)} \cdot \mathbf{x}_L^{(i)} + \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_R^{(i)} = \mathbf{z}^{(i)} \bmod q_1.$$

Therefore

$$\begin{aligned}\mathbf{C}^{(i+1)} \cdot \mathbf{x}^{(i+1)} &= (\mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1}) \cdot (\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i) \\ &= \mathbf{C}_L^{(i)} \cdot \mathbf{x}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_R^{(i)} + \mathbf{C}_R^{(i)} \cdot \mathbf{x}_L^{(i)} \cdot r_i^{-1} + \mathbf{C}_L^{(i)} \cdot \mathbf{x}_R^{(i)} \cdot r_i \\ &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i \\ &= \mathbf{z}^{(i+1)} \bmod q_1.\end{aligned}$$

Finally, since $\left\|\mathbf{x}^{(i)}\right\| \leq \alpha^{(i)}$, it follows that $\left\|\mathbf{x}^{(i+1)}\right\| = \left\|\mathbf{x}_L^{(i)} + \mathbf{x}_R^{(i)} \cdot r_i\right\| \leq 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}} = \alpha^{(i+1)}$. □

D.2 Special Soundness - Proof of Theorem 2

Proof. The case of $n \leq 2$ is trivial. Recall that $\alpha^{(i)} = (2\gamma_{\mathcal{R}})^i \cdot \alpha$. Let $\hat{\alpha}^{(\ell)} = \alpha^{(\ell)} = (2\gamma_{\mathcal{R}})^\ell \cdot \alpha$. For $i \in \{0, \ldots, \ell-1\}$, define $\hat{\alpha}^{(i)} = 4\gamma_{\mathcal{R}}^3 \cdot \hat{\alpha}^{(i+1)}$, so that $\hat{\alpha}^{(0)} = (4\gamma_{\mathcal{R}}^3)^\ell \cdot \hat{\alpha}^{(\ell)} = (8\gamma_{\mathcal{R}}^4)^\ell \cdot \alpha$. In the following, assume that $n > 2$. We need to show that if $(\mathbf{x}_c^{(i)}, \mathbf{z}_{LR}^{(i)}, \mathbf{z}_{RL}^{(i)}, \mathbf{x}_0^{(i+1)}, \mathbf{x}_1^{(i+1)}, \mathbf{x}_2^{(i+1)})$ satisfies

$$\begin{aligned}(\mathbf{B}\ \mathbf{A}) \cdot \mathbf{x}_c^{(i)} &= \mathbf{y}_c^{(i)} \bmod q_0 \text{ if } k_i = 2, & & \left\|\mathbf{x}_c^{(i)}\right\| \leq \alpha^{(i)}, \\ \begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} \cdot \mathbf{x}_j^{(i+1)} &= \mathbf{y}_j^{(i+1)} \bmod q_0, & \text{and} \quad & \left\|\mathbf{x}_j^{(i+1)}\right\| \leq \hat{\alpha}^{(i+1)}, \\ \mathbf{C}_j^{(i+1)} \cdot \mathbf{x}_j^{(i+1)} &= \mathbf{z}_j^{(i+1)} \bmod q_1, & &\end{aligned}$$

where

$$\begin{aligned}\mathbf{C}_j^{(i+1)} &= \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_{i,j}^{-1} \bmod q_1, \\ \mathbf{y}_j^{(i+1)} &= \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_{i,j} - \begin{pmatrix}\mathbf{B} \cdot r_{i,j} \\ \mathbf{0} \\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0 \\ \mathbf{z}_j^{(i+1)} &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_{i,j}^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_{i,j} \bmod q_1\end{aligned}$$

for distinct challenges $r_{i,0}, r_{i,1}, r_{i,2} \in S$, then we can extract $\mathbf{x}^{(i)}$ satisfying

$$\begin{aligned}\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} &= \mathbf{y}^{(i)} \bmod q_0, & \text{and} \quad & \left\|\mathbf{x}^{(i)}\right\| \leq \hat{\alpha}^{(i)}. \\ \mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} &= \mathbf{z}^{(i)} \bmod q_1, & &\end{aligned}$$

Let

$$\mathbf{X} := \begin{pmatrix}\mathbf{x}_0^{(i+1)} & \mathbf{x}_1^{(i+1)} & \mathbf{x}_2^{(i+1)} \\ \mathbf{x}_0^{(i+1)} \cdot r_{i,0}^{-1} & \mathbf{x}_1^{(i+1)} \cdot r_{i,1}^{-1} & \mathbf{x}_2^{(i+1)} \cdot r_{i,2}^{-1}\end{pmatrix} \qquad \text{and} \qquad \mathbf{V} := \begin{pmatrix}r_{i,0}^{-1} & r_{i,1}^{-1} & r_{i,2}^{-1} \\ 1 & 1 & 1 \\ r_{i,0} & r_{i,1} & r_{i,2}\end{pmatrix}.$$

From the hypothesis, we can derive the following relations:

$$\left(\begin{array}{c|c}\begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}} & \\ \hline & \begin{bmatrix}\mathbf{A} \\ \mathbf{B}\end{bmatrix}_{\searrow n_{i+1}}\end{array}\right) \cdot \mathbf{X} = \begin{pmatrix}\mathbf{0} & \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0} \\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{y}_R^{(i)} - \begin{pmatrix}\mathbf{B} \\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} \\ \mathbf{y}_L^{(i)} - \begin{pmatrix}\mathbf{0} \\ \mathbf{A}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{y}_R^{(i)} - \begin{pmatrix}\mathbf{B} \\ \mathbf{0}\end{pmatrix} \cdot \mathbf{x}_c^{(i)} & \mathbf{0}\end{pmatrix} \cdot \mathbf{V} \bmod q_0,$$

$$\begin{pmatrix} \mathbf{C}_L^{(i)} & \big| & \mathbf{C}_R^{(i)} \end{pmatrix} \cdot \mathbf{X} = \begin{pmatrix} \mathbf{z}_{RL}^{(i)} & \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} & \mathbf{z}_{LR}^{(i)} \end{pmatrix} \cdot \mathbf{V} \bmod q_1.$$

Since $\det(\mathbf{V}) = r_{i,0}^{-1} \cdot r_{i,1}^{-1} \cdot r_{i,2}^{-1} \cdot (r_{i,0} - r_{i,1}) \cdot (r_{i,1} - r_{i,2}) \cdot (r_{i,2} - r_{i,0})$ and S is subtractive, $\mathbf{V}$ is invertible. Let

$$\begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} := \mathbf{X} \cdot \mathbf{V}^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

We have

$$\left(\begin{array}{c|c} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n_{i+1}} & \\ \hline & \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n_{i+1}} \end{array}\right) \cdot \begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_L^{(i)} - \begin{pmatrix} \mathbf{0} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \\ \mathbf{y}_R^{(i)} - \begin{pmatrix} \mathbf{B} \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \end{pmatrix} \bmod q_0,$$

$$\begin{pmatrix} \mathbf{C}_L^{(i)} & \big| & \mathbf{C}_R^{(i)} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_L^{(i)} \\ \mathbf{x}_R^{(i)} \end{pmatrix} = \begin{pmatrix} \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} \end{pmatrix} \bmod q_1,$$

or equivalently

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow n_i} \cdot \mathbf{x}^{(i)} = \mathbf{y}^{(i)} \bmod q_0$$
$$\mathbf{C}^{(i)} \cdot \mathbf{x}^{(i)} = \mathbf{z}^{(i)} \bmod q_1$$

where $\mathbf{x}^{(i)} = (\mathbf{x}_L^{(i)}, \mathbf{x}_c^{(i)}, \mathbf{x}_R^{(i)})$.

It remains to show that $\left\| \mathbf{x}^{(i)} \right\| \le \hat{\alpha}^{(i)}$. Note that

$$\underbrace{\mathbf{V}^{-1} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}_{\mathbf{w}_i} = \begin{pmatrix} \frac{r_{i,0} \cdot (r_{i,1} + r_{i,2})}{(r_{i,0} - r_{i,1}) \cdot (r_{i,2} - r_{i,0})} \\ \frac{r_{i,1} \cdot (r_{i,2} + r_{i,0})}{(r_{i,0} - r_{i,1}) \cdot (r_{i,1} - r_{i,2})} \\ \frac{r_{i,2} \cdot (r_{i,0} + r_{i,1})}{(r_{i,1} - r_{i,2}) \cdot (r_{i,2} - r_{i,0})} \end{pmatrix}$$

where each entry can be simplified to be of the form

$$\frac{-(\zeta^a - 1) \cdot (\zeta^b + \zeta^c - 2)}{(\zeta^a - \zeta^b) \cdot (\zeta^a - \zeta^c)}.$$

By a routine calculation (see e.g. [AL21, Proposition 11]), the norm of the above and hence $\|\mathbf{w}_i\|$ can be upper bounded by $4\gamma_{\mathcal{R}}$. Therefore $\left\| \mathbf{x}^{(i)} \right\| \le 4\gamma_{\mathcal{R}}^3 \cdot \hat{\alpha}^{(i+1)} = \hat{\alpha}^{(i)}$. □

D.3 Efficiency - Proof of Theorem 3

Proof. Note that $\log |\mathcal{R}_{q_0}| < \log |\mathcal{R}_{q_1}| = \log q_1^{\varphi(\rho)} = O_\lambda(\log q_1) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_1}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. It is easy to verify that the prover computes $O_\lambda(n)$ operations over $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$, which takes $O_\lambda(n \cdot \log^2 n)$ time, and that $O_\lambda(\ell)$ elements of $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$ are being communicated, for which the overall description size is at most $O_\lambda(\log^2 n)$. To analyse the computation cost of the verifier, we break down the computation steps, consisting of $O_\lambda(\ell + \sum_{i=0}^{\ell} k_i) = O_\lambda(\log n)$ operations over $\mathcal{R}_{q_0}$ or $\mathcal{R}_{q_1}$, which take time $O_\lambda(\log^3 n)$, into three parts.

First, $O_\lambda(\sum_{i=0}^{\ell-i} k_i)$ operations over $\mathcal{R}_{q_1}$ are contributed by the computation of

$$\mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} \bmod q_1, \ i \in \{0, \ldots, \ell - 1\}.$$

Second, $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$ are contributed by the recursive computation of

$$\mathbf{C}^{(i+1)} = \mathbf{C}_L^{(i)} + \mathbf{C}_R^{(i)} \cdot r_i^{-1} \bmod q_1, \ i \in \{0, \ldots, \ell - 1\}.$$

Since $\mathbf{C}$ is $(k_0, \ldots, k_\ell)$-block-foldable with block-size w, there exists $\mathsf{poly}(\lambda)$-size matrices $\mathbf{M}_\ell$ over $\mathcal{R}_{q_1}^{k_\ell}$ and $(\mathbf{L}_i, \mathbf{R}_i)_{i=0}^{\ell-1}$ over $\mathcal{R}_{q_1}$ such that

$$\mathbf{C}^{(\ell)} = (\mathbf{L}_0 + \mathbf{R}_0 \cdot r_0^{-1}) \circ \ldots \circ (\mathbf{L}_{\ell-1} + \mathbf{R}_{\ell-1} \cdot r_{\ell-1}^{-1}) \circ \mathbf{M}_\ell \bmod q_1.$$

Computing $\mathbf{C}^{(\ell)}$ this way requires $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$.

Third, another $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_0}$ are contributed by the recursive computation of

$$\mathbf{y}^{(i+1)} = \mathbf{y}_L^{(i)} + \mathbf{y}_R^{(i)} \cdot r_i - \begin{pmatrix} \mathbf{B} \cdot r_i \\ \mathbf{0} \\ \mathbf{A} \end{pmatrix} \cdot \mathbf{x}_c^{(i)} \bmod q_0$$

for $i \in \{ 0, \ldots, \ell - 1 \}$. Since $\mathbf{y}$ is $(k_0 - 1, \ldots, k_{\ell-1}, k_\ell + 1)$-block-foldable with block-size h_0, there exists $\mathsf{poly}(\lambda)$-size vectors $\mathbf{m}_\ell$ over $\mathcal{R}_{q_0}^{k_\ell+1}$ and $(\mathbf{l}_i, \mathbf{r}_i)_{i=0}^{\ell-1}$ over $\mathcal{R}_{q_0}$ such that

$$\mathbf{y}^{(\ell)} = (\mathbf{l}_0 + \mathbf{r}_0 \cdot r_0^{-1}) \circ \ldots \circ (\mathbf{l}_{\ell-1} + \mathbf{r}_{\ell-1} \cdot r_{\ell-1}^{-1}) \circ \mathbf{m}_\ell - \sum_{i=0}^{\ell-1} (\mathbf{A} + \mathbf{B} \cdot r_i) \cdot \mathbf{x}_c^{(i)} \bmod q_0.$$

Computing $\mathbf{y}^{(\ell)}$ this way requires $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_0}$.

