SLAP: Succinct Lattice-Based Polynomial Commitments from Standard Assumptions

Martin R. Albrecht
martin.albrecht@{kcl.ac.uk,sandboxaq.com}
King's College London and SandboxAQ

Giacomo Fenzi
giacomo.fenzi@epfl.ch
EPFL

Oleksandra Lapiha
sasha.lapiha.2021@live.rhul.ac.uk
Royal Holloway, University of London

Ngoc Khanh Nguyen
khanh.nguyen@epfl.ch
EPFL

Abstract

Recent works on lattice-based extractable polynomial commitments can be grouped into two classes: (i) non-interactive constructions that stem from the functional commitment by Albrecht, Cini, Lai, Malavolta and Thyagarajan (CRYPTO 2022), and (ii) lattice adaptations of the Bulletproofs protocol (S&P 2018). The former class enjoys security in the standard model, albeit a knowledge assumption is desired. In contrast, Bulletproof-like protocols can be made secure under falsifiable assumptions, but due to technical limitations regarding subtractive sets, they only offer inverse-polynomial soundness error. This issue becomes particularly problematic when transforming these protocols to the non-interactive setting using the Fiat-Shamir paradigm.

In this work, we propose the first lattice-based non-interactive extractable polynomial commitment scheme which achieves polylogarithmic proof size and verifier runtime (in the length of the committed message) under standard assumptions. At the core of our work lies a new tree-based commitment scheme, along with an efficient proof of polynomial evaluation inspired by FRI (ICALP 2018). Natively, the construction is secure under a "multi-instance version" of the Power-Ring BASIS assumption (Eprint 2023/846). We then fully reduce security to the Module-SIS assumption by introducing several re-randomisation techniques which can be of independent interest.

1 Introduction

Zero-knowledge succinct non-interactive arguments of knowledge (zkSNARKs) [Kil92; Mic94] are a cryptographic primitive that allows a prover to produce a short proof that a statement is true without revealing any information beyond the validity of the statement itself. A particularly successful paradigm in the construction of zkSNARKs, which has been evident since the "canonical" construction of zkSNARKs [Mic94], is that of combining an *information theoretical proof system* with a *cryptographic compiler*. Originally, this was done via the combination of probabilistic checkable proof (PCPs) [BFLS91] and vector commitment schemes [Mer90]. While the zkSNARKs obtained from these ingredients were not concretely efficient (mostly due to the inefficiency of the PCP), [BCS16] iterated on this approach, introducing interactive oracle proofs (IOPs), and these efforts lead to many concretely deployed constructions (see e.g. [BBHR18b; BCRSVW19; AHIV22; GLSTW21; COS20]). Currently, some of the most efficient and widely deployed zkSNARKs are based on a similar ingredient combination, namely (i) polynomial IOPs (PIOPs) [CHMMVW20]; (ii) and polynomial commitment schemes [KZG10].

In this work, we focus on the cryptographic component of the above recipe, namely polynomial commitment schemes. A polynomial commitment scheme is a generalisation of vector commitments in which a prover is able to commit to any polynomial of bounded degree $f := \sum_{i=0}^{d} f_i \cdot \mathsf{X}^i$ over a ring $\mathcal{R}$, and then later produce a proof π that $f(u) = z$ for some public u, z. For the purpose of this work, we are concerned with polynomial commitment schemes that are *succinct* in both the size of the proof π and in the verification time, i.e. we wish both to be polylogarithmic in d. We aim the verification time to be polylogarithmic *without preprocessing* dependent on u, as that reflects the usage of polynomial commitment schemes in many PIOPs. Further, to obtain a SNARK, we will require that π is a *proof of knowledge*, and call a polynomial commitment scheme with this property *extractable*.

The literature on polynomial commitments [KZG10; BBBPWM18; BMMTV21; Lee21; BFS20; WTsTW17] is vast, but most of the existing construction rely on classical computational assumptions, and are thus insecure against quantum adversaries. If we require post-quantum security, the only concretely and asymptotically efficient polynomial commitment schemes that are known are based on the FRI IOP of Proximity [BBHR18a], compiled into an argument via the BCS construction [BCS16] (and thus relying on the random oracle heuristic for its computational security).

A natural question is whether using post-quantum computational assumptions with "more structure" can lead to more efficient plausibly post-quantum secure polynomial commitment schemes. Constructions based on hard lattice problems have been successfully deployed in various areas of cryptography, as evidenced by the NIST PQC competition, which concluded by standardising lattice-based solutions for both key encapsulation mechanisms and signatures.

Constructions from functional commitments. Recently, a number of lattice-based polynomial commitment schemes were introduced, mostly as a result of promising work on linear functional commitments [ACLMT22; WW23b; CP23; PPS21; BCFL22; FLV23; CLM23]. At a high level, by interpreting the evaluation $f(u) = z$ as the linear relation $\langle \mathbf{f}, \mathsf{pow}(u) \rangle = z$, where $\mathsf{pow}(u) = (1, u, \ldots, u^d)$, those schemes naturally lead to polynomial commitments. A significant limitation in that paradigm is that, without preprocessing, the verification cost (when performed naively[1]) is linear in d and thus not succinct. Since in most PIOPs of interest u cannot be preprocessed, this limits

[1] The recent work of Fisch, Liu and Vesely [FLV23] manages to avoid this issue.

the applicability of those schemes. Furthermore, only [ACLMT22; BCFL22; CLM23; FLV23] offer extractability, albeit under a knowledge k-M-ISIS assumption [ACLMT22]. However, this knowledge k-M-ISIS assumption has been recently shown to be (at least "morally") broken [WW23a].

Split-and-fold protocols. Another line of research on lattice-based (interactive) polynomial commitments with succinct verification [BCS23; CLM23] stems from the lattice adaptation [BLNS20] of the Bulletproofs interactive protocol [BBBPWM18]. Unfortunately, both constructions inherit the inverse-polynomial soundness error from [BLNS20]. Although parallel repetition can be applied in the interactive setting for soundness amplification [AF22], this strategy incurs a super-polynomial security loss when applying the Fiat-Shamir transformation in the random oracle model, as shown by Attema, Fehr and Klooß [AFK22].

Independently, Fenzi and Nguyen [FN23] extended a commitment scheme introduced by Wee and Wu [WW23b] to efficiently prove polynomial evaluations with a split-and-fold approach from FRI [BBHR18a]. The authors pick an exponential-sized challenge space, which results in negligible soundness error in one-shot. As a downside, due to the enormous norm growth of extracted witnesses along with slack, which has direct influence on the proof system modulus and other lattice parameters, [FN23] only manages to achieve *quasi*-polylogarithmic proof and verifier complexity. Also, their scheme inherits several practical inefficiencies from [WW23b], i.e. common reference string (CRS) and committing time that are undesirably large, as they are quadratic in the degree of the committed polynomial. Further, the binding property of the commitment scheme relies on the non-standard Power-Ring-BASIS (PRISIS) assumption[2], which was not reduced to more well-understood lattice problems.

In this work, we iterate on this previous construction, and study the following open question.

Can we construct a non-interactive, extractable polynomial commitment scheme with polylogarithmic communication and verifier complexity; (quasi)-linear prover time; negligible knowledge soundness error and whose security relies on standard lattice assumptions?

1.1 Our Contributions

We present SLAP, the first lattice-based polynomial commitment scheme that achieves the above goals.

Merkle-PRISIS commitment scheme. Our starting point is the PRISIS-based commitment scheme of Fenzi and Nguyen [FN23] which is compressing and supports arbitrarily large messages. We will use it as a subroutine to build a Merkle tree. Concretely, for a message $\mathbf{f} = (f_j)_{j\in[d]} \in \mathcal{R}^d$ where $d = 2^h$, we consider vectors $\mathbf{t}_{h,j} := f_j \cdot \mathbf{e}$ to be the leaves of the Merkle tree, where $\mathbf{e} \in \mathcal{R}^n$ is a fixed vector defined later to argue binding. Then, given the i-th layer $(\mathbf{t}_{i,j})_{j\in[2^i]}$ of the tree, we commit to the pairs of the form $(\mathbf{t}_{i,2j-1}, \mathbf{t}_{i,2j}) \in \mathcal{R}^{2n}$ to obtain PRISIS commitments $(\mathbf{t}_{i-1,j})_{i\in[2^{i-1}]} \in \mathcal{R}^n$. The final commitment is the root $\mathbf{t}_{1,1} \in \mathcal{R}^n$.

An immediate advantage over the original construction of [FN23] is a quasi-linear commitment time. Indeed, we only apply the PRISIS commitment scheme for constant-sized messages. Hence, the only non-constant cost comes from actually building the Merkle tree, which is quasi-linear[3].

[2] Technically, the construction can be based either on PRISIS, or more general PowerBASIS assumption.

[3] The *quasi* part comes from the fact that performing operations over $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^N + 1)$ takes polylog(d) time.

Furthermore, since the commitment key for each layer is of constant size, the common reference string has size only $\text{polylog}(d)$.

Security under Module-SIS via new re-randomisation techniques. The binding property of our commitment scheme holds under a "multi-instance" version of the PRISIS assumption, which we call h-PRISIS. Recall that a PRISIS problem [FN23] belongs to the class of "SIS-with-hints" problems [ACLMT22; BLNS23; WW23b], i.e. given a matrix $\mathbf{A}$ with some a hint aux, find a short non-zero vector $\mathbf{s}$ such that $\mathbf{A} \cdot \mathbf{s} = \mathbf{0}$. In the multi-instance version of "SIS-with-hints", the adversary is given $h \geq 2$ pairs of challenges $(\mathbf{A}_i, \mathsf{aux}_i)_{i \in [h]}$ generated as above, and the goal is to find a short non-zero vector such that $[\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_h] \cdot \mathbf{s} = \mathbf{0}$.

In this work, we introduce new re-randomisation techniques to argue that for a certain type of "SIS-with-hints" problems, the multi-instance version is no easier than the single-instance version. In particular, in Section 3.2, we give a reduction showing that h-PRISIS is no easier than PRISIS for $h = \mathsf{poly}(\lambda)$. These re-randomisation tricks and reductions might be of independent interest.

Finally, we apply the result from [FN23] which says that a single-instance PRISIS for "small parameters" is no easier than Module-SIS [LS15]. Since we work with such parameters when building a Merkle tree of arity two, we conclude that the binding property of the Merkle-PRISIS commitment scheme holds under the Module-SIS assumption.

Polynomial evaluation protocol and negligible soundness. As a next step, we augment the new commitment scheme with a $\log d$-round *interactive* polynomial evaluation proof, that applies a "split-and-fold" approach from FRI [BBHR18a]. The protocol has polylogarithmic communication and verifier complexity, while also simultaneously achieving negligible knowledge soundness error *without parallel repetition*.

The crucial difference from [FN23] is that here we apply soundness amplification via batching, namely proving multiple statements of the form $f_i(u) = z_i$ simultaneously. We note that a similar approach was applied in the setting of groups of unknown order [BHRRS21]. But unlike in [BHRRS21], to argue knowledge soundness, we do not aim to find a short (right-)inverse but instead we prove that our protocol satisfies coordinate-wise special soundness [FN23]. This results in concretely smaller blow-up in parameters. Finally, we show how to achieve zero-knowledge by applying the standard Fiat-Shamir-with-aborts paradigm for lattices [BTT22].

Our construction natively supports polynomials over the standard power-of-two cyclotomic rings $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^N + 1)$. In order to make it compatible with current Polynomial IOPs, the commitment scheme should be able to commit and prove evaluations over finite fields. To this end, we apply the $\mathbb{Z}_q$-to-$\mathcal{R}_q$ transformation demonstrated by Lyubashevsky, Nguyen and Plançon [LNP22] to map a polynomial evaluation statement over $\mathbb{Z}_q$ to one over $\mathcal{R}_q$, where the polynomial in the latter case has degree N times smaller. This technique is generic and could be applied to subsequent lattice-based constructions for more efficiency.

1.2 Related Works

We provide a literature review on the existing publicly-verifiable succinct interactive proof systems from lattices. Bootle, Lyubashevsky, Nguyen and Seiler [BLNS20] introduced the first lattice-based adaptation of the Bulletproofs protocol [BCCGP16; BBBPWM18] over polynomial rings $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$, achieving polylogarithmic proof sizes. This methodology was later improved in the context of soundness analysis [ACK21; AL21], and generalised to the bilinear module setting [BCS21].

The core protocol, unfortunately, has the following three drawbacks. First, the verification time is linear in the witness size. Second, soundness error of the protocol is only $1/\mathsf{poly}(\lambda)$, and thus soundness amplification is necessary. The reason for this limitation is a technical requirement on the challenge space, where any two distinct challenges have to be invertible over the ring, and what is more, its (scaled) inverse has to be short (such sets are called subtractive). As demonstrated in [AL21], such sets can only have polynomial size. Thirdly, we need to account for slack and a huge norm blow-up when performing knowledge extraction. In the case of lattice Bulletproofs, this boils down to inverting a 3×3 Vandermonde matrix for each round, and thus the extracted witness suffers a blow-up in the order of $\mathsf{poly}(\lambda)^{3h}$, where h is the number of rounds. Since, for security purposes, the proof system modulus q has to be larger than the norm of the extracted witness, we conclude that q must be super-polynomial.

Recently, Bootle, Chiesa and Sotiraki [BCS23] and Cini, Lai and Malavolta [CLM23] independently proposed variants of the lattice Bulletproofs protocol which achieve polylogarithmic verification time. The work of [BCS23] avoids linear-time verification via a delegation protocol inspired by Dory [Lee21]. On the other hand, [CLM23] introduces additional power structure on the Ajtai commitment [Ajt96], which enables succinct verification at the cost of a new assumption called Vanishing-SIS. The aforementioned works still inherit the latter two problems above.

Bünz and Fisch [BF22] considered a modified protocol, where instead of subtractive sets, the challenges are simply integers in the range $[0, 2^{\lambda-1})$. Then, using a new knowledge extraction strategy called "almost special soundness", the authors manage to achieve negligible soundness error – also in the non-interactive setting. However, the protocol still maintains linear-time verification and suffers from large extracted norm growth.

Fenzi and Nguyen [FN23] moved away from the lattice Bulletproofs template, and instead considered a commitment scheme based on a power variant of the BASIS assumption [WW23b]. The algebraic structure of the commitment allows proving polynomial evaluations using the FRI-type split-and-fold approach, rather than the one from Bulletproofs. An immediate consequence of this change is the norm blow-up in the order of $\mathsf{poly}(\lambda)^{h}$, since the knowledge extractor now only needs to account for the norm growth from inverting a 2×2 Vandermonde matrix. This comes at the cost of a trusted setup. The authors propose two concrete instantiations, which either achieve polylogarithmic verification time (using subtractive sets), or negligible soundness error (using an exponential-size challenge space) – but not both.

Beullens and Seiler [BS23] proposed an interactive proof, which successfully combines ideas from lattice-based batch arguments [BBCdGL18; NS22] with algebraic techniques from the non-interactive zero-knowledge (NIZK) framework by Lyubashevsky, Nguyen and Plançon [LNP22]. The protocol achieves asymptotically polylogarithmic proof size, and concretely ≈ 50KB proofs for circuits of size 2^{20}. The key to achieving such small proofs in practice is proving exactly that the short witness $\mathbf{s}$ satisfies a relation $\|\mathbf{s}\| = \beta$ without any blow-up in the extracted norm. As in [LNP22], the approach is to prove that $\|\mathbf{s}\| \ll \sqrt{q}$ via an *approximate range proof* [LNS21; GHL22], and $\|\mathbf{s}\|^2 = \beta^2 \bmod q$ as a quadratic equation over $\mathbb{Z}_q$. Here, Beullens and Seiler observed that both claims are folding-friendly and can thus be proven efficiently using recursion. As a downside, the protocol does not support succinct verification due to the approximate range proof part.

In this work, we address the three limitations stated above: (i) our construction achieves polylogarithmic verification time, (ii) the protocol enjoys negligible soundness error via a batching

Table 1: Comparison of lattice-based publicly verifiable (interactive) polynomial commitments for polynomials of degree at most d with polylogarithmic communication complexity.

scheme	assumption	TP	soundness error	time		size		stretch × slack
				prover	verifier	crs	proof	
[BLNS20]	M-SIS	✓	$1/\mathsf{poly}(\lambda)$	$O(d)$	$O(d)$	$O(1)$	$O(\log d)$	$O(d^{3\log N})$
[BCS23]	M-SIS	✓	$1/\mathsf{poly}(\lambda)$	$O(d)$	$O(\log^2 d)$	$O(1)$	$O(\log^2 d)$	$O(d^{6\log N})$
[CLM23]	vSIS	✓	$1/\mathsf{poly}(\lambda)$	$O(d)$	$O(\log d)$	$O(1)$	$O(\log d)$	$O(d^{4\log N})$
[BF22]	(M-)SIS	✓	$\mathsf{negl}(\lambda)$	$O(d)$	$O(d)$	$O(1)$	$O(\log d)$	$O(d^{\log d+2\lambda})$
[BS23]	M-SIS	✓	$\mathsf{negl}(\lambda)$	$O(d)$	$O(d)$	$O(1)$	$O(\log d)$	$O(1)$
[FN23]	PowerBASIS	✗	$1/\mathsf{poly}(\lambda)$	$O(d^2)$	$O(\log d)$	$O(d^2)$	$O(\log d)$	$O(d^{\log N})$
SLAP	M-SIS	✗	$\mathsf{negl}(\lambda)$	$O(d)$	$O(\log^2 d)$	$O(\log d)$	$O(\log^2 d)$	$O(d^{\log N})$

We count the runtime (resp. sizes) in the number of operations (resp. elements) in $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^N+1)$, which take time (resp. size) polylog(d) each. We ignore the terms related polynomially in the security parameter λ. The "TP" column specifies whether the scheme has transparent setup. The "stretch × slack" column denotes the term $\gamma_{\mathsf{stretch}} \cdot \gamma_{\mathsf{slack}}$ defined in (1). For presentation, we only include the terms that are super-polynomial in d.

argument, and (iii) thanks to the FRI split-and-fold structure, we suffer the norm blow-up in the order of $\mathsf{poly}(\lambda)^h$ rather than $\mathsf{poly}(\lambda)^{3h}$ as in the Bulletproofs-type protocols. Unfortunately, the algebraic structure of our commitment comes at a price of a trusted setup.

We summarise the comparison with prior works in Table 1. To estimate concrete efficiency from asymptotic statements, we estimate so-called "stretch" and "slack" of the protocols. That is, in every lattice-based proof system there is a part where the prover wants to prove knowledge of a short vector $\mathbf{s}$ such that $\mathbf{A}\cdot\mathbf{s} = \mathbf{t}$ and $\|\mathbf{s}\| \le \beta$. However, due to technical reasons, it is the case that one can only extract a slightly larger vector $\mathbf{z}$, along with a scalar c for which

$$\mathbf{A}\cdot\mathbf{z} = c\cdot\mathbf{t} \quad \text{and} \quad \|\mathbf{z}\| \le \gamma_{\mathsf{stretch}}\cdot\beta \quad \text{and} \quad \|c\| \le \gamma_{\mathsf{slack}}. \tag{1}$$

In the literature, $\gamma_{\mathsf{stretch}}$ is called stretch, and c is the slack. The term $\gamma_{\mathsf{stretch}}\cdot\gamma_{\mathsf{slack}} \in \mathbb{R}_+$ often indicates how efficient the underlying protocol is, because, for security, the proof system modulus has to be larger than $(\gamma_{\mathsf{stretch}}\cdot\gamma_{\mathsf{slack}})\cdot\beta$. For instance, [BS23] achieves $\gamma_{\mathsf{stretch}}, \gamma_{\mathsf{slack}} \in O(1)$, which results in very small proof sizes.

1.3 Technical Overview

We provide a brief overview of our techniques. First, let us recall some notation. We write $\mathbb{Z}_2^h$ for the set of binary strings of length h, and let $\mathbb{Z}_2^{\le h} := \cup_{0\le j\le h}\mathbb{Z}_2^j$. For $\mathbf{b} = (b_1, \ldots, b_h) \in \mathbb{Z}_2^h$ and $j \in [h]$ we let $\mathbf{b}_{:j} = (b_1, \ldots, b_j) \in \mathbb{Z}_2^j$. Let λ be a security parameter, q be an odd prime, and N be a power-of-two. Define the polynomial rings $\mathcal{R} := \mathbb{Z}[X]/(X^N+1)$ and $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^N+1)$. Letting $\delta \ge 2$ be a (fixed) base and $n \ge 1$, we define the gadget matrix as $\mathbf{G}_n := \begin{bmatrix}1 & \delta & \cdots & \delta^{\tilde{q}}\end{bmatrix} \otimes \mathbf{I}_n \in \mathcal{R}_q^{n\times n\tilde{q}}$ where $\tilde{q} := \lfloor \log_\delta q \rfloor + 1$. For simplicity, we omit the subscript n and write $\mathbf{G} := \mathbf{G}_n$ when clear from context. In our context, a *trapdoor* for a matrix $\mathbf{B}$ is a short matrix such that $\mathbf{B}\cdot\mathbf{T} = \mathbf{G}$. In particular, knowledge of a trapdoor of $\mathbf{B}$ enables to sample (random) short preimages of $\mathbf{B}$, i.e.

short $\mathbf{v}$ such that $\mathbf{B} \cdot \mathbf{v} = \mathbf{t}$ for a given image $\mathbf{t}$ [MP12]. We also let $\mathbf{e}$ be a fixed vector, that will be used later to argue binding.

1.3.1 Merkle-PRISIS commitment schemes

A polynomial commitment scheme naturally consists of two components: (i) a commitment scheme; (ii) an evaluation protocol. In order to achieve succinct verification, the commitment scheme has to be *compressing*. Further, in order to commit to *arbitrary* polynomials, we would want the commitment scheme to work for any message over $\mathcal{R}_q$ and not only elements of small norm. Our starting point is the following two-to-one commitment scheme, whose security relies on BASIS-style assumptions introduced in [WW23b] and revisited in [FN23].

The common reference string consists of a triple $(\mathbf{A}, w, \mathbf{T})$, where $\mathbf{T}$ is a trapdoor to the matrix $\mathbf{B} := \left[\begin{array}{cc|c} \mathbf{A} & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w \cdot \mathbf{A} & -\mathbf{G} \end{array}\right]$ and $w \in \mathcal{R}_q^\times$. To commit to a vector $\mathbf{f} = (f_0, f_1) \in \mathcal{R}_q^2$, the committer sets $\mathbf{t}_b = f_b \cdot \mathbf{e}$ and uses $\mathbf{T}$ to sample short vectors $\mathbf{s}_0, \mathbf{s}_1, \hat{\mathbf{t}}$ such that

$$\mathbf{B} \cdot \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \hat{\mathbf{t}} \end{bmatrix} = \begin{bmatrix} -\mathbf{t}_0 \\ -\mathbf{t}_1 \end{bmatrix} \quad .$$

The final commitment is then set to be $\mathbf{t} := \mathbf{G} \cdot \hat{\mathbf{t}}$. To verify an opening, which consists of $\mathbf{s}_0, \mathbf{s}_1$, the verifier simply checks whether the induced constraints are satisfied, namely by checking that the openings are short and that $w^b \cdot \mathbf{A} \cdot \mathbf{s}_b + \mathbf{t} = \mathbf{t}_b$ for $b \in \{0, 1\}$. The commitment scheme can be naturally extended to handle messages of arbitrary length, and indeed variants of this construction are at the core of the results in both [WW23b] and [FN23]. However, these natural extensions incur some drawbacks. First, the common reference string has size *quadratic* in the message length. Second, binding holds under BASIS-style assumptions in parameter regimes that are not known to reduce to standard assumptions.

Conversely, the commitment scheme that we sketched is binding under a version of the PRISIS assumption of arity 2, which was shown in [FN23] to reduce to the *standard* Module-SIS assumption. The commitment scheme that we introduce in the work, which we refer to as Merkle-PRISIS commitment, addresses both of these drawbacks by applying the sketched commitments scheme iteratively in a Merkle-tree fashion. It is compressing for messages of arbitrary length in $\mathcal{R}_q$, binding follows under the Module-SIS assumption, has polylogarithmic common reference string, and quasi-linear commitment time.

For simplicity, assume that $\ell = 2^h$. The common reference string for the Merkle-PRISIS commitment scheme (when used with messages of length ℓ) consists of h common reference strings for the base commitment scheme, which we denote as $(\mathbf{A}_i, w_i, \mathbf{T}_i)_{i \in [h]}$. To commit to a message $\mathbf{f} \in \mathcal{R}_q^\ell$, which we index by $\mathbb{Z}_2^h$, the committer applies the basic commitment scheme to $(f_{\mathbf{b},0}, f_{\mathbf{b},1}) \in \mathcal{R}_q^2$ for $\mathbf{b} \in \mathbb{Z}_2^{h-1}$, obtaining $\ell/2$ commitments $\mathbf{t}_\mathbf{b}$ (and corresponding openings). Recall that this is done by setting $\mathbf{t}_{\mathbf{b},b_h} := f_{\mathbf{b},b_h} \cdot \mathbf{e}$ and then sampling an appropriate preimage. In the next layer, we apply again the commitment scheme to these resulting commitments, *without scaling by* $\mathbf{e}$. This process is repeated h times, until a single commitment is obtained. Denoting by $\mathbf{t}$ this final commitment, and by $(\mathbf{s}_\mathbf{b})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}}$ the openings, checking a valid opening involves checking shortness of the openings

```
PRISIS.Sample_ℓ(A):
1. Sample[a] $\mathbf{a}^\top \leftarrow \mathcal{R}_q^m$ and set $\bar{\mathbf{A}} := \begin{bmatrix} \mathbf{a}^\top \\ \mathbf{A} \end{bmatrix}$. We write $\mathsf{PRISIS}_\ell := \mathsf{BASIS}[\mathsf{PRISIS.Sample}_\ell]$.
2. Sample $w \leftarrow \mathcal{R}_q^\times$.
3. Set
   $\mathbf{B} := \left[\begin{array}{ccc|c} w^0 \cdot \bar{\mathbf{A}} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & w^{\ell-1} \cdot \bar{\mathbf{A}} & -\mathbf{G} \end{array}\right]$ and $\mathsf{aux} := w$ .
4. Return $\mathbf{B}, \mathsf{aux}$.
```

[a] This vector $\mathbf{a}$ will be used in concert with $\mathbf{e}$ to argue binding later. Refer to Section 4.1 for details.

Figure 1: Sampling algorithm for the PRISIS assumption.

and that the following equation for each of the ℓ authentication paths $\mathbf{b} \in \mathbb{Z}_2^h$

$$\sum_{j \in [h]} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}:j} + \mathbf{f}_\mathbf{b} \cdot \mathbf{e} = \mathbf{t} \ .$$

It is easy to verify that the commitment scheme has the claimed efficiency properties. What might be surprising is that, despite the fact that the inner instantiations of the commitment schemes are not *individually* binding, we show that the overall scheme is binding under a multi-instance version of the PRISIS assumption, which we discuss next, and reduce to standard assumptions.

1.3.2 Power-Ring-BASIS Assumption

We first recall the BASIS family of assumptions, introduced in [WW23b]. Informally, each of the assumptions states that it should be hard for an adversary to find a short non-zero element in the kernel of a random matrix *even when given a trapdoor* to a matrix related to the target matrix.

Definition 1.1 (Informal)**.** *Let* Samp *be an algorithm that, when given* $\mathbf{A} \in \mathcal{R}_q^{n \times m}$*, outputs* $(\mathbf{B}, \mathsf{aux}) \leftarrow \mathsf{Samp}(\mathbf{A})$*. The* BASIS[Samp] *assumption states that an efficient adversary, given access to* $(\mathbf{A}, \mathbf{B}, \mathsf{aux}, \mathbf{T})$*, where* $\mathbf{A}$ *is a random matrix and* $\mathbf{T}$ *is a trapdoor for* $\mathbf{B}$*, is not able to compute a short non-zero solution* $\mathbf{z}$ *to* $\mathbf{A} \cdot \mathbf{z} \equiv \mathbf{0} \bmod q$*.*

The choice of the Samp algorithm affects the hardness of the assumption. In this work, we consider the PRISIS assumption, introduced in [FN23], which is obtained from BASIS when the sampling algorithm is defined as in Figure 1.

As mentioned before, binding of the Merkle-PRISIS commitment scheme follows from a *multi-instance* version of the PRISIS_2 assumption. We give a general definition for multi-instance BASIS assumptions, in which the adversary is given a number of BASIS instances, and aims to find a short non-zero solution to the matrix obtained by concatenating the challenge matrices of the individual instances.