Last, the remaining $O_\lambda(\ell + k_\ell)$ operations over $\mathcal{R}_{q_1}$ are contributed by the recursive computation of

$$\begin{aligned} \mathbf{z}^{(i+1)} &= \mathbf{z}^{(i)} - \mathbf{C}_c^{(i)} \cdot \mathbf{x}_c^{(i)} + \mathbf{z}_{RL}^{(i)} \cdot r_i^{-1} + \mathbf{z}_{LR}^{(i)} \cdot r_i \bmod q_1, \text{ and} \\ \alpha^{(i+1)} &= 2 \cdot \alpha^{(i)} \cdot \gamma_{\mathcal{R}} \end{aligned}$$

for $i \in \{ 0, \ldots, \ell - 1 \}$ and well as the final check

$$\begin{aligned} \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}_{\searrow k_\ell} \cdot \mathbf{x}^{(\ell)} &= \mathbf{y}^{(\ell)} \bmod q_0, \\ \mathbf{C}^{(\ell)} \cdot \mathbf{x}^{(\ell)} &= \mathbf{z}^{(\ell)} \bmod q_1, \end{aligned} \qquad \text{and} \qquad \left\| \mathbf{x}^{(\ell)} \right\| \leq \alpha^{(\ell)}.$$

□

E Knowledge-based Argument for Well-formedness of vSIS Commitments

Let $\mathcal{R}, s, \eta, m, q_1, q_3, \alpha, \beta, \delta, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (Section 3.2) parametrised by (η, m, q_3, β), in Fig. 3 we give a formal description of $\mathit{\Pi}_1^{\mathsf{know}}$.

F Proofs for Knowledge-based Arguments

F.1 Completeness - Proof of Theorem 7

Proof. Condition b_0. We first consider the condition b_0 in the verification algorithm. Recall that $c_0 = c_{0,0} + c_{0,1} \bmod q_3$ where

$$\begin{aligned} c_{0,0} &= \bar{c}_{\mathbf{M}} \cdot c_{\mathbf{x}} + \bar{c}_{q_0} \cdot c_{\mathbf{r}} - \hat{c}_{\mathbf{y}} \bmod q_3, \\ c_{0,1} &= \bar{c}_{\mathbf{I}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{x}} \cdot c_{\mathbf{I}} \bmod q_3. \end{aligned}$$

Substituting the expressions of each component, we have

$$\begin{aligned} c_{0,0} &= \mathbf{f}_0^{\mathtt{T}} \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} + \mathbf{f}_0^{\mathtt{T}} \cdot q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} - \mathbf{f}_0^{\mathtt{T}} \cdot \mathbf{y} \bmod q_3 \\ &= \mathbf{f}_0^{\mathtt{T}} \cdot (\mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} + q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} - \mathbf{y}) \bmod q_3 \\ &= \mathbf{f}_0^{\mathtt{T}} \cdot (\mathbf{M} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} - \mathbf{I}_s) \cdot \mathbf{x} + q_0 \cdot (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} - \mathbf{I}_t) \cdot \mathbf{r}) \bmod q_3 \\ &= \sum_{i,j \in [s], k \in [t], i \neq j} f_{0,k} \cdot M_{k,i} \cdot v^{j-i} \cdot x_j + \sum_{i,j,k \in [t], i \neq j} f_{0,k} \cdot q_0 \cdot v^{j-i} \cdot r_j \bmod q_3 \end{aligned}$$

```
$\mathsf{Setup}(1^\lambda, \mathsf{pp})$
    $v \leftarrow \mathsf{pp};\ \mathbf{t} \leftarrow\!\$\ \mathcal{T}$
    $(\mathbf{D}, \mathsf{td}) \leftarrow \mathsf{TrapGen}(1^\lambda)$
    $\mathbf{u}_i \leftarrow \mathsf{SampPre}(\mathsf{td}, \mathbf{t} \cdot v^i),\ \forall i \in \pm[s]$
    $\mathsf{pp} := v$
    $\mathsf{crs} := \big(\mathbf{D}, \mathbf{t}, v, (\mathbf{u}_i)_{i \in \pm[s]}\big)$
    return $\mathsf{crs}$

$\mathsf{Prove}(\mathsf{crs}, (\epsilon, c_{\mathbf{z}}), \mathbf{z})$
    $\mathbf{u} := \sum_{i \in \pm[s]} \mathbf{u}_i \cdot z_i$
    return $\pi := \mathbf{u}$

$\mathsf{PreVerify}(\mathsf{crs}, \epsilon)$
    return $\mathsf{pp}_\epsilon := (\mathbf{D}, \mathbf{t})$

$\mathsf{Verify}(\mathsf{crs}_\epsilon, c_{\mathbf{z}}, \pi)$
    $b_0 := \left(\mathbf{D} \cdot \mathbf{u} \overset{?}{\equiv} \mathbf{t} \cdot c_{\mathbf{z}} \bmod q_3\right)$
    $b_1 := \left(\|\mathbf{u}\| \overset{?}{\leq} \delta\right)$
    return $b_0 \wedge b_1$
```

Fig. 3. Our argument system Π_1^{know}.

where the last equality is due to $\mathbf{M} \cdot \mathbf{x} + q_0 \cdot \mathbf{r} = \mathbf{y}$, and

$$\begin{aligned}
c_{0,1} &= \mathbf{f}_1^{\mathsf{T}} \cdot \mathbf{I} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{x} - \mathbf{x}^{\mathsf{T}} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{I} \cdot \mathbf{f}_1 \bmod q_3 \\
&= \mathbf{f}_1^{\mathsf{T}} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{x} - \mathbf{f}_1^{\mathsf{T}} \cdot \mathbf{v} \cdot (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathsf{T}} \cdot \mathbf{x} \bmod q_3 \\
&= \mathbf{f}_1^{\mathsf{T}} \cdot ((\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^{\mathsf{T}} - \mathbf{v} \cdot (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathsf{T}}) \cdot \mathbf{x} \bmod q_3 \\
&= \sum_{i,j \in [s]} f_{1,i} \cdot (h_i \cdot v^{j-i} - v^{i-j} \cdot h_j) \cdot x_j \bmod q_3 \\
&= \sum_{i,j \in [s]} f_{1,j} \cdot h_j \cdot v^{i-j} \cdot x_i - \sum_{i,j \in [s]} f_{1,i} \cdot v^{i-j} \cdot h_j \cdot x_j \bmod q_3 \\
&= \sum_{i,j \in [s]} v^{i-j} \cdot h_j \cdot (f_{1,j} \cdot x_i - f_{1,i} \cdot x_j) \bmod q_3.
\end{aligned}$$

Since

$$\mathbf{D}_0 \cdot \mathbf{u}_{0,i} = \mathbf{t}_0 \cdot v^i \bmod q_3$$

for all $i \in \pm[\max\{s, t\}]$, it follows that

$$\mathbf{D}_0 \cdot \mathbf{u}_0 = \mathbf{t}_0 \cdot c_0 \bmod q_3.$$

Furthermore, we observe the following norm bounds:

(i) $\|\mathbf{u}_{0,j}\| \leq \beta$ for all $j \in \pm[\max\{s, t\}]$,
(ii) $\|\mathbf{h}\| \leq q_1/2$,
(iii) $\|\mathbf{f}_0\|, \|\mathbf{f}_1\| \leq q_2/2$,
(iv) $\|\mathbf{M}\|, \|\mathbf{y}\| \leq q_0/2$,
(v) $\|\mathbf{x}\| \leq \alpha$, and
(vi) $\|\mathbf{r}\| \leq \frac{1}{q_0} \cdot (s \cdot \|\mathbf{M}\| \cdot \|\mathbf{x}\| \cdot \gamma_{\mathcal{R}} + \|\mathbf{y}\|) \leq s \cdot \alpha \cdot \gamma_{\mathcal{R}}$.

Therefore

$$\begin{aligned}
\|\mathbf{u}_0\| &\leq s^2 \cdot t \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 + t^3 \cdot q_2 \cdot q_0 \cdot \beta \cdot s \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 + s^2 \cdot \beta \cdot q_1 \cdot q_2 \cdot \alpha \cdot \gamma_{\mathcal{R}}^3 \\
&\leq (s+t)^4 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \\
&\leq \delta_0.
\end{aligned}$$

Conditions b_1*,* b_2*, and* b_3. We next consider the conditions b_1, b_2, and b_3 in the verification algorithm. Clearly, it holds that

$$\mathbf{D}_1 \cdot \mathbf{u}_1 = \mathbf{D}_1 \cdot \left(\sum_{j \in [s]} \mathbf{u}_{1,j} \cdot x_j \right) = \sum_{j \in [s]} \mathbf{t}_1 \cdot v^j \cdot x_j = \mathbf{t}_1 \cdot c_{\mathbf{x}} \bmod q_3,$$

$$\mathbf{D}_2 \cdot \mathbf{u}_2 = \mathbf{D}_2 \cdot \left(\sum_{j \in [s]} \mathbf{u}_{2,-j} \cdot h_j \cdot x_j \right) = \sum_{j \in [s]} \mathbf{t}_2 \cdot v^{-j} \cdot h_j \cdot x_j = \mathbf{t}_2 \cdot \bar{c}_{\mathbf{x}} \bmod q_3,$$

$$\mathbf{D}_3 \cdot \mathbf{u}_3 = \mathbf{D}_3 \cdot \left(\sum_{j \in [t]} \mathbf{u}_{3,j} \cdot r_j \right) = \sum_{j \in [t]} \mathbf{t}_3 \cdot v^j \cdot r_j = \mathbf{t}_3 \cdot c_{\mathbf{r}} \bmod q_3.$$

Furthermore, since $\|\mathbf{u}_{1,j}\| \leq \beta$ for $j \in [s]$, $\|\mathbf{u}_{2,j}\| \leq \beta$ for $j \in -[s]$, $\|\mathbf{u}_{3,j}\| \leq \beta$ for $j \in [t]$, $\|\mathbf{h}\| \leq q_1/2$, $\|\mathbf{x}\| \leq \alpha$, and $\|r\| \leq s \cdot \alpha \cdot \gamma_{\mathcal{R}}$, we have

$$\begin{aligned} \|\mathbf{u}_1\| &\leq s \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_1, \\ \|\mathbf{u}_2\| &\leq s \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_2, \\ \|\mathbf{u}_3\| &\leq s^2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2 \leq \delta_3. \end{aligned}$$

Putting everything together yields the claim. □

F.2 Knowledge Soundness - Proof of Theorem 8

Proof. Fix a PPT prover $\mathcal{P}^*$. Consider an algorithm $\mathcal{B}_1 = \mathcal{B}^{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, runs $\pi \leftarrow \mathcal{P}^*(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, parses $c_{\mathbf{x}}$ from stmt and $\mathbf{u}_1$ from π, and outputs $(c_{\mathbf{x}}, \mathbf{u}_1)$. Similarly, consider the algorithms $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ and $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ which do almost the same, except that $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ parses $\bar{c}_{\mathbf{x}}$ from stmt and $\mathbf{u}_2$ from π and outputs $(\bar{c}_{\mathbf{x}}, \mathbf{u}_2)$, and $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ parses $c_{\mathbf{r}}$ from stmt and $\mathbf{u}_3$ from π and outputs $(c_{\mathbf{r}}, \mathbf{u}_3)$. Let $\mathcal{E}_{\mathcal{B}_1}^{k\text{-}R\text{-}\mathsf{ISIS},1}$, $\mathcal{E}_{\mathcal{B}_2}^{k\text{-}R\text{-}\mathsf{ISIS},2}$, and $\mathcal{E}_{\mathcal{B}_3}^{k\text{-}R\text{-}\mathsf{ISIS},3}$ be the knowledge extractors whose existence are guaranteed by Assumptions 1, 2, and 3. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, does the following:

- run $\mathbf{x}_1^\dagger \leftarrow \mathcal{E}_{\mathcal{B},1}^{k\text{-}R\text{-}\mathsf{ISIS},1}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{x}_2^\dagger \leftarrow \mathcal{E}_{\mathcal{B},2}^{k\text{-}R\text{-}\mathsf{ISIS},2}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{r}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},3}^{k\text{-}R\text{-}\mathsf{ISIS},3}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- checks that $\mathbf{x}_1^\dagger \circ \mathbf{h} = \mathbf{x}_2^\dagger$,
- checks that $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, and
- outputs $\mathbf{x}^\dagger := \mathbf{x}_1^\dagger$ if both checks pass.

Fix any adversary $\mathcal{A}$ and consider the following experiment Exp:

```
$\mathsf{Exp}(1^\lambda)$
$\mathsf{pp} \leftarrow \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$
$(\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs})$
$(\pi, \mathsf{wit}^\dagger) \leftarrow (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$
$\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$
$\mathbf{return}$ $\mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}, \pi) = 1 \wedge (\mathsf{stmt}, \mathsf{wit}^\dagger) \notin \mathit{\Psi}_{\mathsf{pp}}$
```

We claim that $\Pr\left[\mathsf{Exp}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$, which proves the theorem.

To prove the claim, consider a modified experiment Exp' where in the setup $\mathsf{Setup}(1^\lambda, \mathsf{pp})$ the matrices $\mathbf{D}_0, \mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ are sampled uniformly at random and the $\mathsf{SampPre}$ steps are replaced with sampling from SampD subject to the appropriate constraints. By the properties of $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$, Exp' is statistically close to Exp. Therefore it suffices to show that $\Pr\left[\mathsf{Exp}'(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$.

We now examine $\mathsf{wit}^\dagger$ generated during the execution of $\mathsf{Exp}'(1^\lambda)$. Parse $\mathsf{stmt} = (\mathbf{M}, \mathbf{y}, c_\mathbf{x}, \bar{c}_\mathbf{x})$ and $\mathsf{wit}^\dagger = \mathbf{x}^\dagger$. First, suppose that $\mathcal{E}_{\mathcal{P}^*}$ returns something, i.e. $\mathbf{x}_2^\dagger = \mathbf{x}_1^\dagger \circ \mathbf{h}$ and $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, then by Conditions b_1, b_2, and b_3 of the verification algorithm and Assumptions 1, 2, and 3 we have

$$\begin{aligned} c_\mathbf{x} &= \mathbf{v}^\mathsf{T} \cdot \mathbf{x}_1^\dagger \bmod q_3, & & \left\|\mathbf{x}_1^\dagger\right\| \leq \alpha_1^* \\ \bar{c}_\mathbf{x} &= \bar{\mathbf{v}}^\mathsf{T} \cdot \mathbf{x}_2^\dagger \bmod q_3, & & \left\|\mathbf{x}_2^\dagger\right\| \leq \alpha_2^* \\ c_\mathbf{r} &= \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r}^\dagger \bmod q_3, & \text{and} \quad & \left\|\mathbf{r}^\dagger\right\| \leq \alpha_3^*. \end{aligned}$$

It remains to show that $\mathbf{x}_1^\dagger \circ \mathbf{h} = \mathbf{x}_2^\dagger$ and $\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger = \mathbf{y}$, so that $\mathcal{E}_{\mathcal{P}^*}$ returns something, with overwhelming probability.

Examining the condition b_0 in the verification algorithm, we observe

$$\begin{aligned} & \mathbf{D}_0 \cdot \mathbf{u}_0 \\ = \; & \mathbf{t}_0 \cdot (\bar{c}_\mathbf{M} \cdot c_\mathbf{x} + \bar{c}_{q_0} \cdot c_\mathbf{r} - \hat{c}_\mathbf{y} + \bar{c}_\mathbf{I} \cdot c_\mathbf{x} - \bar{c}_\mathbf{x} \cdot c_\mathbf{I}) \bmod q_3 \\ = \; & \mathbf{t}_0 \cdot (\mathbf{f}_0^\mathsf{T} \cdot \mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}_1^\dagger + \mathbf{f}_0^\mathsf{T} \cdot q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r}^\dagger \\ & \quad + \mathbf{f}_1^\mathsf{T} \cdot (\bar{\mathbf{v}} \circ \mathbf{h}) \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}_1^\dagger - \mathbf{f}_0^\mathsf{T} \cdot \mathbf{y} - \bar{\mathbf{v}}^\mathsf{T} \cdot \mathbf{x}_2^\dagger \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{f}_1) \bmod q_3 \\ = \; & \mathbf{t}_0 \cdot \mathbf{f}_0^\mathsf{T} \cdot (\mathbf{M} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} \cdot \mathbf{r}^\dagger - \mathbf{y}) \\ & \quad + \mathbf{t}_0 \cdot \mathbf{f}_1^\mathsf{T} \cdot (\mathsf{diag}(\mathbf{h}) \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} \cdot \mathbf{x}_1^\dagger - \mathbf{v} \cdot \bar{\mathbf{v}}^\mathsf{T} \cdot \mathbf{x}_2^\dagger) \bmod q_3 \\ = \; & \mathbf{t}_0 \cdot \mathbf{f}_0^\mathsf{T} \cdot (\mathbf{M} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}_1^\dagger + q_0 \cdot (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^\mathsf{T} - \mathbf{I}_t) \cdot \mathbf{r}^\dagger + (\mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger - \mathbf{y})) \\ & \quad + \mathbf{t}_0 \cdot \mathbf{f}_1^\mathsf{T} \cdot (\mathsf{diag}(\mathbf{h}) \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}_1^\dagger - (\mathbf{v} \cdot \bar{\mathbf{v}}^\mathsf{T} - \mathbf{I}_s) \cdot \mathbf{x}_2^\dagger + (\mathbf{h} \circ \mathbf{x}_1^\dagger - \mathbf{x}_2^\dagger)) \bmod q_3. \end{aligned}$$

Let

$$\begin{aligned} \mathbf{u}_0^\dagger \coloneqq \; & \sum_{i,j \in [s], k \in [t] : i \neq j} f_{0,k} \cdot M_{k,i} \cdot \mathbf{u}_{0,j-i} \cdot x_{1,j}^\dagger + \sum_{i,j,k \in [t] : i \neq j} f_{0,k} \cdot q_0 \cdot \mathbf{u}_{0,j-i} \cdot r_j^\dagger \\ & + \sum_{i,j \in [s] : i \neq j} f_{1,i} \cdot h_i \cdot \mathbf{u}_{0,j-i} \cdot x_{1,j}^\dagger + \sum_{i,j \in [s] : i \neq j} f_{1,i} \cdot \mathbf{u}_{0,i-j} \cdot x_{2,j}^\dagger, \\ \mathbf{e}_0^\dagger \coloneqq \; & \mathbf{M} \cdot \mathbf{x}_1^\dagger + q_0 \cdot \mathbf{r}^\dagger - \mathbf{y}, \\ \mathbf{e}_1^\dagger \coloneqq \; & \mathbf{h} \circ \mathbf{x}_1^\dagger - \mathbf{x}_2^\dagger. \end{aligned}$$

We have

$$\mathbf{D}_0 \cdot (\mathbf{u}_0 - \mathbf{u}_0^\dagger) = \mathbf{t}_0 \cdot (\mathbf{f}_0^\mathsf{T} \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\mathsf{T} \cdot \mathbf{e}_1^\dagger) \bmod q_3.$$

Suppose, contrary to our claim, that $(\mathbf{e}_0^\dagger, \mathbf{e}_1^\dagger) \neq \mathbf{0}$ with non-negligible probability. Then one (or both) of the following must be true:

(i) $\mathbf{f}_0^\mathsf{T} \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\mathsf{T} \cdot \mathbf{e}_1^\dagger = \mathbf{0}$ with non-negligible probability.
(ii) $\mathbf{f}_0^\mathsf{T} \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\mathsf{T} \cdot \mathbf{e}_1^\dagger \neq \mathbf{0}$ with non-negligible probability.

If Case (i) is true, then we also have with non-negligible probability

$$\mathbf{f}_0^\mathsf{T} \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^\mathsf{T} \cdot \mathbf{e}_1^\dagger = \mathbf{0} \bmod q_2.$$

Note that

$$\begin{aligned} \left\|\mathbf{e}_0^\dagger\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_\mathcal{R} + q_0 \cdot \alpha_3^* + q_0/2 \leq s \cdot q_0 \cdot \alpha^* \cdot \gamma_\mathcal{R}, \\ \left\|\mathbf{e}_1^\dagger\right\| &\leq q_1/2 \cdot \alpha_1^* \cdot \gamma_\mathcal{R} + \alpha_2^* \leq q_1 \cdot \alpha^* \cdot \gamma_\mathcal{R}. \end{aligned}$$

Therefore $\left\|(\mathbf{e}_0^\dagger, \mathbf{e}_1^\dagger)\right\| \leq s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_\mathcal{R} \leq \beta_{q_2}^*$. This would, however, violate Assumption 4. We thus conclude that Case (i) is impossible.

If Case (ii) is true, we observe that

$$\begin{aligned}
\left\|\mathbf{u}_0^\dagger\right\| &\leq s^2 \cdot t \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^3 + t^3 \cdot q_2 \cdot q_0 \cdot \beta \cdot \alpha_3^* \cdot \gamma_{\mathcal{R}}^2 \\
&\quad + s^2 \cdot q_2 \cdot q_1 \cdot \beta \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^3 + s^2 \cdot q_2 \cdot \beta \cdot \alpha_2^* \cdot \gamma_{\mathcal{R}}^2 \\
&\leq (s+t)^3 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \\
&\leq \beta_{q_3}^*/2, \\
\left\|\mathbf{u}_0 - \mathbf{u}_0^\dagger\right\| &\leq \beta_{q_3}^*, \\
\left\|\mathbf{f}_0^{\mathsf{T}} \cdot \mathbf{e}_0^\dagger + \mathbf{f}_1^{\mathsf{T}} \cdot \mathbf{e}_1^\dagger\right\| &\leq (s+t) \cdot q_2 \cdot s \cdot q_0 \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2 \\
&\leq (s+t)^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2 \\
&\leq \beta_{q_3}^*.
\end{aligned}$$

This would, however, violate Assumption 0. We thus conclude that Case (ii) is impossible.

Since both cases are impossible, we conclude that $(\mathbf{e}_0^\dagger, \mathbf{e}_1^\dagger) \neq \mathbf{0}$ with non-negligible probability. □

F.3 Efficiency - Proof of Theorem 9

Proof. Note that, $\log |\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_3}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. The common reference string

$$\mathsf{crs} = \begin{pmatrix} \mathbf{D}_0,\ \mathbf{t}_0,\ (\mathbf{u}_{0,j})_{j \in I_0}, \\ \mathbf{D}_1,\ \mathbf{t}_1,\ (\mathbf{u}_{1,j})_{j \in I_1}, \\ \mathbf{D}_2,\ \mathbf{t}_2,\ (\mathbf{u}_{2,j})_{j \in I_2}, \\ \mathbf{D}_3,\ \mathbf{t}_3,\ (\mathbf{u}_{3,j})_{j \in I_3}, \\ v,\quad \mathbf{h},\quad \mathbf{f}_0,\quad \mathbf{f}_1 \end{pmatrix}$$

has description size at most

$$(4 \cdot \eta \cdot (m+1) + 6 \cdot (s+t)) \cdot |\mathcal{R}_{q_3}| = O_\lambda(n \cdot \log n).$$

A proof $(c_{\mathbf{r}}, \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ has description size at most

$$(4m+1) \cdot \log |\mathcal{R}_{q_3}| = O_\lambda(\log^2 n).$$

Preprocessing requires $O(n)$ $\mathcal{R}_{q_3}$ operations, which cost $O_\lambda(n \cdot \log^2 n)$ bit operations. After preprocessing, verification requires $O_\lambda(m)$ $\mathcal{R}_{q_3}$ operations, which cost $O_\lambda(\log^3 n)$ bit operations.

It remains to show that prover time is $O_\lambda(n \cdot \log^3 n)$. It suffices to analyse the time needed for computing $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$ since they dominate the prover computation. Recall that

$$\begin{aligned}
\mathbf{u}_{0,0} &= \sum_{i \in [s], k \in [t]} f_{0,k} \cdot M_{k,i} \cdot \sum_{j \in [s] : j \neq i} \mathbf{u}_{0,j-i} \cdot x_j + \sum_{i,k \in [t]} f_{0,k} \cdot p \cdot \sum_{j \in [t] : j \neq i} \mathbf{u}_{0,j-i} \cdot r_j, \\
\mathbf{u}_{0,1} &= \sum_{j \in [s]} h_j \cdot f_{1,j} \cdot \sum_{i \in [s] : i \neq j} \mathbf{u}_{0,i-j} \cdot x_i - \sum_{i \in [s]} f_{1,i} \cdot \sum_{j \in [s] : j \neq i} \mathbf{u}_{0,i-j} \cdot h_j \cdot x_j.
\end{aligned}$$

It is clear that once the terms

$$\begin{matrix} \displaystyle\sum_{j \in [s] : j \neq i} \mathbf{u}_{0,j-i} \cdot x_j, & & \displaystyle\sum_{j \in [t] : j \neq i} \mathbf{u}_{0,j-i} \cdot r_j, \\ \displaystyle\sum_{i \in [s] : i \neq j} \mathbf{u}_{0,i-j} \cdot x_i, & \text{and} & \displaystyle\sum_{j \in [s] : j \neq i} \mathbf{u}_{0,i-j} \cdot h_j \cdot x_j \end{matrix}$$

are computed, $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$ can be computed with $O_\lambda(n)$ $\mathcal{R}_{q_3}$ operations. We examine the cost for computing the first term, i.e. $\sum_{j \in [s] : j \neq i} \mathbf{u}_{0,j-i} \cdot x_j$.

Observe that $\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$ can be written in the form

$$\begin{pmatrix} \mathbf{0} & \mathbf{u}_{0,1} & \mathbf{u}_{0,2} & \cdots & \cdots & \mathbf{u}_{0,s-1} \\ \mathbf{u}_{0,-1} & \mathbf{0} & \mathbf{u}_{0,1} & \ddots & & \vdots \\ \mathbf{u}_{0,-2} & \mathbf{u}_{0,-1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \mathbf{u}_{0,1} & \mathbf{u}_{0,2} \\ \vdots & & \ddots & \mathbf{u}_{0,-1} & \mathbf{0} & \mathbf{u}_{0,1} \\ \mathbf{u}_{0,-(s-1)} & \cdots & \cdots & \mathbf{u}_{0,-2} & \mathbf{u}_{0,-1} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ x_s \end{pmatrix}$$

which can be expressed as a sum of m matrix-vector products where each of the m matrices is an s-by-s Toeplitz matrix over $\mathcal{R}_{q_3}$. It is well-known (see e.g. [GL96]) that multiplying an s-by-s Toeplitz matrix to a vector takes $O(s \cdot \log s)$ operations over the base ring, i.e. $\mathcal{R}_{q_3}$. Therefore $\sum_{j\in[s]:j\neq i} \mathbf{u}_{0,j-i} \cdot x_j$ can be computed using $O_\lambda(n \cdot \log n \cdot m \cdot \log q) = O_\lambda(n \cdot \log^3 n)$ bit operations.