Definition 1.2 (Informal)**.** *Let* Samp *be an algorithm as before. Let* $\mathbf{A}_1, \ldots, \mathbf{A}_h$ *be random matrices in* $\mathcal{R}_q$*, and suppose further that* $\mathbf{B}_i, \mathsf{aux}_i \leftarrow \mathsf{Samp}(\mathbf{A}_i)$ *and that* $\mathbf{T}_i$ *is a trapdoor for* $\mathbf{B}_i$*. The h-*

$\mathsf{BASIS}[\mathsf{Samp}]$ *assumption states that no efficient adversary, given* $(\mathbf{A}_i, \mathbf{B}_i, \mathsf{aux}_i, \mathbf{T}_i)_i$*, can find a short non-zero vector* $\mathbf{z}$ *such that* $[\mathbf{A}_1|\ldots|\mathbf{A}_h] \cdot \mathbf{z} \equiv \mathbf{0} \bmod q$*.*

The PRISIS version is defined as $h\text{-}\mathsf{PRISIS}_\ell := h\text{-}\mathsf{BASIS}[\mathsf{PRISIS.Sample}_\ell]$. In Section 3.2 we show that this multi-instance version of PRISIS is as hard as the single-instance version. Combining this with the result in [FN23, Sec 3.1], which reduces PRISIS_2 to Module-SIS, implies that binding of Merkle-PRISIS follows from standard assumption. We alternatively show a tighter reduction directly from $h\text{-}\mathsf{PRISIS}_2$ to Module-SIS in Section 3.1.

1.3.3 Evaluation protocol

Next, we focus on the second component of our polynomial commitment scheme, namely the evaluation protocol. First, we present a Σ-protocol, resembling that in [FN23], that reduces checking an opening of a committed degree d polynomial to that of one of degree $d/2$. Our final protocol will apply this Σ-protocol recursively, which will suffice to achieve polylog(d) communication and verifier complexity.

Again for simplicity, assume that $d = 2^h - 1$, and let $(\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}$ be a Merkle-PRISIS common reference string as before. Given a public commitment $\mathbf{t}$, an evaluation point u and a claimed image z, the prover aims to show knowledge of a polynomial f of degree at most d such that $f(u) = z$, and that $\mathbf{t}$ is a commitment to f.

The protocol, as the previous work, follows a FRI-inspired split-and-fold approach. The prover will split the witness vector into odd and even components, and send over some evaluations and partial openings. The verifier will sample randomness which will be used to update the witness to a random linear combination of those components. Prover and verifier will jointly (and efficiently) updated their reference string and commitment to one of the folded polynomial. We describe the Σ-protocol in Figure 2.

Correctness, follows easily, if not for some cumbersome notation. We index the coefficients of f, g with binary strings as before. Thus, the $\mathbf{j}$-th coefficient of g is $g_\mathbf{j} = \alpha_0 f_{0,\mathbf{j}} + \alpha_1 f_{1,\mathbf{j}}$ for $\mathbf{j} \in \mathbb{Z}_2^{h-1}$. Now, by expanding the verification equations, we have

$$\begin{aligned}
g_\mathbf{j} \cdot \mathbf{e} &= (\alpha_0 \cdot f_{0,\mathbf{j}} + \alpha_1 \cdot f_{1,\mathbf{j}}) \cdot \mathbf{e} \\
&= \alpha_0 \cdot \left(\mathbf{t} - w_1^0 \cdot \mathbf{A}_1 \cdot \mathbf{s}_0 - \sum_{t=1}^{h-1} w_{1+t}^{j_t} \cdot \mathbf{A}_{1+t} \cdot \mathbf{s}_{0,\mathbf{j}_{:t}}\right) + \alpha_1 \cdot \left(\mathbf{t} - w_1^1 \cdot \mathbf{A}_1 \cdot \mathbf{s}_1 - \sum_{t=1}^{h-1} w_{1+t}^{j_t} \cdot \mathbf{A}_{1+t} \cdot \mathbf{s}_{1,\mathbf{j}_{:t}}\right) \\
&= \alpha_0 \cdot \big(\mathbf{t} - w_1^0 \cdot \mathbf{A}_1 \cdot \mathbf{s}_0\big) + \alpha_1 \cdot \big(\mathbf{t} - w_1^1 \cdot \mathbf{A}_1 \cdot \mathbf{s}_1\big) - \sum_{t=1}^{h-1} w_{1+t}^{j_t} \cdot \mathbf{A}_{1+t} \cdot (\alpha_0 \cdot \mathbf{s}_{0,\mathbf{j}_{:t}} + \alpha_1 \cdot \mathbf{s}_{1,\mathbf{j}_{:t}}) \\
&= \mathbf{t}' - \sum_{t=1}^{h-1} \cdot w_{1+t}^{j_t} \cdot \mathbf{A}_{1+t} \cdot \mathbf{z}_{\mathbf{j}_{:t}} \quad .
\end{aligned}$$

Note also that $g(u^2) = \alpha_0 \cdot f_0(u^2) + \alpha_1 \cdot f_1(u^2) = \alpha_0 \cdot z_0 + \alpha_1 \cdot z_1$. Further, the updated openings are scaled by monomials, and thus remain short. We are able to straightforwardly generalise the protocol to an arbitrary "folding factor", that we denote as k. This parameter regulates in how many components the polynomial is divided into in a round of the protocol. More concretely, the original degree d polynomial is split into 2^k polynomials of degree roughly $d/2^k$, which are then folded as in the original protocol. Applying the protocol recursively logarithmically many times (in

Basic Σ-Protocol

Prover		**Verifier**
$f(\mathsf{X}) = f_0(\mathsf{X}^2) + \mathsf{X} f_1(\mathsf{X}^2)$ $z_i \coloneqq f_i(u^2)$ for $i \in \mathbb{Z}_2$		
	$\xrightarrow{z_0, z_1, \mathbf{s}_0, \mathbf{s}_1}$	Check: $z_0 + uz_1 =_? z$ Check: $\mathbf{s}_0, \mathbf{s}_1$ short
	$\xleftarrow{\alpha_0, \alpha_1}$	$\alpha_0, \alpha_1 \leftarrow \{ X^i : i \in \mathbb{Z} \}$
$g(\mathsf{X}) \coloneqq \alpha_0 f_0(\mathsf{X}) + \alpha_1 f_1(\mathsf{X})$ $\mathbf{z_b} \coloneqq \alpha_0 \mathbf{s}_{\mathbf{b},0} + \alpha_1 \mathbf{s}_{\mathbf{b},1}$ for $\mathbf{b} \in \mathbb{Z}_2^{\leq h-1}$		
	$\xrightarrow{g, (\mathbf{z_b})_\mathbf{b}}$	$\mathsf{crs}' \coloneqq (\mathbf{A}_{1+t}, w_{1+t}, \mathbf{T}_{1+t})_{t \in [h-1]}$ $\mathbf{t}' \coloneqq \alpha_0 \cdot \big(\mathbf{t} - w_1^0 \mathbf{A}_1 \mathbf{s}_0\big) + \alpha_1 \cdot \big(\mathbf{t} - w_1^1 \mathbf{A}_1 \mathbf{s}_1\big)$ $u' \coloneqq u^2$ $z' \coloneqq \alpha_0 \cdot z_0 + \alpha_1 \cdot z_1$ Check: $g(u') = z'$ Check: $\mathsf{Open}(\mathsf{crs}', \mathbf{t}', g, (\mathbf{z_b})_\mathbf{b}) = 1$

Figure 2: Σ-protocol to check evaluations of a degree d polynomial committed under Merkle-PRISIS.

d), we are able to obtain an interactive evaluation protocol with communication complexity and verification complexity polylogarithmic.

Knowledge soundness of the Σ protocol follows from techniques similar to those in [FN23], inheriting the limitation that the knowledge soundness error is $\frac{2^k}{2N}$, where $N = \mathsf{poly}(\lambda)$ and thus non-negligible. In the interactive setting, we could boost this via parallel repetition, but since our aim will to construct non-interactive arguments through the Fiat-Shamir transform another approach is required.

To this end, we combine the amortisation techniques from [BBCdGL18; BHRRS21]. Rather than proving a single claim $f(u) = z$, we consider a protocol for a bundle of r claims $\{ f_i(u) = z_i \}_{i \in [r]}$ (note that the evaluation points are *the same* across claims). The new Σ-protocol takes as input r polynomials of degree d, applies the same "split" strategy to obtain $r \cdot 2^k$ polynomials of degree roughly $d/2^k$. These polynomials are combined into r new ones via a redundant linear combination, which induces a new bundle of claim to be recursively proven.

We apply this new Σ-protocol ℓ times recursively, and analyse the resulting protocol using coordinate-wise special soundness. We show that the knowledge soundness error is roughly $\ell \cdot r 2^k / (2\,N)^r$. Setting r large enough, we can thus achieve negligible knowledge soundness error. Applying the Fiat-Shamir transformation is then sound, and we obtain a non-interactive protocol for proving multiple polynomial evaluations. The single polynomial case is then handled by proving the *same* claim multiple times with the resulting protocol. An appropriate setting of parameters then implies our main result.

Our protocols, as they are, natively provide evaluations proofs of polynomials over $\mathcal{R}_q$. We present a new generic technique to make use of such evaluations proof to provide evaluations proofs over $\mathbb{Z}_q$. The techniques, allow to prove evaluation of degree d polynomials over $\mathbb{Z}_q$ by making use of evaluation protocols of degree d/N in $\mathcal{R}_q$, leading to significant savings in practice. At a high level, we make use of the observation in [LNP22] that $\mathcal{R}$ has an automorphism $\sigma : \mathcal{R} \to \mathcal{R}$ such that, for $a, b \in \mathcal{R}$, the constant coefficient of $a \cdot \sigma(b)$ equals the inner product of the coefficient vectors of a and b. We then make use of this fact to "pack" the coefficients of the original polynomial in $\mathbb{Z}_q$ in one of smaller degree over $\mathcal{R}_q$, embedding the original claim in the constant coefficient of this new polynomial.

2 Preliminaries

We denote the security parameter by λ, which is implicitly given to all algorithms unless specified otherwise. Further, we write $\mathsf{negl}(\lambda)$ (resp. $\mathsf{poly}(\lambda)$) to denote an unspecified negligible function (resp. polynomial) in λ. In this work, we implicitly assume that the vast majority of the key parameters, e.g. the ring dimension, and the dimensions of matrices and vectors, are $\mathsf{poly}(\lambda)$. However, the modulus used in this work may be super-polynomial in λ.

For $a, b \in \mathbb{N}$ with $a < b$, write $[a, b] \coloneqq \{a, a+1, \ldots, b\}, [a] \coloneqq [1, a]$. For $q \in \mathbb{N}$ write $\mathbb{Z}_q$ for the integers modulo q. We denote vectors with lowercase boldface (e.g. $\mathbf{u}, \mathbf{v}$) and matrices with uppercase boldface (e.g. $\mathbf{A}, \mathbf{B}$). For a vector $\mathbf{x}$ of length n, we write x_i or $\mathbf{x}[i]$ for its i-th entry. Similarly, we define $\mathbf{x}_{:i} \coloneqq (x_1, \ldots, x_i)$, $\mathbf{x}_{i:} \coloneqq (x_i, \ldots, x_n)$ and $\mathbf{x}_{i:j} \coloneqq (x_i, \ldots, x_j)$ for $i, j \in [n]$. Given two vectors $\mathbf{u}, \mathbf{v}$, we denote by $(\mathbf{u}, \mathbf{v})$ its concatenation. Also, ε is an empty string. Given two matrices $\mathbf{A}, \mathbf{B}$ we write $[\mathbf{A} \mid \mathbf{B}]$ for their concatenation and $[\mathbf{A} \| \mathbf{B}]$ for stacking them on top of each other i.e. $[\mathbf{A} \| \mathbf{B}] \coloneqq [\mathbf{A}^T \mid \mathbf{B}^T]^T$.

Bits-to-integer conversion. Let $k \geq 1$. For a vector $\mathbf{b} \in \mathbb{Z}_2^k$, we define the bits-to-integer conversion function $\mathsf{int}(\mathbf{b}) \coloneqq \sum_{i=1}^{k} b_i \cdot 2^{i-1} \in [0, 2^k - 1]$. Clearly, if $\mathbf{u} \in \mathbb{Z}_2^k$ and $\mathbf{v} \in \mathbb{Z}_2^l$ then $\mathsf{int}((\mathbf{u}, \mathbf{v})) = \mathsf{int}(\mathbf{u}) + 2^k \cdot \mathsf{int}(\mathbf{v})$.

Norms. We define the ℓ_p norm on $\mathbb{C}^n$ as $\|\mathbf{x}\|_p = (\sum_i |x_i|^p)^{1/p}$ for $p < \infty$ and $\|\mathbf{x}\|_\infty \coloneqq \max_i |x_i|$. Unless otherwise specified, we use $\|\cdot\|$ for the ℓ_2 norm. We let the norm of a matrix be defined as the norm taken over the concatenation of columns of the matrix.

2.1 Lattices

A subset $\Lambda \subseteq \mathbb{R}^m$ is a lattice if: (i) $\mathbf{0} \in \Lambda$, and for $\mathbf{x}, \mathbf{y} \in \Lambda$, $\mathbf{x} + \mathbf{y} \in \Lambda$, and (ii) for every $\mathbf{x} \in \Lambda$, there exists $\varepsilon > 0$ such that $\{\mathbf{y} \in \mathbb{R}^m : \|\mathbf{x} - \mathbf{y}\| < \varepsilon\} \cap \Lambda = \{\mathbf{x}\}$. We say $\mathbf{B} \in \mathbb{R}^{n \times m}$ is a basis for Λ if its columns are linearly independent and $\Lambda = \mathcal{L}(\mathbf{B}) \coloneqq \{\mathbf{B} \cdot \mathbf{z} : \mathbf{z} \in \mathbb{Z}^m\}$. The span (as a vector space) of the basis of a lattice is the span of a lattice denoted as $\mathrm{Span}(\Lambda)$. We also let Λ^* be the dual lattice defined as $\Lambda^* = \{\mathbf{w} \in \mathrm{Span}(\Lambda) : \langle \Lambda, \mathbf{w} \rangle \subseteq \mathbb{Z}\}$. We denote by $\tilde{\mathbf{B}}$ the Gram-Schmidt orthogonalisation of $\mathbf{B}$. The Gram-Schmidt norm of $\mathbf{B}$ is defined as $\|\tilde{\mathbf{B}}\| \coloneqq \max_{i \in [k]} \|\tilde{\mathbf{b}}_i\|$ where $\tilde{\mathbf{b}}_i$ is the i-th column of $\tilde{\mathbf{B}}$.

2.2 Discrete Gaussian Distributions

Let $\sigma > 0$ be a parameter and Λ be a m-dimensional lattice. We then define the discrete Gaussian distribution $\mathcal{D}_{\Lambda,\sigma,\mathbf{c}}$ over a lattice coset $\mathbf{c} + \Lambda$ as follows.

$$\rho_{\sigma,\mathbf{c}}(\mathbf{z}) \coloneqq \exp\left(-\frac{\pi \|\mathbf{z} - \mathbf{c}\|^2}{\sigma^2}\right) \text{ and } \mathcal{D}_{\Lambda,\sigma,\mathbf{c}}(\mathbf{z}) \coloneqq \frac{\rho_{\sigma,\mathbf{c}}(\mathbf{z})}{\sum_{\mathbf{x} \in \Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})} \ .$$

When $\mathbf{c} = \mathbf{0}$ or $\Lambda = \mathbb{Z}^m$, we will omit either from the notation. We naturally extend this notion for lattices over the ring of integers of number fields (see below), and for matrices by sampling column-wise.

Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ be a matrix over some ring $\mathcal{R}_q$ and take any $\mathbf{u} \in \mathcal{R}_q^n$. We write $\mathbf{s} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{u})$ to denote sampling $\mathbf{s} \leftarrow \mathcal{D}_\sigma^m$ conditioned on $\mathbf{A} \cdot \mathbf{s} \equiv \mathbf{u} \bmod q$.

```
RejSamp:
$(\mathbf{u}, \mathbf{v}) \leftarrow h$
$\mathbf{z} \leftarrow \mathcal{D}^{mN}_{\Lambda,\sigma,\mathbf{v}+\mathbf{u}}$
return $(\mathbf{u}, \mathbf{v}, \mathbf{z})$ with prob. $\min\left(\frac{\mathcal{D}^m_\sigma(\mathbf{z})}{M \cdot \mathcal{D}^m_{\sigma,\mathbf{v}}(\mathbf{z})}, 1\right)$
```

```
SimRS:
$(\mathbf{u}, \mathbf{v}) \leftarrow h$
$\mathbf{z} \leftarrow \mathcal{D}^{mN}_{\Lambda,\sigma,\mathbf{u}}$
return $(\mathbf{u}, \mathbf{v}, \mathbf{z})$ with prob. $\frac{1}{M}$
```

Figure 3: Rejection sampling [BTT22].

We make use of the following lemmas on discrete Gaussians, bounding the norm of vectors sampled from that distribution, and the entropy of the distribution.

Lemma 2.1 (Implicit in Cor 5.5 in [Pei07])**.** *For any parameter m, a lattice $\Lambda \subset \mathbb{R}^m$ and a constant $\sigma > 0$ we have:*

$$\Pr_{\mathbf{u} \leftarrow \mathcal{D}_{\Lambda,\sigma}}\left[\|\mathbf{u}\|_\infty > \sigma \cdot \omega(\sqrt{\log m})\right] = \mathsf{negl}(m) \ .$$

Lemma 2.2 (Implicit in Lem 2.10 in [PR06])**.** *Let Λ be a full rank lattice in $\mathbb{R}^N$. Let $\varepsilon > 0$ and $\sigma \geq \eta_\varepsilon(\Lambda)$. Let $\mathbf{y} \in \Lambda$ then*

$$\Pr_{\mathbf{y}' \leftarrow \mathcal{D}_{\Lambda,\sigma}}\left[\mathbf{y} = \mathbf{y}'\right] \leq (\sigma^N \det(\Lambda^*)(1 - \varepsilon))^{-1} \ .$$

2.3 Smoothing Parameter

The smoothing parameter $\eta_\varepsilon(\Lambda)$ of a lattice is the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^*) \leq 1 + \varepsilon$. Below we recall the standard upper-bound on the smoothing parameter [MR07; GPV08].

Lemma 2.3. *Let $\Lambda \subset \mathbb{R}^m$ be a lattice, and let $\varepsilon > 0$. Then, for every basis $\mathbf{B}$ of Λ,*

$$\eta_\varepsilon(\Lambda) \leq \|\tilde{\mathbf{B}}\| \cdot \sqrt{\frac{\ln(2\,m \cdot (1 + 1/\varepsilon))}{\pi}} \ .$$

2.4 Rejection Sampling

We recall the generalised version of rejection sampling for discrete Gaussian over arbitrary lattices as introduced recently in [BTT22] (here we skip the general case for ellipsoidal Gaussians).

Lemma 2.4 (Rejection Sampling [BTT22])**.** *Take any $\alpha, T > 0$ and $\varepsilon \leq 1/2$. Let $\Lambda \subseteq \mathcal{R}^m$ be a lattice over some ring $\mathcal{R}$ and $\sigma \geq \max(\alpha \cdot T, \eta_\varepsilon(\Lambda))$ be a parameter. Let $h : \mathcal{R}^m \times \mathcal{R}^m \to [0, 1]$ be a probability distribution which returns $(\mathbf{u}, \mathbf{v})$ where the vector $\mathbf{v}$ satisfies $\|\mathbf{v}\| \leq T$.*[4] *Further, define $M := \exp(\frac{\pi}{\alpha^2} + 1)$ and $\varepsilon := 2\frac{1+\varepsilon}{1-\varepsilon}\exp(-\alpha^2 \cdot \frac{\pi-1}{\pi^2})$. Then, the statistical distance between distributions* RejSamp *and* SimRS *defined in Figure 3 is at most $\frac{\varepsilon}{2M} + \frac{2\varepsilon}{M}$. Moreover, the probability that* RejSamp *outputs something is at least $\frac{1-\varepsilon}{M} \cdot \left(1 - \frac{4\varepsilon}{(1+\varepsilon)^2}\right)$.*

2.5 Power-of-Two Cyclotomic Rings

Let N be a power-of-two and $\mathcal{K} = \mathbb{Q}[X]/(X^N + 1)$ be the $2N$-th cyclotomic field. Denote $\mathcal{R} := \mathbb{Z}[X]/(X^N + 1)$ to be the ring of integers of $\mathcal{K}$. We write $\mathcal{N}(\mathfrak{f})$ for the algebraic norm over $\mathbb{Q}$ of the

[4] One may think of $\mathbf{u}$ (resp. $\mathbf{v}$) as the public (resp. private) shift.

ideal $\mathfrak{f} \subset \mathcal{R}$. for the algebraic norm of For an odd prime q, we write $\mathcal{R}_q := \mathcal{R}/(q)$. We denote $\mathcal{R}_q^\times$ to be the set of invertible elements in $\mathcal{R}_q$. Let ℓ be a divisor of N such that $q \equiv 2N/\ell + 1 \bmod 4N/\ell$. Then, by [LS18, Corollary 1.2], the polynomial $X^N + 1$ factors as

$$X^N + 1 \equiv \prod_{i=1}^{N/\ell} (X^\ell - r_i) \bmod q \ ,$$

for distinct $r_i \in \mathbb{Z}_q^\times$, where all $X^\ell - r_i$ are irreducible in the ring $\mathbb{Z}_q[X]$. Hence, the ideal $\langle q \rangle$ of $\mathcal{R}$ can be written as a product of prime ideals:

$$\langle q \rangle = \prod_{i=1}^{N/\ell} \langle q, X^\ell - r_i \rangle \ .$$

Moreover, by the Chinese Remainder Theorem and a union bound, the probability of $w \leftarrow \mathcal{R}_q$ being non-invertible can be bounded by $N/(\ell \cdot q^\ell)$. In this work, we fix $q \equiv 5 \bmod 8$, so that $\ell = N/2$. In particular, if $N = O(\lambda)$ then $\mathcal{R}_q$ splits into exponentially large fields.

For $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{x} \in \mathcal{R}_q^m$, we define the q-ary lattices (or lattice cosets) as $\Lambda_{\mathbf{u}}^\perp(\mathbf{A}) := \{\mathbf{z} \in \mathcal{R}^m : \mathbf{A} \cdot \mathbf{z} \equiv \mathbf{u} \bmod q\}$. We omit the subscript $\mathbf{u}$ if $\mathbf{u} = \mathbf{0}$.

Lemma 2.5 (Prop. 2 of [AL21]). *In the power-of-2 cyclotomic ring $\mathcal{R}$ of degree N*

$$\max_{a_0,\ a_1 \in \mathcal{R}} \frac{\|a_0 \cdot a_1\|_\infty}{\|a_0\|_\infty \cdot \|a_1\|_\infty} \leq N \ .$$

Next, we define $\mathcal{X} := \{1, X, \ldots, X^{2N-1}\}$ to be the set of monomials in $\mathcal{R}$. We use the result from [BCKLN14, Lemma 3.1] which says that the (scaled) inverse of any two distinct monomials has small coefficients.

Lemma 2.6 (Lemma 3.1 of [BCKLN14]). *Let $0 \leq i < j < 2N$. Then, $2/(X^i - X^j) \bmod (X^N + 1)$ has coefficients in $\{-1, 0, 1\}$.*

2.6 Module-SIS

We recall the standard Module-SIS [LS15] assumption.

Definition 2.7 (Module-SIS). *Let $q = q(\lambda)$, $n = n(\lambda)$, $m = m(\lambda)$, $\beta = \beta(\lambda)$ and $N = N(\lambda)$. We say that the $\mathsf{MSIS}_{n,m,\mathcal{R}_q,\beta}$ assumption holds if for any PPT adversary $\mathcal{A}$, the following holds:*

$$\Pr\left[\ \mathbf{A} \cdot \mathbf{z} = \mathbf{0} \wedge 0 < \|\mathbf{z}\| \leq \beta \ \middle|\ \begin{array}{c} \mathbf{A} \leftarrow \mathcal{R}_q^{n \times m} \\ \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A}) \end{array} \right] \leq \mathsf{negl}(\lambda) \ .$$

2.7 Leftover Hash Lemmas over Rings

We will rely on a leftover hash lemma over rings, both explicitly and implicitly when performing trapdoor sampling.

Lemma 2.8 (Cor. 4.2 of [LPR13]). *For a cyclotomic number field of degree N, integers $q \geq 2$ and $n \leq m \leq poly(\lambda)$. Let $\mathbf{A} = [\mathbf{Id}_n | \bar{\mathbf{A}}] \in \mathcal{R}_q^{n \times m}$, where $\bar{\mathbf{A}}$ is sampled uniformly at random. Then with probability $1 - 2^{-\Omega(N)}$ over the choice of $\bar{\mathbf{A}}$ the distribution of $\mathbf{A} \cdot \mathbf{u} \bmod q$ where $\mathbf{u} \leftarrow \mathcal{D}_{\mathcal{R}^m,\sigma}$ with $\sigma \geq 2\,N \cdot q^{n/m + 2/(N \cdot m)}$ is within statistical distance $2^{-\Omega(N)}$ of the uniform distribution over $\mathcal{R}_q^n$.*

Corollary 2.9. *Let $\mathcal{S}_{n,m} \subset \mathcal{R}_q^{n\times m}$, $m \geq n$ be a space of all matrices in $\mathcal{R}_q^{n\times m}$ that contain an invertible $n \times n$ submatrix. Let matrix $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{S}_{n,m})$, and a vector $\mathbf{x} \leftarrow \mathcal{D}_{\mathcal{R}^m,\sigma}$ with $\sigma \geq 2\,N \cdot q^{n/m+2/(N\cdot m)}$ then*

$$\Delta\left((\mathbf{A}, \mathbf{A}\cdot\mathbf{x} \bmod q),\ (\mathcal{U}(\mathcal{S}_{n,m}),\ \mathcal{U}(\mathcal{R}_q^n))\right) \leq 2^{-\Omega(N)}\ .$$

Since the above statement is only for matrices containing an invertible submatrix, we need to prove that for our parameters at least a constant fraction of all matrices are in $S(n,m)$.

Lemma 2.10 (Lemma 2.5 of [BJRW23])**.** *Let n, k, m, q be positive integers such that q is unramified prime and factors as $\langle q\rangle = \prod_{i=1}^{\kappa} \mathfrak{p}_i$ in $\mathcal{R}$. Let $1 \leq k \leq m$ and $\mathbf{a}_1, \ldots, \mathbf{a}_k \in \mathcal{R}_q^m$ be a set of $\mathcal{R}_q$ linearly independent vectors. Then*

$$\Pr_{b\leftarrow\mathcal{U}(\mathcal{R}_q^m)}[\mathbf{a}_0, \ldots, \mathbf{a}_{k-1}, \mathbf{b} \textit{ are } \mathcal{R}_q\textit{-lin. indep.}] = \prod_{i=1}^{\kappa}\left(1 - \frac{1}{\mathcal{N}(\mathfrak{p}_i)^{m-k}}\right)\ .$$

As a result for any $1 \leq n \leq m$, it holds that

$$\Pr_{(\mathbf{a}_i)_i\leftarrow\mathcal{U}(\mathcal{R}_q^m)^n}[\mathbf{a}_1, \ldots, \mathbf{a}_n \textit{ are } \mathcal{R}_q\textit{-lin. indep.}] = \prod_{k=0}^{n-1}\prod_{i=1}^{\kappa}\left(1 - \frac{1}{\mathcal{N}(\mathfrak{p}_i)^{m-k}}\right)\ .$$

Remark 2.11. In this work, we pick x^N+1 to split into two factors, so $\kappa = 2$ and $\forall i : \mathcal{N}(\mathfrak{p}_i) = q^{N/2}$. As a consequence, elements in $\mathcal{R}_q$ are invertible and linear independence holds with overwhelming probability.

Remark 2.12. Let $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{n\times m})$ and $1 \leq n \leq m$. Then with overwhelming probability all matrix elements are invertible and its rows are linearly independent. Then with overwhelming probability the Gaussian elimination algorithm can find an inverse for some $n \times n$ submatrix of $\mathbf{A}$. Therefore, $\mathbf{A} \in S(n,m)$ with overwhelming probability.