By splitting the other terms as sums of Toeplitz-vector products, we conclude that the computation of $\mathbf{u}_{0,0}$ and $\mathbf{u}_{0,1}$, and hence the the overall prover computation, takes time

$$O_\lambda(n \cdot \log^3 n).$$

□

G Proofs for Applications

G.1 Completeness - Proof of Theorem 13

Proof. Since $\mathbf{x} \in \{0,1\}^s$, observe that

$$\|\mathbf{z}\| \leq \left\| \sum_{0\leq i,j\leq s: i-j=k} h_j \cdot x_j \cdot (x_i - 1) \right\| \leq s \cdot \|\mathbf{h}\|.$$

For $\mathbf{h}$ generated by $\mathsf{Gen}^{\mathsf{str}}$, Lemma 6 implies that $\|\mathbf{h}\| \leq (q_1/2)^{\ell+1} \cdot \gamma_{\mathcal{R}}^{\ell}$. For $\mathbf{h}$ generated by $\mathsf{Gen}^{\mathsf{unstr}}$, we have $\mathbf{h} \in \mathcal{R}_{q_1}^s$ and thus $\|\mathbf{h}\| \leq q_1/2$. □

G.2 Knowledge-Soundness - Proof of Theorem 14

Proof. Fix a PPT prover $\mathcal{P}^*$ and let $\mathcal{P}_0^*$ and $\mathcal{P}_1^*$ be wrappers of $\mathcal{P}^*$ which interact with $\Pi'.\mathsf{Verify}$ and $\Pi''.\mathsf{Verify}$ respectively. By the knowledge-soundness of Π' and Π'', there exist knowledge extractors $\mathcal{E}_{\mathcal{P}_0^*}^{\Pi'}$ and $\mathcal{E}_{\mathcal{P}_1^*}^{\Pi''}$ respectively. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}) = ((\mathbf{v}, \mathbf{h}), (\mathbf{M}, \mathbf{y}))$, does the following:

- Obtain $(c_{\mathbf{x}}, \bar{c}_{\mathbf{x}})$ from $\mathcal{P}^*$.
- Compute $c_{\mathbf{z}} := \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1} \rangle) \bmod q_3$.
- Obtain $\mathbf{x}^\dagger$ by running $\mathcal{E}_{\mathcal{P}_0^*}^{\Pi'}$ on $(\mathsf{crs}', ((\mathbf{M}, \mathbf{y}), (c_{\mathbf{x}}, \bar{c}_{\mathbf{x}}))$.
- If $\mathbf{x}^\dagger \in \{0,1\}^s$, output $\mathbf{x}^\dagger$, else continue.
- Compute $\hat{z}_0 := -\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger$.
- If $\hat{z}_0 = 0$, output $((x_i^\dagger - 1) \cdot x_i^\dagger)_{i\in[s]}$, else continue.
- Obtain $\hat{\mathbf{z}}_{-0} = (\hat{z}_{-s}, \ldots, \hat{z}_{-1}, \hat{z}_1, \ldots, \hat{z}_s)$ by running $\mathcal{E}_{\mathcal{P}_1^*}^{\Pi''}$ on $(\mathsf{crs}'', (\epsilon, c_{\mathbf{z}}))$.
- Define $\hat{\mathbf{z}} := (\hat{z}_{-s}, \ldots, \hat{z}_{-1}, \hat{z}_0, \hat{z}_1, \ldots, \hat{z}_s)$.
- Compute $\mathbf{z}^\dagger := \left(\sum_{0\leq i,j\leq s: i-j=k} h_j \cdot x_j^\dagger \cdot (x_i^\dagger - 1)\right)_{-s\leq k\leq s}$.
- Output $\hat{\mathbf{z}} - \mathbf{z}^\dagger$.

We claim that with overwhelming probability $\mathcal{E}_{\mathcal{P}^*}$ outputs $\mathbf{x}^\dagger$ such that $((\mathbf{M}, \mathbf{y}), \mathbf{x}^\dagger) \in \varPsi^{\texttt{str-bin-sat}}$ (resp. $\varPsi^{\texttt{bin-sat}}$).

First, by the knowledge-soundness of $\varPi'$, we have with overwhelming probability that $\mathbf{M} \cdot \mathbf{x}^\dagger = \mathbf{y} \bmod q_0$ and that $\mathbf{x}^\dagger$ satisfies

$$\left\langle \mathbf{v}, \mathbf{x}^\dagger \right\rangle = c_{\mathbf{x}} \bmod q_3, \qquad \left\langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x}^\dagger \right\rangle = \bar{c}_{\mathbf{x}} \bmod q_3, \qquad \left\| \mathbf{x}^\dagger \right\| \leq \alpha'.$$

It remains to argue that $\mathbf{x}^\dagger \in \{0,1\}^s$ with overwhelming probability.

Suppose towards a contradiction that $\mathbf{x}^\dagger \notin \{0,1\}^s$ with non-negligible probability. By the knowledge-soundness of $\varPi''$, with overwhelming probability $\hat{\mathbf{z}}_{-0}$ satisfies

$$\begin{aligned}
\langle (\bar{\mathbf{v}} || \mathbf{v}), \hat{\mathbf{z}}_{-0} \rangle &= c_{\mathbf{z}} \bmod q_3 \\
&= \bar{c}_{\mathbf{x}} \cdot (c_{\mathbf{x}} - \langle \mathbf{v}, \mathbf{1} \rangle) \bmod q_3 \\
&= \left(\sum_j \bar{v}_j \cdot h_j \cdot x_j^\dagger \right) \cdot \left(\sum_i v_i \cdot (x_i^\dagger - 1) \right) \bmod q_3 \\
&= \sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger + \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3, \\
\langle (\bar{\mathbf{v}} || 1 || \mathbf{v}), \hat{\mathbf{z}} \rangle &= \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3,
\end{aligned}$$

and $\|\hat{\mathbf{z}}\| \leq \alpha''$, where in the third equality we have used that the extracted vector $\mathbf{x}^\dagger$ satisfies $\left\langle \mathbf{v}, \mathbf{x}^\dagger \right\rangle = c_{\mathbf{x}} \bmod q_3$, and $\left\langle \bar{\mathbf{v}} \circ \mathbf{h}, \mathbf{x}^\dagger \right\rangle = \bar{c}_{\mathbf{x}} \bmod q_3$. On the other hand, $\mathbf{z}^\dagger$ satisfies

$$\left\langle (\bar{\mathbf{v}} || 1 || \mathbf{v}), \mathbf{z}^\dagger \right\rangle = \sum_{i,j,i \neq j} v^{i-j} \cdot h_j \cdot (x_i^\dagger - 1) \cdot x_j^\dagger \bmod q_3$$

with $z_0^\dagger = 0$ and $\left\| \mathbf{z}^\dagger \right\| \leq s \cdot \|\mathbf{h}\| \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^2 \leq s \cdot (q_1/2)^{\ell+1} \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^{\ell+2}$ (resp. $s \cdot q_1/2 \cdot (\alpha' + 1)^2 \cdot \gamma_{\mathcal{R}}^2$). Therefore,

$$\left\langle (\bar{\mathbf{v}} || 1 || \mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad \left\| \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\| \leq \beta_{q_3}.$$

One (or both) of the following two cases must be true

(i) $\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger = 0$ with non-negligible probability.
(ii) $\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger \neq 0$ with non-negligible probability,

If Case (i) is true, we have

$$\sum_i h_i \cdot (x_i^\dagger - 1) \cdot x_i^\dagger = 0 \bmod q_1 \qquad \text{and} \qquad 0 < \left\| ((x_i^\dagger - 1) \cdot x_i^\dagger)_{i \in [s]} \right\| \leq \beta_{q_1}$$

with non-negligible probability. This contradicts Assumption 0. If Case (ii) is true, we have

$$\left\langle (\bar{\mathbf{v}} || 1 || \mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q \qquad \text{and} \qquad 0 < \left\| \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\| \leq \beta_{q_3}$$

with non-negligible probability. This contradicts Assumption 1. Since none of the two cases could be true, we must have $\mathbf{x}^\dagger \in \{0,1\}^s$, as claimed. □

G.3 Efficiency - Proof of Theorem 15

Proof. Note that $\log |\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_{q_3}$ operation takes at most $O_\lambda(\log^2 n)$ bit operations.

Notice that $\mathbf{z}$ can be computed in time $O_\lambda(n \cdot \log^3 n)$, exploiting fast multiplication algorithms for Toeplitz matrices (similarly to what described in Appendix F.3). All claims about the unstructured case then follow from Theorems 9 and 12.

For the structured case, we need to argue that crs has a short description size. Note that crs can be succinctly described by $(v, \tilde{\mathbf{h}}) \in \mathcal{R}_{q_3}^\times \times (\mathcal{R}_{q_3}^\times)^{\tilde{n}}$ where $\tilde{n} = \sum_{i=0}^{\ell-1} (k_i + 1) + k_\ell$ and $n = \sum_{i=0}^{\ell} k_i$ with $k_i \in \{1, 2\}$. We thus conclude that crs has description size $O_\lambda(\log^2 n)$. The rest of the claims for the structured case then follow from Theorems 3 and 6. □

H Construction and Proofs for R1CS Argument

Let $\mathcal{R}, s_1, s_2, t, \eta, m, q_0, q_1, q_2, q_3, \alpha, \beta, \delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6, \mathcal{T}$ depend on λ. Using the lattice trapdoor algorithms (Section 3.2) parametrised by (η, m, q_3, β), in Fig. 4, we construct an argument system for $\varPsi^{\mathtt{R1CS}}$.

H.1 Completeness

Theorem 16 (Completeness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold. For*

$$\begin{aligned} &\delta_0 \geq 6 \cdot t^2 \cdot s^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^4, && \delta_1 \geq s_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}, \\ &\delta_2 \geq s \cdot t \cdot q_0 \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3, && \delta_3 \geq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2, \\ &\delta_4 \geq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2, && \delta_5 \geq 2 \cdot s^2 \cdot q_0 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^4, \\ &\delta_6 \geq 6 \cdot s^2 \cdot t^2 \cdot q_0^2 \cdot q_1 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^6, \end{aligned}$$

$\varPi^{\mathtt{R1CS}}$ in Figure 4 is complete.

Proof. Condition b_0. We first consider the condition b_0 in the verification algorithm. Recall that $c_0 = c_{0,\mathbf{E}} + c_{0,\mathbf{F}} + c_{0,\mathbf{G}} \bmod q_3$ where

$$\begin{aligned} c_{0,\mathbf{E}} &= \bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},1} \cdot \bar{c}_{\mathbf{e}} \bmod q_3, \\ c_{0,\mathbf{F}} &= \bar{c}_{\mathbf{F}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},2} \cdot \bar{c}_{\mathbf{f}} \bmod q_3 \\ c_{0,\mathbf{G}} &= \bar{c}_{\mathbf{G}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},3} \cdot \bar{c}_{\mathbf{g}} \bmod q_3 \end{aligned}$$

Substituting the expressions of each component, we have

$$\begin{aligned} c_{0,\mathbf{E}} &= (\boldsymbol{\ell}_1 \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \mathbf{v}_t^{\mathtt{T}} \cdot \boldsymbol{\ell}_1 \cdot (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \mathbf{x} \bmod q_3 \\ &= (\boldsymbol{\ell}_1 \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \boldsymbol{\ell}_1^{\mathtt{T}} \cdot \mathbf{v}_t \cdot (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \mathbf{x} \bmod q_3 \\ &= \sum_{\substack{i \in [t], j,k \in [s] \\ j \neq k}} E_{i,j} \cdot h_i \cdot \ell_{1,i} \cdot v^{k-j} \cdot x_k - \sum_{\substack{i,k \in [t], j \in [s] \\ k \neq i}} E_{i,j} \cdot x_j \cdot h_i \cdot v^{k-i} \cdot \ell_{1,k} \bmod q_3, \end{aligned}$$

$$\begin{aligned} c_{0,\mathbf{F}} &= \boldsymbol{\ell}_2^{\mathtt{T}} \cdot \mathbf{F} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \boldsymbol{\ell}_2 \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{F} \cdot \mathbf{x} \bmod q_3 \\ &= \boldsymbol{\ell}_2^{\mathtt{T}} \cdot (\mathbf{F} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{F} \cdot \mathbf{x}) \bmod q_3 \\ &= \boldsymbol{\ell}_2^{\mathtt{T}} \cdot (\mathbf{F} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} - \mathbf{I}_s) \cdot \mathbf{x} - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} - \mathbf{I}_t) \cdot \mathbf{F} \cdot \mathbf{x}) \bmod q_3 \\ &= \sum_{\substack{i \in [t], j,k \in [s] \\ j \neq k}} \ell_{2,i} \cdot F_{i,j} \cdot v^{k-j} \cdot x_k - \sum_{\substack{k,i \in [t], j \in [s] \\ i \neq k}} \ell_{2,k} \cdot v^{i-k} \cdot F_{i,j} \cdot x_j \bmod q_3, \end{aligned}$$

and

$$\begin{aligned} c_{0,\mathbf{G}} &= \boldsymbol{\ell}_3^{\mathtt{T}} \cdot \mathbf{G} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \boldsymbol{\ell}_3 \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{G} \cdot \mathbf{x} \bmod q_3 \\ &= \boldsymbol{\ell}_3^{\mathtt{T}} \cdot (\mathbf{G} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} \cdot \mathbf{x} - \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{G} \cdot \mathbf{x}) \bmod q_3 \\ &= \boldsymbol{\ell}_3^{\mathtt{T}} \cdot (\mathbf{G} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathtt{T}} - \mathbf{I}_s) \cdot \mathbf{x} - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathtt{T}} - \mathbf{I}_t) \cdot \mathbf{G} \cdot \mathbf{x}) \bmod q_3 \\ &= \sum_{\substack{i \in [t], j,k \in [s] \\ j \neq k}} \ell_{3,i} \cdot G_{i,j} \cdot v^{k-j} \cdot x_k - \sum_{\substack{k,i \in [t], j \in [s] \\ i \neq k}} \ell_{3,k} \cdot v^{i-k} \cdot G_{i,j} \cdot x_j \bmod q_3 \end{aligned}$$