2.8 Trapdoor Sampling

Gadget matrix. Let $\delta \geq 2$. We set $\tilde{q} := \lfloor \log_\delta q\rfloor + 1$, $\mathbf{g}^\top = [1, \delta, \ldots, \delta^{\tilde{q}-1}] \in \mathcal{R}_q^{1\times\tilde{q}}$ and $\mathbf{G}_n := \mathbf{I}_n \otimes \mathbf{g}^\top \in \mathcal{R}_q^{n\times(n\cdot\tilde{q})}$. When the dimensions are clear from context we simply write $\mathbf{G}$. Write $\mathbf{G}_n^{-1} : \mathcal{R}_q^{n\times t} \to \mathcal{R}_q^{(n\cdot\tilde{q})\times t}$ for the inverse function that takes a matrix of entries in $\mathcal{R}_q$, and decomposes each entry w.r.t. the base δ. We also write $\mathbf{g}^{-1}$ for $\mathbf{G}_1^{-1}$. Next, we recall the trapdoor generation from [MP12; FN23] and make use of Remark 2.12.

Lemma 2.13 (Trapdoor Generation)**.** *Let $\mathcal{R}_q$ split into fields of super-polynomial size. Let $N, n > 0, t = n \cdot \tilde{q}$ and $\mathbf{G}_n \in \mathcal{R}_q^{n\times t}$ be the gadget matrix. Take $m \geq t + n$. Then, there is a PPT algorithm $\mathsf{TrapGen}(n,m)$ that with an overwhelming probability returns two matrices $(\mathbf{A}, \mathbf{R}) \in \mathcal{R}_q^{n\times m} \times \mathcal{R}_q^{m\times t}$ such that $\mathbf{A}\cdot\mathbf{R} \equiv \mathbf{G}_n \bmod q$ and $\|\mathbf{R}\| \leq \sigma \cdot \sqrt{2\,t\cdot(m-t)\cdot N}$ where $\sigma > 2\,N \cdot q^{\frac{n}{m-t}+\frac{2}{N\cdot(m-t)}}$. Moreover, $\mathbf{A}$ is statistically close to a uniformly random matrix in $\mathcal{R}_q^{n\times m}$.*

The next lemma [MP12] states that given a short $\mathbf{G}$-trapdoor matrix $\mathbf{R}$ for $\mathbf{A}$, we can efficiently sample preimages of $\mathbf{A}$ according to the discrete Gaussian distribution. We further merge the result with the tail-bound inequality from [MR07].

Lemma 2.14 (Preimage Sampling)**.** *Let $\mathcal{R}_q$ split into fields of super-polynomial size. Let $N, n, m > 0$, $t = n \cdot \tilde{q}$ and $k = \max(n, m)$. Then, there exists a PPT algorithm $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \sigma)$ that takes as input a matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$, a $\mathbf{G}_n$-trapdoor $\mathbf{R} \in \mathcal{R}_q^{m\times t}$ for $\mathbf{A}$ with a tag $\mathbf{H}$, a target vector $\mathbf{v} \in \mathcal{R}_q^n$ in the column-span of $\mathbf{A}$, and a Gaussian parameter σ, and outputs a vector $\mathbf{u} \in \mathcal{R}_q^m$ such that $\mathbf{A} \cdot \mathbf{u} \equiv \mathbf{v} \bmod q$. Further, if $\sigma \geq \delta \cdot \|\mathbf{R}\| \cdot \omega(N \cdot \sqrt{\log(k \cdot N)})$ then the statistical distance between the following distributions is negligible:*

$$\{\mathbf{u} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \sigma)\} \text{ and } \left\{\mathbf{u} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{v})\right\} \ ,$$

and in particular, $\Pr[\|\mathbf{s}\| > \sigma \cdot \sqrt{m \cdot N} : \mathbf{s} \leftarrow \mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}, \sigma)]$ is negligible. We extend this algorithm for matrices, i.e. for a matrix $\mathbf{V} \in \mathcal{R}_q^{n\times \ell}$ with columns $\mathbf{v}_1, \ldots, \mathbf{v}_\ell$, we define $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{V}, \sigma)$ to be the algorithm which returns a matrix $\mathbf{S} \in \mathcal{R}_q^{m\times \ell}$, where the i-th column is the output of $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}_i, \sigma)$.

2.9 Ring-LWE

The Ring-LWE problem is to distinguish Ring-LWE samples from uniform.

Definition 2.15 (Ring-LWE)**.** *Let χ_s, χ_e be distributions on $\mathcal{R}$. The Ring-LWE problem, denoted, $\mathsf{RLWE}_{\mathcal{R}_q,\chi_s,\chi_e}$ is to distinguish $\{a_i, a_i \cdot s + e_i\}$ from uniform where $a_i \leftarrow \mathcal{U}(\mathcal{R}_q), e_i \leftarrow \chi_e$ and $s \leftarrow \chi_s$.*

The problem is considered hard for $\chi_s, \chi_e \coloneqq \mathcal{D}^2_{\mathcal{R},\sigma}$ with $\sigma \in \mathsf{poly}(N)$ and $q \in \mathsf{poly}(N)$ [SSTX09; LPR10]. By a hybrid argument we also have that $\{a \cdot s_j + e_j\}_{j\in\mathsf{poly}(\lambda)}$ for a fixed a and varying $s_j, e_j \leftarrow \mathcal{D}^2_{\mathcal{R},\sigma}$ is indistinguishable from uniform, cf. [PW08]. We refer to this setting as multi-instance $\mathsf{RLWE}_{\mathcal{R},\mathcal{D}_{\mathcal{R},\sigma},\mathcal{D}_{\mathcal{R},\sigma}}$.

2.10 NTRU Lattices

Definition 2.16 (NTRU)**.** *Let $q \geq 2 \in \mathbb{Z}$ and $\beta > 0$ a real number. A $(\mathcal{R}_q, \chi_f, \chi_g)$-NTRU instance is an element $h \in \mathcal{R}_q$ such that there exist $(f, g) \in \mathcal{R}^2 \setminus \{(0, 0)\}$ with $g \cdot h \equiv f \bmod q$ and $f, g \leftarrow \chi_f, \chi_g$. The $(\mathcal{R}_q, \chi_f, \chi_g)$-NTRU problem, denoted $\mathsf{NTRU}_{\mathcal{R}_q,\chi_f,\chi_g}$, is to distinguish NTRU instances from uniform.*

Our main construction uses the main result of Stehlé and Steinfeld [SS13], which says that there is an efficient algorithm that outputs $(h, \mathbf{T}_{\mathsf{NTRU}})$ such that h is statistically close to a uniform distribution over $\mathcal{R}_q^\times$ and $\mathbf{T}_{\mathsf{NTRU}}$ is a trapdoor for h.

Lemma 2.17 (Statistical NTRU Trapdoor Generation [SS13])**.** *Let $q = \omega(N)$ such that $q \equiv 5 \bmod 8$. Take $\varepsilon \in (0, 1/3)$ and $\sigma \geq \max(\sqrt{N \ln(8Nq)} \cdot q^{1/2+\varepsilon}, \omega(N^{3/2} \ln^{3/2} N))$. There is a PPT algorithm $\mathsf{NTRU.TrapGen}(\mathcal{R}_q, \sigma)$ which with an overwhelming probability outputs $h \in \mathcal{R}_q$ and a basis $\mathbf{T}_{\mathsf{NTRU}}$ of the lattice*

$$\Lambda_h \coloneqq \{(u, v) \in \mathcal{R}^2 : u + v \cdot h \equiv 0 \bmod q\}$$

such that $\|\tilde{\mathbf{T}}_{\mathsf{NTRU}}\| \leq N \cdot \sigma$. Furthermore, the statistical distance between the distribution of h and uniform over $\mathcal{R}_q^\times$ is at most $2^{10\,N} \cdot q^{-\lfloor \varepsilon N \rfloor}$.

When, as in Section 3.2, $\chi_f = \chi_g = \mathcal{D}_{\mathcal{R},\sigma}$ and when $\sigma < \sqrt{q}$ the problem is considered to be computationally hard for some choices of parameters [HPS98; ABD16; CJL16; KF17; DW21]. In particular, if $\sigma \ll \sqrt{q}$ then distinguishing NTRU is exponential in $\tilde{\Theta}(N \cdot \log(\sigma)/\log^2(q))$ under known algorithms.

2.11 Commitment Scheme

We recall the notion of a commitment scheme with relaxed binding, as introduced in [ALS20].

Definition 2.18. *Let* $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *be a triple of PPT algorithms. We say that* CM *is a commitment scheme over* $\mathcal{M}$ *with slack space* $\mathcal{S}$ *if it has the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$ *takes a security parameter* λ *(specified in unary) and outputs a common reference string* crs.
- $\mathsf{Commit}(\mathsf{crs}, m) \to (C, \mathsf{st})$ *takes a common reference string* crs *a message* $m \in \mathcal{M}$ *and outputs a commitment* C *and decommitment state* st.
- $\mathsf{Open}(\mathsf{crs}, C, m, \mathsf{st}, c)$ *takes a common reference string* crs, *a commitment* C, *a message* $m \in \mathcal{M}$, *a decommitment state* st *and a relaxation factor* $c \in \mathcal{S}$ *and outputs a bit indicating whether* C *is a valid commitment to* m *under* crs.[5]

We define the key properties of the commitment scheme: correctness, (relaxed) binding and hiding. In the following, we denote the message space as $\mathcal{M}$ and the slack space as $\mathcal{S}$.

Definition 2.19 (Completeness)**.** *A commitment scheme* $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *satisfies completeness if there exists a global relaxation factor* $c^* \in \mathcal{S}$ *such that for every* $m \in \mathcal{M}$*:*

$$\Pr\left[\mathsf{Open}(\mathsf{crs}, C, m, \mathsf{st}, c^*) = 1 \,\middle|\, \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ C, \mathsf{st} \leftarrow \mathsf{Commit}(\mathsf{crs}, m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \ .$$

For the notion of relaxed binding, we assume that the adversary comes up with two different openings with *the same* relaxation factor.

Definition 2.20 (Relaxed Binding)**.** *A commitment scheme* $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *satisfies relaxed binding if for every PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[\begin{array}{l} m \neq m' \wedge m, m' \in \mathcal{M} \wedge \\ \mathsf{Open}(\mathsf{crs}, C, m, \mathsf{st}, c) \\ = \mathsf{Open}(\mathsf{crs}, C, m', \mathsf{st}', c) = 1 \end{array} \,\middle|\, \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (C, (m, \mathsf{st}), (m, \mathsf{st}'), c) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array}\right] = \mathsf{negl}(\lambda) \ .$$

Definition 2.21 (Hiding)**.** *A commitment scheme* $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *satisfies hiding if for every (stateful) PPT adversary* $\mathcal{A}$*:*

$$\Pr\left[b' = b \,\middle|\, \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{crs}) \\ b \leftarrow \{0,1\} \\ C, \mathsf{st} \leftarrow \mathsf{Commit}(\mathsf{crs}, m_b) \\ b' \leftarrow \mathcal{A}(C) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

2.12 Polynomial Commitment Scheme

Polynomial commitment schemes extend commitments with the ability to prove evaluations of the committed polynomial.

Definition 2.22. *Let* $\mathsf{PC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}, \mathsf{Verify})$ *be a tuple of algorithms.* PC *is a polynomial commitment scheme over a ring* R *with degree bound* d *and slack space* $\mathcal{S}$ *if:*

[5] We implicitly assume that if $c \notin \mathcal{S}$ then Open automatically returns 0.

- $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ *is a commitment scheme over*

$$\mathcal{M} \coloneqq \left\{ (f_0, f_1, \dots, f_d) \in R^{d+1} : \sum_{i=0}^{d} f_i \cdot \mathsf{X}^i \in R[\mathsf{X}] \right\}$$

with slack space $\mathcal{S}$.
- $\mathsf{Eval}(\mathsf{crs}, C, u, \mathsf{st}) \to \pi$ *takes a common reference string* crs*, a commitment C, an evaluation point $u \in R$, auxiliary state* st *and outputs an evaluation proof π.*
- $\mathsf{Verify}(\mathsf{crs}, C, u, z, \pi) \to 0/1$ *takes a common reference string* crs*, a commitment C, an evaluation point $u \in \mathcal{R}$, a claimed image $z \in R$, an evaluation proof π, and outputs a bit indicating whether π is a valid evaluation proof that the polynomial committed to in C evaluates to z at the point u.*

We also consider a setting in which Eval *and* Verify *are replaced with an interactive two-party protocol between a prover and a verifier, and refer to that setting as an interactive polynomial commitment scheme.*

Further, we require that the evaluations procedure satisfy evaluation completeness and knowledge soundness. For simplicity, we give these definitions for non-interactive polynomial commitments, the interactive variant follows similarly.

Definition 2.23 (Evaluation Completeness)**.** *We say that a polynomial commitment scheme* $\mathsf{PC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}, \mathsf{Verify})$ *satisfies completeness if for every polynomial $f \in R^{\leq d}[\mathsf{X}]$ and any evaluation point $u \in R$:*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, C, u, f(u), \pi) = 0 \;\middle|\; \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ C, \mathsf{st} \leftarrow \mathsf{Commit}(\mathsf{crs}, f) \\ \pi \leftarrow \mathsf{Eval}(\mathsf{crs}, C, u, \mathsf{st}) \end{array} \right] = \mathsf{negl}(\lambda) \ .$$

Definition 2.24 (Knowledge Soundness)**.** *We say that a polynomial commitment scheme* $\mathsf{PC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}, \mathsf{Verify})$ *is knowledge sound with knowledge error ε if for all stateful PPT adversaries $\mathcal{P}^*$, there exists an expected PPT extractor $\mathcal{E}$ such that*

$$\Pr\left[\begin{array}{l} o = \mathsf{Open}(\mathsf{crs}, C, f, \mathsf{st}, c) \\ b = 1 \wedge \\ (o \neq 1 \vee f(u) \neq z) \end{array} \;\middle|\; \begin{array}{r} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (C, u, z, \pi) \leftarrow \mathcal{P}^*(\mathsf{crs}) \\ b = \mathsf{Verify}(\mathsf{crs}, C, u, z, \pi) \\ (f, \mathsf{st}, c) \leftarrow \mathcal{E}^{\mathcal{P}^*}(\mathsf{crs}, C, u, z, \pi) \end{array} \right] \leq \varepsilon(\lambda) \ .$$

2.13 Interactive Proofs

We recall the notion of interactive proofs [GMR85]. Let $\mathsf{R} \subseteq \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^*$ be a ternary relation. If $(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \mathsf{R}$, we say that $\mathbb{i}$ is an index, $\mathbb{x}$ is a statement and $\mathbb{w}$ is a witness for $\mathbb{x}$. We denote $\mathsf{R}(\mathbb{i}, \mathbb{x}) = \{\mathbb{w} : \mathsf{R}(\mathbb{i}, \mathbb{x}, \mathbb{w}) = 1\}$. In this work, we only consider NP relations R for which a witness w can be verified in time $\mathsf{poly}(|\mathbb{i}|, |\mathbb{x}|)$ for all $(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \mathsf{R}$.

A proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ for relation R consists of three PPT algorithms: the Setup algorithm, prover $\mathcal{P}$, and the verifier $\mathcal{V}$. The latter two are interactive and stateful. We write $(tr, b) \leftarrow \langle \mathcal{P}(\mathbb{i}, \mathbb{x}, \mathbb{w}), \mathcal{V}(\mathbb{i}, \mathbb{x}) \rangle$ for running $\mathcal{P}$ and $\mathcal{V}$ on inputs $\mathbb{i}, \mathbb{x}, \mathbb{w}$ and $\mathbb{i}, \mathbb{x}$ respectively and getting communication transcript tr and the verifier's decision bit b. We use the convention that $b = 0$ means reject and $b = 1$ means accept the prover's claim of knowing $\mathbb{w}$ such that $(\mathbb{x}, \mathbb{w}) \in R$. If tr contains a $\perp$ then we say that $\mathcal{P}$ aborts. Unless stated otherwise, we will assume that the first and the last message are sent from a prover. Hence, the protocol between $\mathcal{P}$ and $\mathcal{V}$ has an odd number of rounds. A Σ-protocol is a three-round protocol. Further, we say a protocol is *public coin* if the verifier's challenges are chosen uniformly at random independently of the prover's messages.

Definition 2.25 (Completeness)**.** *A proof system* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for the relation* R *has statistical completeness with correctness error* $\varepsilon(\lambda)$ *if for all adversaries* $\mathcal{A}$*,*

$$\Pr\left[b = 0 \land (\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \mathsf{R} \;\middle|\; \begin{array}{r} \mathbb{i} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbb{x}, \mathbb{w}) \leftarrow \mathcal{A}(\mathbb{i}) \\ (tr, b) \leftarrow \langle \mathcal{P}(\mathbb{i}, \mathbb{x}, \mathbb{w}), \mathcal{V}(\mathbb{i}, \mathbb{x}) \rangle \end{array} \right] \leq \varepsilon(\lambda) + \mathsf{negl}(\lambda) \ .$$

Definition 2.26 (Knowledge Soundness)**.** *A proof system* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *for the relation* R *is knowledge sound with knowledge error* $\varepsilon(\lambda)$ *if there exists an expected PPT extractor* $\mathcal{E}$ *such that for any stateful PPT adversary* $\mathcal{P}^*$*:*

$$\Pr\left[b = 1 \land (\mathbb{i}, \mathbb{x}, \mathbb{w}) \notin \mathsf{R} \;\middle|\; \begin{array}{r} \mathbb{i} \leftarrow \mathsf{Setup}(1^\lambda) \\ (\mathbb{x}, \mathsf{st}) \leftarrow \mathcal{P}^*(\mathbb{i}) \\ (tr, b) \leftarrow \langle \mathcal{P}^*(\mathbb{i}, \mathbb{x}, \mathsf{st}), \mathcal{V}(\mathbb{i}, \mathbb{x}) \rangle \\ \mathbb{w} \leftarrow \mathcal{E}^{\mathcal{P}^*}(\mathbb{i}, \mathbb{x}) \end{array} \right] \leq \varepsilon(\lambda) + \mathsf{negl}(\lambda) \ .$$

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

2.14 Coordinate-Wise Special Soundness

We recall the notion of coordinate-wise special soundness defined in [FN23]. Let S be a set and $\ell \in \mathbb{N}$. Namely, take two vectors $\mathbf{x} := (x_1, \ldots, x_\ell), \mathbf{y} := (y_1, \ldots, y_\ell) \in S^\ell$. Then, we define the following relation "$\equiv_i$" for fixed $i \in [\ell]$ as:

$$\mathbf{x} \equiv_i \mathbf{y} \iff x_i \neq y_i \land \forall\, j \in [\ell] \backslash \{i\}, x_j = y_j \ .$$

That is, vectors $\mathbf{x}$ and $\mathbf{y}$ have the same values in all coordinates apart from the i-th one. For $\ell = 1$, the relations boils down to cheking whether two elements are distinct. Further, we can define the set

$$\mathsf{SS}(S, \ell) := \left\{ (\mathbf{x}_1, \ldots, \mathbf{x}_{\ell+1}) \in \left(S^\ell\right)^{\ell+1} : \begin{array}{l} \exists\, k \in [\ell+1],\ \forall\, i \in [\ell], \\ \exists\, j \in [\ell+1] \backslash \{k\},\ \mathbf{x}_k \equiv_i \mathbf{x}_j \end{array} \right\} \ .$$

As a simple example, $((0,0),(1,0),(0,1)) \in \mathsf{SS}(\mathbb{Z}_2, 2)$ – the vector $(0,0)$ differs from $(1,0)$ (resp. $(0,1)$) exactly in the first (resp. second) coordinate. Note that for $\ell = 1$, this set contains pairs of distinct elements in S.

We are ready to define the notion of coordinate-wise special soundness.

Definition 2.27 (Coordinate-Wise Special Soundness)**.** *Let* $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ *be a public-coin* $(2\mu + 1)$*-round interactive proof system for relation* R*, where in each round the verifier picks a uniformly random challenge from* S^ℓ*. A tree of transcripts is a set of* $K = (\ell+1)^\mu$ *arranged in the following tree structure. The nodes in the tree correspond to the prover's messages and the edges correspond to the verifier's challenges. Each node at depth* i *has exactly* $\ell + 1$ *children corresponding to* $\ell + 1$ *distinct challenges which, as a vector, lie in* $\mathsf{SS}(S, \ell)$*. Every transcript corresponds to exactly one path from the root to a leaf node.*

We say that Π *is* ℓ*-coordinate-wise special sound if there is a polynomial time algorithm that given an index* $\mathbb{i}$*, statement* $\mathbb{x}$ *and the tree of transcripts, outputs a witness* $\mathbb{w} \in \mathsf{R}(\mathbb{i}, \mathbb{x})$*.*

The following lemma states that coordinate-wise special sound protocols are knowledge sound.

Lemma 2.28 (Lemma 2.30 of [FN23]). *Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be public-coin $(2\mu + 1)$-round interactive proof system for relation R and suppose the challenge space of $\mathcal{V}$ in each round is S^{ℓ}. If Π is ℓ-coordinate-wise special sound and $\ell^{\mu} = \mathsf{poly}(\lambda)$, then it is knowledge sound with knowledge error $\mu\ell/|S|$.*

3 Power-Ring-BASIS Assumption

Our construction of the polynomial commitment will rely on the multi-instance version of the PRISIS (Power-Ring-BASIS) assumption [FN23] which is a special case of the BASIS assumption introduced by Wee and Wu [WW23b].[6] Recall that $\mathbf{G}_n$ is a gadget matrix with base δ. We fix the modulus q and set $\tilde{q} := \lfloor \log_\delta q \rfloor + 1$. Here, we consider a multi-instance version of BASIS, where the adversary is given h instances $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{T}_i, \mathsf{aux}_i)$ of BASIS, and it has to find a short non-zero solution to the concatenated matrix $[\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_h]$.

We also analyse hardness of the h-PRISIS assumption for $h = \mathsf{poly}(\lambda)$. First, we show that for specific parameters, $h\text{-}\mathsf{PRISIS}_{n,m,\mathcal{R}_q,2,\sigma,\beta}$ is secure under the Module-SIS assumption. This implies that our polynomial commitment scheme in Section 5 is secure under standard assumptions. Furthermore, we prove that h-PRISIS can generally be reduced to a single instance PRISIS which can be of independent interest.

Definition 3.1 (h-BASIS). *Let $h \geq 1$ and $q, n, m, n', m', \ell, N, \sigma, \beta$ be the lattice parameters. Let Samp be a PPT algorithm, which given a matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, outputs a matrix $\mathbf{B} \in \mathcal{R}_q^{n' \times m'}$ along with auxiliary information aux. We say the $h\text{-}\mathsf{BASIS}_{n,m,n',m',\mathcal{R}_q,\ell,\sigma,\beta}$ assumption holds w.r.t. Samp if for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{l|r} [\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_h] \cdot \mathbf{z} = \mathbf{0} & \forall j \in [h], \mathbf{A}_j \leftarrow \mathcal{R}_q^{n \times m}, \\ 0 < \|\mathbf{z}\| \leq \beta & (\mathbf{B}_j, \mathsf{aux}_j) \leftarrow \mathsf{Samp}(\mathbf{A}_j) \\ & \mathbf{T}_j \leftarrow {\mathbf{B}_j}_{\sigma}^{-1}(\mathbf{G}_{n'}) \\ & \mathbf{z} \leftarrow \mathcal{A}\left((\mathbf{A}_j, \mathbf{B}_j, \mathbf{T}_j, \mathsf{aux}_j)_{j \in [h]}\right) \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

The PRISIS assumption is defined by the following sampling algorithm Samp.

Definition 3.2 (h-PRISIS). *The $h\text{-}\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma,\beta}$ assumption is an instantiation of the h-BASIS assumption with the following sampling algorithm Samp. That is, $\mathsf{Samp}(\mathbf{A})$ samples a row $\mathbf{a}^{\mathsf{T}} \leftarrow \mathcal{R}_q^m$ and sets $\bar{\mathbf{A}}$ as*

$$\bar{\mathbf{A}} := \begin{bmatrix} \mathbf{a}^{\mathsf{T}} \\ \mathbf{A} \end{bmatrix} \in \mathcal{R}_q^{(n+1) \times m} \ . \tag{2}$$

Then, it samples $w \leftarrow \mathcal{R}_q^{\times}$, and outputs

$$\mathbf{B} := \left[\begin{array}{ccc|c} w^0 \cdot \bar{\mathbf{A}} & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & w^{\ell-1} \cdot \bar{\mathbf{A}} & -\mathbf{G}_{n+1} \end{array}\right] \quad \text{and} \quad \mathsf{aux} := w.$$

[6] BASIS stands for Basis-Augmented Shortest Integer Solution.

3.1 h-PRISIS Assumption for $\ell = 2$

We provide an efficient reduction from the $h\text{-}\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma,\beta}$ assumption, where $\ell = 2$, to MSIS. Since we reduce security of our constructions to this case, we obtain a polynomial commitment from standard lattice assumptions.

Recall that Fenzi and Nguyen [FN23] already gave such a reduction for the single instance case ($h = 1$). We adapt their proof strategy to prove hardness of h-PRISIS. To this end, we fist provide a technical lemma from [FN23, Lemma 3.4], which says that if one can find a short solution to a specific linear equation, then one can also build a PRISIS trapdoor.

Lemma 3.3 ([FN23]). *Let $n, m, N > 0$ and $\alpha \geq 1$. Denote $t = n \cdot \tilde{q}$. Then, there exists an efficient deterministic algorithm, that given as input a matrix $\mathbf{A}^\star \in \mathcal{R}_q^{n\times m}$, invertible $\mathbf{W}_1, \mathbf{W}_2, \mathbf{H} \in \mathcal{R}_q^{n\times n}$ and two matrices $\mathbf{T}_1, \mathbf{T}_2 \in \mathcal{R}_q^{m\times t}$, which satisfy $\left\|[\mathbf{T}_1 \| \mathbf{T}_2]\right\| \leq \alpha$ and*

$$\mathbf{W}_1 \cdot \mathbf{A}^\star \cdot \mathbf{T}_1 - \mathbf{W}_2 \cdot \mathbf{A}^\star \cdot \mathbf{T}_2 = \mathbf{H} \cdot \mathbf{G}_n \ ,$$

outputs a tag $\mathbf{H}^* \in \mathsf{GL}(2n, \mathcal{R}_q)$ and a $\mathbf{G}_{2n}$-trapdoor $\mathbf{S}$ for the matrix $\mathbf{B}$ defined as:

$$\mathbf{B} \coloneqq \begin{bmatrix} \mathbf{W}_1 \cdot \mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{W}_2 \cdot \mathbf{A}^\star & -\mathbf{G} \end{bmatrix}$$

with a tag $\mathbf{H}^*$, where $\|\mathbf{S}\| \leq \sqrt{2 \cdot (\alpha^2 + t^2 \cdot N)}$.

We now follow the footsteps of the proof of [FN23, Lemma 3.6] to prove hardness of h-PRISIS.

Lemma 3.4 (h-$\mathsf{PRISIS} \implies \mathsf{MSIS}$). *Let $n > 0, m \geq n$ and denote $t = (n+1) \cdot \tilde{q}$. Let $q = \omega(N)$ satisfy $q \equiv 5 \bmod 8$. Take $\varepsilon \in (0, 1/3)$ and $\sigma_0 \geq \max(\sqrt{N \ln(8\,N \cdot q)} \cdot q^{1/2+\varepsilon},\ \omega(N^{3/2} \cdot \ln^{3/2} \cdot N))$ such that $2^{10\,N} q^{-\lfloor \varepsilon \cdot N \rfloor}$ is negligible. Let $\tau \coloneqq \max(2 \cdot (n+1), 2\,m + t)$ and*

$$\sigma_1 \geq \delta \sqrt{t \cdot N \cdot (N^2 \cdot \sigma_0^2 \cdot m + 2\,t)} \cdot \omega(\sqrt{N \cdot \log(\tau N)}) \ .$$

Then, for $h = \mathsf{poly}(\lambda)$, $h\text{-}\mathsf{PRISIS}_{n,m,\mathcal{R}_q,2,\sigma_1,\beta}$ is hard under the $\mathsf{MSIS}_{n,hm,\mathcal{R}_q,\beta}$ assumption.

Proof. Suppose there is a PPT algorithm $\mathcal{A}$ that wins $h\text{-}\mathsf{PRISIS}_{n,m,\mathcal{R}_q,2,\sigma_1,\beta}$ with probability ε. We revisit the h-PRISIS security game and introduce a game hop where we plug in the NTRU trapdoors inside the auxiliary information $w_1, \ldots, w_h$. We define ε_i to be the probability that $\mathcal{A}$ wins $\mathsf{Game}\ i$.