Since

$$\mathbf{D}_0 \cdot \mathbf{u}_{0,i} = \mathbf{t}_0 \cdot v^i \bmod q_3$$

for all $i \in \pm[\max\{s, t\}]$, it follows that

$$\mathbf{D}_0 \cdot \mathbf{u}_0 = \mathbf{t}_0 \cdot c_0 \bmod q_3.$$

Furthermore, we observe the following norm bounds:

```
$\mathsf{Setup}(1^\lambda, \mathsf{pp})$
$(v, \mathbf{h}, \boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3) \leftarrow\!\!\$\; \mathcal{R}_q^\times \times \mathcal{R}_{q_1}^t \times \mathcal{R}_{q_2}^t \times \mathcal{R}_{q_2}^t \times \mathcal{R}_{q_2}^t$
$I_0 \coloneqq \pm[\max\{\, s, t \,\}],\ I_1 \coloneqq [s_1 + 1; s],$
$I_2 \coloneqq -[t],\ I_3 \coloneqq [t],\ I_4 \coloneqq [t],\ I_5 \coloneqq [t]$
$I_6 \coloneqq \pm[t]$
$\mathbf{for}\ i \in \{0, 1, 2, 3, 4, 5, 6\}\ \mathbf{do}$
    $(\mathbf{D}_i, \mathsf{td}_i) \leftarrow \mathsf{TrapGen}(1^\lambda), \quad \mathbf{t}_i \leftarrow\!\!\$\; \mathcal{T}$
    $\mathbf{u}_{i,j} \leftarrow \mathsf{SampPre}(\mathsf{td}_i, \mathbf{t}_i \cdot v^j),\ \forall j \in I_i$
$\mathsf{crs} \coloneqq \begin{pmatrix} (\mathbf{D}_i,\ \mathbf{t}_i,\ (\mathbf{u}_{i,j})_{j \in I_i})_{i=0}^6, \\ v, \quad \mathbf{h}, \quad \boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3 \end{pmatrix}$
$\mathbf{return}\ \mathsf{crs}$
```

```
$\mathsf{PreVerify}(\mathsf{crs}, (\mathbf{x}_1, \mathbf{E}, \mathbf{F}, \mathbf{G}))$
$c_{\mathbf{x}_1} \coloneqq \mathbf{v}_1^\intercal \cdot \mathbf{x}_1 \bmod q_3$
$\bar{c}_{\mathbf{E}} \coloneqq (\boldsymbol{\ell}_1 \circ \mathbf{h})^\intercal \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \bmod q_3$
$\bar{c}_{\mathbf{F}} \coloneqq \boldsymbol{\ell}_2^\intercal \cdot \mathbf{F} \cdot \bar{\mathbf{v}} \bmod q_3$
$\bar{c}_{\mathbf{G}} \coloneqq \boldsymbol{\ell}_3^\intercal \cdot \mathbf{G} \cdot \bar{\mathbf{v}} \bmod q_3$
$c_{\mathbf{I},1} \coloneqq \mathbf{v}_t^\intercal \cdot \boldsymbol{\ell}_1 \bmod q_3$
$\bar{c}_{\mathbf{I},2} \coloneqq \bar{\mathbf{v}}_t^\intercal \cdot \boldsymbol{\ell}_2 \bmod q_3$
$\bar{c}_{\mathbf{I},3} \coloneqq \bar{\mathbf{v}}_t^\intercal \cdot \boldsymbol{\ell}_3 \bmod q_3$
$\bar{c}_{\mathbf{I},6} \coloneqq \bar{\mathbf{v}}_t^\intercal \cdot \mathbf{h} \bmod q_3$
$\mathsf{crs}_{\mathbf{E},\mathbf{F},\mathbf{G}} \coloneqq \begin{pmatrix} (\mathbf{D}_i, \mathbf{t}_i)_{i=0}^6, \\ c_{\mathbf{x}_1}, \bar{c}_{\mathbf{E}}, \bar{c}_{\mathbf{F}}, \bar{c}_{\mathbf{G}}, \\ c_{\mathbf{I},1}, \bar{c}_{\mathbf{I},2}, \bar{c}_{\mathbf{I},3}, \bar{c}_{\mathbf{I},6} \end{pmatrix}$
$\mathbf{return}\ \mathsf{crs}_{\mathbf{x}_1,\mathbf{E},\mathbf{F},\mathbf{G}}$
```

```
$\mathsf{Verify}(\mathsf{crs}_{\mathbf{x}_1,\mathbf{E},\mathbf{F},\mathbf{G}}, \pi)$
$c_{\mathbf{x}} \coloneqq c_{\mathbf{x}_1} + c_{\mathbf{x}_2} \bmod q_3$
$c_{0,\mathbf{E}} \coloneqq \bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},1} \cdot \bar{c}_{\mathbf{e}} \bmod q_3$
$c_{0,\mathbf{F}} \coloneqq \bar{c}_{\mathbf{F}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},2} \cdot c_{\mathbf{f}} \bmod q_3$
$c_{0,\mathbf{G}} \coloneqq \bar{c}_{\mathbf{G}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},3} \cdot c_{\mathbf{g}} \bmod q_3$
$c_0 \coloneqq c_{0,\mathbf{E}} + c_{0,\mathbf{F}} + c_{0,\mathbf{G}} \bmod q_3$
$(c_1, c_2, c_3, c_4, c_5) \coloneqq (c_{\mathbf{x}_2}, \bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{g}}, c_{\mathbf{r}})$
$c_6 \coloneqq \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},\mathbf{6}} \cdot c_{\mathbf{g}} \bmod q_3$
$\mathbf{for}\ i \in \{0, 1, 2, 3, 4, 5, 6\}\ \mathbf{do}$
    $b_i \coloneqq (\mathbf{D}_i \cdot \mathbf{u}_i \overset{?}{\equiv} \mathbf{t}_i \cdot c_i \bmod q_3\ \wedge\ \|\mathbf{u}_i\| \overset{?}{\leq} \delta_i)$
$\mathbf{return}\ b_0 \wedge b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5 \wedge b_6$
```

```
$\mathsf{Prove}(\mathsf{crs}, (\mathbf{E}, \mathbf{F}, \mathbf{G}), \mathbf{x}_2)$
$c_{\mathbf{x}_2} \coloneqq \mathbf{v}_2^\intercal \cdot \mathbf{x}_2 \bmod q_3$
$c_{\mathbf{r}} \coloneqq \mathbf{v}_t^\intercal \cdot \mathbf{r} \bmod q_3$
$\bar{c}_{\mathbf{e}} \coloneqq (\bar{\mathbf{v}}_t \circ \mathbf{h})^\intercal \cdot \mathbf{E} \cdot \mathbf{x} \bmod q_3$
$c_{\mathbf{f}} \coloneqq \mathbf{v}_t^\intercal \cdot \mathbf{F} \cdot \mathbf{x} \bmod q_3$
$c_{\mathbf{g}} \coloneqq \mathbf{v}_t^\intercal \cdot \mathbf{G} \cdot \mathbf{x} \bmod q_3$
$\mathbf{u}_{\mathbf{E}} \coloneqq \sum_{i \in [t], j \in [s]} E_{i,j} \cdot h_i \cdot \ell_{1,i} \cdot \sum_{k \in [s] : k \neq j} \mathbf{u}_{0,k-j} \cdot x_k$
    $- \sum_{i \in [t], j \in [s]} E_{i,j} \cdot x_j \cdot h_i \cdot \sum_{k \in [t] : k \neq i} \mathbf{u}_{0,k-i} \cdot \ell_{1,k}$
$\mathbf{u}_{\mathbf{F}} \coloneqq \sum_{i \in [t], j \in [s]} F_{i,j} \cdot \ell_{2,i} \cdot \sum_{k \in [s] : k \neq j} \mathbf{u}_{0,k-j} \cdot x_k$
    $- \sum_{i \in [t], j \in [s]} F_{i,j} \cdot x_j \cdot \sum_{k \in [t] : k \neq i} \mathbf{u}_{0,i-k} \cdot \ell_{2,k}$
$\mathbf{u}_{\mathbf{G}} \coloneqq \sum_{i \in [t], j \in [s]} G_{i,j} \cdot \ell_{3,i} \cdot \sum_{k \in [s] : k \neq j} \mathbf{u}_{0,k-j} \cdot x_k$
    $- \sum_{i \in [t], j \in [s]} G_{i,j} \cdot x_j \cdot \sum_{k \in [t] : k \neq i} \mathbf{u}_{0,i-k} \cdot \ell_{3,k}$
$\mathbf{u}_0 \coloneqq \mathbf{u}_{\mathbf{E}} + \mathbf{u}_{\mathbf{F}} + \mathbf{u}_{\mathbf{G}}$
$\mathbf{u}_1 \coloneqq \sum_{j \in [s_1+1; s]} \mathbf{u}_{1,j} \cdot x_j$
$\mathbf{u}_2 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{2,-i} \cdot E_{i,j} \cdot h_i \cdot x_j$
$\mathbf{u}_3 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{3,i} \cdot F_{i,j} \cdot x_j$
$\mathbf{u}_4 \coloneqq \sum_{i \in [t]} \sum_{j \in [s]} \mathbf{u}_{4,i} \cdot G_{i,j} \cdot x_j$
$\mathbf{u}_5 \coloneqq \sum_{i \in [t]} \mathbf{u}_{5,i} \cdot r_i$
$\mathbf{u}_6 \coloneqq \sum_{i,j \in [t], i \neq j} \mathbf{u}_{6,i-j} \cdot (e_j \cdot f_i + q_0 \cdot r_i - g_i) \cdot h_j$
$\mathbf{return}\ \pi \coloneqq \begin{pmatrix} c_{\mathbf{x}_2}, c_{\mathbf{r}}, \bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{g}}, \\ \mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_6 \end{pmatrix}$
```

Fig. 4. Our argument system $\Pi^{\mathtt{R1CS}}$.

(i) $\|\mathbf{u}_{0,j}\| \leq \beta$ for all $j \in \pm[\max\{\, s, t \,\}]$,
(ii) $\|\mathbf{h}\| \leq q_1/2$,
(iii) $\|\boldsymbol{\ell}_1\|, \|\boldsymbol{\ell}_2\|, \|\boldsymbol{\ell}_3\| \leq q_2/2$,
(iv) $\|\mathbf{E}\|, \|\mathbf{F}\|, \|\mathbf{G}\| \leq q_0/2$, and
(v) $\|\mathbf{x}\| \leq \alpha$.

Therefore

$$\begin{aligned}
\|\mathbf{u}_0\| &\leq 2 \cdot s^2 \cdot t^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^4 + 2 \cdot s^2 \cdot t^2 \cdot q_0 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3 + 2 \cdot s^2 \cdot t^2 \cdot q_0 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \\
&\leq 6 \cdot s^2 \cdot t^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^4 \\
&\leq \delta_0.
\end{aligned}$$

Conditions b_1, b_2, b_3, b_4, and b_5. We next consider the conditionsb_1, b_2, b_3, b_4, and b_5 in the verification algorithm. Clearly, it holds that

$$\begin{aligned}
\mathbf{D}_1 \cdot \mathbf{u}_1 &= \mathbf{D}_1 \cdot \left(\sum_{j \in [s_1+1;s]} \mathbf{u}_{1,j} \cdot x_j \right) = \sum_{j \in [s_1+1;s]} \mathbf{t}_1 \cdot v^j \cdot x_j = \mathbf{t}_1 \cdot c_{\mathbf{x}} \bmod q_3, \\
\mathbf{D}_2 \cdot \mathbf{u}_2 &= \mathbf{D}_2 \cdot \left(\sum_{i \in [t], j \in [s]} \mathbf{u}_{2,-i} \cdot E_{i,j} \cdot h_i \cdot x_j \right) = \sum_{i \in [t], j \in [s]} \mathbf{t}_2 \cdot v^{-i} \cdot E_{i,j} \cdot h_i \cdot x_j = \mathbf{t}_2 \cdot \bar{c}_{\mathbf{e}} \bmod q_3, \\
\mathbf{D}_3 \cdot \mathbf{u}_3 &= \mathbf{D}_3 \cdot \left(\sum_{i \in [t], j \in [s]} \mathbf{u}_{3,i} \cdot F_{i,j} \cdot x_j \right) = \sum_{i \in [t], j \in [s]} \mathbf{t}_3 \cdot v^i \cdot F_{i,j} \cdot x_j = \mathbf{t}_3 \cdot c_{\mathbf{f}} \bmod q_3, \\
\mathbf{D}_4 \cdot \mathbf{u}_4 &= \mathbf{D}_4 \cdot \left(\sum_{i \in [t], j \in [s]} \mathbf{u}_{4,i} \cdot G_{i,j} \cdot x_j \right) = \sum_{i \in [t], j \in [s]} \mathbf{t}_4 \cdot v^i \cdot G_{i,j} \cdot x_j = \mathbf{t}_4 \cdot c_{\mathbf{g}} \bmod q_3, \\
\mathbf{D}_5 \cdot \mathbf{u}_5 &= \mathbf{D}_5 \cdot \left(\sum_{i \in [t]} \mathbf{u}_{5,i} \cdot r_i \right) = \sum_{i \in [t]} \mathbf{t}_5 \cdot v^i \cdot r_i = \mathbf{t}_5 \cdot c_{\mathbf{r}} \bmod q_3.
\end{aligned}$$