Game 1: This is the standard h-PRISIS security game. To recall, for $j \in [h]$, the challenger samples $\mathbf{a}_j \leftarrow \mathcal{R}_q^m$, $\mathbf{A}_j \leftarrow \mathcal{R}_q^{n\times m}$ and sets $\mathbf{A}_j^\star$ as in (2). Then, it generates an invertible element $w_j \leftarrow \mathcal{R}_q^\times$ and computes the matrix:

$$\mathbf{B}_j \coloneqq \begin{bmatrix} \mathbf{A}_j^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \cdot \mathbf{A}_j^\star & -\mathbf{G} \end{bmatrix} \ .$$

Then, it samples $\mathbf{T}_j \leftarrow \mathbf{B}_{j\,\sigma_1}^{-1}\left(\mathbf{G}_{2(n+1)}\right)$ and outputs $(\mathbf{A}_j, \mathbf{B}_j, \mathbf{T}_j, w_j)_{j\in[h]}$ to the adversary $\mathcal{A}$. By definition, $\varepsilon_1 = \varepsilon$.

Game 2: For all $j \in [h]$, we substitute $w_j \leftarrow \mathcal{R}_q$ by running $(w_j,\ \mathbf{T}^{(j)}_{\mathsf{NTRU}}) \leftarrow \mathsf{NTRU.TrapGen}(\mathcal{R}_q, \sigma_0)$. By Lemma 2.17 and the hybrid argument on h, we get $\varepsilon_5 \geq \varepsilon_4 - \mathsf{negl}(\lambda)$.

Assume there is an adversary that wins Game_2. We now show how to build h-PRISIS trapdoors $(\mathbf{T}_j)_{j\in[h]}$ given the Module-SIS matrix $[\mathbf{A}_1 \mid \ \cdots \ \mid \mathbf{A}_h]$ and the NTRU trapdoors $(\mathbf{T}^{j}_{\mathsf{NTRU}})_{j\in[h]}$. To this end, we will show how to find short matrices $\mathbf{S}_{j,1}, \mathbf{S}_{j,2}$ such that:

$$\mathbf{A}^{\star}_j \cdot \mathbf{S}_{j,1} - w \cdot \mathbf{A}^{\star}_j \mathbf{S}_{j,2} = \mathbf{G}$$

where $\mathbf{A}^{\star}_j$ is computed as in Eq. (2). Let $\mathbf{g}_i$ be the i-th column of $\mathbf{G}$ and fix $j \in [h]$. Assuming that $\mathbf{A}^{\star}_j$ is full-rank (cf. Lemma 2.10) and using linear algebra, we can find a (possibly large) vector $\mathbf{t}$ such that $\mathbf{A}^{\star}_j \cdot \mathbf{t} = \mathbf{g}_i$. Now, using the NTRU trapdoor $\mathbf{T}_{\mathsf{NTRU}}$ and the standard Nearest Plane algorithm [LLL82; Bab85], we can sample vectors $(\mathbf{s}_{1,i}, \mathbf{s}_{2,i})$ such that:

$$\mathbf{s}_{1,i} - w_j \cdot \mathbf{s}_{2,i} = \mathbf{t} \text{ and } \|(\mathbf{s}_{1,i}, \mathbf{s}_{2,i})\| \leq N \cdot \sigma_0 \cdot \sqrt{m \cdot N/2} \ .$$

Hence

$$\mathbf{A}^{\star}_j \cdot \mathbf{s}_{1,i} - w_j \cdot \mathbf{A}^{\star}_j \cdot \mathbf{s}_{2,i} = \mathbf{A}^{\star}_j \cdot (\mathbf{s}_{1,i} - w_j \cdot \mathbf{s}_{2,i}) = \mathbf{A}^{\star} \cdot \mathbf{t} = \mathbf{g}_i \ .$$

Thus, we obtain the matrices $\mathbf{S}_{j,1}, \mathbf{S}_{j,2}$ by concatenation where

$$\big\|[\mathbf{S}_{j,1} \| \mathbf{S}_{j,2}]\big\| \leq \alpha \coloneqq N \cdot \sigma_0 \cdot \sqrt{m \cdot t \cdot N/2} \ .$$

Therefore, by Lemma 3.3, we can construct a $\mathbf{G}_{2(n+1)}$-trapdoor $\mathbf{S}_j$ for $\mathbf{B}_j$ such that

$$\|\mathbf{S}_j\| \leq \sqrt{2\,(\alpha^2 + t^2 \cdot N)} = \sqrt{t \cdot N \cdot (N^2\,\sigma_0^2 \cdot m + 2\,t)} \ .$$

Hence, the reduction $\mathcal{B}$ can build the trapdoor $(\mathbf{S}_j)_{j\in[h]}$ as above and then randomise the trapdoor for $\mathbf{B}_j$ by sampling fresh preimages as $\mathbf{T}_j \leftarrow \mathsf{SamplePre}(\mathbf{B}_j, \mathbf{S}_j, \mathbf{G}_{2(n+1)}, \sigma_1)$ for $j \in [h]$. Finally it sends the tuple to $\mathcal{A}$ and returns what it outputs. Since $\sigma_1 \geq \delta \cdot \|\mathbf{S}_j\| \cdot \omega(\sqrt{N \cdot \log(\tau N)})$, by Lemma 2.14 $\mathcal{B}$ wins the Module-SIS game with probability at least $\varepsilon_2 - \mathsf{negl}(\lambda)$, which concludes the proof. □

3.2 h-PRISIS Assumption for $\ell = O(1)$

In this section, we show that if PRISIS is hard then h-PRISIS is hard for $\ell \in O(1) > 2$ and $h = \mathsf{poly}(\lambda)$. In particular, we will prove the following theorem.

Theorem 3.5 (h-$\mathsf{PRISIS} \implies \mathsf{PRISIS}$)**.** *Let $n, m, \mathcal{R}_q, \ell, \sigma_T$ be* PRISIS *parameters. Let the ring $\mathcal{R}_q$ split into fields of superpolynomial size. Let $\ell, n \in O(1)$. Let $\sigma_x > 0$ and $\beta > 0$ be real numbers. Let δ be a gadget matrix base and set $\tilde{q} \coloneqq \lfloor \log_\delta q \rfloor + 1$. Let $m \geq 2n > 0$, let $h = \mathsf{poly}(\lambda)$, let*

$$\beta' \geq 4\,h \cdot \beta \cdot N^{9/2} \cdot m^{5/2} \cdot q^{4n/m + 8/(Nm)} \cdot \omega(\log N) \ ,$$

let

$$\sigma_{Ti} \geq N^{6+2\ell} \cdot \sigma_x^{2\ell} \cdot 2^{\ell+2} \cdot q^{4n/m+8/(Nm)} \cdot m^2 \cdot \delta \cdot \sigma_T \cdot \sqrt{(m\,\ell + n\,\tilde{q}) \cdot n\,\tilde{q}\,\ell \cdot \log(m \cdot N)} \cdot \omega(\log^{3/2+\ell}(N)) \ .$$

Then h-$\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma_{Ti},\beta}$ is hard under the $\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma_T,\beta'}$ assumption, under the $\mathsf{RLWE}_{\mathcal{R}_q,\mathcal{D}_{\mathcal{R},\sigma_x}}$ and the $\mathsf{NTRU}_{\mathcal{R}_q,\mathcal{D}_{\mathcal{R},\sigma_x},\mathcal{D}_{\mathcal{R},\sigma_x}}$ assumptions.

3.2.1 Overview

We will describe the key ideas of the proof and then glue them together in the end of the section. First, we note that for an arbitrary matrix $\mathbf{R} \in \mathcal{R}^{m/2 \times m/2}$ the following holds:

$$\begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{R} \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & -\mathbf{R} \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} = \mathbf{Id}_m \ .$$

If $\mathbf{R}$ has a small norm, both the matrix and its inverse are small. Similarly, we can place $\mathbf{R}$ in the bottom left corner:

$$\begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ \mathbf{R} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ -\mathbf{R} & \mathbf{Id}_{m/2} \end{bmatrix} = \mathbf{Id}_m \ .$$

Randomising A. Using these, we are going to transform a given PRISIS matrix into multiple almost independent uniform matrices. A PRISIS instance given by the challenger consists of $(\mathbf{A}, \mathbf{B}, w, \mathbf{T})$ such that $\mathbf{B} \cdot \mathbf{T} = \mathbf{G}_{n\,\ell}$, i.e:

$$\begin{bmatrix} \mathbf{A} & & & -\mathbf{G}_n \\ & \ddots & & \vdots \\ & & w^{\ell-1} \cdot \mathbf{A} & -\mathbf{G}_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_{0,0} \\ \vdots \\ \mathbf{T}_{0,(\ell-1)} \\ \mathbf{T}_{0,\ell} \end{bmatrix} = \mathbf{G}_{n\,\ell} \ .$$

We split $\mathbf{A} = [\mathbf{A}_L | \mathbf{A}_R]$ in the middle. Similarly, for an arbitrary trapdoor block $\mathbf{T}_{0,j}$ such that $0 \leq j \leq \ell - 1$, let $\mathbf{T}_{0,j} = [\mathbf{T}_L \| \mathbf{T}_R]$. The block $\mathbf{T}_{0,\ell}$ is left unchanged in this step. We sample $\mathbf{R}_1, \mathbf{R}_2$ from a distribution we define later on. We rerandomise $\mathbf{A}$ and $\mathbf{T}_{0,j}$ in the following way:

$$\mathbf{A}' := [\mathbf{A}_L \mid \mathbf{A}_R] \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{R}_1 \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ \mathbf{R}_2 & \mathbf{Id}_{m/2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_L + (\mathbf{A}_L \cdot \mathbf{R}_1 + \mathbf{A}_R) \cdot \mathbf{R}_2 \mid \mathbf{A}_L \cdot \mathbf{R}_1 + \mathbf{A}_R \end{bmatrix} \ ,$$

and accordingly

$$\mathbf{T}'_{0,j} := \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ -\mathbf{R}_2 & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & -\mathbf{R}_1 \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_L \\ \mathbf{T}_R \end{bmatrix} = \begin{bmatrix} \mathbf{T}_L - \mathbf{R}_1 \cdot \mathbf{T}_R \\ \mathbf{T}_R - \mathbf{R}_2 \cdot (\mathbf{T}_L - \mathbf{R}_1 \cdot \mathbf{T}_R) \end{bmatrix} \ .$$

We can verify that

$$\mathbf{A}' \cdot \mathbf{T}'_{0,j} = \mathbf{A} \cdot \mathbf{T}_{0,j} \ .$$

We use the Leftover Hash Lemma to prove that the new challenge matrix $\mathbf{A}'$ looks uniformly random and independent of the initial instance. Then we can apply this rerandomisation h times to generate h new instances.

That is, we will eventually sample fresh short matrices $\mathbf{R}_{i1}, \mathbf{R}_{i2}$ for each of the h-PRISIS matrices. We denote

$$\mathbf{M}_i := \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{R}_{i1} \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ \mathbf{R}_{i2} & \mathbf{Id}_{m/2} \end{bmatrix} \ .$$

Then its inverse is equal to

$$\mathbf{M}_i^{-1} = \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ -\mathbf{R}_{i2} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & -\mathbf{R}_{i1} \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \ .$$

Randomising w. We will also need to rerandomise the scalar w. For this, we use an approach inspired by the NTRU rerandomisation in [PS21]. We set $w_i \coloneqq \frac{w}{x_i}$ for some short x_i. Then we have:

$$\begin{bmatrix} \mathbf{A} & & & -\mathbf{G}_n \\ & \ddots & & \vdots \\ & & w^{\ell-1}/x_i^{\ell-1} \cdot \mathbf{A} & -\mathbf{G}_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_{0,0} \\ \vdots \\ x_i^{\ell-1} \cdot \mathbf{T}_{0,(\ell-1)} \\ \mathbf{T}_{0,\ell} \end{bmatrix} = \mathbf{G}_{n\,\ell} \ .$$

We will show that this is indistinguishable from uniform $\{w_i\}$ if NTRU and RLWE are hard.

Randomising $\mathbf{T}$. We apply both transformations to the initial trapdoor $\mathbf{T}$ and compute h new trapdoors. We define

$$\mathbf{T}'_i = \begin{bmatrix} \mathbf{T}'_{i,0} \\ \vdots \\ \mathbf{T}'_{i,(\ell-1)} \\ \mathbf{T}'_{i,\ell} \end{bmatrix} \coloneqq \begin{bmatrix} \mathbf{M}_i^{-1} \cdot \mathbf{T}_{0,0} \\ \vdots \\ x_i^{\ell-1} \cdot \mathbf{M}_i^{-1} \cdot \mathbf{T}_{0,(\ell-1)} \\ \mathbf{T}_{0,\ell} \end{bmatrix} \ .$$

The columns of the new trapdoors do not look like spherical Gaussian vectors, so we use them in the SamplePre algorithm from Lemma 2.14 and generate well-distributed preimages of vectors in $\mathbf{G}_{n\,\ell}$.

3.2.2 Technical Lemmas

Lemma 3.6 (Randomising $\mathbf{A}$). *Let $\mathcal{R}_q$ split into fields of superpolynomial size. Let $\mathbf{A}_L, \mathbf{A}_R \leftarrow \mathcal{U}(\mathcal{R}_q^{n\times m/2})$ with $m/2 \geq n$. Let $\mathbf{R}_1, \mathbf{R}_2 \in \mathcal{R}^{m/2\times m/2}$ have columns sampled independently of $\mathcal{D}_{\mathcal{R}^{m/2},\sigma_R}$ with $\sigma_R \geq 2\,N \cdot q^{2n/m+4/(N\cdot m)}$. Let*

$$\begin{aligned} LHS &\coloneqq \mathbf{A}_L, \mathbf{A}_R,\ \mathbf{A}_L \cdot \mathbf{R}_1 + \mathbf{A}_R, \mathbf{A}_L + (\mathbf{A}_L \cdot \mathbf{R}_1 + \mathbf{A}_R) \cdot \mathbf{R}_2, \\ RHS &\coloneqq \mathcal{U}(\mathcal{R}_q^{n\times m/2}),\ \mathcal{U}(\mathcal{R}_q^{n\times m/2}),\ \mathcal{U}(\mathcal{R}_q^{n\times m/2}),\ \mathcal{U}(\mathcal{R}_q^{n\times m/2}) \ , \end{aligned}$$

then $\Delta(LHS, RHS) \leq 2^{-\Omega(N)}$.

Proof. Using Corollary 2.9 and the fact that adding $\mathbf{A}_R$ is a bijective map we get that $\Delta((\mathbf{A}_L, \mathbf{A}_L \cdot \mathbf{R}_1 + \mathbf{A}_R), (\mathbf{A}_L, \mathbf{A}_F)) \leq 2^{-\Omega(N)}$, where $\mathbf{A}_F$ is a fresh uniformly random matrix. With the same argument we obtain

$$\Delta((\mathbf{A}_L, \mathbf{A}_R, \mathbf{A}_F, \mathbf{A}_L + \mathbf{A}_F \cdot \mathbf{R}_2), (\mathbf{A}_L, \mathbf{A}_R, \mathbf{A}_F, \mathbf{A}'_F)) \leq 2^{-\Omega(N)} \ ,$$

where $\mathbf{A}'_F$ is a fresh uniformly random matrix. Now applying the triangle inequality we get the statement of the Lemma. □

We now analyse the norms of the trapdoors we output. Let us drop the index i to make this part of the document more readable.

Lemma 3.7 (Norm of $\mathbf{T}$). *For the trapdoor $\mathbf{T}' \coloneqq \left[\mathbf{T}'_0 \| \dots \| \mathbf{T}'_{(\ell-1)} \| \mathbf{T}'_\ell\right] \in \mathcal{R}^{(m\ell+n\tilde{q})\times \ell n\tilde{q}}$ constructed above with $\mathbf{R}_1, \mathbf{R}_2 \leftarrow (\mathcal{D}_{\mathcal{R},\sigma_R}^{m/2\times m/2})^2$ and $x \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$. For $\sigma_R \cdot N \cdot m > 2$ we have $s_1(\mathbf{T}') \leq \|\mathbf{T}'\| \leq$*

$$\sqrt{(N\,m\,\ell + N\,n\,\tilde{q}) \cdot N\,n\,\tilde{q}\,\ell} \cdot (N \cdot 2N \cdot \sigma_x^2)^\ell \cdot (N\,m)^2 \cdot \sigma_T\,\sigma_R^2 \cdot \omega(\log^{\ell+3/2} N)$$

with overwhelming probability.

Proof. We compute the norm of each block of the resulting trapdoor separately. By Lemma 2.1 each coefficient of the initial trapdoor matrix is bounded by $\sigma_T \cdot \omega(\sqrt{\log N})$ in the infinity norm with overwhelming probability.

In particular, the infinity norm of $\mathbf{T}_{0,\ell}$ is bounded by $\sigma_T \cdot \omega(\sqrt{\log N})$. For every remaining block $\mathbf{T}_{0,i} \coloneqq [\mathbf{T}_L \| \mathbf{T}_R]$ with $0 \le i \le \ell - 1$ the transformation is of the following type :

$$\begin{aligned} x^i \cdot \begin{bmatrix} \mathbf{T}'_L \\ \mathbf{T}'_R \end{bmatrix} &\coloneqq x^i \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ -\mathbf{R}_2 & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{S}_L \\ \mathbf{S}_R \end{bmatrix} \\ &= x^i \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ -\mathbf{R}_2 & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & -\mathbf{R}_1 \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_L \\ \mathbf{T}_R \end{bmatrix} \quad . \end{aligned}$$

Analysing this expression, we first note that $\mathbf{S}_R = \mathbf{T}_R$, hence $\|\mathbf{S}_R\|_\infty \le \sigma_T \cdot \omega(\sqrt{\log N})$. As for $\mathbf{S}_L = \mathbf{T}_L - \mathbf{R}_1 \mathbf{T}_R$, we can bound its infinity norm as follows:

$$\begin{aligned} \|\mathbf{S}_L\|_\infty &\le \|\mathbf{T}_L\|_\infty + \max_{ij} \left\| \sum_{k=0}^{m/2-1} r_{ik}^{(1)} \cdot t_{kj}^{(R)} \right\|_\infty \\ &\le \|\mathbf{T}_L\|_\infty + \sum_{k=1}^{m/2} N \cdot \|\mathbf{R}_1\|_\infty \cdot \|\mathbf{T}_R\|_\infty \\ &\le \sigma_T \cdot \omega(\sqrt{\log N}) + \frac{Nm}{2} \cdot \sigma_R \cdot \omega(\sqrt{\log N}) \cdot \sigma_T \cdot \omega(\sqrt{\log N}) \\ &\le (\sigma_T + \frac{N\, m\, \sigma_T\, \sigma_R}{2}) \cdot \omega(\log N) \quad . \end{aligned}$$

Here we denote an element of $\mathbf{R}_1$ at position (i,j) as $r_{ij}^{(1)}$ and an element of $\mathbf{T}_R$ as $t_{ij}^{(R)}$. The transitions above use Lemmata 2.1, 2.5 and the triangle inequality. Secondly, we bound the results of the second matrix multiplication. As before, $\mathbf{T}'_L = \mathbf{S}_L$ so $\|\mathbf{T}'_L\|_\infty = \|\mathbf{S}_L\|_\infty$. And we have $\mathbf{T}'_R = \mathbf{S}_R - \mathbf{R}_2 \cdot \mathbf{S}_L$. Similarly, we analyse its norm:

$$\begin{aligned} \left\|\mathbf{T}'_R\right\|_\infty &\le \|\mathbf{S}_R\|_\infty + \sum_{k=1}^{m/2} N \cdot \|\mathbf{R}_2\|_\infty \cdot \|\mathbf{S}_L\|_\infty \\ &\le \sigma_T \cdot \omega(\sqrt{\log N}) + \frac{Nm}{2} \cdot \sigma_R \cdot (\sigma_T + \frac{Nm\sigma_T\sigma_R}{2}) \cdot \omega(\log^{3/2} N) \\ &= \left(\sigma_T + \frac{\sigma_R\sigma_T Nm}{2} + \frac{(Nm)^2\sigma_T\sigma_R^2}{4}\right) \cdot \omega(\log^{3/2} N) \quad . \end{aligned}$$

Finally, we use the ring expansion factor and Remark 3.10 to bound the value $\left\|x^i \cdot \mathbf{T}'_R\right\|_\infty \le N^i \cdot \|x\|_\infty^i \cdot \|\mathbf{T}'_R\|_\infty \le N^i \cdot (2N \cdot \sigma_x^2 \cdot \omega(\log N))^i \cdot \|\mathbf{T}'_R\|_\infty$. An equivalent bound holds for $\mathbf{T}'_L$.

We conclude that the quality of the trapdoor is as claimed using a geometric progression inequality $\sum_{1^{\ell-1}} N^i \cdot (2N \cdot \sigma_x^2 \cdot \omega(\log N))^i \le N^\ell \cdot (2N \cdot \sigma_x^2 \cdot \omega(\log N))^\ell$. □

Lemma 3.8 (Randomising w)**.** *Let $\mathcal{R}_q$ be such that it splits into superpolynomially large fields. Let σ_x be a real number > 0. Given $\{w_i\}_{1 \le i \le k}$. If both $\mathsf{NTRU}_{\mathcal{R}_q, \mathcal{D}_{\mathcal{R},\sigma_x}, \mathcal{D}_{\mathcal{R},\sigma_x}}$ and $\mathsf{RLWE}_{\mathcal{R}_q, \mathcal{D}_{\mathcal{R},\sigma_x}, \mathcal{D}_{\mathcal{R},\sigma_x}}$ are hard, it is hard to decide if $w_i \leftarrow \mathcal{U}(\mathcal{R}_q)$ or $w_i \coloneqq w/x_i$ where $w \leftarrow \mathcal{U}(\mathcal{R}_q)$, $x_i \coloneqq f \cdot s_i + g \cdot e_i$ with $f, g, s_i, e_i \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$.*

Proof. We proceed in a sequence of hybrids.

Game 1: This is setting where we have uniformly random elements $\{w_1, w_2, \ldots, w_k\}$.

Game 2: We rewrite our elements as $\{u/u_1, u/u_2, \ldots, u/u_k\}$, where $u \leftarrow \mathcal{U}(\mathcal{R}_q)$ and $u_i \leftarrow \mathcal{U}(\mathcal{R}_q^\times)$. The two distributions are within negligible statistical distance by our assumption on $\mathcal{R}_q$ which implies that $w_i \leftarrow \mathcal{U}(\mathcal{R}_q)$ are invertible with overwhelming probability.

Game 3: We rewrite our elements again as $\{u/b_1, u/b_2, \ldots, u/b_k\}$ where $b_i \coloneqq a \cdot s_i + e_i$ for $a \leftarrow \mathcal{U}(\mathcal{R}_q)$ and $s_i, e_i \leftarrow \mathcal{D}^2_{\mathcal{R},\sigma_x}$. Under the Ring-LWE assumption for $\mathsf{RLWE}_{\mathcal{R}_q,\mathcal{D}_{\mathcal{R},\sigma_x},\mathcal{D}_{\mathcal{R},\sigma_x}}$ this game hop is undetectable. To show this, we plant a multi-instance Ring-LWE challenge which is either $(a, b_i \coloneqq a \cdot s_i + e_i)$ or (a, u_i) with $u_i \leftarrow \mathcal{U}(\mathcal{R}_q)$.

Game 4: We replace a with $h = \frac{f}{g}$ where $f, g \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$. If this game hop can be detected by the adversary we can break the NTRU assumption with $\chi_f = \chi_g = \mathcal{D}_{\mathcal{R},\sigma_x}$. To do that we plant an NTRU challenge into h. If h is uniform we have the distribution from Game 3. If $h = \frac{f}{g}$, we obtain $\{u/(f/g \cdot s_i + e_i)\}$.

Game 5: We see that:

$$u/(h \cdot s_i + e_i) = u/(f/g \cdot s_i + e_i) = u \cdot g/(f \cdot s_i + g \cdot e_i) \ .$$

Therefore, the expression obtained in Game 4 is the distribution $\{w/x_i\}$ from the proposition statement with $x_i \coloneqq f \cdot s_i + g \cdot e_i$ and $w \coloneqq u \cdot g$. □

Remark 3.9. Note that we require $\|x_i^{\ell-1}\| < q$ and that NTRU for $h = x_1/x_2$ is hard. We pick $\ell \in O(1)$ as per the discussion in Section 2.10.

Remark 3.10. By Lemma 2.1 and Lemma 2.5 when $f, g, s_i, e_i \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$ we have:

$$\Pr\left[\|f \cdot s_i + g \cdot e_i\| > 2N \cdot \sigma_x^2 \cdot \omega(\log N)\right] < \mathsf{negl}(\lambda) \ .$$

3.2.3 Proof

We are now ready to prove Theorem 3.5. The result follows by repeatedly applying Lemma 3.6 h times and randomising ring scalars using Lemma 3.8. The norm bounds on the trapdoors are obtained from Lemma 3.7. This allows us to construct well-distributed trapdoors using Lemma 2.14.

Proof of Theorem 3.5. Assume that we have a PPT adversary $\mathcal{A}$ that solves the h-PRISIS problem with non-negligible probability ε. We aim to use $\mathcal{A}$ to solve any PRISIS instance $(\mathbf{A}, \mathbf{B}, \mathsf{aux}, \mathbf{T})$ where $\mathsf{aux} = w$.

We proceed through a series of hybrids in which we transform the h-PRISIS input in order to plant the given PRISIS instance within. Let us denote the adversary's winning probability in Game i with ε_i.

Game 1: (Standard h-PRISIS security game) We sample an h-PRISIS instance $(\{\,\mathbf{A}_i\,\},\{\,\mathbf{B}_i\,\},\{\,w_i\,\},\{\,\mathbf{T}_i\,\})$ honestly and hand it to th adversary $\mathcal{A}$. For every index i the inputs have the following distributions:

1. $\forall i \in [h], \mathbf{A}_i \leftarrow \mathcal{U}(\mathcal{R}_q^{n\times m})$,
2. $\mathsf{aux}_i = w_i \leftarrow \mathcal{U}(\mathcal{R}_q)$,
3. $\mathbf{B}_i = \begin{bmatrix} \mathbf{A}_i & & -\mathbf{G}_n \\ & \ddots & \vdots \\ & w_i^{\ell-1}\cdot\mathbf{A}_i & -\mathbf{G}_n \end{bmatrix}$,
4. $\mathbf{T}_i \leftarrow \mathbf{B}_{i\,\sigma_{Ti}}^{-1}(\mathbf{G}_{n\cdot\ell})$.

By definition $\varepsilon_1 = \varepsilon$.

Game 2: (Replacing the scalar distribution) When sampling our h-PRISIS, we replace the auxiliary ring elements w_i with $\frac{w}{x_i}$ where $w \leftarrow \mathcal{U}(\mathcal{R}_q)$ and $\forall i : x_i = f\cdot s_i + g\cdot e_i$ with $f, g, \{s_i\}, \{e_i\} \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$. If the adversary's advantage against h-PRISIS is different when given $\{\frac{w}{x_i}\}$ instead of uniformly random ring elements we can build a distinguisher for these two distributions.

Given a challenge tuple $\{w_i\}$ we do the following. We generate h tuples $(\tilde{\mathbf{A}}_i, \tilde{\mathbf{T}}_i) \leftarrow \mathsf{TrapGen}(n, m)$ satisfying conditions of Lemma 2.13. Using $\tilde{\mathbf{T}}_i$ we generate trapdoors for $\tilde{\mathbf{B}}_i$ constructed from $\tilde{\mathbf{A}}_i$ and w_i. If the adversary has the expected advantage then the distingusher detected the uniform distribution, otherwise it detected $\{\frac{w}{x_i}\}$.

By Lemma 3.8 such a distinguisher does not exist under the NTRU and RLWE assumptions. Which allows us to conclude that the adversary cannot distinguish between $\{\frac{w}{x_i}\}$ and the uniform distribution of scalars. Therefore, adversary's winning probability $\varepsilon_2 \geq \varepsilon_1 - \mathsf{negl}(\lambda)$.

Game 3: (Replacing the matrix distributions) We receive a PRISIS instance, $(\mathbf{A}, w, \mathbf{T})$ and sample $\mathbf{R}_{i,1}, \mathbf{R}_{i,2} \in \mathcal{R}^{m/2\times m/2}$ with columns from $\mathcal{D}_{\mathcal{R}^{m/2},\sigma_R}$. Recall

$$\mathbf{M}_i := \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{R}_{i1} \\ \mathbf{0} & \mathbf{Id}_{m/2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{Id}_{m/2} & \mathbf{0} \\ \mathbf{R}_{i2} & \mathbf{Id}_{m/2} \end{bmatrix} .$$

We set our h-PRISIS matrices $\mathbf{A}_i := \mathbf{A}\cdot\mathbf{M}_i$ for $1 \leq i \leq h$. Since $\mathbf{A} \leftarrow \mathcal{U}(\mathcal{R}_q^{n\times m})$ each $\mathbf{A}_i$ is statistically close to uniformly random and independent of $\mathbf{A}$ by Lemma 3.6.

We also sample $\{x_i\}$ where $\forall i : x_i = f\cdot s_i + g\cdot e_i$ with $f, g, \{s_i\}, \{e_i\} \leftarrow \mathcal{D}_{\mathcal{R},\sigma_x}$ as in Game 2.

Moreover, for each i we sample $\mathbf{T}'_i = \left(\mathbf{T}'_{i,1}, \ldots, \mathbf{T}'_{i,(\ell-1)}, \mathbf{T}'_{i,\ell}\right)$ where $\mathbf{T}'_{i,\ell} = \mathbf{T}_{0,\ell}$ and $\mathbf{T}'_{i,j} = x_i^j \cdot \mathbf{M}_i^{-1} \cdot \mathbf{T}_{0,j}$ when $j \leq \ell - 1$. Here $\mathbf{T}_{0,j}$ is the challenge trapdoor blocks. As a result for each i we have the following:

$$\begin{bmatrix} \mathbf{A}_i & & -\mathbf{G}_n \\ & \ddots & \vdots \\ & w_i^{\ell-1}\cdot\mathbf{A}_i & -\mathbf{G}_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}'_{i,0} \\ \vdots \\ \mathbf{T}'_{i,\ell} \end{bmatrix} = \mathbf{G}_{n\cdot\ell} .$$

The trapdoors $\mathbf{T}'_i$ we have computed have a bounded norm, but they are not distributed according to a spherical Discrete Gaussian. We use the Preimage Sampling Algorithm defined in Lemma 2.14 to resample them. This way we compute $\mathbf{T}_i := \mathsf{SamplePre}(\mathbf{B}_i, \mathbf{T}'_i, \mathbf{G}_{n\,\ell}, \sigma_{Ti})$.

Lemma 3.7 establishes a bound on the norm of $\mathbf{T}'_i$, so by Lemma 2.14 the final trapdoor $\mathbf{T}_i$ can have a standard deviation parameter $\sigma_{Ti} \geq \delta \cdot \sqrt{(N\,m\,\ell + N\,n\,\tilde{q}) \cdot N\,n\,\tilde{q}\,\ell} \cdot (N \cdot 2N \cdot \sigma_x^2))^{\ell} \cdot (N\,m)^2 \cdot \sigma_T\,\sigma_R^2 \cdot \omega(\log^{3/2+\ell}(N) \cdot N \cdot \sqrt{\log(m \cdot N)})$. We take into account $\sigma_R \geq 2 \cdot N \cdot q^{2n/m+4/(Nm)}$ to obtain the final bound on σ_{Ti}. Since the bound hold with overwhelming probability the distribution of $\mathbf{T}_i$ is statistically close to $\mathbf{B}_{i\,\sigma_{Ti}}^{-1}(\mathbf{G}_{n \cdot \ell})$ and the advantage of the adversary can only change by a negligible value.

Overall, the adversary's winning probability in Game 3, ε_3 is negligibly close to ε_2.

Adversary's output. The values constructed in Game 3 are now related to the initial PRISIS instance. Upon receiving these values the adversary replies with a nonzero vector $\mathbf{v} = (\mathbf{v}_1, \ldots, \mathbf{v}_h)$ such that:

$$[\mathbf{A}_1 \;\; | \;\; \cdots \;\; | \, \mathbf{A}_h] \cdot \mathbf{v} = \sum_{i=1}^{h} \mathbf{A}_i \cdot \mathbf{v}_i = 0 \bmod q \;\; .$$

We can rewrite this equality as $\mathbf{A} \cdot \sum_{i=1}^{h} \mathbf{M}_i \cdot \mathbf{v}_i = 0 \bmod q$. Then $\mathbf{u} := \sum_{i=1}^{h} \mathbf{M}_i \cdot \mathbf{v}_i$ is a candidate solution for the PRISIS problem for $\mathbf{A}$.

To confirm that $\mathbf{u}$ is a valid solution we need to prove the following two statements. First, that the norm of $\mathbf{u}$ is upper bounded by β' and second, that $\mathbf{u} \neq \mathbf{0}$. Applying the techniques from Lemma 3.7 we see that $\left\| \sum_{i=1}^{h} \mathbf{M}_i \cdot \mathbf{v}_i \right\| \leq h \cdot (N\,m)^{5/2} \cdot \beta \cdot \sigma_R^2 \cdot \omega(\log N) < \beta'$.

It remains to prove that $\mathbf{u} \neq \mathbf{0}$. Let us assume that the adversary's goal is to output a solution $\mathbf{v}$ that $\sum_{i=1}^{h} \mathbf{M}_i \cdot \mathbf{v}_i = 0$. Since $\mathcal{A}$ does not know the values of random matrices $\mathbf{M}_i$, their chance to succeed is bounded by $\max_{\mathbf{v} \neq 0} \Pr\left[\sum_i \mathbf{M}_i \cdot \mathbf{v}_i = 0\right]$.

Let us analyse this expression for an arbitrary fixed nonzero vector $\mathbf{v}$. W.l.o.g. we assume that $\mathbf{v}_1 = \mathbf{v}_{:m} \neq 0$. Then $\Pr\left[\sum_{i=1}^{h} \mathbf{M}_i \cdot \mathbf{v}_i = \mathbf{0}\right] \leq \max_{\mathbf{c}} \Pr\left[\mathbf{M}_1 \cdot \mathbf{v}_1 = \mathbf{c}\right]$. For any value of $\mathbf{c}$ the probability above can be expressed as follows:

$$P := \Pr\left[\begin{bmatrix} \mathbf{v}_{:m/2} + \mathbf{R}_{1,1} \cdot (\mathbf{R}_{1,2} + \mathbf{Id}_{m/2}) \cdot \mathbf{v}_{(m/2+1):m} \\ \mathbf{R}_{1,2} \cdot \mathbf{v}_{:m/2} + \mathbf{v}_{(m/2+1):m} \end{bmatrix} = \mathbf{c} \right] \;\; .$$

We know $\exists 1 \leq j \leq m$ such that $\mathbf{v}[j] \neq 0$. There are two cases to consider: $1 \leq j \leq m/2$ and $m/2 < j \leq m$. In the first case

$$\begin{aligned} P &\leq \Pr\left[\mathbf{R}_{1,2} \cdot \mathbf{v}_{:m/2} = \mathbf{c}_{(m/2+1):} - \mathbf{v}_{(m/2+1):m}\right] \\ &\leq \Pr\left[r \cdot \mathbf{v}[j] = c'\right] \leq \max_{c'} \Pr\left[r \cdot \mathbf{v}[j] = c'\right] \;\; . \end{aligned}$$

Here r is the element of matrix $\mathbf{R}_{1,2}$ at position $\{1, j\}$ and $c' = c(\mathbf{c}, \mathbf{v}, \mathbf{R}_{1,2})$ is a value determined by all other entries of $\mathbf{R}_{1,2}$ not including r and vectors $\mathbf{v}, \mathbf{c}$.

The ring $\mathcal{R}$ is an integral domain. Hence, $\forall a, b$ the equation $a \cdot x = b$ has not more than one solution. Therefore, $\forall \varepsilon \in (1/2, 1)$ if $\sigma_R > \eta_\varepsilon(\mathbb{Z}^N)$ then by Lemma 2.2

$$P \leq \max_{c''} \Pr\left[r = c'' | r \leftarrow \mathcal{D}_{\mathcal{R}, \sigma_R}\right] \leq \frac{1}{\sigma_R^N (1 - \varepsilon)} = \mathsf{negl}(\lambda) \;\; .$$

Note that by Lemma 2.3 $\sigma_R \geq 2 \cdot N \cdot q^{2n/m+4/(Nm)} > \sqrt{\frac{\ln(2N(1+1/\varepsilon))}{\pi}} > \eta_\varepsilon(\mathbb{Z}^N)$. In the second case $m/2 < j \leq m$ and

$$P \leq \Pr\left[\mathbf{R}_{1,1} \cdot (\mathbf{R}_{1,2} + \mathbf{Id}_{m/2}) \cdot \mathbf{v}_{(m/2+1):m} = \mathbf{c}_{:m/2}\right]$$

$$
\begin{aligned}
&\leq \max_{\mathbf{c}'} \big(\Pr\big[\mathbf{R}_{1,1} \cdot \mathbf{v}' = \mathbf{c}' \mid (\mathbf{R}_{1,2} + \mathbf{Id}_{m/2}) \cdot \mathbf{v}_{(m/2+1):m} = \mathbf{v}' \neq 0\big] \cdot \Pr\big[\mathbf{v}' \neq 0\big] \\
&+ \Pr\big[\mathbf{v}' = 0\big] \big) \\
&\leq \frac{1}{\sigma_R^N(1-\varepsilon)} + \frac{1}{\sigma_R^N(1-\varepsilon)} = \mathsf{negl}(\lambda) \ ,
\end{aligned}
$$

where for the last transition we apply Lemmas 2.2 and 2.3 again.

We conclude that $\mathbf{u}$ is a valid solution to $(\mathbf{A}, \mathbf{B}, \mathsf{aux}, \mathbf{T})$ instance of $\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma_T,\beta'}$ problem with overwhelming probability. Hence, the winning probability of this adversary against $\mathsf{PRISIS}_{n,m,\mathcal{R}_q,\ell,\sigma_T,\beta'}$ is at least $\varepsilon_3 - \mathsf{negl}(\lambda) \geq \varepsilon - \mathsf{negl}(\lambda)$. □

4 Merkle-PRISIS Commitment Scheme

In this section we define a new compressing commitment scheme which combines the BASIS construction [FN23; WW23b] with Merkle trees (of arity two). This approach significantly reduces the size of the common reference string, as well as the prover running time.

Let $\ell = 2^h$ be the length of the committed message. The message space is $\mathcal{M} := \mathcal{R}_q^\ell$. We let γ be the parameter controlling the norm of the opening vectors. Further, we define the slack space to be the set $\mathcal{S} := \mathcal{R}_q^\times$. We define $\mathbf{G} := \mathbf{G}_n \in \mathcal{R}_q^{n \times t}$ and the decomposition base δ as in Section 2. Also, $\mathbf{e}_1 := (1, 0, \ldots, 0) \in \mathcal{R}_q^n$. The commitment scheme is presented in Figure 4.

4.1 Security Analysis

In the following, we show that the Merkle-PRISIS commitment scheme from Figure 4 satisfies completeness, relaxed binding and hiding.

Lemma 4.1 (Completeness). *Suppose $n, N, \beta_s \geq 1$, define $t := n \cdot \tilde{q}$. Let $m \geq t + n$, $m' := 2m + t$, $n' := 2t$ and $t' := \max(n', m')$*[7] *and $s > 2\,N \cdot q^{\frac{n}{m-t} + \frac{2}{N \cdot (m-t)}}$. Take*

$$
\begin{aligned}
\sigma_0 \geq& 2\,\delta s\, N \cdot \omega(\sqrt{t \cdot (m-t) \cdot \log(t'N)}) \quad \textit{and} \\
\sigma_1 \geq& \delta\, \sigma_0\, N \cdot \omega(\sqrt{m'\, n' \cdot \log(t'N)}) \ .
\end{aligned}
$$

If $\gamma \geq \sigma_1 \sqrt{m'N}$ for $j \in [h]$, then the Merkle-PRISIS commitment scheme satisfies completeness.

Proof. We first focus on the verification equation in Item 2. Fix $\mathbf{b} \in \mathbb{Z}_2^h$. Then by Item 4 of the Commit algorithm, we have for every $j \in [h]$:

$$
w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}_{:j}} + \mathbf{t}_{\mathbf{b}_{:j}} = \mathbf{t}_{\mathbf{b}_{:j-1}} \ ,
$$

where $\mathbf{b}_0$ is defined as the empty string ε. By expanding this equation we obtain

$$
\sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}_{:j}} + f_{\mathbf{b}} \cdot \mathbf{e}_1 = \sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}_{:j}} + \mathbf{t}_{\mathbf{b}_{:h}} = \mathbf{t}_\varepsilon = C \ .
$$

For the second verification check, set a global relaxation factor $c^* = 1 \in \mathcal{S}$. First, note that the matrix $\tilde{\mathbf{R}} \in \mathcal{R}_q^{m' \times n'}$ in Item 5 satisfies $\|\tilde{\mathbf{R}}\| \leq 2\,s \cdot \sqrt{t \cdot (m-t) \cdot N}$ with high probability by

[7] Clearly, $m' \geq n'$. Nevertheless, we define t' this way to explicitly show how we can apply Lemma 2.14.

Merkle-PRISIS Commitment Scheme

$\mathsf{Setup}(1^\lambda)$

```
For $j = h, \ldots, 1$:
    $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m)$.
    $w \leftarrow \mathcal{R}_q^\times$.
    Let $\mathbf{R}_i \coloneqq \mathbf{R} \cdot \mathbf{G}^{-1}(w^{-i} \cdot \mathbf{G})$ for $i \in [0, 1]$.
    Set
        $$\mathbf{B} \coloneqq \left[\begin{array}{cc|c} \mathbf{A} & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w \cdot \mathbf{A} & -\mathbf{G} \end{array}\right], \quad \tilde{\mathbf{R}} \coloneqq \left[\begin{array}{cc} \mathbf{R}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_1 \\ \hline \mathbf{0} & \mathbf{0} \end{array}\right] .$$
    Sample $\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \tilde{\mathbf{R}}, \mathbf{G}_{2n}, \sigma_0)$
    Let $(\mathbf{A}_j, w_j, \mathbf{T}_j) \coloneqq (\mathbf{A}, w, \mathbf{T})$.
Return $\mathsf{crs} \coloneqq (\mathbf{A}_j, w_j, \mathbf{T}_j)_{j \in [h]}$.
```

$\mathsf{Commit}(\mathsf{crs}, \mathbf{f} = (f_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^h} \in \mathcal{R}_q^\ell)$

```
Set $\mathbf{t}_{\mathbf{b}} \coloneqq f_{\mathbf{b}} \cdot \mathbf{e}_1$ for $\mathbf{b} \in \mathbb{Z}_2^h$.
For $j = h, \ldots, 1$:
    For $\mathbf{b} \in \mathbb{Z}_2^{j-1}$:
        $\begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \cdot \mathbf{A}_j & -\mathbf{G} \end{array}\right], \begin{bmatrix} -\mathbf{t}_{(\mathbf{b},0)} \\ -\mathbf{t}_{(\mathbf{b},1)} \end{bmatrix}, \mathbf{T}_j, \sigma_1 \right)$
        Set $\mathbf{t}_{\mathbf{b}} \coloneqq \mathbf{G} \cdot \hat{\mathbf{t}}_{\mathbf{b}}$.
Return $(C \coloneqq \mathbf{t}_\varepsilon, \mathsf{st} \coloneqq (\mathbf{s}_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}})$.
```

$\mathsf{Open}(\mathsf{crs}, C, \mathbf{f}, \mathsf{st}, c)$

```
Parse $C \coloneqq \mathbf{t}$, $\mathbf{f} \coloneqq (f_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^h}$, $\mathsf{st} \coloneqq (\mathbf{s}_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}}$ and $c \in \mathcal{S}$.
Return 1 if and only if for all $\mathbf{b} \in \mathbb{Z}_2^h$,
    • $\sum_{j=1}^h w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}_{:j}} + f_{\mathbf{b}} \cdot \mathbf{e}_1 = \mathbf{t}$.
    • $\forall j \in [h], \left\| c \cdot \mathbf{s}_{\mathbf{b}_{:j}} \right\| \leq \gamma$.
```

Figure 4: Merkle-PRISIS commitment scheme for arbitrary messages of length $\ell = 2^h$ over $\mathcal{R}_q$ with the slack space being $\mathcal{S} \coloneqq \mathcal{R}_q^\times$.

Lemma 2.13. Hence $\sigma_0 \geq \delta \cdot \|\tilde{\mathbf{R}}\| \cdot \omega(\sqrt{N \log(t'N)})$ for $t' = \max(n', m')$ and thus we can apply Lemma 2.14 to deduce that with an overwhelming probability $\|\mathbf{T}_j\| \leq \sigma_0 \cdot \sqrt{m'\, n'\, N}$ for all $j \in [h]$. Similarly, we have $\sigma_1 \geq \delta \cdot \|\mathbf{T}_j\| \cdot \omega(\sqrt{N \cdot \log(t'N)})$ and thus $\|\mathbf{s}_{\mathbf{b}}\| \leq \sigma_1 \cdot \sqrt{m' \cdot N} \leq \gamma$ with an overwhelming probability for any $\mathbf{b} \in \mathbb{Z}_2^{\leq h}$, which concludes the proof. □

Lemma 4.2 (Relaxed Binding). *Define $t := n \cdot \tilde{q}$ and let $m \geq t + n$ and $n' = 2\,t$. Take $m' := 2\,m + t$, $n' := 2\,t$ and $t' := \max(n', m')$ and $s > 2\,N \cdot q^{\frac{n}{m-t} + \frac{2}{N \cdot (m-t)}}$. If $\sigma_0 \geq 2\,\delta s \cdot N \cdot \omega(\sqrt{t \cdot (m-t) \cdot \log(t'N)})$ then under the h-$\mathsf{PRISIS}_{n-1,m,\mathcal{R}_q,2,\sigma_0,2\gamma\sqrt{h}}$ assumption the $\mathsf{PowerBASIS}$ commitment scheme satisfies binding.*

Proof. Let $\mathcal{A}$ be an adversary for the relaxed binding game which succeeds with probability ε. We prove the statement using the hybrid argument. We define ε_i to be the probability that $\mathcal{A}$ wins Game i.

Game 0: This is the standard relaxed binding game. By definition $\varepsilon_0 = \varepsilon$.

Game 1: Here, for each $j \in [h]$ we swap the $\mathsf{SamplePre}$ algorithm with sampling truly from a discrete Gaussian distribution. Since $\sigma_0 \geq \delta \|\tilde{\mathbf{R}}\| \cdot \omega(\sqrt{N \log t'N})$, we can argue as in Lemma 4.1 that $\varepsilon_1 \geq \varepsilon_0 - \mathsf{negl}(\lambda)$.

Game 2: In this game, we do not run $\mathsf{TrapGen}$ anymore, but instead the matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ is selected uniformly at random. By Lemma 2.13, we deduce that $\varepsilon_2 \geq \varepsilon_1 - \mathsf{negl}(\lambda)$.

Game 3: The challenger first samples a vector $\mathbf{b}^* \leftarrow \mathbb{Z}_2^h$. Then, given the output $\mathbf{t}, \mathbf{c}, (\mathbf{f}, \mathsf{st}), (\mathbf{f}', \mathsf{st}')$ from the adversary, it aborts if $f_{\mathbf{b}^*} \neq f'_{\mathbf{b}^*}$. Thus, $\varepsilon_3 \geq \frac{1}{\ell} \cdot \varepsilon_2$.

We claim that $\varepsilon_3 = \mathsf{negl}(\lambda)$ under the h-PRISIS assumption. Suppose we are given the h-PRISIS instance $(\mathbf{A}_j, \mathbf{B}_j, \mathbf{T}_j, w_j)_{j \in [h]}$ from the challenger and we want to find a short non-zero solution for the matrix $[\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_h]$. Recall that for $j = 1, \ldots, h$:

$$\mathbf{B}_j := \left[\begin{matrix} \bar{\mathbf{A}}_j & \mathbf{0} \\ \mathbf{0} & w_j \bar{\mathbf{A}}_j \end{matrix} \;\middle|\; \begin{matrix} -\mathbf{G} \\ -\mathbf{G} \end{matrix}\right]$$

where $\bar{\mathbf{A}}_j := \begin{bmatrix} \mathbf{a}_j^\intercal \\ \mathbf{A}_j \end{bmatrix}$. Now, we need to prepare the correctly formed input $(\mathbf{A}_j^*, w_j^*, \mathbf{T}_j^*)_{j \in [h]}$ to send to the adversary $\mathcal{A}$. Let us fix $j \in [h]$ and set $b := b_j^*$ [8]. Define $\mathbf{A}_j^* := w_j^b \bar{\mathbf{A}}_j$, $w_j^* := w_j^{1-2b}$ and

$$\mathbf{T}_j^* := \begin{bmatrix} \mathbf{T}_{b,b} & \mathbf{T}_{b,1-b} \\ \mathbf{T}_{1-b,b} & \mathbf{T}_{1-b,1-b} \\ \mathbf{T}_{2,b} & \mathbf{T}_{2,1-b} \end{bmatrix} \quad \text{where } \mathbf{T}_j := \begin{bmatrix} \mathbf{T}_{0,0} & \mathbf{T}_{0,1} \\ \mathbf{T}_{1,0} & \mathbf{T}_{1,1} \\ \mathbf{T}_{2,0} & \mathbf{T}_{2,1} \end{bmatrix}.$$

Note that by careful inspection we get for any $b \in \mathbb{Z}_2$:

$$\left[\begin{matrix} \mathbf{A}_j^* & \mathbf{0} \\ \mathbf{0} & w_j^* \mathbf{A}_j^* \end{matrix} \;\middle|\; \begin{matrix} -\mathbf{G} \\ -\mathbf{G} \end{matrix}\right] \cdot \mathbf{T}_j^* = \left[\begin{matrix} w_j^b \bar{\mathbf{A}}_j & \mathbf{0} \\ \mathbf{0} & w_j^{1-b} \bar{\mathbf{A}}_j \end{matrix} \;\middle|\; \begin{matrix} -\mathbf{G} \\ -\mathbf{G} \end{matrix}\right] \cdot \begin{bmatrix} \mathbf{T}_{b,b} & \mathbf{T}_{b,1-b} \\ \mathbf{T}_{1-b,b} & \mathbf{T}_{1-b,1-b} \\ \mathbf{T}_{2,b} & \mathbf{T}_{2,1-b} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}.$$

[8] Recall that $\mathbf{b}^*$ is the vector that was selected by the challenger in Game 3.

Hence, the common reference string $\mathsf{crs} := (\mathbf{A}^*_j, w^*_j, \mathbf{T}^*_j)_{j\in[h]}$ is well-formed and distributed identically as the one output by Setup. Thus, we send crs to $\mathcal{A}$.

Suppose, the adversary outputs the commitment $\mathbf{t}$, the relaxation factor $c \in \mathcal{S}$ and two pairs $(\mathbf{f}, \mathsf{st} = (\mathbf{s}_\mathbf{b})_\mathbf{b}), (\mathbf{f}', \mathsf{st}' = (\mathbf{s}'_\mathbf{b})_\mathbf{b})$ where $f_{\mathbf{b}^*} \neq f'_{\mathbf{b}^*}$. Then, from the verification equations we know

$$\sum_{j=1}^{h} w^{*b^*_j}_j \cdot \mathbf{A}^*_j \cdot \mathbf{s}_{\mathbf{b}^*_{:j}} + f_{\mathbf{b}^*} = \mathbf{t} \quad \text{and} \quad \sum_{j=1}^{h} w^{*b^*_j}_j \cdot \mathbf{A}^*_j \cdot \mathbf{s}'_{\mathbf{b}^*_{:j}} + f'_{\mathbf{b}^*} = \mathbf{t} \ .$$

Thus, by subtracting both equations we get

$$\sum_{j=1}^{h} w^{*b^*_j}_j \cdot \mathbf{A}^*_j \cdot (\mathbf{s}_{\mathbf{b}^*_{:j}} - \mathbf{s}'_{\mathbf{b}^*_{:j}}) + (f_{\mathbf{b}^*} - f'_{\mathbf{b}^*}) \cdot \mathbf{e}_1 = \mathbf{0} \ .$$

Finally, note that by construction of $\mathbf{A}^*_j$ and w^*_j and the fact that $b^*_j \in \mathbb{Z}_2$:

$$w^{*b^*_j}_j \cdot \mathbf{A}^*_j = w_j^{(1-2b^*_j)b^*_j + b^*_j} \cdot \bar{\mathbf{A}}_j = \bar{\mathbf{A}}_j \ .$$

Therefore, we found a non-zero solution for the matrix $[\mathbf{A}_1 \mid \cdots \mid \mathbf{A}_h]$ since $f_{\mathbf{b}^*} \neq f'_{\mathbf{b}^*}$. Even though it may not be short, we have

$$\left\| c \cdot \begin{bmatrix} \mathbf{s}_{\mathbf{b}^*_{:1}} - \mathbf{s}'_{\mathbf{b}^*_{:1}} \\ \vdots \\ \mathbf{s}_{\mathbf{b}^*_{:h}} - \mathbf{s}'_{\mathbf{b}^*_{:h}} \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} c(\mathbf{s}_{\mathbf{b}^*_{:1}} - \mathbf{s}'_{\mathbf{b}^*_{:1}}) \\ \vdots \\ c(\mathbf{s}_{\mathbf{b}^*_{:h}} - \mathbf{s}'_{\mathbf{b}^*_{:h}}) \end{bmatrix} \right\|^2 \leq \sum_{j=1}^{h} (2\gamma)^2 = 4h\gamma^2 \ .$$

Hence, we found a non-zero solution for h-PRISIS with norm at most $2\gamma\sqrt{h}$, since $c \in \mathcal{R}^\times_q$. □

Hiding. In order to argue hiding, we first note that the underlying PRISIS commitment is hiding (see [FN23, Lemma 4.3]), and thus the bottom non-leaf commitments $(\mathbf{t}_\mathbf{b})_{\mathbf{b}\in\mathbb{Z}_2^{h-1}}$ look pseudo-random. Therefore, any commitments computed in higher nodes (in particular, the root) do not leak any information about the message $\mathbf{f}$. Since the methodology is folklore and we do not instantiate the hiding variant of our commitment, we leave a formal treatment out of scope of this paper.

Efficiency. If we assume that $n, m, N \in \mathsf{poly}(\lambda)$, then the common reference string contains $\log \ell \cdot \mathsf{poly}(\lambda)$ elements in $\mathcal{R}_q$, while the prover and the verifier make $\ell \cdot \mathsf{poly}(\lambda)$ ring operations.

5 Proof of Polynomial Evaluation

We use the construction in Figure 4 to build our polynomial commitment scheme. Namely, given a polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ of degree at most $d := 2^h - 1$ over $\mathcal{R}_q$, we commit to f by committing to its coefficient vector $\mathbf{f} = (f_\mathbf{b})_{\mathbf{b}\in\mathbb{Z}_2^h} \in \mathcal{R}_q^{d+1}$ to obtain a commitment $\mathbf{t} \in \mathcal{R}_q^n$, along with the decommitment state $\mathsf{st} = (\mathbf{s}_\mathbf{b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}}$, where each $\mathbf{s}_\mathbf{b} \in \mathcal{R}_q^m$. Here, we represent a polynomial f as

$$f(\mathsf{X}) = \sum_{\mathbf{b}\in\mathbb{Z}_2^h} f_\mathbf{b} \cdot \mathsf{X}^{\mathsf{int}(\mathbf{b})} \ .$$

We say that the $\mathbf{b}$-th coefficient of f is $f_{\mathbf{b}}$.

An essential property of polynomial commitments is the ability to show that the committed polynomial was evaluated correctly, i.e. $f(u) = z$ for public u and z in $\mathcal{R}_q$. In other words, we consider the following relation:

$$\mathsf{R}_{h,\beta} := \left\{ \begin{array}{c|c} \Big((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j \in [h]}, & \forall\ \mathbf{b} \in \mathbb{Z}_2^h,\ \|\mathbf{s}_{\mathbf{b}}\| \leq \beta \\ (\mathbf{t}, u, z), & \wedge \sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}:j} + f_{\mathbf{b}} = \mathbf{t} \\ (f, (\mathbf{s}_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}}) \Big) & \wedge f(u) = z \end{array} \right\} . \tag{3}$$

5.1 Compressed Σ-Protocol

The main intuition for proving evaluations can be described with the following Σ-protocol. First, we define $k \in [h]$ to be the folding factor and $l := h - k$.