Furthermore, since $\|\mathbf{u}_{k,i}\| \leq \beta$ for $k \in [5], j \in \pm[\max\{\, s, t \,\}]$, $\|\mathbf{h}\| \leq q_1/2$, $\|\mathbf{x}\| \leq \alpha$, and $\|\mathbf{r}\| \leq 2 \cdot s^2 \cdot q_0 \cdot \alpha^2 \cdot \gamma_{\mathcal{R}}^3$ we have

$$\begin{aligned}
\|\mathbf{u}_1\| &\leq s_2 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}} \leq \delta_1, \\
\|\mathbf{u}_2\| &\leq s \cdot t \cdot \cdot q_0 \cdot q_1 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^3 \leq \delta_2, \\
\|\mathbf{u}_3\| &\leq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2 \leq \delta_3, \\
\|\mathbf{u}_4\| &\leq s \cdot t \cdot q_0 \cdot \alpha \cdot \beta \cdot \gamma_{\mathcal{R}}^2 \leq \delta_4, \\
\|\mathbf{u}_5\| &\leq 2 \cdot s^2 \cdot q_0 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^4 \leq \delta_5.
\end{aligned}$$

Condition b_6. Finally, we consider the condition b_6 in the verification algorithm. Recall that $c_6 = \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{g}} \bmod q_3$ where

$$\begin{aligned}
\bar{c}_{\mathbf{e}} &= (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \mathbf{x} \bmod q_3, \\
c_{\mathbf{f}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{F} \cdot \mathbf{x} \bmod q_3 \\
c_{\mathbf{g}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{G} \cdot \mathbf{x} \bmod q_3 \\
c_{\mathbf{r}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} \bmod q_3 \\
\bar{c}_{\mathbf{I},6} &= \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{h} \bmod q_3.
\end{aligned}$$

Substituting the expressions of each component, we have

$$\begin{aligned}
c_6 &= (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{E} \cdot \mathbf{x} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{F} \cdot \mathbf{x} + q_0 \cdot \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{h} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} - \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{h} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{G} \cdot \mathbf{x} \bmod q_3 \\
&= (\bar{\mathbf{v}}_t \circ \mathbf{h})^{\mathtt{T}} \cdot \mathbf{e} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{f} + q_0 \cdot \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{h} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r} - \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{h} \cdot \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{g} \bmod q_3
\end{aligned}$$

$$\begin{aligned} &= \mathbf{e}^{\mathsf{T}} \cdot (\bar{\mathbf{v}}_t \circ \mathbf{h}) \cdot \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{f} + q_0 \cdot \mathbf{h}^{\mathsf{T}} \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{r} - \mathbf{h}^{\mathsf{T}} \cdot \bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{g} \bmod q_3 \\ &= \sum_{i,j \in [t], i \neq j} v^{i-j} \cdot (e_j \cdot f_i + q_0 \cdot r_i - g_i) \cdot h_j. \end{aligned}$$

where in the last equality we have used that $e_i \cdot f_i + q_0 \cdot r_i - g_i = 0$. Furthermore, since $\|\mathbf{u}_{6,i}\| \leq \beta$ for $j \in \pm[\max\{s, t\}]$, $\|\mathbf{h}\| \leq q_1/2$, $\|\mathbf{x}\| \leq \alpha$, and $\|\mathbf{r}\| \leq 2 \cdot s^2 \cdot q_0 \cdot \alpha^2 \cdot \gamma_{\mathcal{R}}^3$ we have

$$\|\mathbf{u}_6\| \leq 6 \cdot s^2 \cdot t^2 \cdot q_0^2 \cdot q_1 \cdot \alpha^2 \cdot \beta \cdot \gamma_{\mathcal{R}}^6 \leq \delta_6,$$

Putting everything together yields the claim. □

H.2 Knowledge Soundness

Theorem 17 (Knowledge Soundness). *Let (η, m, q_3, β) be such that the properties of lattice trapdoor algorithms described in Section 3.2 hold. Let $w = 1$, $\mathcal{G} := \{ X^j : -s \leq j \leq s \}$, $\mathcal{G}_0 = \{ X^i : i \in \pm[\max\{s,t\}] \}$, $\mathcal{G}_1 = \{ X^i : i \in [s_1; s] \}$, $\mathcal{G}_2 = \{ X^i : i \in -[t] \}$, $\mathcal{G}_3 = \{ X^i : i \in [t] \}$, $\mathcal{G}_4 = \{ X^i : i \in [t] \}$, $\mathcal{G}_5 = \{ X^i : i \in [t] \}$, and $\mathcal{G}_6 = \{ X^i : i \in \pm[t] \}$ be sets of monomials in X. Let $\mathcal{D}$ denote the distribution $\mathsf{SampD}(1^\lambda)$. For $i \in \{1,2,3,4,5,6\}$, let $\mathcal{Z}_i(1^\lambda)$ be almost identical to $\mathsf{Setup}(1^\lambda, \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda))$, except that it is given $(\mathbf{D}_i, \mathbf{t}_i, v, \{ \mathbf{u}_{i,j} \}_{j \in I_i})$ as input and generates the rest of crs. Let*

$$\begin{aligned} \alpha_i^* &\geq \delta_i, \ \forall i \in [6] \\ \alpha^* &:= \max\{ \alpha_1^*, \alpha_2^*, \alpha_3^*, \alpha_4^*, \alpha_5^*, \alpha_6^* \}, \\ q_1 \geq \beta_{q_1}^* &\geq s \cdot q_0 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}} \\ q_2 \geq \beta_{q_2}^* &\geq t \cdot s \cdot q \cdot q_1 \cdot \alpha^* \cdot \gamma_{\mathcal{R}}^2, \\ q_3 \geq \beta_{q_3} &\geq t \cdot s \cdot q_0 \cdot q_1 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}}^3, \\ q_3 \geq \beta_{q_3}^* &\geq \max\{ 2\delta_0, (s+t)^2 \cdot q_0 \cdot q_1 \cdot q_2 \cdot \alpha^* \cdot \beta \cdot \gamma_{\mathcal{R}}^4 \}. \end{aligned}$$

$\Pi^{\mathtt{R1CS}}$ in Fig. 4 is knowledge-sound for $\Psi^{\mathtt{R1CS}}[\alpha_1^*]$ if the following assumptions hold:

Assumption 0. k-R-$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\beta,\beta_{q_3}^*,\mathcal{G}_0,g^*=1,\mathcal{D},\mathcal{T}}$,
Assumption 1. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_1^*,\beta,\delta_1,\mathcal{G}_1,\mathcal{D},\mathcal{T},\mathcal{Z}_1}$,
Assumption 2. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_2^*,\beta,\delta_2,\mathcal{G}_2,\mathcal{D},\mathcal{T},\mathcal{Z}_2}$,
Assumption 3. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_3^*,\beta,\delta_3,\mathcal{G}_3,\mathcal{D},\mathcal{T},\mathcal{Z}_3}$,
Assumption 4. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_4^*,\beta,\delta_4,\mathcal{G}_4,\mathcal{D},\mathcal{T},\mathcal{Z}_4}$,
Assumption 5. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_5^*,\beta,\delta_5,\mathcal{G}_5,\mathcal{D},\mathcal{T},\mathcal{Z}_5}$,
Assumption 6. *knowledge-k-R-*$\mathsf{ISIS}_{\mathcal{R},\eta,m,w,q_3,\alpha_6^*,\beta,\delta_6,\mathcal{G}_6,\mathcal{D},\mathcal{T},\mathcal{Z}_6}$,
Assumption 7. R-$\mathsf{SIS}_{\mathcal{R},t,q_1,\beta_{q_1}^*}$,
Assumption 8. R-$\mathsf{SIS}_{\mathcal{R},3\cdot t,q_2,\beta_{q_2}^*}$, *and*
Assumption 9. $\mathsf{vSIS}_{\mathcal{R},\mathcal{G},1,q_3,\beta_{q_3}}$.

Proof. Fix a PPT prover $\mathcal{P}^*$. Consider an algorithm $\mathcal{B}_1 = \mathcal{B}^{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, runs $\pi \leftarrow \mathcal{P}^*(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, parses $c_{\mathbf{x}_2}$ and $\mathbf{u}_1$ from π, and outputs $(c_{\mathbf{x}_2}, \mathbf{u}_1)$. Similarly, consider the algorithms $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_4 = \mathcal{B}^{\mathcal{P}^*}$, $\mathcal{B}_5 = \mathcal{B}^{\mathcal{P}^*}$, and $\mathcal{B}_6 = \mathcal{B}^{\mathcal{P}^*}$ which do almost the same, except that

- $\mathcal{B}_2 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(\bar{c}_{\mathbf{e}}, \mathbf{u}_2)$ from π ,
- $\mathcal{B}_3 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{f}}, \mathbf{u}_3)$ from π,
- $\mathcal{B}_4 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{g}}, \mathbf{u}_4)$ from π,
- $\mathcal{B}_5 = \mathcal{B}^{\mathcal{P}^*}$ extracts $(c_{\mathbf{r}}, \mathbf{u}_5)$ from π, and
- $\mathcal{B}_6 = \mathcal{B}^{\mathcal{P}^*}$ parses $(\bar{c}_{\mathbf{e}}, c_{\mathbf{f}}, c_{\mathbf{r}}, c_{\mathbf{g}}, \mathbf{u}_6)$ from π, computes $\bar{c}_{\mathbf{I},6} := \bar{\mathbf{v}}_t^{\mathsf{T}} \cdot \mathbf{h}$ and

  $$c_{\mathbf{z}} := \bar{c}_{\mathbf{e}} \cdot c_{\mathbf{f}} + q_0 \cdot \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},6} \cdot c_{\mathbf{g}},$$

  and outputs $(c_{\mathbf{z}}, \mathbf{u}_6)$.

Let $\mathcal{E}_{\mathcal{B}_1}^{k\text{-}R\text{-}\mathsf{ISIS},1}$, $\mathcal{E}_{\mathcal{B}_2}^{k\text{-}R\text{-}\mathsf{ISIS},2}$, $\mathcal{E}_{\mathcal{B}_3}^{k\text{-}R\text{-}\mathsf{ISIS},3}$, $\mathcal{E}_{\mathcal{B}_4}^{k\text{-}R\text{-}\mathsf{ISIS},4}$, $\mathcal{E}_{\mathcal{B}_5}^{k\text{-}R\text{-}\mathsf{ISIS},5}$, and $\mathcal{E}_{\mathcal{B}_6}^{k\text{-}R\text{-}\mathsf{ISIS},6}$ be the knowledge extractors whose existence are guaranteed by Assumptions 1, 2, 3, 4, 5, and 6. Define an extractor $\mathcal{E}_{\mathcal{P}^*}$ which, on input $(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$, does the following:

- run $\mathbf{x}_2^\dagger \leftarrow \mathcal{E}_{\mathcal{B},1}^{k\text{-}R\text{-}\mathsf{ISIS},1}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{e}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},2}^{k\text{-}R\text{-}\mathsf{ISIS},2}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{f}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},3}^{k\text{-}R\text{-}\mathsf{ISIS},3}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{g}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},4}^{k\text{-}R\text{-}\mathsf{ISIS},4}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{r}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},5}^{k\text{-}R\text{-}\mathsf{ISIS},5}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- run $\mathbf{z}_{-0}^\dagger \leftarrow \mathcal{E}_{\mathcal{B},6}^{k\text{-}R\text{-}\mathsf{ISIS},6}(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$,
- check that $\mathbf{e}^\dagger = \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,
- check that $\mathbf{f}^\dagger = \mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$
- check that $\mathbf{g}^\dagger = \mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$
- check that $\left(\mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) \circ \left(\mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) + q_0 \cdot \mathbf{r}^\dagger = \left(\mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right)$, and
- output $\mathbf{x}_2^\dagger$ if all checks pass.

Fix any adversary $\mathcal{A}$ and consider the following experiment Exp:

```
$\mathsf{Exp}(1^\lambda)$
$\mathsf{pp} \leftarrow \mathsf{Gen}^{\mathtt{unstr}}(1^\lambda)$
$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathsf{pp})$
$(\mathsf{stmt}, \mathsf{wit}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{crs})$
$(\pi, \mathsf{wit}^\dagger) \leftarrow (\mathcal{P}^* | \mathcal{E}_{\mathcal{P}^*})(\mathsf{crs}, \mathsf{stmt}, \mathsf{wit})$
$\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}} \leftarrow \mathsf{PreVerify}(\mathsf{crs}, \mathsf{stmt}_{\mathtt{off}})$
$\mathbf{return}\ \mathsf{Verify}(\mathsf{crs}_{\mathsf{stmt}_{\mathtt{off}}}, \mathsf{stmt}_{\mathtt{on}}, \pi) = 1 \wedge (\mathsf{stmt}, \mathsf{wit}^\dagger) \notin \Psi_{\mathsf{pp}}$
```

We claim that $\Pr\left[\mathsf{Exp}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$, which proves the theorem.

To prove the claim, consider a modified experiment Exp' where in the setup $\mathsf{Setup}(1^\lambda, \mathsf{pp})$ the matrices $(\mathbf{D}_i)_{i=0}^6$ are sampled uniformly at random and the $\mathsf{SampPre}$ steps are replaced with sampling from SampD subject to the appropriate constraints. By the properties of $(\mathsf{TrapGen}, \mathsf{SampD}, \mathsf{SampPre})$, Exp' is statistically close to Exp. Therefore it suffices to show that $\Pr\left[\mathsf{Exp}'(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda)$.