The prover starts by splitting the polynomial f into 2^k polynomials of degree at most $d' := 2^l - 1$. Namely, we introduce a $(k+1)$-variate function $\bar{f} : \mathcal{R}_q \times \mathbb{Z}_2^k \to \mathcal{R}_q$ as follows:

$$\bar{f}(\mathsf{X}, \mathsf{I}) := \sum_{\mathbf{j} \in \mathbb{Z}_2^l} f_{(\mathsf{I}, \mathbf{j})} \cdot \mathsf{X}^{\mathsf{int}(\mathbf{j})} .$$

Then, by construction and the fact that $\mathsf{int}((\mathbf{i}, \mathbf{j})) = \mathsf{int}(\mathbf{i}) + 2^k \cdot \mathsf{int}(\mathbf{j})$:

$$\begin{aligned} z = f(u) &= \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \left(\sum_{\mathbf{j} \in \mathbb{Z}_2^l} f_{(\mathbf{i}, \mathbf{j})} \cdot u^{2^k \cdot \mathsf{int}(\mathbf{j})} \right) \cdot u^{\mathsf{int}(\mathbf{i})} \\ &= \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \bar{f}(u^{2^k}, \mathbf{i}) \cdot u^{\mathsf{int}(\mathbf{i})} \\ &= \sum_{\mathbf{i} \in \mathbb{Z}_2^k} z_{\mathbf{i}} \cdot u^{\mathsf{int}(\mathbf{i})} \quad , \end{aligned}$$

where for each $\mathbf{i} \in \mathbb{Z}_2^k$, we define $z_{\mathbf{i}} := \bar{f}(u^{2^k}, \mathbf{i}) \in \mathcal{R}_q$. The partial evaluations $(z_{\mathbf{i}})_{\mathbf{i}}$ are then sent to the verifier. The prover also outputs the partial openings $(\mathbf{s}_{\mathbf{i}})_{\mathbf{i} \in \mathbb{Z}_2^{\leq k}}$ in the clear; later we will explain the meaning behind this move. After the first round, the verifier already checks whether:

$$z = \sum_{\mathbf{i} \in \mathbb{Z}_2^k} z_{\mathbf{i}} \cdot u^{\mathsf{int}(\mathbf{i})} \quad \text{and} \quad \|\mathbf{s}_{\mathbf{i}}\| \leq \beta \text{ for } \forall \mathbf{i} \in \mathbb{Z}_2^{\leq k} . \tag{4}$$

Now, the verifier outputs the challenge vector $\boldsymbol{\alpha} = (\alpha_{\mathbf{i}})_{\mathbf{i}} \leftarrow \mathcal{X}^{2^k}$, and the prover computes the folded polynomial of degree d'

$$g(\mathsf{X}) := \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot \bar{f}(\mathsf{X}, \mathbf{i}) .$$

So far, this protocol focused on proving the evaluation $f(u) = z$. We additionally need to prove knowledge of the opening Merkle-PRISIS commitment $\mathbf{t}$. Let $\mathbf{j} \in \mathbb{Z}_2^l$. Then, the $\mathbf{j}$-th coefficient of g satisfies

$$g_{\mathbf{j}} \cdot \mathbf{e}_1 = \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot f_{(\mathbf{i}, \mathbf{j})} \cdot \mathbf{e}_1$$

$$= \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot \left(\mathbf{t} - \sum_{t=1}^{k} w_t^{i_t} \cdot \mathbf{A}_t \cdot \mathbf{s}_{\mathbf{i}_{:t}} - \sum_{t=1}^{l} w_{k+t}^{j_t} \cdot \mathbf{A}_{k+t} \cdot \mathbf{s}_{(\mathbf{i},\mathbf{j}_{:t})} \right) .$$

Thus, we obtain

$$\sum_{t=1}^{l} w_{k+t}^{j_t} \mathbf{A}_{k+t} \cdot \left(\sum_{\mathbf{i} \in \mathbb{Z}_2^k} \boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{s}_{(\mathbf{i},\mathbf{j}_{:t})} \right) + g_{\mathbf{j}} \, \mathbf{e}_1 = \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot \left(\mathbf{t} - \sum_{t=1}^{k} w_t^{i_t} \, \mathbf{A}_t \cdot \mathbf{s}_{\mathbf{i}_{:t}} \right) \tag{5}$$

where the right-hand side can be computed by the verifier given the initial commitment $\mathbf{t}$ and partial openings $\mathbf{s}_{\mathbf{i}_{:t}}$. Hence, by setting for $\mathbf{j} \in \mathbb{Z}_2^{\leq l}$:

$$\mathbf{z}_{\mathbf{j}} \coloneqq \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \boldsymbol{\alpha}_{\mathbf{i}} \cdot \mathbf{s}_{(\mathbf{i},\mathbf{j})}, \quad \mathbf{t}' \coloneqq \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot \left(\mathbf{t} - \sum_{t=1}^{k} w_t^{i_t} \cdot \mathbf{A}_t \cdot \mathbf{s}_{\mathbf{i}_{:t}} \right) \quad \text{and}$$
$$z' \coloneqq \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\mathbf{i}} \cdot z_{\mathbf{i}}$$

we can check that:

$$\left((\mathbf{A}_{k+t}, \; w_{k+t}, \; \mathbf{T}_{k+t})_{t \in [l]}, \; (\mathbf{t}', \; u^{2^k}, \; z'), \; (g, \; (\mathbf{z}_{\mathbf{j}})_{\mathbf{j} \in \mathbb{Z}_2^{\leq l}}) \right) \in \mathsf{R}_{l, 2^k \beta} \; .$$

Thus, in the third round the prover outputs $(g, (\mathbf{z}_{\mathbf{j}})_{\mathbf{j} \in \mathbb{Z}_2^{\leq l}})$, and the verifier checks the claim above, along with (4). We highlight that the newly formed statement $(\mathbf{t}', u^{2^k}, z')$ can be constructed directly by the verifier.

Similarly as in [FN23], one can show that the soundness error of the protocol above is $2^k/(2N)$. In order to amplify soundness, we directly consider proving r polynomial evaluations $f_\iota(u) = z_\iota$ at the same point u for $\iota = 1, \ldots, r$, where r will be the amplification parameter. Hence, in the setting of our commitment scheme, we are interested in the following ternary relation:

$$\mathsf{R}_{h,\beta}^{(r)} \coloneqq \left\{ \left(\begin{array}{c} (\mathbf{A}_j, \; w_j, \; \mathbf{T}_j)_{j \in [h]}, \\ ((\mathbf{t}_\iota)_{\iota \in [r]}, \; u, \; (z_\iota)_{\iota \in [r]}), \\ (f_\iota, (\mathbf{s}_{\iota,\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}})_{\iota \in [r]} \end{array} \right) \middle| \begin{array}{c} \forall \iota \in [r], \; f_\iota(u) = z_\iota \wedge \forall \mathbf{b} \in \mathbb{Z}_2^h, \\ \wedge \sum_{j=1}^{h} w_j^{b_j} \mathbf{A}_j \cdot \mathbf{s}_{\iota,\mathbf{b}_{:j}} + f_{\iota,\mathbf{b}} = \mathbf{t}_\iota \\ \wedge \forall j \in [h], \|\mathbf{s}_{\iota,\mathbf{b}_{:j}}\| \leq \beta \end{array} \right\} . \tag{6}$$

To handle proving multiple polynomial evaluations at the same point u the prover defines functions $\bar{f}_\iota$ with respect to the polynomials f_ι for $\iota \in [r]$ as before, and computes $z_{\iota,\mathbf{i}} \coloneqq \bar{f}_\iota(u^{2^k}, \mathbf{i})$ for $\mathbf{i} \in \mathbb{Z}_2^k$. It outputs $(z_{\iota,\mathbf{i}})_{\iota,\mathbf{i}}$ along with the partial openings $(\mathbf{s}_{\iota,\mathbf{i}})_{\iota \in [r], \mathbf{i} \in \mathbb{Z}_2^{\leq k}}$. The verifier replies with a challenge $(\boldsymbol{\alpha}_{\iota,\mathbf{i}})_{\iota \in [r], \mathbf{i} \in \mathbb{Z}_2^k} \leftarrow (\mathcal{X}^r)^{r 2^k}$ where $\boldsymbol{\alpha}_{\iota,\mathbf{i}} \coloneqq (\alpha_{\iota,\mathbf{i},1}, \ldots, \alpha_{\iota,\mathbf{i},r}) \in \mathcal{X}^r$. Next, the prover computes r polynomials $g_1, \ldots, g_r$ defined as:

$$g_\kappa(\mathsf{X}) \coloneqq \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\iota,\mathbf{i},\kappa} \cdot \bar{f}_\iota(\mathsf{X}, \mathbf{i}) \quad \text{for } \kappa = 1, \ldots, r.$$

Using the same strategy as in (5), we deduce that the $\mathbf{j}$-th coefficient of g_κ satisfies:

$$\sum_{t=1}^{l} w_{k+t}^{j_t} \cdot \mathbf{A}_{k+t} \cdot \left(\sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \boldsymbol{\alpha}_{\iota,\mathbf{i},\kappa} \cdot \mathbf{s}_{\iota,(\mathbf{i},\mathbf{j}_{:t})} \right) + g_{\kappa,\mathbf{j}} \cdot \mathbf{e}_1$$

$$= \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\iota,\mathbf{i},\kappa} \cdot \left(\mathbf{t}_\iota - \sum_{t=1}^{k} w_t^{i_t} \cdot \mathbf{A}_t \cdot \mathbf{s}_{\iota,\mathbf{i}_{:t}} \right) \ .$$

Hence, by defining for $\mathbf{j} \in \mathbb{Z}_2^{\leq l}$ and $\kappa \in [r]$:

$$\begin{aligned}
\mathbf{z}_{\kappa,\mathbf{j}} &:= \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\iota,\mathbf{i},\kappa} \cdot \mathbf{s}_{\iota,(\mathbf{i},\mathbf{j})}, \\
\mathbf{t}'_\kappa &:= \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\iota,\mathbf{i},\kappa} \cdot \left(\mathbf{t}_\iota - \sum_{t=1}^{k} w_t^{i_t} \cdot \mathbf{A}_t \cdot \mathbf{s}_{\iota,\mathbf{i}_{:t}} \right), \\
z'_\kappa &:= \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha_{\iota,\mathbf{i},\kappa} \cdot z_{\iota,\mathbf{i}} \ ,
\end{aligned}$$

we obtain

$$\begin{pmatrix} (\mathbf{A}_{k+t}, w_{k+t},\ \mathbf{T}_{k+t})_{t \in [l]}, \\ \left((\mathbf{t}'_\kappa)_{\kappa \in [r]},\ u^{2^k},\ (z'_\kappa)_{\kappa \in [r]} \right), \\ \left(g_\kappa,\ (\mathbf{z}_{\kappa,\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq l}} \right)_{\kappa \in [r]} \end{pmatrix} \in \mathsf{R}^{(r)}_{l, r2^k\beta} \ . \tag{7}$$

Hence, the prover sends $(g_\kappa, (\mathbf{z}_{\kappa,\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}})_{\kappa \in [r]}$, and the verifier checks (7), and whether for all $\iota \in [r]$:

$$z_\iota = \sum_{\mathbf{i} \in \mathbb{Z}_2^k} z_{\iota,\mathbf{i}} \cdot u^{\mathsf{int}(\mathbf{i})} \quad \text{and} \quad \|\mathbf{s}_{\iota,\mathbf{i}}\| \leq \beta \text{ for } \forall \mathbf{i} \in \mathbb{Z}_2^{\leq k} \ . \tag{8}$$

As we will show in the more general case, soundness error of this Σ-protocol is $r2^k/(2N)^r$ which is negligible for e.g. $N = \mathsf{poly}(\lambda)$, and $r, k = O(\log d)$.

5.2 Succinct Arguments via Recursion

In order to achieve succinct proofs and verification, we extend the Σ-protocol above as follows. Concretely, instead of checking (7) manually, the verifier recursively runs the Σ-protocol ℓ times (or until the degrees of the r committed polynomials are zero). This yields a $(2\ell+1)$-th round protocol where $\ell \leq h/k$ (here we assume that k is divisible by h). We recall the notation in Table 2 and describe the resulting interactive proof in Figure 5.

Security analysis. In the following, we prove completeness and coordinate-wise special soundness of the protocol in Figure 5.

Lemma 5.1 (Completeness)**.** *The protocol in Figure 5 satisfies perfect completeness.*

Proof. Intuitively, completeness follows from the discussion on the Σ-protocol in Section 5.1 applied inductively. Nevertheless, due to an overwhelming amount of notation, we carefully show completeness via the following claims.

Claim 5.2. *Let $0 \leq \tau \leq \ell$. Then, for every $\kappa \in [r]$, $f_{\tau,\kappa}(u_\tau) = z_{\tau,\kappa}$.*

Interactive Protocol for $\mathsf{R}^{(r)}_{h,\beta}$

```
$\mathcal{P}\left((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, ((\mathbf{t}_{0,\iota})_{\iota\in[r]}, u_0, (z_{0,\iota})_{\iota\in[r]}), (f_{0,\iota}, (\mathbf{s}_{0,\iota,\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})_{\iota\in[r]}\right)$
1. Set $l_0 = h$.
2. For $\tau \in [\ell]$:
    (a) Set $l_\tau \coloneqq l_{\tau-1} - k$.
    (b) Compute $u_\tau \coloneqq u_{\tau-1}^{2^k}$.
    (c) For $\iota \in [r]$ and $\mathbf{i} \in \mathbb{Z}_2^k$:
        i. Set $\bar{f}_{\tau-1,\iota}(\mathsf{X}, \mathbf{i}) \coloneqq \sum_{\mathbf{j}\in\mathbb{Z}_2^{l_\tau}} f_{\tau-1,\iota,(\mathbf{i},\mathbf{j})} \cdot \mathsf{X}^{\mathsf{int}(\mathbf{j})} \in \mathcal{R}_q[\mathsf{X}]$.
        ii. Set $z_{\tau-1,\iota,\mathbf{i}} \coloneqq \bar{f}_{\tau-1,\iota}(u_\tau, \mathbf{i})$.
    (d) Send $\left((z_{\tau-1,\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^k}, (\mathbf{s}_{\tau-1,\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq k}}\right)_{\iota\in[r]}$ to the verifier.
    (e) Receive $(\boldsymbol{\alpha}^{(\tau)}_{\iota,\mathbf{i}})_{\iota\in[r],\mathbf{i}\in\mathbb{Z}_2^k} \leftarrow (\mathcal{X}^r)^{r2^k}$ from the verifier.
    (f) For $\kappa \in [r]$:
        i. Compute $f_{\tau,\kappa}(\mathsf{X}) \coloneqq \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} \bar{f}_{\tau-1,\iota}(\mathsf{X}, \mathbf{i})$
        ii. Compute $\mathbf{s}_{\tau,\kappa,\mathbf{j}} \coloneqq \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} \mathbf{s}_{\tau-1,\iota,(\mathbf{i},\mathbf{j})}$ for $\mathbf{j} \in \mathbb{Z}_2^{\leq l_\tau}$
3. Send $((f_{\ell,\kappa}) \in \mathcal{R}_q^{\leq 2^{h-k\ell}-1}[\mathsf{X}], (\mathbf{s}_{\ell,\kappa,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq h-k\ell}})_{\kappa\in[r]}$ to the verifier.

$\mathcal{V}((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, ((\mathbf{t}_{0,\iota})_{\iota\in[r]}, u_0, (z_{0,\iota})_{\iota\in[r]}))$
1. Set $l_0 = h$ and $\beta_0 \coloneqq \beta$.
2. For $\tau \in [\ell]$:
    (a) Set $l_\tau \coloneqq l_{\tau-1} - k$.
    (b) Compute $u_\tau \coloneqq u_{\tau-1}^{2^k}$.
    (c) Receive $\left((z_{\tau-1,\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^k}, (\mathbf{s}_{\tau-1,\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq k}}\right)_{\iota\in[r]}$ from the prover.
    (d) Check for $\iota \in [r]$:
        i. $z_{\tau-1,\iota} = \sum_{\mathbf{i}\in\mathbb{Z}_2^k} z_{\tau-1,\iota,\mathbf{i}} \cdot u_\tau^{\mathsf{int}(\mathbf{i})}$
        ii. $\|\mathbf{s}_{\tau-1,\iota,\mathbf{i}}\| \leq (r2^k)^{\tau-1}\beta$ for all $\mathbf{i} \in \mathbb{Z}_2^{\leq k}$
    (e) Sample $(\boldsymbol{\alpha}^{(\tau)}_{\iota,\mathbf{i}})_{\iota\in[r],\mathbf{i}\in\mathbb{Z}_2^k} \leftarrow (\mathcal{X}^r)^{r2^k}$ and send it to the prover.
    (f) For $\kappa \in [r]$:
        i. For $\iota \in [r], \mathbf{i} \in \mathbb{Z}_2^k$, $\tilde{\mathbf{t}}_{\iota,\mathbf{i},\mathbf{k}} \coloneqq \mathbf{t}_{\tau-1,\iota} - \sum_{t=1}^k w^{i_t}_{(\tau-1)k+t} \mathbf{A}_{(\tau-1)k+t} \mathbf{s}_{\tau-1,\iota,\mathbf{i}:t}$
        ii. Set $\mathbf{t}_{\tau,\kappa} \coloneqq \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} \tilde{\mathbf{t}}_{\iota,\mathbf{i},\mathbf{k}}$.
        iii. Set $z_{\tau,\kappa} \coloneqq \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} z_{\tau-1,\iota,\mathbf{i}}$.
    (g) Set $\beta_\tau \coloneqq r2^k \cdot \beta_{\tau-1}$.
3. Receive $((f_{\ell,\kappa}) \in \mathcal{R}_q^{\leq 2^{h-k\ell}-1}[\mathsf{X}], (\mathbf{s}_{\ell,\kappa,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq h-k\ell}})_{\kappa\in[r]}$ from the prover.
4. Check:
    $\begin{pmatrix} (\mathbf{A}_{k\ell+t}, w_{k\ell+t}, \mathbf{T}_{k\ell+t})_{t\in[h-k\ell]}, \\ ((\mathbf{t}_{\ell,\kappa})_{\kappa\in[r]}, u_\ell, (z_{\ell,\kappa})_{\kappa\in[r]}), \\ ((f_{\ell,\kappa}), (\mathbf{s}_{\ell,\kappa,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq h-k\ell}})_{\kappa\in[r]} \end{pmatrix} \in \mathsf{R}^{(r)}_{h-k\ell,(r2^k)^\ell\beta}$.
```

Figure 5: Interactive protocol for $\mathsf{R}^{(r)}_{h,\beta}$ with notation from Table 2. Intuitively, index τ keeps track of the number of iterations of the compressed Σ-protocol, while indices ι and κ are used as in Section 5.1.

Table 2: Overview of parameters and notation.

Parameter	**Explanation**
q	proof system modulus
N	degree of the cyclotomic ring $\mathcal{R} \coloneqq \mathbb{Z}[X]/(X^N + 1)$
d	degree of the committed polynomial $f \in \mathcal{R}_q[\mathsf{X}]$
$\mathcal{X}$	Set of signed monomials $\pm X^i$ of $\mathcal{R}$
n	height of the matrices $\mathbf{A}_j$
m	width of the matrices $\mathbf{A}_j$
δ	decomposition base of the gadget matrix $\mathbf{G}$
$\tilde{q}$	$\lfloor \log_\delta q \rfloor + 1$
h	positive integer such that $d + 1 = 2^h$
k	folding factor of the folding protocol, divisor of h
l	$h - k$
ℓ	$\leq h/k$
β	initial norm of the witness openings
γ	extracted norm

Proof. For $\tau = 0$, this is equivalent to $f_{0,\kappa}(u_0) = z_{0,\kappa}$ which is true by assumption on the input given to the prover $\mathcal{P}$. For $\tau \geq 1$, by construction (cf. Items 2(c)i, 2(c)ii and 2(f)i of the prover algorithm) we have:

$$f_{\tau,\kappa}(u_\tau) = \sum_{\iota=1}^{r} \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} \bar{f}_{\tau-1,\iota}(u_\tau, \mathbf{i}) = \sum_{\iota=1}^{r} \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\tau)}_{\iota,\mathbf{i},\kappa} z_{\tau-1,\iota,\mathbf{i}} = z_{\tau,\kappa} \ ,$$

which concludes the proof. □

Claim 5.3. *Let $\tau \in [\ell]$. Then, for every $\iota \in [r]$, $z_{\tau-1,\iota} = \sum_{\mathbf{i}\in\mathbb{Z}_2^k} z_{\tau-1,\iota,\mathbf{i}} \cdot u_\tau^{\mathsf{int}(\mathbf{i})}$.*

Proof. Using the claim above we deduce that

$$\begin{aligned} z_{\tau-1,\iota} = f_{\tau-1,\iota}(u_{\tau-1}) &= \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \left(\sum_{\mathbf{j}\in\mathbb{Z}_2^{l_\tau}} f_{\tau-1,\iota,(\mathbf{i},\mathbf{j})} \cdot u_{\tau-1}^{2^k\cdot\mathsf{int}(\mathbf{j})} \right) \cdot u_{\tau-1}^{\mathsf{int}(\mathbf{i})} \\ &= \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \bar{f}_{\tau-1,\iota}(u_{\tau-1}^{2^k}, \mathbf{i}) \cdot u_{\tau-1}^{\mathsf{int}(\mathbf{i})} \\ &= \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \bar{f}_{\tau-1,\iota}(u_\tau, \mathbf{i}) \cdot u_{\tau-1}^{\mathsf{int}(\mathbf{i})} \\ &= \sum_{\mathbf{i}\in\mathbb{Z}_2^k} z_{\tau-1,\iota,\mathbf{i}} \cdot u^{\mathsf{int}(\mathbf{i})} \end{aligned}$$

by Items 2b, 2(c)i and 2(c)ii of the prover algorithm. □

Claim 5.4. *Let $0 \leq \tau \leq \ell$ and $l_\tau \coloneqq h - \tau k$. Then, for every $\kappa \in [r]$ and $\mathbf{j} \in \mathbb{Z}_2^{\leq l_\tau}$, $\|\mathbf{s}_{\tau,\kappa,\mathbf{j}}\| \leq (r2^k)^\tau \beta$.*

Proof. We prove the statement by induction on τ. For $\tau = 0$, the claim holds by assumption on $(\mathbf{s}_{0,\kappa,\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}}$ as in Equation (6). Now suppose the statement is true for some $\tau - 1$. Using the fact that each $\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}$ is of the form $\pm X^j$, we conclude that (cf. Item 2(f)ii of the prover algorithm)

$$\begin{aligned}\|\mathbf{s}_{\tau,\kappa,\mathbf{j}}\| &\leq \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k} \|\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}\mathbf{s}_{\tau-1,\iota,(\mathbf{i},\mathbf{j})}\| \\ &\leq \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k} \|\mathbf{s}_{\tau-1,\iota,(\mathbf{i},\mathbf{j})}\| \leq (r2^k)\cdot(r2^k)^{\tau-1}\beta = (r2^k)^{\tau}\beta \ .\end{aligned}$$

□

Claim 5.5. *Let $0 \leq \tau \leq \ell$ and $l_\tau := h - \tau k$. Take any $\kappa \in [r]$ and $\mathbf{j} = (j_1, \ldots, j_{l_\tau}) \in \mathbb{Z}_2^{l_\tau}$. Then,*

$$\sum_{t=1}^{l_\tau} w^{j_t}_{\tau k+t}\mathbf{A}_{\tau k+t}\mathbf{s}_{\tau,\kappa,\mathbf{j}_{:t}} + f_{\tau,\kappa,\mathbf{j}}\mathbf{e}_1 = \mathbf{t}_{\tau,\kappa} \ .$$

Proof. We prove the statement by induction. Let $\tau = 0$. Then, it is equivalent to:

$$\sum_{t=1}^{h} w^{j_t}_{t}\cdot\mathbf{A}_{t}\cdot\mathbf{s}_{0,\kappa,\mathbf{j}_{:t}} + f_{0,\kappa,\mathbf{j}} = \mathbf{t}_{0,\kappa}$$

for $\mathbf{j} \in \mathbb{Z}_2^h$. This is true by definition of the witness in Equation (6).

Now, suppose the statement holds for some $\tau - 1 \geq 0$. Take any $\mathbf{j} \in \mathbb{Z}_2^{l_\tau}$. By the induction hypothesis we can write for any $\mathbf{i} \in \mathbb{Z}_2^k$:

$$\mathbf{t}_{\tau-1,\kappa} = \sum_{t=1}^{k} w^{i_t}_{(\tau-1)k+t}\cdot\mathbf{A}_{(\tau-1)k+t}\cdot\mathbf{s}_{\tau-1,\kappa,\mathbf{i}_{:t}} + \sum_{t=1}^{l_\tau} w^{j_t}_{\tau k+t}\cdot\mathbf{A}_{\tau k+t}\cdot\mathbf{s}_{\tau-1,\kappa,(\mathbf{i},\mathbf{j}_{:t})} + f_{\tau-1,\kappa,(\mathbf{i},\mathbf{j})}\cdot\mathbf{e}_1 \ .$$

Thus, by definition of $\mathbf{t}_{\tau,\kappa}$ in Item 2(f)ii of the verifier algorithm, and also by Items 2(f)i and 2(f)ii in the prover algorithm:

$$\begin{aligned}\mathbf{t}_{\tau,\kappa} =& \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k}\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}\cdot\left(\mathbf{t}_{\tau-1,\iota} - \sum_{t=1}^{k} w^{i_t}_{(\tau-1)k+t}\cdot\mathbf{A}_{(\tau-1)k+t}\cdot\mathbf{s}_{\tau-1,\iota,\mathbf{i}_{:t}}\right) \\ =& \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k}\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}\cdot\left(\sum_{t=1}^{l_\tau} w^{j_t}_{\tau k+t}\cdot\mathbf{A}_{\tau k+t}\cdot\mathbf{s}_{\tau-1,\kappa,(\mathbf{i},\mathbf{j}_{:t})} + f_{\tau-1,\kappa,(\mathbf{i},\mathbf{j})}\cdot\mathbf{e}_1\right) \\ =& \sum_{t=1}^{l_\tau} w^{j_t}_{\tau k+t}\cdot\mathbf{A}_{\tau k+t}\cdot\left(\sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k}\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}\cdot\mathbf{s}_{\tau-1,\iota,(\mathbf{i},\mathbf{j}_{:t})}\right) + \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k}\alpha^{(\tau)}_{\iota,\mathbf{i},\kappa}\cdot f_{\tau-1,\iota,(\mathbf{i},\mathbf{j})}\cdot\mathbf{e}_1 \\ =& \sum_{t=1}^{l_\tau} w^{j_t}_{\tau k+t}\cdot\mathbf{A}_{\tau k+t}\cdot\mathbf{s}_{\tau,\kappa,\mathbf{j}_{:t}} + f_{\tau,\kappa,\mathbf{j}}\cdot\mathbf{e}_1\end{aligned}$$

which concludes the proof. □

Finally, correctness holds by applying all the claims above. □

To argue coordinate-wise special soundness, we consider a relaxed relation:

$$\tilde{\mathsf{R}}^{(r)}_{h,\gamma,\xi} := \left\{ \left(\begin{matrix} (\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, \\ ((\mathbf{t}_\iota)_{\iota\in[r]}, u, (z_\iota)_{\iota\in[r]}), \\ (f_\iota, (\mathbf{s}_{\iota,\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})_{\iota\in[r]} \end{matrix} \right) \middle| \begin{matrix} \forall\iota\in[r], f_\iota(u) = z \wedge \forall\mathbf{b}\in\mathbb{Z}_2^h, \\ \wedge \sum_{j=1}^h w_j^{b_j}\mathbf{A}_j\mathbf{s}_{\iota,\mathbf{b}_{:j}} + f_{\iota,\mathbf{b}}\mathbf{e}_1 = \mathbf{t}_\iota \\ \wedge\forall j\in[h], \|\xi\cdot\mathbf{s}_{\iota,\mathbf{b}_{:j}}\| \leq \gamma \end{matrix} \right\} . \tag{9}$$

Let us note the difference from the original relation $\mathsf{R}^{(r)}_{h,\beta}$ in (6). Namely, we do not require the opening vectors to have norm at most β, and thus they do not need to be short anymore. One can also see the connection between the relaxed notion and the slack space of the commitment in Section 4. Also, we observe that $\tilde{\mathsf{R}}^{(r)}_{h,\beta,1} = \mathsf{R}^{(r)}_{h,\beta}$ which is the relation appearing in Item 4 of Figure 5.

Let us consider the tree of transcripts as in Definition 2.27. Informally, the following lemma describes a procedure to extract a witness corresponding to a (non-leaf) node, given witnesses corresponding to its $r2^k+1$ children. Thus, we deduce that our protocol is $r2^k$-coordinate-wise special sound.