We now examine $\mathsf{wit}^\dagger$ generated during the execution of $\mathsf{Exp}'(1^\lambda)$. Parse $\mathsf{stmt} = (\mathbf{x}_1, (\mathbf{E}, \mathbf{F}, \mathbf{G}))$ and $\mathsf{wit}^\dagger = \mathbf{x}_2^\dagger$. First, suppose that $\mathcal{E}_{\mathcal{P}^*}$ returns something, i.e.

(i) $\mathbf{e}^\dagger = \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,

(ii) $\mathbf{f}^\dagger = \mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$,

(iii) $\mathbf{g}^\dagger = \mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}$, and

(iv) $\left(\mathbf{E} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) \circ \left(\mathbf{F} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right) + q_0 \cdot \mathbf{r}^\dagger = \left(\mathbf{G} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2^\dagger \end{pmatrix}\right)$,

then by Conditions b_1, b_2, b_3, b_4, and b_5 of the verification algorithm and Assumptions 1, 2, 3, 4, and 5, we have

$$\begin{aligned}
c_{\mathbf{x}_2} &= \mathbf{v}_2^{\mathtt{T}} \cdot \mathbf{x}_2^\dagger \bmod q_3, & & & \left\|\mathbf{x}_2^\dagger\right\| &\leq \alpha_1^* \\
\bar{c}_{\mathbf{e}} &= \bar{\mathbf{v}}_t^{\mathtt{T}} \cdot \mathbf{e}^\dagger \bmod q_3, & & & \left\|\mathbf{e}^\dagger\right\| &\leq \alpha_2^* \\
c_{\mathbf{f}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{f}^\dagger \bmod q_3, & & & \left\|\mathbf{f}^\dagger\right\| &\leq \alpha_3^* \\
c_{\mathbf{g}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{g}^\dagger \bmod q_3, & & & \left\|\mathbf{g}^\dagger\right\| &\leq \alpha_4^* \\
c_{\mathbf{r}} &= \mathbf{v}_t^{\mathtt{T}} \cdot \mathbf{r}^\dagger \bmod q_3, & &\text{and} & \left\|\mathbf{r}^\dagger\right\| &\leq \alpha_5^*,
\end{aligned}$$

Let us first show that Item (i), Item (ii), and Item (iii) hold.

Let $\mathbf{x}^\dagger = \begin{pmatrix}\mathbf{x}_1 \\ \mathbf{x}_2^\dagger\end{pmatrix}$. Examining the condition b_0 in the verification algorithm, we observe

$$\begin{aligned}
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{E}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{E}} \cdot c_{\mathbf{x}} - c_{\mathbf{I},1} \cdot \bar{c}_{\mathbf{e}}) \\
&= \mathbf{t}_0 \cdot ((\boldsymbol{\ell}_1 \circ \mathbf{h})^{\mathsf{T}} \cdot \mathbf{E} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{x}^\dagger - \mathbf{v}_t^{\mathsf{T}} \cdot \boldsymbol{\ell}_1 \cdot \bar{\mathbf{v}}_t^{\mathsf{T}} \cdot \mathbf{e}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_1^{\mathsf{T}} \cdot (\mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\mathbf{v}_t \cdot \bar{\mathbf{v}}_t^\dagger - \mathbf{I}_t) \cdot \mathbf{e}^\dagger + \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \mathbf{x}^\dagger - \mathbf{e}^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{F}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{F}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},2} \cdot c_{\mathbf{f}}) \\
&= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_2^{\mathsf{T}} \cdot \mathbf{F} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{x}^\dagger - \bar{\mathbf{v}}_t^{\mathsf{T}} \cdot \boldsymbol{\ell}_2 \cdot \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{f}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_2^{\mathsf{T}} \cdot (\mathbf{F} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathsf{T}} - \mathbf{I}_t) \cdot \mathbf{f}^\dagger + \mathbf{F} \cdot \mathbf{x}^\dagger - \mathbf{f}^\dagger) \bmod q_3, \\
\mathbf{D}_0 \cdot \mathbf{u}_{0,\mathbf{G}} &= \mathbf{t}_0 \cdot (\bar{c}_{\mathbf{G}} \cdot c_{\mathbf{x}} - \bar{c}_{\mathbf{I},3} \cdot c_{\mathbf{g}}) \\
&= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_3^{\mathsf{T}} \cdot \mathbf{G} \cdot \bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} \cdot \mathbf{x}^\dagger - \bar{\mathbf{v}}_t^{\mathsf{T}} \cdot \boldsymbol{\ell}_3 \cdot \mathbf{v}_t^{\mathsf{T}} \cdot \mathbf{g}^\dagger) \\
&= \mathbf{t}_0 \cdot \boldsymbol{\ell}_3^{\mathsf{T}} \cdot (\mathbf{G} \cdot (\bar{\mathbf{v}} \cdot \mathbf{v}^{\mathsf{T}} - \mathbf{I}_s) \cdot \mathbf{x}^\dagger - (\bar{\mathbf{v}}_t \cdot \mathbf{v}_t^{\mathsf{T}} - \mathbf{I}_t) \cdot \mathbf{g}^\dagger + \mathbf{G} \cdot \mathbf{x}^\dagger - \mathbf{g}^\dagger) \bmod q_3.
\end{aligned}$$

Let

$$\begin{aligned}
\mathbf{u}_{0,\mathbf{E}}^\dagger &\coloneqq \sum_{i \in [t], j,k \in [s] : k \neq j} \ell_{1,i} \cdot h_i \cdot E_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k \in [t] : k \neq i} \ell_{1,k} \cdot \mathbf{u}_{0,k-i} \cdot e_i^\dagger \\
\mathbf{u}_{0,\mathbf{F}}^\dagger &\coloneqq \sum_{i \in [t], j,k \in [s] : k \neq j} \ell_{2,i} \cdot F_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k \in [t] : k \neq i} \ell_{2,i} \cdot \mathbf{u}_{0,i-k} \cdot f_k^\dagger \\
\mathbf{u}_{0,\mathbf{G}}^\dagger &\coloneqq \sum_{i \in [t], j,k \in [s] : k \neq j} \ell_{3,i} \cdot G_{i,j} \cdot \mathbf{u}_{0,k-j} \cdot x_k^\dagger + \sum_{i,k \in [t], j \in [s] : i \neq j} G_{i,j} \cdot x_j^\dagger \cdot \ell_{3,i} \cdot \mathbf{u}_{0,i-k} \cdot h_k,
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{w}_1^\dagger &\coloneqq \mathsf{diag}(\mathbf{h}) \cdot \mathbf{E} \cdot \mathbf{x}^\dagger - \mathbf{e}^\dagger \\
\mathbf{w}_2^\dagger &\coloneqq \mathbf{F} \cdot \mathbf{x}^\dagger - \mathbf{f}^\dagger \\
\mathbf{w}_3^\dagger &\coloneqq \mathbf{G} \cdot \mathbf{x}^\dagger - \mathbf{g}^\dagger
\end{aligned}$$

We have

$$\begin{aligned}
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{E}} - \mathbf{u}_{0,\mathbf{E}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_1^{\mathsf{T}} \cdot \mathbf{w}_1^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{F}} - \mathbf{u}_{0,\mathbf{F}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_2^{\mathsf{T}} \cdot \mathbf{w}_2^\dagger) \bmod q_3 \\
\mathbf{D}_0 \cdot (\mathbf{u}_{0,\mathbf{G}} - \mathbf{u}_{0,\mathbf{G}}^\dagger) &= \mathbf{t}_0 \cdot (\boldsymbol{\ell}_3^{\mathsf{T}} \cdot \mathbf{w}_3^\dagger) \bmod q_3.
\end{aligned}$$

Suppose, contrary to our claim, that $\mathbf{w}^\dagger \coloneqq (\mathbf{w}_1^\dagger, \mathbf{w}_2^\dagger, \mathbf{w}_3^\dagger) \neq \mathbf{0}$ with non-negligible probability. Then, one (or both) of the following must be true:

(1) $\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^\dagger = \mathbf{0}$ with non-negligible probability
(2) $\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^\dagger \neq \mathbf{0}$ with non-negligible probability,

where $\boldsymbol{\ell} \coloneqq (\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \boldsymbol{\ell}_3)$. If Case (1) is true, then we also have with non-negligible probability

$$\boldsymbol{\ell}^{\mathsf{T}} \cdot \mathbf{w}^\dagger = \mathbf{0} \bmod q_2.$$

Note that

$$\begin{aligned}
\left\|\mathbf{w}_1^\dagger\right\| &\leq t \cdot s \cdot q_1/2 \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}}^2 + \alpha_2^* \leq t \cdot s \cdot q_0 \cdot q_1 \cdot \gamma_{\mathcal{R}}^2 \cdot \alpha^* \\
\left\|\mathbf{w}_2^\dagger\right\| &\leq s \cdot q_0/2 \cdot \alpha_1^* \cdot \gamma_{\mathcal{R}} + \alpha_3^* \leq s \cdot q_0 \cdot \gamma_{\mathcal{R}} \cdot \alpha^*
\end{aligned}$$

$$\left\|\mathbf{w}_3^\dagger\right\| \le s\cdot q_0/2\cdot \alpha_1^*\cdot\gamma_{\mathcal{R}} + \alpha_3^* \le s\cdot q_0\cdot\gamma_{\mathcal{R}}\cdot\alpha^*,$$

Therefore $\left\|\mathbf{w}^\dagger\right\| \le t\cdot s\cdot q_0\cdot q_1\cdot\alpha^*\cdot\gamma_{\mathcal{R}}^2 \le \beta^*_{q_2}$. This would, however, violate Assumption 8. We thus conclude that Case (1) is impossible.

If Case (2) is true, we observe that for each $j\in\{\mathbf{E},\mathbf{F},\mathbf{G}\}$

$$\begin{aligned}\left\|\mathbf{u}_{0,j}^\dagger\right\| &\le 2\cdot t^2\cdot s\cdot q_0/2\cdot q_1/2\cdot q_2/2\cdot\beta\cdot\alpha^*\cdot\gamma_{\mathcal{R}}^4\\ &\le (s+t)^2\cdot q_0\cdot q_1\cdot q_2\cdot\alpha^*\cdot\beta\cdot\gamma_{\mathcal{R}}^4\\ &\le \beta^*_{q_3}/6,\\ \left\|\mathbf{u}_{0,j}-\mathbf{u}_{0,j}^\dagger\right\| &\le \beta^*_{q_3}/3\end{aligned}$$

Therefore

$$\left\|\mathbf{u}_{0,\mathbf{E}}-\mathbf{u}_{0,\mathbf{E}}^\dagger+\mathbf{u}_{0,\mathbf{F}}-\mathbf{u}_{0,\mathbf{F}}^\dagger+\mathbf{u}_{0,\mathbf{G}}-\mathbf{u}_{0,\mathbf{G}}^\dagger\right\| \le \beta^*_{q_3}.$$

Moreover

$$\left\|\boldsymbol{\ell}^{\mathsf{T}}\cdot\mathbf{w}^\dagger\right\| \le (t+s)^2\cdot q_2\cdot s\cdot q_0\cdot q_1\cdot\alpha^*\cdot\gamma_{\mathcal{R}} \le \beta^*_{q_3}.$$

This would, however, violate Assumption 0. We thus conclude that Case (2) is impossible.

It remains to show that Item (iv) also holds, so that $\mathcal{E}_{\mathcal{P}^*}$ returns something with overwhelming probability. Suppose, for the sake of contradiction, that this is not the case. Let $\hat{\mathbf{e}} := \mathbf{E}\cdot\mathbf{x}^\dagger$, i.e. $\mathbf{e}^\dagger = \mathsf{diag}(\mathbf{h})\cdot\hat{\mathbf{e}}$. Compute $z_0^\dagger := -\sum_{i\in[t]}(\hat{e}_i\cdot f_i^\dagger + q_0\cdot r_i^\dagger + g_i^\dagger)\cdot h_i$. Then

$$\begin{aligned}\left\|(\hat{e}_i\cdot f_i^\dagger + q_0\cdot r_i^\dagger + g_i^\dagger)_i\right\| &\le s\cdot q_0/2\cdot\alpha_1^*\cdot\alpha_3^*\cdot\gamma_{\mathcal{R}} + q_0\cdot\alpha_5^* + \alpha_4^*\\ &\le s\cdot q_0\cdot(\alpha^*)^2\cdot\gamma_{\mathcal{R}}\\ &\le \beta^*_{q_1}\end{aligned}$$

By Condition b_6 of the verification algorithm and Assumption 6, we have

$$\begin{aligned}(\bar{\mathbf{v}}||\mathbf{v})^{\mathsf{T}}\cdot\mathbf{z}_{-0}^\dagger &= c_{\mathbf{z}} \bmod q_3\\ &= \bar{c}_{\mathbf{e}}\cdot c_{\mathbf{f}} + q_0\cdot\bar{c}_{\mathbf{I},6}\cdot c_{\mathbf{r}} - \bar{c}_{\mathbf{I},6}\cdot c_{\mathbf{g}} \bmod q_3\\ &= \left(\sum_{j\in[t]} v^{-j}\cdot e_j^\dagger\right)\cdot\left(\sum_{i\in[t]} v^i\cdot f_i^\dagger\right) + q_0\cdot\left(\sum_{j\in[t]} v^{-j}\cdot h_j\right)\cdot\left(\sum_{i\in[t]} v^i\cdot r_i^\dagger\right)\\ &\qquad -\left(\sum_{i\in[t]} g_i^\dagger\cdot v^i\right)\cdot\left(\sum_{j\in[t]} h_j\cdot v^{-j}\right)\\ &= \sum_{i,j\in[t]}\hat{e}_j\cdot f_i^\dagger\cdot h_j\cdot v^{i-j} + q_0\cdot\sum_{i,j\in[t]} r_i^\dagger\cdot h_j\cdot v^{i-j} - \sum_{i,j\in[t]} g_i^\dagger\cdot h_j\cdot v^{i-j}\\ &= \sum_{i,j\in[t]}\left(\hat{e}_j\cdot f_i^\dagger + q_0\cdot r_i^\dagger - g_i^\dagger\right)\cdot h_j\cdot v^{i-j}.\end{aligned}$$

If $\mathbf{z}_{-0}^\dagger = (z_{-s}^\dagger,\ldots,z_{-1}^\dagger,z_1^\dagger,\ldots,z_s^\dagger)$, and we let $\mathbf{z}^\dagger := (z_{-s}^\dagger,\ldots,z_{-1}^\dagger,z_0^\dagger,z_1^\dagger,\ldots,z_s^\dagger)$, we obtain

$$(\bar{\mathbf{v}}||1||\mathbf{v})^{\mathsf{T}}\cdot\mathbf{z}^\dagger = \sum_{i,j\in[t],i\neq j}(\hat{e}_j\cdot f_i^\dagger + q_0\cdot r_i^\dagger - g_i^\dagger)\cdot h_j\cdot v^{i-j} \bmod q_3$$

and

$$\left\|\mathbf{z}^\dagger\right\| \le \max\{\alpha_6^*, t\cdot s\cdot q_0\cdot\alpha^*\cdot\gamma_{\mathcal{R}}^2\} \le \beta_{q_3}/2$$

On the other hand, if we define $\hat{\mathbf{z}}_{-0} = (\hat{z}_{-s},\ldots,\hat{z}_{-1},\hat{z}_0,\hat{z}_1,\ldots,\hat{z}_s)$ as

$$\hat{z}_0 := 0$$

$$\hat{z}_k := \sum_{i,j\in[t], i-j=k} \left(\hat{e}_j \cdot f_i^\dagger + q_0 \cdot r_i^\dagger - g_i^\dagger\right) \cdot h_j \quad \text{for } k \in \pm[s]$$

we have that

$$(\bar{\mathbf{v}}||1||\mathbf{v})^{\mathtt{T}} \cdot \hat{\mathbf{z}} = \sum_{i,j\in[t], i\neq j} (\hat{e}_j^\dagger \cdot f_i^\dagger + q_0 \cdot r_i^\dagger - g_i^\dagger) \cdot h_j \cdot v^{i-j} \bmod q_3,$$

and

$$\begin{aligned} \|\hat{\mathbf{z}}\| &\le t \cdot q_0/2 \cdot \alpha_2^* \cdot \alpha_3^* \cdot q_1 \cdot \gamma_{\mathcal{R}}^3 + q_0 \cdot q_1 \cdot \alpha_5 \cdot \gamma_{\mathcal{R}}^2 + q_0 \cdot \alpha_4 \cdot \gamma_{\mathcal{R}} \\ &\le t \cdot q_0 \cdot q_1 \cdot (\alpha^*)^2 \cdot \gamma_{\mathcal{R}}^3 \\ &\le \beta_{q_3}/2 \end{aligned}$$

Therefore

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\| \le \beta_{q_3}.$$

One (or both) of the following two cases must be true

(i) $\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) = 0$ with non-negligible probability.
(ii) $\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) \neq 0$ with non-negligible probability,

If Case (i) is true, we have

$$\sum_{i\in[t]} h_i \cdot (\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger) = 0 \bmod q_1 \qquad \text{and} \qquad 0 < \left\|(\hat{e}_i \cdot f_i^\dagger + q_0 \cdot r_i^\dagger + g_i^\dagger)_{i\in[s]}\right\| \le \beta_{q_1}$$

with non-negligible probability. This contradicts Assumption 7. If Case (ii) is true, we have

$$\left\langle (\bar{\mathbf{v}}||1||\mathbf{v}), \hat{\mathbf{z}} - \mathbf{z}^\dagger \right\rangle = 0 \bmod q_3 \qquad \text{and} \qquad 0 < \left\|\hat{\mathbf{z}} - \mathbf{z}^\dagger\right\| \le \beta_{q_3}$$

with non-negligible probability. This contradicts Assumption 9. Since none of the two cases could be true, we must have $(\mathbf{E} \cdot \mathbf{x}^\dagger) \circ (\mathbf{F} \cdot \mathbf{x}^\dagger) = \mathbf{G} \cdot \mathbf{x}^\dagger \bmod q_0$, as claimed. □

H.3 Efficiency

Theorem 18. *Let* $n = \max\{|\mathbf{E}|, |\mathbf{F}|, |\mathbf{G}|, s+t\}$, $\eta, \alpha, \beta, \gamma_{\mathcal{R}} = \mathsf{poly}(\lambda)$ *be a fixed polynomial in* λ, $(q_0, q_1, q_2, q_3) = (s, s^2, t \cdot s^4, (s+t)^{14}) \cdot \mathsf{poly}(\lambda)$, *and* $m = \log n \cdot \mathsf{poly}(\lambda)$. *Then* $\Pi^{\mathtt{bin\text{-}sat}}$ *has (i) common reference string size* $O_\lambda(n \cdot \log n)$, *(ii) proof size* $O_\lambda(\log^2 n)$, *(iii) prover time* $O_\lambda(n \cdot \log^3 n)$, *(iv) preprocessing time* $O_\lambda(n \cdot \log^2 n)$, *and (v) verifier time* $O_\lambda(\log^3 n)$ *after preprocessing.*

Proof. Note that $\log|\mathcal{R}_{q_3}| = \log q_3^{\varphi(\rho)} = O_\lambda(\log q_3) = O_\lambda(\log n)$, and an $\mathcal{R}_q$ operation takes at most $O_\lambda(\log^2 n)$ bit operations. Notice that $\mathbf{u}_{\mathbf{E}}, \mathbf{u}_{\mathbf{F}}, \mathbf{u}_{\mathbf{G}}, \mathbf{u}_{\mathbf{z}}$ can be computed in time $O_\lambda(n \cdot \log^3 n)$, exploiting fast multiplication algorithms for Toeplitz matrices (similarly to what described in Appendix F.3). All claims then follow by the same calculations as in Theorem 9. □

I Argument for Succinct-R1CS

In this section, we describe a folding-based succinct argument for succinct-R1CS [BCG+19], which captures computations involving iterative executions of small circuits, with quasi-linear-time prover and polylogarithmic-time verifier without preprocessing. The high-level idea of the construction is identical to that in Section 8, except that here we will consider linear relations represented by not just a single, but multiple, foldable matrices. To avoid distraction by having too many variables, we only provide a sketch of the construction.

Recall that a succinct-R1CS instance is given by $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{y})$ and a witness $\mathbf{x}$ satisfies

$$\begin{gathered} (\mathbf{A} \cdot \mathbf{x}) \circ (\mathbf{B} \cdot \mathbf{x}) = (\mathbf{C} \cdot \mathbf{x}) \bmod q_0 \\ \mathbf{D} \cdot \mathbf{x} = \mathbf{y} \bmod q_0 \end{gathered}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ representing the "time constraints" are of the form

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_0 & \mathbf{A}_1 & & \\ & \mathbf{A}_0 & \mathbf{A}_1 & \\ & & \ddots & \ddots \\ & & & \mathbf{A}_0 \; \mathbf{A}_1 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} \mathbf{B}_0 & \mathbf{B}_1 & & \\ & \mathbf{B}_0 & \mathbf{B}_1 & \\ & & \ddots & \ddots \\ & & & \mathbf{B}_0 \; \mathbf{B}_1 \end{pmatrix}, \qquad \mathbf{C} = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 & & \\ & \mathbf{C}_0 & \mathbf{C}_1 & \\ & & \ddots & \ddots \\ & & & \mathbf{C}_0 \; \mathbf{C}_1 \end{pmatrix}$$

and $(\mathbf{D}, \mathbf{y})$ represents the "boundary constraints". In the following, we outline a folding protocol for a variant of succinct-R1CS over $\mathcal{R}$ where $\mathbf{x}$ additionally satisfies a bounded-norm constraint $\|\mathbf{x}\| \leq \alpha$ and $\mathbf{D}$ (after removing the first and last block-columns) is foldable.

Let $s = w(n+2)$ denote the number of columns in $\mathbf{A}$ (and hence also in $\mathbf{B}$, $\mathbf{C}$, and $\mathbf{D}$). Similar to the strategy for proving R1CS, we let the prover commit to $\mathbf{a} = \mathbf{A} \cdot \mathbf{x}$, $\mathbf{b} = \mathbf{B} \cdot \mathbf{x}$, and $\mathbf{c} = \mathbf{C} \cdot \mathbf{x}$ as

- $\bar{c}_{\mathbf{h} \circ \mathbf{a}} = \bar{\mathbf{v}}^{\mathtt{T}} \cdot (\mathbf{h} \circ \mathbf{a}) \bmod q_3$,
- $c_{\mathbf{b}} = \mathbf{v}^{\mathtt{T}} \cdot \mathbf{b} \bmod q_3$, and
- $c_{\mathbf{c}} = \mathbf{v}^{\mathtt{T}} \cdot \mathbf{c} \bmod q_3$

respectively, where $\mathbf{h}$ is a foldable vector of norm $q_0 \ll \|\mathbf{h}\| \ll q_3$, and prove that

$$\begin{pmatrix} \mathbf{A} & -\mathbf{I} & & \\ \mathbf{B} & & -\mathbf{I} & \\ \mathbf{C} & & & -\mathbf{I} \\ \mathbf{D} & & & \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{y} \end{pmatrix} \bmod q_0, \tag{9}$$

$$\begin{pmatrix} \mathbf{0} & (\bar{\mathbf{v}} \circ \mathbf{h})^{\mathtt{T}} & & \\ \mathbf{0} & & \mathbf{v}^{\mathtt{T}} & \\ \mathbf{0} & & & \mathbf{v}^{\mathtt{T}} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \bar{c}_{\mathbf{h} \circ \mathbf{a}} \\ c_{\mathbf{b}} \\ c_{\mathbf{c}} \end{pmatrix} \bmod q_3, \tag{10}$$

and $\|(\mathbf{x}, \mathbf{a}, \mathbf{b}, \mathbf{c})\| \approx 0$. Observe that Eq. (9) is equivalent to $\mathbf{A} \cdot \mathbf{x} = \mathbf{a}$, $\mathbf{B} \cdot \mathbf{x} = \mathbf{b}$, $\mathbf{C} \cdot \mathbf{x} = \mathbf{c}$, and $\mathbf{D} \cdot \mathbf{x} = \mathbf{y}$ all modulo q_0, whereas Eq. (10) ensures that the commitments $\bar{c}_{\mathbf{h} \circ \mathbf{a}}$, $c_{\mathbf{b}}$, and $c_{\mathbf{c}}$ are well-formed. Then, we let the prover prove that $\mathbf{a} \circ \mathbf{b} = \mathbf{c}$ by proving the existence of

$$\mathbf{z} = \left(\sum_{0 \leq i,j, \leq s : j-i=k} h_i a_i b_j - h_i c_j \right)_{-s \leq k \leq s}$$

which satisfies

$$(\bar{\mathbf{v}} \mid \mathbf{v})^{\mathtt{T}} \cdot \mathbf{z} = \bar{c}_{\mathbf{h} \circ \mathbf{a}} \cdot c_{\mathbf{b}} - \bar{c}_{\mathbf{h}} \cdot c_{\mathbf{c}} \bmod q_3 \qquad \text{and} \qquad \|\mathbf{z}\| \approx 0. \tag{11}$$

Since Eqs. (10) and (11) are represented by foldable matrices, an adaption of the folding protocol in Section 6 applies. For Eq. (9), we need to handle one technical issue: The matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are not in the block-bidiagonal form which is supported by the folding protocol in Section 6. Taking $\mathbf{A}$ as an example, we observe that we have one $\mathbf{A}_0$ block extra at the top left, and one $\mathbf{A}_1$ block extra at the bottom right. To deal with this issue, we let the prover reveal the first and last blocks of $\mathbf{x}$, so that the verifier can subtract the contributions of these blocks from $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$. Letting $\mathbf{A}'$, $\mathbf{B}'$, $\mathbf{C}'$, and $\mathbf{D}'$ be derived from their counterparts with the first and last block-columns removed, we obtain a relation of the form

$$\begin{pmatrix} \mathbf{A}' & -\mathbf{I} & & \\ \mathbf{B}' & & -\mathbf{I} & \\ \mathbf{C}' & & & -\mathbf{I} \\ \mathbf{D}' & & & \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{y}_a \\ \mathbf{y}_b \\ \mathbf{y}_c \\ \mathbf{y}_d \end{pmatrix} \bmod q_0,$$

where $\mathbf{A}' = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{B}' = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{C}' = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_0 \end{bmatrix}_{\searrow n}$, $\mathbf{D}'$, $\mathbf{y}_a$, $\mathbf{y}_b$, $\mathbf{y}_c$, and $\mathbf{y}_d$ are foldable. We can therefore adapt the folding protocol in Section 6 to prove the statement.