Lemma 5.6 (Extraction From a Non-leaf Node)**.** *Let $\tau\in[\ell]$ and $\beta^*,\gamma,\xi > 0$. Define $\mathbb{i} := (\mathbf{A}_{k(\tau-1)+t}, w_{k(\tau-1)+t}, \mathbf{T}_{k(\tau-1)+t})_{t\in[h-k(\tau-1)]}$, and the statement $\mathbb{x} := ((\mathbf{t}_\iota)_{\iota\in[r]}, u, (z_\iota)_{\iota\in[r]})$. Consider $r2^k+1$ triples $(\mathbb{a}, \mathbb{c}_\mu, \mathbb{z}_\mu)_{\mu\in[0,r2^k]}$ defined as:*[9]

$$\begin{aligned} \mathbb{a} &:= \left((z_{\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^k}, (\mathbf{s}_{\iota,\mathbf{i}})_{\mathbf{i}\in\mathbb{Z}_2^{\leq k}}\right)_{\iota\in[r]} \\ \mathbb{c}_\mu &:= (\boldsymbol{\alpha}^{(\mu)}_{\iota,\mathbf{i}})_{\iota\in[r],\mathbf{i}\in\mathbb{Z}_2^k} \in (\mathcal{X}^r)^{r2^k} \\ \mathbb{z}_\mu &:= ((f^{(\mu)}_{\kappa,\mathbf{j}})_{\mathbf{j}\in\mathbb{Z}_2^{h-\tau k}}, (\mathbf{s}^{(\mu)}_{\kappa,\mathbf{j}})_{\mathbf{j}\in\mathbb{Z}_2^{\leq h-\tau k}})_{\kappa\in[r]} \end{aligned}$$

which satisfy for all $\iota\in[r]$: $(\mathbb{c}_\mu)_\mu \in \mathsf{SS}(\mathcal{X}^r, r2^k)$,

$$z_\iota = \sum_{\mathbf{i}\in\mathbb{Z}_2^k} z_{\iota,\mathbf{i}}\cdot u^{\mathsf{int}(\mathbf{i})} \quad \textit{and} \quad \|\mathbf{s}_{\iota,\mathbf{i}}\| \leq \beta^* \textit{ for all } \mathbf{i}\in\mathbb{Z}_2^{\leq k} \quad ,$$

and for $\mu\in[0,r2^k]$:

$$\left(\begin{matrix} (\mathbf{A}_{k\tau+t}, w_{k\tau+t}, \mathbf{T}_{k\tau+t})_{t\in[h-k\tau]}, \\ ((\mathbf{t}^{(\mu)}_\kappa)_{\kappa\in[r]}, u^{2^k}, (z^{(\mu)}_\kappa)_{\kappa\in[r]}) \quad , \\ ((f^{(\mu)}_\kappa), (\mathbf{s}^{(\mu)}_{\kappa,\mathbf{j}})_{\mathbf{j}\in\mathbb{Z}_2^{\leq h-k\tau}})_{\kappa\in[r]} \end{matrix} \right) \in \mathsf{R}^{(r)}_{h-k\tau,\gamma,\xi}$$

where

$$\begin{aligned} \mathbf{t}^{(\mu)}_\kappa &:= \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\mu)}_{\iota,\mathbf{i},\kappa} \left(\mathbf{t}_\iota - \sum_{t=1}^k w^{i_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\mathbf{s}_{\iota,\mathbf{i}_{:t}} \right) \quad , \\ z^{(\mu)}_\kappa &:= \sum_{\iota=1}^r \sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\mu)}_{\iota,\mathbf{i},\kappa} z_{\iota,\mathbf{i}} \quad . \end{aligned}$$

[9] One can think of $\mathbb{a}$ as the message sent by the prover at a particular (non-leaf) node of the tree, $(\mathbb{c}_\mu)_\mu$ as the labels on the edges to its children, and $(\mathbb{z}_\mu)_\mu$ to be the witnesses already extracted in the children nodes. For example, if the children are leaves then $(\mathbb{z}_\mu)_\mu$ are simply the prover's last messages.

Then, there exists an efficient deterministic algorithm that given triples $(\mathbb{a}, \mathbb{c}_\mu, \mathbb{z}_\mu)_{\mu\in[0,r2^k]}$*, outputs* $\mathbb{w}$ *such that*

$$(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \tilde{\mathsf{R}}^{(r)}_{h-k(\tau-1),\gamma^*,2\xi} \quad \textit{where} \quad \gamma^* := \max\left(2\xi\beta^*, 2N\gamma\right) \ .$$

Proof. Without loss of generality, we can reorder the $r2^k+1$ triples $(\mathbb{a}, \mathbb{c}_\mu, \mathbb{z}_\mu)_{\mu\in[0,r2^k]}$ as $(\mathbb{a}, \mathbb{c}_0, \mathbb{z}_0)$ and $(\mathbb{a}, \mathbb{c}_{\rho,\boldsymbol{\nu}}, \mathbb{z}_{\rho,\boldsymbol{\nu}})_{\rho\in[r],\boldsymbol{\nu}\in\mathbb{Z}_2^k}$, where we now denote

$$\mathbb{c}_{\rho,\boldsymbol{\nu}} := (\boldsymbol{\alpha}^{(\rho,\boldsymbol{\nu})}_{\iota,\mathbf{i}})_{\iota\in[r],\mathbf{i}\in\mathbb{Z}_2^k} \in (\mathcal{X}^r)^{r2^k},$$
$$\mathbb{z}_{\rho,\boldsymbol{\nu}} := \left((f^{(\rho,\boldsymbol{\nu})}_{\kappa,\mathbf{j}})_{\mathbf{j}\in\mathbb{Z}_2^{h-\tau k}}, (\mathbf{s}^{(\rho,\boldsymbol{\nu})}_{\kappa,\mathbf{j}})_{\mathbf{j}\in\mathbb{Z}_2^{\le h-\tau k}} \right)_{\kappa\in[r]},$$

such that for any $\rho\in[r], \boldsymbol{\nu}\in\mathbb{Z}_2^k$, we have

$$\forall(\iota,\mathbf{i}) \neq (\rho,\boldsymbol{\nu}), \boldsymbol{\alpha}^{(0)}_{\iota,\mathbf{i}} = \boldsymbol{\alpha}^{(\rho,\boldsymbol{\nu})}_{\iota,\mathbf{i}} \quad \text{and} \quad \boldsymbol{\alpha}^{(0)}_{\rho,\boldsymbol{\nu}} \neq \boldsymbol{\alpha}^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu}}.$$

Similarly, we denote $\mathbf{t}^{(\rho,\boldsymbol{\nu})}_\kappa$ and $z^{(\rho,\boldsymbol{\nu})}_\kappa$.

Let us fix $\rho\in[r]$ and $\mathbf{j}\in\mathbb{Z}_2^{h-k(\tau-1)}$. We will construct $h-k(\tau-1)$ vectors $\bar{\mathbf{s}}_1,\ldots,\bar{\mathbf{s}}_{h-k(\tau-1)}\in\mathcal{R}_q^m$ and a coefficient $\bar{f}_{\rho,\mathbf{j}}\in\mathcal{R}_q$ such that

$$\begin{aligned} &\sum_{t=1}^{h-k(\tau-1)} w^{j_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\bar{\mathbf{s}}_t + \bar{f}_{\rho,\mathbf{j}}\mathbf{e}_1 = \mathbf{t}_\rho \ , \\ &\|2\xi\cdot\bar{\mathbf{s}}_t\| \le \gamma^* \text{ for } t\in[h-k(\tau-1)] \ . \end{aligned} \tag{10}$$

By repeating the same argument for all $\mathbf{j}\in\mathbb{Z}_2^{h-k(\tau-1)}$, if the polynomial $\bar{f}_\rho := \sum_{\mathbf{j}\in\mathbb{Z}_2^{h-k(\tau-1)}} \bar{f}_{\rho,\mathbf{j}}\cdot\mathsf{X}^{\mathsf{int}(\mathbf{j})}$ satisfies $\bar{f}_\rho(u) = z_\rho$, then the statement follows.

To this end, write $\mathbf{j} := (\boldsymbol{\nu},\mathbf{j}^*)$ where $\boldsymbol{\nu}\in\mathbb{Z}_2^k$ and $\mathbf{j}^*\in\mathbb{Z}^{h-\tau k}$. Consider the two transcripts: $(\mathbb{a},\mathbb{c}_0,\mathbb{z}_0)$ and $(\mathbb{a},\mathbb{c}_{\rho,\boldsymbol{\nu}},\mathbb{z}_{\rho,\boldsymbol{\nu}})$. We know that $\boldsymbol{\alpha}^{(0)}_{\rho,\boldsymbol{\nu}} \neq \boldsymbol{\alpha}^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu}}$. Let $\eta\in[r]$ be an index such that $\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} \neq \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}$. Note that η is independent of $\mathbf{j}^*$. By definition of (9), we get:

$$\sum_{t=1}^{h-\tau k} w^{j^*_t}_{\tau k+t}\mathbf{A}_{\tau k+t}\mathbf{s}^{(0)}_{\eta,\mathbf{j}^*_{:t}} + f^{(0)}_{\eta,\mathbf{j}^*}\mathbf{e}_1 = \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(0)}_{\iota,\mathbf{i},\eta}\left(\mathbf{t}_\iota - \sum_{t=1}^{k} w^{i_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\mathbf{s}_{\iota,\mathbf{i}_{:t}}\right)$$

and

$$\sum_{t=1}^{h-\tau k} w^{j^*_t}_{\tau k+t}\mathbf{A}_{\tau k+t}\mathbf{s}^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*_{:t}} + f^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*}\mathbf{e}_1 = \sum_{\iota=1}^{r}\sum_{\mathbf{i}\in\mathbb{Z}_2^k} \alpha^{(\rho,\boldsymbol{\nu})}_{\iota,\mathbf{i},\eta}\left(\mathbf{t}_\iota - \sum_{t=1}^{k} w^{i_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\mathbf{s}_{\iota,\mathbf{i}_{:t}}\right) .$$

By subtracting the two equations we obtain:

$$\sum_{t=1}^{h-\tau k} w^{j^*_t}_{\tau k+t}\mathbf{A}_{\tau k+t}\left(\mathbf{s}^{(0)}_{\eta,\mathbf{j}^*_{:t}} - \mathbf{s}^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*_{:t}}\right) + \left(f^{(0)}_{\eta,\mathbf{j}^*} - f^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*}\right)\mathbf{e}_1 = \left(\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} - \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}\right)\left(\mathbf{t}_\rho - \sum_{t=1}^{k} w^{\nu_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\mathbf{s}_{\rho,\boldsymbol{\nu}_{:t}}\right) \ .$$

Hence, we define vectors $\bar{\mathbf{s}}_1,\ldots,\bar{\mathbf{s}}_{h-k(\tau-1)}$ as:

$$\bar{\mathbf{s}}_t := \mathbf{s}_{\rho,\boldsymbol{\nu}_{:t}} \text{ for } t\in[k] \quad \text{and} \quad \bar{\mathbf{s}}_{k+t} := \frac{\mathbf{s}^{(0)}_{\eta,\mathbf{j}^*_{:t}} - \mathbf{s}^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*_{:t}}}{\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} - \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}} \text{ for } t\in[h-k\tau]$$

and

$$\bar{f}_{\rho,\mathbf{j}} \coloneqq \frac{f^{(0)}_{\eta,\mathbf{j}^*} - f^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*}}{\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} - \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}} \in \mathcal{R}_q \ .$$

Then, since $\mathbf{j} \coloneqq (\boldsymbol{\nu}, \mathbf{j}^*)$, we obtain the first part of (10). As for the latter part, note that

$$\|2\xi \cdot \bar{\mathbf{s}}_t\| \leq 2\xi\beta^* \leq \gamma^*$$

for $t \in [k]$, and using Lemma 2.6

$$\|2\xi \cdot \bar{\mathbf{s}}_{k+t}\| \leq \left\| \frac{2}{\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} - \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}} \right\|_1 \cdot \left\| \eta \cdot \left(\mathbf{s}^{(0)}_{\eta,\mathbf{j}^*_{:t}} - \mathbf{s}^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*_{:t}} \right) \right\| \leq 2N\gamma \leq \gamma^*$$

for $t \in [\ell + 1 - \tau]$. Thus, (10) holds.

Finally, we need to show that $\bar{f}_\rho$ satisfies $\bar{f}_\rho(u) = z_\rho$. We rewrite $\bar{f}_\rho$ as

$$\bar{f}_\rho = \sum_{\boldsymbol{\nu} \in \mathbb{Z}_2^k} \sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} \bar{f}_{\rho,(\boldsymbol{\nu},\mathbf{j}^*)} \cdot \mathsf{X}^{2^k \mathsf{int}(\mathbf{j}^*)} \cdot \mathsf{X}^{\mathsf{int}(\boldsymbol{\nu})}$$

Recall we have $z_\rho = \sum_{\boldsymbol{\nu} \in \mathbb{Z}_2^k} z_{\rho,\boldsymbol{\nu}} \cdot u^{\mathsf{int}(\boldsymbol{\nu})}$. Thus, it is sufficient to show that for every $\boldsymbol{\nu} \in \mathbb{Z}_2^k$ we have

$$z_{\rho,\boldsymbol{\nu}} = \sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} \bar{f}_{\rho,(\boldsymbol{\nu},\mathbf{j}^*)} \cdot u^{2^k \mathsf{int}(\mathbf{j}^*)} \ .$$

Fix $\boldsymbol{\nu}$ and define $\eta \in [r]$ as before. From (9) we deduce that

$$\sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} f^{(0)}_{\eta,\mathbf{j}^*} \cdot u^{2^k \mathsf{int}(\mathbf{j}^*)} = f^{(0)}_\eta(u^{2^k}) = z^{(0)}_\eta = \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha^{(0)}_{\iota,\mathbf{i},\eta} z_{\iota,\mathbf{i}}$$

$$\sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} f^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*} \cdot u^{2^k \mathsf{int}(\mathbf{j}^*)} = f^{(\rho,\boldsymbol{\nu})}_\eta(u^{2^k}) = z^{(\rho,\boldsymbol{\nu})}_\eta = \sum_{\iota=1}^{r} \sum_{\mathbf{i} \in \mathbb{Z}_2^k} \alpha^{(\rho,\boldsymbol{\nu})}_{\iota,\mathbf{i},\eta} z_{\iota,\mathbf{i}} \ .$$

Therefore

$$z_{\rho,\boldsymbol{\nu}} = \sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} \left(\frac{f^{(0)}_{\eta,\mathbf{j}^*} - f^{(\rho,\boldsymbol{\nu})}_{\eta,\mathbf{j}^*}}{\alpha^{(0)}_{\rho,\boldsymbol{\nu},\eta} - \alpha^{(\rho,\boldsymbol{\nu})}_{\rho,\boldsymbol{\nu},\eta}} \right) \cdot u^{2^k \mathsf{int}(\mathbf{j}^*)} = \sum_{\mathbf{j}^* \in \mathbb{Z}_2^{\ell+1-\tau}} \bar{f}_{\rho,(\boldsymbol{\nu},\mathbf{j}^*)} \cdot u^{2^k \mathsf{int}(\mathbf{j}^*)}$$

which concludes the proof. □

We are ready to prove coordinate-wise special soundness.

Lemma 5.7 (Coordinate-Wise Special Soundness)**.** *Define* $\gamma^* \coloneqq (2^{k+1} r N)^\ell \beta$*. If* $(r 2^k + 1)^\ell = \mathsf{poly}(\lambda, d)$*, then the interactive proof in Figure 5 is* $r2^k$*-coordinate-wise special sound w.r.t. the relation* $\tilde{\mathsf{R}}^{(r)}_{h,\gamma^*,2^\ell}$*.*

Proof. We prove the statement using Lemma 5.6 inductively, starting from the bottom non-leaf nodes, i.e. for $\tau = \ell, \ell-1, \ldots, 1$. For example, if $\tau = \ell$ then using the notation from Lemma 5.6, $\gamma := (r2^k)^\ell\beta$, $\beta^* := (r2^k)^{\ell-1}\beta$ and $\xi = 1$ (cf. Item 2(d)ii of the verifier algorithm). Hence, the extractor outputs a witness for the relation $\tilde{\mathsf{R}}^{(r)}_{h-k(\ell-1),\gamma_\ell,2}$ where

$$\gamma_\ell := \max(2\beta^*, 2N\gamma) = 2\,N \cdot (r2^k)^\ell\beta \ .$$

Then by induction on τ, one can extract a witness for the relation $\tilde{\mathsf{R}}^{(r)}_{h-k(\tau-1),\gamma_\tau,2^{\ell-\tau+1}}$ where $\gamma_\tau := (2N)^{\ell-\tau+1}(r2^k)^\ell\beta$. The statement follows by setting $\tau = 1$. □

The result above in particular says that one can extract a relaxed opening for the Merkle-PRISIS commitment with the relaxation factor $2^{h/k-1} = 2^\ell \in \mathcal{R}_q^\times$. Further, by Lemma 2.28, this implies that the Fiat-Shamir transformation of the protocol in Figure 5 is knowledge sound in the random oracle model with knowledge error $(Q+1)\ell r2^k/(2N)^r$, where Q is the number of random oracle queries made by an adversary.

Efficiency. We analyse the efficiency of the protocol in the next lemma.

Lemma 5.8 (Efficiency)**.** *The total communication complexity of the protocol in Figure 5 (in bits) can be bounded by*

$$\underbrace{(\ell+1)\cdot(2^kN\lceil\log q\rceil)}_{\textit{partial evaluations}} + \underbrace{2^{k+1}mN\cdot\sum_{i=0}^{\ell-1}\lceil\log 2(r2^k)^i\beta\rceil}_{\textit{short openings}} + \underbrace{\ell\cdot r^2 2^k\lceil\log 2N\rceil}_{\textit{verifier messages}}$$
$$+ \underbrace{2^{h-k\ell}N\lceil\log q\rceil + 2^{h-k\ell+1}\lceil\log 2(r2^k)^\ell\beta\rceil}_{\textit{final message}} \ .$$

Further, accounting both in terms of operations over $\mathcal{R}_q$, the prover runs in time $O(r^2md + r2^{h-k\ell})$ and the verifier in time $O\left(\ell\cdot 2^kn(km+r^2) + r2^{h-k\ell}\right)$.

Proof. We start with the communication complexity. Note that the size of each i-th of the first ℓ messages (counting from zero) can be naively bounded by

$$2^kN\lceil\log q\rceil + 2^{k+1}mN\lceil\log 2(r2^k)^i\beta\rceil \ .$$

Size of the last message can be naively bounded by

$$2^{h-k\ell}N\lceil\log q\rceil + 2^{h-k\ell+1}\lceil\log 2(r2^k)^\ell\beta\rceil \ .$$

Meanwhile the total size of the verifier messages is $\ell\cdot r^2 2^k\lceil\log 2N\rceil$.

Next, consider the prover runtime in a single iteration of the loop in Item 2 for some $\tau\in[\ell]$. The main bottleneck is the procedure in Item 2(f)ii which takes $r\cdot 2^{k+l_\tau+1}\cdot m = r\cdot 2^{l_{\tau-1}+1}\cdot m$ operations over $\mathcal{R}_q$. Since we run that line r times, we conclude that the total runtime in a single iteration of the loop is $O(r^2m2^{l_{\tau-1}})$. Hence, the total prover runtime can be bounded by

$$O\left(\sum_{\tau=1}^{\ell} r^2m2^{l_{\tau-1}} + r2^{h-k\ell}\right) = O\left(r^2m\sum_{\tau=1}^{\ell}2^{h-(\tau-1)k} + r2^{h-k\ell}\right) = O(r^2md + r2^{h-k\ell}) \ .$$

Table 3: Parameters for the polynomial commitment scheme obtained from Figure 4 and running the protocol in Figure 5 for proofs of evaluation.

Parameters	**Instantiation**
δ	$q^{1/O(1)}$
t	$n\tilde{q}$
m	$\geq t+n$
m'	$2m+t$
n'	$2t$
t'	$\max(n',m')$
s	$> 2N \cdot q^{\frac{n}{m-t}+\frac{2}{N\cdot(m-t)}}$
σ_0	$\geq 2\delta s N \cdot \omega(\sqrt{t(m-t)\log t'N})$
σ_1	$\geq \delta\sigma_0 N \cdot \omega(\sqrt{m'n'\log t'N})$
β	$\geq \sigma_1\sqrt{m'N}$
γ	$(2^{k+1}rN)^{\ell}\beta$
k	$O(\log\log d)$
r	$O(\log\lambda)$
ℓ	$h/k = O\left(\frac{\log d}{\log\log d}\right)$

As for the verifier, excluding reading the last message, the main bottleneck is computing the new $\mathbf{t}_{\tau,\kappa}$ in Item 2(f)ii. First, the verifier can compute all the necessary partial sums $\sum_{t=1}^{k} w^{i_t}_{(\tau-1)k+t}\mathbf{A}_{(\tau-1)k+t}\mathbf{s}_{\tau-1,\iota,\mathbf{i}_{:t}}$ in $O(2^k knm)$ operations over $\mathcal{R}_q$. Then, calculating $\mathbf{t}_{\tau,\kappa}$ takes $O(rn2^k)$ time. Since we compute that for every $\kappa \in [r]$, a single iteration of the loop in Item 2f takes $O(2^k n(km+r^2))$ operations. Hence, by iterating over all possible $\tau \in [\ell]$, the verifier runtime can be bounded by $O(\ell \cdot 2^k n(km+r^2) + r2^{h-k\ell})$. □

5.3 Succinct Polynomial Commitment Scheme

Finally, by combining the results above, we obtain a polynomial commitment scheme with polylogarithmic evaluation proofs, quasi-linear prover runtime, and polylogarithmic verifier runtime.

Namely, we use the construction in Figure 4; given a polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ of degree at most $d := 2^h - 1$ over $\mathcal{R}_q$, we commit to f by committing to its coefficient vector $\mathbf{f} = (f_{\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^h} \in \mathcal{R}_q^{d+1}$ to obtain a commitment $\mathbf{t} \in \mathcal{R}_q^n$, together with the decommitment state $\mathsf{st} = (\mathbf{s}_{\mathbf{b}})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}}$. Then, the Eval protocol runs the Fiat-Shamir trasnformed protocol[10] in Figure 5, and Verify verifies the proof. The following theorem summarises our results.

Theorem 5.9 (Polynomial Commitment Scheme)**.** *Let $n, m, N, d \in \mathsf{poly}(\lambda)$ be lattice parameters. Define $\mathsf{PC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}, \mathsf{Verify})$ where $\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}$ are as in Figure 4 and $\mathsf{Eval}, \mathsf{Verify}$ are defined in Figure 5. Take the parameters from Table 3. Then, PC is an interactive polynomial commitment scheme which satisfies evaluation completeness and relaxed binding under the h-$\mathsf{PRISIS}_{n-1,m,\mathcal{R}_q,2,\sigma_1,2\gamma\sqrt{h}}$ assumption. Further,*

[10] Note that the prover in Figure 5 starts with proving r evaluations of r (not necessarily distinct) polynomials $(f_{0,\iota})_{\iota\in[r]}$ at a single point u_0, while for Eval we only require proving one evaluation for a single polynomial f. We can thus naively let Eval run $\mathcal{P}$ for $f = f_{0,1} = \ldots = f_{0,r}$.

- *the evaluation proof consists of* $O(\log^2 d) \cdot \mathsf{poly}(\lambda)$ *elements in* $\mathcal{R}_q$,
- *running time of* Eval *is* $O(d) \cdot \mathsf{poly}(\lambda)$ *operations over* $\mathcal{R}_q$,
- *running time of* Verify *is* $O(\log^2 d) \cdot \mathsf{poly}(\lambda)$ *operations over* $\mathcal{R}_q$,
- PC *is knowledge sound in the random oracle model, with knowledge error* $O\left(\frac{\log^2 d \cdot \log \lambda}{\lambda^{\log N+1}}\right) = \mathsf{negl}(\lambda)$.

Proof. First, evaluation completeness follows from Lemmas 4.1 and 5.1. Then, relaxed binding follows from Lemma 4.2. Note that the parameter γ is chosen with respect to the extracted openings in Lemma 5.7.

The proof sizes and the running times of Eval and Verify come from Lemma 5.8 for $k = O(\log\log d)$, $r = O(\log \lambda)$ and $\ell = O\left(\frac{\log d}{\log\log d}\right)$. The knowledge error can be directly deduced from Lemma 5.7 and Lemma 2.28 while keeping in mind that

$$(r2^k+1)^\ell = O\left(\ell \cdot (r2^k)^\ell\right) = O\left(\frac{\log d}{\log\log d} \cdot (\log\lambda \cdot \log d)^{\frac{\log d}{\log\log d}}\right) = O\left(\frac{\log d}{\log\log d} \cdot d^2\right) = \mathsf{poly}(d) \ .$$

□

Batching. It is easy to see that the protocol in Figure 5 naturally supports proving multiple polynomial evaluations at a single point; indeed, we consider proving r evaluations simultaneously from the very start. Unfortunately, apart from small optimisations in the last round as in [FN23, Section 5.4.2], we do not see how to batch polynomial evaluations proofs for multiple distinct points.

Non-interactive polynomial commitments. $\mathsf{Eval}, \mathsf{Verify}$ can be made non-interactive using the Fiat-Shamir transformation. One can show using the framework from [AFK22] that coordinate-wise special sound protocols maintain knowledge soundness after performing the Fiat-Shamir transformation, with the linear reduction loss in the number of random oracle queries. Since our protocol satisfies coordinate-wise special soundness, this yields a secure non-interactive polynomial commitment scheme.

5.4 Honest-Verifier Zero-Knowledge

We provide a linear-size zero-knowledge proof for relation $\mathsf{R}_{h,\beta}$ in (16). The protocol follows the standard Fiat-Shamir with aborts paradigm [Lyu09; BTT22], and combined with Section 5.2 yields a succinct zero-knowledge proof of polynomial evaluation.

Construction. Suppose the prover is given a witness $(f, (\mathbf{s_b})_\mathbf{b})$ such that

$$\left((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, (\mathbf{t}, u, z), (f, (\mathbf{s_b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})\right) \in \mathsf{R}_{h,\beta} \ .$$

Denote the "leaf commitments" $\mathbf{t_b} := f_\mathbf{b} \cdot \mathbf{e}_i$ for $\mathbf{b} \in \mathbb{Z}_2^h$. Next, we define so-called "partial commitments" $\mathbf{t_b}$ which can be directly computed from the openings.

HVZK Σ-Protocol for $\mathsf{R}_{h,\beta}$

```
Prover                                                                  Verifier
$g \leftarrow \mathcal{R}_q^{\leq d}[\mathsf{X}]$
$\mathbf{v}_{\mathbf{b}} \coloneqq g_{\mathbf{b}} \cdot \mathbf{e}_1$ for $\mathbf{b} \in \mathbb{Z}_2^h$
$z_g \coloneqq g(u)$
for $j = h, \ldots, 1$ :
  for $\mathbf{b} \in \mathbb{Z}_2^{j-1}$ :
    $\begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \cdot \mathbf{A}_j & -\mathbf{G} \end{array}\right], \begin{bmatrix} -\mathbf{v}_{(\mathbf{b},0)} \\ -\mathbf{v}_{(\mathbf{b},1)} \end{bmatrix}, \mathbf{T}_j, \sigma \right)$
    $\mathbf{v}_{\mathbf{b}} \coloneqq \mathbf{G}\hat{\mathbf{v}}_{\mathbf{b}}$
                          $\xrightarrow{\mathbf{v} \coloneqq \mathbf{v}_\varepsilon, z_g}$
                                                                        $\alpha \leftarrow \mathcal{X}$
                          $\xleftarrow{\alpha}$
$h \coloneqq g + \alpha f$
for $j = h, \ldots, 1$ :
  $\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} \coloneqq \begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix}$ for $\mathbf{b} \in \mathbb{Z}_2^{j-1}$
  $\mathbf{z}_j^* \coloneqq \left( \begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} \right)_{\mathbf{b} \in \mathbb{Z}_2^{j-1}}, \mathbf{s}_j^* \coloneqq \left( \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix} \right)_{\mathbf{b} \in \mathbb{Z}_2^{j-1}}$
  $\rho \leftarrow [0,1)$
  if $\rho > \min\left( \frac{\mathcal{D}_\sigma^{m_j' N}(\mathbf{z}_j^*)}{M \cdot \mathcal{D}_{\sigma,\alpha\mathbf{s}_j^*}^{m_j' N}(\mathbf{z}_j^*)}, 1 \right)$ :
    abort
                          $\xrightarrow{h, (\mathbf{z}_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}}}$
                                                                        Check for all $\mathbf{b} \in \mathbb{Z}_2^h$:
                                                                        $h(u) = z_g + \alpha z$
                                                                        $\sum_{j=1}^{h} w_j^{b_j} \mathbf{A}_j \mathbf{z}_{\mathbf{b}:j} + h_{\mathbf{b}}$
                                                                            $= \mathbf{v} + \alpha \mathbf{t}$
                                                                        $\|\mathbf{z}_{\mathbf{b}}\| \leq \beta_z$
```

Figure 6: The honest-verifier zero-knowledge Σ-protocol for $\mathsf{R}_{h,\beta}$. Here, $m_j' \coloneqq 2^{j-1}(2m + n\tilde{q})$ for $j \in [h]$. The vectors $\hat{\mathbf{t}}_{\mathbf{b}}$ are binary decompositions of partial commitments defined in Equation (12).

Lemma 5.10 (Partial Commitments)**.** *Let $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}_2^{j-1}$. Then, for any $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_2^{h-j+1}$ we have:*

$$\begin{aligned}\mathbf{t}_{\mathbf{b}} &:= \sum_{i=1}^{h-j+1} w_{i+j-1}^{x_i} \mathbf{A}_{i+j-1} \mathbf{s}_{(\mathbf{b},\mathbf{x}_i)} + f_{(\mathbf{b},\mathbf{x})} \mathbf{e}_1 \\ &= \sum_{i=1}^{h-j+1} w_{i+j-1}^{x'_i} \mathbf{A}_{i+j-1} \mathbf{s}_{(\mathbf{b},\mathbf{x}'_i)} + f_{(\mathbf{b},\mathbf{x}')} \mathbf{e}_1 \ .\end{aligned} \tag{11}$$

In particular, we have

$$\mathbf{t}_{\mathbf{b}} = \mathbf{A}_j \mathbf{s}_{(\mathbf{b},0)} + \mathbf{t}_{(\mathbf{b},0)} \quad \textit{and} \quad \mathbf{t}_{\mathbf{b}} = w_j \mathbf{A}_j \mathbf{s}_{(\mathbf{b},1)} + \mathbf{t}_{(\mathbf{b},1)} \ .$$

Proof. The second part follows directly from the definition of the partial commitments $\mathbf{t}_{\mathbf{b}}$. For the former one, we observe that (11) is equivalent to the following

$$\begin{aligned}&\left(\sum_{i=1}^{j-1} w_i^{b_i} \mathbf{A}_i \mathbf{s}_{\mathbf{b}:i}\right) + \sum_{i=1}^{h-j+1} w_{i+j-1}^{x_i} \mathbf{A}_{i+j-1} \mathbf{s}_{(\mathbf{b},\mathbf{x}_i)} + f_{(\mathbf{b},\mathbf{x})} \mathbf{e}_1 \\ &\qquad = \left(\sum_{i=1}^{j-1} w_i^{b_i} \mathbf{A}_i \mathbf{s}_{\mathbf{b}:i}\right) + \sum_{i=1}^{h-j+1} w_{i+j-1}^{x'_i} \mathbf{A}_{i+j-1} \mathbf{s}_{(\mathbf{b},\mathbf{x}'_i)} + f_{(\mathbf{b},\mathbf{x}')} \mathbf{e}_1 \ .\end{aligned}$$

Finally, note that both sides are equal to $\mathbf{t}$ by assumption on the relation $\mathsf{R}_{h,\beta}$. □

Next, we define the binary decompositions of the partial commitments as

$$\hat{\mathbf{t}}_{\mathbf{b}} := \mathbf{G}^{-1}(\mathbf{t}_{\mathbf{b}}) \ . \tag{12}$$

Then, by construction and Lemma 5.10 we have for $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}_2^{j-1}$:

$$\begin{bmatrix} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \mathbf{A}_j & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix} = \begin{bmatrix} -\mathbf{t}_{(\mathbf{b},0)} \\ -\mathbf{t}_{(\mathbf{b},1)} \end{bmatrix} \ .$$

We are ready to describe our protocol. The prover first picks a uniformly random masking polynomial $g \in \mathcal{R}_q^{\leq d}[\mathsf{X}]$ and runs the commit algorithm from Section 4. That is, the prover sets the leaf commitments $\mathbf{v}_{\mathbf{b}} := g_{\mathbf{b}} \cdot \mathbf{e}_1$ for $\mathbf{b} \in \mathbb{Z}_2^h$, and then computes for $j = h, \ldots, 1$ and $\mathbf{b} \in \mathbb{Z}^{j-1}$:

$$\begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \mathbf{A}_j & -\mathbf{G} \end{bmatrix}_{\sigma}^{-1} \left(\begin{bmatrix} -\mathbf{v}_{(\mathbf{b},0)} \\ -\mathbf{v}_{(\mathbf{b},1)} \end{bmatrix} \right) \quad \text{and} \quad \mathbf{v}_{\mathbf{b}} := \mathbf{G} \cdot \hat{\mathbf{v}}_{\mathbf{b}} \ .$$

In order to perform this operation efficiently, the prover makes use of the trapdoors $(\mathbf{T}_j)_j$. Further, the prover outputs the commitment $\mathbf{v} := \mathbf{v}_{\varepsilon}$ and the evaluation $z_g := g(u) \in \mathcal{R}_q$. Next, given a challenge $\alpha \leftarrow \mathcal{X}$ from the verifier, the prover computes $h := g + \alpha f$ and

$$\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} := \begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix} \text{ for } \mathbf{b} \in \mathbb{Z}_2^{j-1} \ , \tag{13}$$

and outputs $(h, (\mathbf{z_b})_\mathbf{b})$. Finally, the verifier checks whether

$$\Big((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, (\mathbf{t}_g + \alpha\mathbf{t}, u, z_g + \alpha z), (h, (\mathbf{z_b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})\Big) \in \mathsf{R}_{h,\beta_z}$$

for some suitable bound β_z stated later.

Note that revealing vectors $(\mathbf{z_b})_\mathbf{b}$ as of now may reveal some information about the secrets $(\mathbf{s_b})_\mathbf{b}$. To prevent that, we apply rejection sampling. However, naively applying the procedure for each $\mathbf{z_b}$ (there are $O(d)$ of them) would result in the non-abort probability being negligible. In our final protocol in Figure 6 we show how to increase the non-abort probability to $1/\mathsf{poly}(\lambda)$.

First, for $\mathbf{b} \in \mathbb{Z}_2^h$, define $\hat{\mathbf{z}}_\mathbf{b} := \mathbf{G}^{-1}(h_\mathbf{b} \cdot \mathbf{e}_1)$. Then, by construction we have:

$$\mathbf{G} \cdot \hat{\mathbf{z}}_\mathbf{b} = g_\mathbf{b} \cdot \mathbf{e}_1 + \alpha \cdot f_\mathbf{b} \cdot \mathbf{e}_1 = \mathbf{v_b} + \alpha \cdot \mathbf{t_b} \ .$$

Moreover, by (13) we have that for all $\mathbf{b} \in \mathbb{Z}_2^{\leq h-1}$:

$$\mathbf{G} \cdot \hat{\mathbf{z}}_\mathbf{b} = \mathbf{G} \cdot \hat{\mathbf{v}}_\mathbf{b} + \alpha \cdot \mathbf{G} \cdot \hat{\mathbf{t}}_\mathbf{b} = \mathbf{v_b} + \alpha \cdot \mathbf{t_b} \ .$$

Hence, by (13) again, we obtain for all $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}^{j-1}$:

$$\begin{bmatrix} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_\mathbf{b} \end{bmatrix} = - \begin{bmatrix} \mathbf{v}_{(\mathbf{b},0)} + \alpha \cdot \mathbf{t}_{(\mathbf{b},0)} \\ \mathbf{v}_{(\mathbf{b},1)} + \alpha \cdot \mathbf{t}_{(\mathbf{b},1)} \end{bmatrix} = - \begin{bmatrix} \mathbf{G} \cdot \hat{\mathbf{z}}_{(\mathbf{b},0)} \\ \mathbf{G} \cdot \hat{\mathbf{z}}_{(\mathbf{b},1)} \end{bmatrix} \ . \tag{14}$$

So, if we consider the concatenated vectors

$$\mathbf{z}_j^* := \left(\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_\mathbf{b} \end{bmatrix} \right)_{\mathbf{b}\in\mathbb{Z}_2^{j-1}} \in \mathcal{R}_q^{2^{j-1}(2m+n\tilde{q})}$$

and the lattice

$$\Lambda := \Lambda^\perp \left(\mathbf{I}_{2^{j-1}} \otimes \begin{bmatrix} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{bmatrix} \right) \tag{15}$$

then the distribution of $\mathbf{z}_j^*$ is statistically close to a discrete Gaussian over a coset of Λ for a suitable parameter σ. This is the setting where we apply the rejection sampling from Lemma 2.4. Hence, we apply rejection sampling for each $j = 1, \ldots, h$. If $M = O(1)$ is a constant rejection rate, then the non-abort probability is $\approx 1/M^h = 1/\mathsf{poly}(d)$.

Security analysis. We start by showing completeness.

Lemma 5.11 (Completeness)**.** *Suppose the parameters are chosen as in Lemma 4.1. Set*

$$\sigma \geq \max(\sigma_1, \sqrt{\lambda \cdot (\beta^2 + n\tilde{q}/2) \cdot d}) \quad \textit{and} \quad \beta_z \geq \sigma\sqrt{(2m+n\tilde{q})N} + \beta.$$

Then, the protocol in Figure 6 satisfies completeness with correctness error $1 - 1/M^h$.

Proof. We first focus on the verification equations. The first two hold trivially since

$$h(u) = g(u) + \alpha f(u) = z_g + \alpha z$$

and

$$\sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{z}_{\mathbf{b}:j} + h_{\mathbf{b}} = \sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{y}_{\mathbf{b}:j} + g_{\mathbf{b}} + \alpha \cdot \left(\sum_{j=1}^{h} w_j^{b_j} \cdot \mathbf{A}_j \mathbf{s}_{\mathbf{b}:j} + f_{\mathbf{b}} \right) = \mathbf{v} + \alpha \cdot \mathbf{t} \ .$$

Further, we know from Lemma 4.1 that $\|\mathbf{y}_{\mathbf{b}}\| \leq \sigma_1 \sqrt{(2m+n\tilde{q})N}$ with an overwhelming probability for all indices $\mathbf{b}$. Hence,

$$\|\mathbf{z}_{\mathbf{b}}\| \leq \|\mathbf{y}_{\mathbf{b}}\| + \|\alpha \cdot \mathbf{s}_{\mathbf{b}}\| = \|\mathbf{y}_{\mathbf{b}}\| + \|\mathbf{s}_{\mathbf{b}}\| \leq \sigma_1 \cdot \sqrt{(2m+n\tilde{q}) \cdot N} + \beta \leq \beta_z \ .$$

We now focus on the non-abort probability. First, the conditions in Lemma 4.1 allow us to argue that the correctness error will not change (up to a negligible additive term) if instead of using $\mathsf{SamplePre}$ we directly sample preimages from the discrete Gaussian distribution. Further, by picking σ_1 as in Lemma 4.1 we make sure that

$$\sigma_1 \geq \eta_\varepsilon \left(\Lambda^\perp \left(\begin{bmatrix} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \mathbf{A}_j & -\mathbf{G} \end{bmatrix} \right) \right) \quad \text{for all } j \in [h] \ .$$

for some negligible ε. Moreover, looking at the Gram-Schmidt basis of Λ in (15) and using the bound in Lemma 2.3 we deduce that $\sigma \geq \sigma_1 \geq \eta_\varepsilon(\Lambda)$. Also,

$$\|\alpha \mathbf{s}_j^*\|^2 = \|\mathbf{s}_j^*\|^2 \leq (2\beta^2 + n\tilde{q}) \cdot 2^{j-1} \leq (\beta^2 + n\tilde{q}/2) \cdot d \ .$$

Therefore, $\sigma \geq \sqrt{\lambda} \cdot \|\alpha \mathbf{s}_j^*\|$. Thus, we can apply Lemma 2.4 for each $j \in [h]$ and deduce that the non-abort probability is indeed $(1/M)^h - \mathsf{negl}(\lambda)$. □

As usual, for soundness we consider a relaxed relation of $\mathsf{R}_{h,\beta}$, where the witness is only a relaxed opening of the commitment:

$$\mathsf{R}^*_{h,\gamma} := \left\{ \left(\begin{array}{c} (\mathbf{A}_j, w_j, \mathbf{T}_j)_{j \in [h]}, \\ (\mathbf{t}, u, z), \\ (f, (\mathbf{s}_{\mathbf{b}})_{\mathbf{b} \in \mathbb{Z}_2^{\leq h}}) \end{array} \right) \middle| \begin{array}{c} \forall \mathbf{b} \in \mathbb{Z}_2^h, \|2 \cdot \mathbf{s}_{\mathbf{b}}\| \leq \gamma \\ \wedge \sum_{j=1}^h w_j^{b_j} \cdot \mathbf{A}_j \cdot \mathbf{s}_{\mathbf{b}:j} + f_{\mathbf{b}} = \mathbf{t} \\ \wedge f(u) = z \end{array} \right\} \ . \tag{16}$$

Lemma 5.12 (Special Soundness)**.** *The protocol in Figure 6 is special sound w.r.t. the relation* $\mathsf{R}^*_{h,2\beta_z N}$.

Proof. Consider two accepting transcripts

$$((\mathbf{v}, z_g), \alpha, (h, (\mathbf{z}_{\mathbf{b}})_{\mathbf{b}})) \quad \text{and} \quad ((\mathbf{v}, z_g), \alpha', (h', (\mathbf{z}'_{\mathbf{b}})_{\mathbf{b}}))$$

where $\alpha \neq \alpha'$. We can define

$$\bar{f}(\mathsf{X}) := \frac{h(\mathsf{X}) - h'(\mathsf{X})}{\alpha - \alpha'} \quad \text{and} \quad \bar{\mathbf{s}}_{\mathbf{b}} := \frac{\mathbf{z}_{\mathbf{b}} - \mathbf{z}'_{\mathbf{b}}}{\alpha - \alpha'} \quad \text{for } \mathbf{b} \in \mathbb{Z}_2^{\leq h}.$$

By the first verification equation we know that $\bar{f}(u) = z$. From the second verification equation we get for every $\mathbf{b} \in \mathbb{Z}_2^h, \sum_{j=1}^h w_j^{b_j} \mathbf{A}_j \bar{\mathbf{s}}_{\mathbf{b}:j} + \bar{f}_{\mathbf{b}} = \mathbf{t}$. Finally, we use the property of monomials to deduce that

$$\|2 \cdot \mathbf{z}_{\mathbf{b}}\| \leq \left\| \frac{2}{\alpha - \alpha'} \right\|_1 \cdot \|\mathbf{z}_{\mathbf{b}} - \mathbf{z}'_{\mathbf{b}}\| \leq 2\beta_z N$$

which concludes the proof. □

Remark 5.13. The lemma above implies that the knowledge error of the protocol in Figure 6 is $1/(2N)$, which is not negligible. A simple approach to amplify soundness would be to have the verifier sample many challenges $(\alpha_1, \ldots, \alpha_r) \leftarrow \mathcal{X}$, and for each $i \in [r]$ the prover would output $(h^{(i)}, (\mathbf{z}_{\mathbf{b}}^{(i)})_{\mathbf{b}})$ the same way as before, but corresponding to the challenge α_i. One can show that the protocol would still be special sound; thus the knowledge error becomes $1/(2N)^r$ at the cost of having the proof size almost r times larger.

We now turn to honest-verifier zero-knowledge. That is, we show how to simulate the transcripts when the verifier behaves honestly.

Lemma 5.14 (Honest-Verifier Zero-Knowledge)**.** *Let σ and other parameters be chosen as in Lemma 5.11. Then, the output distributions of $\mathcal{T}$ and $\mathcal{S}$ in Figure 7 are statistically indistinguishable.*

Proof. We prove the statement via a standard hybrid argument.

- Hyb_0 is identical to $\mathcal{T}$ as in Figure 7, but it additionally computes $\hat{\mathbf{z}}_{\mathbf{b}} := \mathbf{G}^{-1}(h_{\mathbf{b}} \cdot \mathbf{e}_1)$ for $\mathbf{b} \in \mathbb{Z}_2^h$.
- Hyb_1 is identical to Hyb_0, but now we compute $\mathbf{v}_\varepsilon := \mathbf{G}\hat{\mathbf{z}}_\varepsilon - \alpha\mathbf{t}$ and $z_g := h(u) - \alpha z$. By construction, the output distribution of Hyb_1 is identical to Hyb_0 since $\hat{\mathbf{z}}_\varepsilon = \hat{\mathbf{v}}_\varepsilon + \alpha\hat{\mathbf{t}}_\varepsilon$ and thus $\mathbf{G}\hat{\mathbf{z}}_\varepsilon = \mathbf{v}_\varepsilon + \alpha\mathbf{t}$. Similarly, $h(u) = z_g + \alpha z$ from the verification equations.
- Hyb_2 is identical to Hyb_1, but now for each $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}_2^{j-1}$ we compute

  $$\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} := \begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_{\mathbf{b}} \end{bmatrix}$$

  where

  $$\begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_{\mathbf{b}} \end{bmatrix} \leftarrow \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \cdot \mathbf{A}_j & -\mathbf{G} \end{array}\right]_\sigma^{-1} \left(\begin{bmatrix} -\mathbf{v}_{(\mathbf{b},0)} \\ -\mathbf{v}_{(\mathbf{b},1)} \end{bmatrix} \right) .$$

  By Lemma 2.14, Hyb_1 and Hyb_2 are statistically close.
- Hyb_3 is identical to Hyb_2, but here for each $j \in [h]$ we directly compute

  $$\mathbf{z}_j^* := \mathbf{y}_j^* + \alpha \cdot \mathbf{s}_j^*$$

  where

  $$\mathbf{y}_j^* \leftarrow \left(\mathbf{I}_{2^{j-1}} \otimes \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j \cdot \mathbf{A}_j & -\mathbf{G} \end{array}\right] \right)_\sigma^{-1} \left(- \left(\begin{bmatrix} \mathbf{v}_{(\mathbf{b},0)} \\ \mathbf{v}_{(\mathbf{b},1)} \end{bmatrix} \right)_{\mathbf{b} \in \mathbb{Z}_2^{j-1}} \right) .$$

  Here, we use the following simple fact: for any matrices $\mathbf{D}$ and $\mathbf{E}$, and image vectors $\mathbf{y}_0, \mathbf{y}_1$, the distributions below are identical:

  $$\left\{ \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \middle| \mathbf{x}_0 \leftarrow \mathbf{D}_\sigma^{-1}(\mathbf{y}_0), \mathbf{x}_1 \leftarrow \mathbf{E}_\sigma^{-1}(\mathbf{y}_1) \right\}$$

  and

  $$\left\{ \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \middle| \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{E} \end{bmatrix}_\sigma^{-1} \left(\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \end{bmatrix} \right) \right\} .$$

  Thus, the distributions of Hyb_2 and Hyb_3 are identical.

```
$\mathcal{T}((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, f, (\mathbf{s_b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}}, \mathbf{t}, u, z)$
$g \leftarrow \mathcal{R}_q^{\leq d}[\mathsf{X}]$
$\mathbf{v_b} \coloneqq g_\mathbf{b} \cdot \mathbf{e}_1$ for $\mathbf{b} \in \mathbb{Z}_2^h$
$z_g \coloneqq g(u)$
for $j = h, \ldots, 1$:
    for $\mathbf{b} \in \mathbb{Z}_2^{j-1}$:
        $\begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_\mathbf{b} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{array}\right], \begin{bmatrix} -\mathbf{v}_{(\mathbf{b},0)} \\ -\mathbf{v}_{(\mathbf{b},1)} \end{bmatrix}, \mathbf{T}_j, \sigma \right)$
        $\mathbf{v_b} \coloneqq \mathbf{G}\hat{\mathbf{v}}_\mathbf{b}$
$\alpha \leftarrow \mathcal{X}$
$h \coloneqq g + \alpha f$
for $j = h, \ldots, 1$ :
    $\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_\mathbf{b} \end{bmatrix} \coloneqq \begin{bmatrix} \mathbf{y}_{(\mathbf{b},0)} \\ \mathbf{y}_{(\mathbf{b},1)} \\ \hat{\mathbf{v}}_\mathbf{b} \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_\mathbf{b} \end{bmatrix}$ for $\mathbf{b} \in \mathbb{Z}_2^{j-1}$
    $\mathbf{z}_j^* \coloneqq \left( \begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_\mathbf{b} \end{bmatrix} \right)_{\mathbf{b}\in\mathbb{Z}_2^{j-1}}, \mathbf{s}_j^* \coloneqq \left( \begin{bmatrix} \mathbf{s}_{(\mathbf{b},0)} \\ \mathbf{s}_{(\mathbf{b},1)} \\ \hat{\mathbf{t}}_\mathbf{b} \end{bmatrix} \right)_{\mathbf{b}\in\mathbb{Z}_2^{j-1}}$
    $\rho \leftarrow [0, 1)$
    if $\rho > \min\left( \frac{\mathcal{D}_\sigma^{m'_j N}(\mathbf{z}_j^*)}{M \cdot \mathcal{D}_{\sigma,\alpha\mathbf{s}_j^*}^{m'_j N}(\mathbf{z}_j^*)}, 1 \right)$ :
        abort
return $((\mathbf{v}_\varepsilon, z_g), \alpha, (h, \mathbf{z_b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})$
```

```
$\mathcal{S}((\mathbf{A}_j, w_j, \mathbf{T}_j)_{j\in[h]}, \mathbf{t}, u, z)$
$h \leftarrow \mathcal{R}_q^{\leq d}[\mathsf{X}]$
$\hat{\mathbf{z}}_\mathbf{b} \coloneqq \mathbf{G}^{-1}(h_\mathbf{b} \cdot \mathbf{e}_1)$ for $\mathbf{b} \in \mathbb{Z}_2^h$
for $j = h, \ldots, 1$:
    for $\mathbf{b} \in \mathbb{Z}_2^{j-1}$:
        $\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_\mathbf{b} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left( \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{array}\right], -\begin{bmatrix} \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},0)} \\ \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},1)} \end{bmatrix}, \mathbf{T}_j, \sigma \right)$
    $\rho \leftarrow [0, 1)$
    if $\rho > 1/M$ :
        abort
$\alpha \leftarrow \mathcal{X}$
$\mathbf{v}_\varepsilon \coloneqq \mathbf{G}\hat{\mathbf{z}}_\varepsilon - \alpha\mathbf{t}$
$v \coloneqq h(u) - \alpha z$
return $((\mathbf{v}_\varepsilon, z_g), \alpha, (h, \mathbf{z_b})_{\mathbf{b}\in\mathbb{Z}_2^{\leq h}})$
```

Figure 7: Simulating the transcripts from the Σ-protocol described in Figure 7.

- For $j = h, \ldots, 1$, Hyb_{4+h-j} is identical to Hyb_{3+h-j}, but now we apply rejection sampling for the vector $\mathbf{z}_j^*$: For $\mathbf{b} \in \mathbb{Z}_2^{j-1}$

$$\mathbf{z}_j^* := \left(\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix}\right) \leftarrow \left(\mathbf{I}_{2^{j-1}} \otimes \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{array}\right]\right)_{\sigma}^{-1} \left(-\left(\begin{bmatrix} \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},0)} \\ \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},1)} \end{bmatrix}\right)\right) \quad ,$$

and with probability $1 - 1/M$ we abort. By the generalised rejection sampling (cf. Lemma 2.4) and Equation (14) we can argue that Hyb_{4+h-j} and Hyb_{3+h-j} are statistically close. Note that the conditions on σ are satisfied by the proof of Lemma 5.11.

- Hyb_{h+4} reverses the change from Hyb_3. That is, for $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}_2^{j-1}$ we compute:

$$\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} \leftarrow \left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{array}\right]_{\sigma}^{-1} \left(-\begin{bmatrix} \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},0)} \\ \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},1)} \end{bmatrix}\right) \quad ,$$

Thus, the output distributions of Hyb_{h+3} and Hyb_{h+4} are identical.

- Hyb_{h+5} reverses the change from Hyb_2. So, for $j \in [h]$ and $\mathbf{b} \in \mathbb{Z}_2^{j-1}$ we compute:

$$\begin{bmatrix} \mathbf{z}_{(\mathbf{b},0)} \\ \mathbf{z}_{(\mathbf{b},1)} \\ \hat{\mathbf{z}}_{\mathbf{b}} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{cc|c} \mathbf{A}_j & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w_j\mathbf{A}_j & -\mathbf{G} \end{array}\right], -\begin{bmatrix} \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},0)} \\ \mathbf{G}\hat{\mathbf{z}}_{(\mathbf{b},1)} \end{bmatrix}, \mathbf{T}_j, \sigma\right) \quad .$$

As before, by Lemma 2.14 we deduce that Hyb_4 and Hyb_3 are statistically close.

- Hyb_{h+6} is identical to Hyb_{h+5}, except now we generate the polynomial $h \leftarrow \mathcal{R}_q^{\leq d}[\mathsf{X}]$ uniformly at random. Since in Hyb_{h+5} coefficients g_i were sampled uniformly at random from $\mathcal{R}_q$ and not used anywhere else apart from computing h_i, we conclude that the output distributions of Hyb_{h+6} and Hyb_{h+5} are identical.

Finally, the output distribution of Hyb_{h+6} is identical to the one by $\mathcal{S}$ which ends the proof. □

5.5 Polynomial Evaluations over Integers

We provide a new method for proving evaluations over $\mathbb{Z}_q$ using our framework which natively operates over the ring $\mathcal{R}_q$.

Suppose $f(u) = z$ over $\mathbb{Z}_q$ and f has degree at most d, which is divisible by the ring dimension N. Then, we can write

$$\begin{aligned} z &= \sum_{i=0}^{d-1} f_i u^i \\ &= \sum_{i=0}^{d/N-1} \sum_{j=0}^{N-1} f_{iN+j} u^{iN+j} \\ &= \sum_{i=0}^{d/N-1} \left(\sum_{j=0}^{N-1} f_{iN+j} u^j\right) \cdot \left(u^N\right)^i \quad . \end{aligned}$$

Let $\sigma_{-1} : \mathcal{R} \to \mathcal{R}$ be the Galois automorphism, which maps $X \mapsto X^{-1}$. We will use the following key observation from [LNP22].

Lemma 5.15 ([LNP22])**.** *For any polynomials $a, b \in \mathcal{R}$, the constant coefficient of $a \cdot \sigma_{-1}(b) \in \mathcal{R}$ is equal to $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{Z}$, where $\mathbf{a}$ (resp. $\mathbf{b}$) is the coefficient vector of a (resp. b).*

Hence, if we define the following $\mathcal{R}_q$-elements:

$$\mathsf{u} := \sum_{j=0}^{N-1} u^j \cdot X^j, \qquad \mathsf{f}_i := \sum_{j=0}^{N-1} f_{iN+j} X^j \quad \text{for } i = 0, 1, \ldots, d/N - 1,$$

then the constant coefficient of

$$\mathsf{z} := \sum_{i=0}^{d/N-1} \sigma_{-1}(\mathsf{u})\mathsf{f}_i \cdot \left(u^N\right)^i$$

is indeed equal to z. Moreover, the equation above is a polynomial evaluation statement $\mathsf{f}(\mathsf{u}) = \mathsf{z}$ over $\mathcal{R}_q$, where the polynomial f has coefficients $(\sigma_{-1}(\mathsf{u})\mathsf{f}_0, \ldots, \sigma_{-1}(\mathsf{u})\mathsf{f}_{d/N-1})$, the evaluation point is $\mathsf{u} := u^N$ and the image is z defined earlier. Hence, the prover can first send $\mathsf{z} \in \mathcal{R}_q$ in the clear[11], and then proceed with proving knowledge of f such that $\mathsf{f}(\mathsf{u}) = \mathsf{z}$. The verifier stays the same as before, with an additional check that the constant coefficient of z is z. The advantage here is that we only prove evaluations of polynomials of degree at most d/N rather than d, which has significant implications in practice.

What we have left to do is to make sure that the polynomial u is invertible over $\mathcal{R}_q$, since otherwise we cannot extract $f \in \mathbb{Z}_q[\mathsf{X}]$ from $\mathsf{f} \in \mathcal{R}_q[\mathsf{X}]$. To this end, note that

$$\mathsf{u} \cdot (uX - 1) = \left(\sum_{j=0}^{N-1} (uX)^j\right) \cdot (uX - 1) = (uX)^N - 1 = -(u^N + 1) \ .$$

This means that u is invertible if and only if u is not a primitive $2N$-th root of unity. Since $q \equiv 5 \bmod 8$, no such roots exist.

6 Concrete Instantiation

MSIS hardness. The $\mathsf{MSIS}_{n,m,\mathcal{R}_q,\beta}$ problem for a uniformly random matrix $\mathbf{A}$ is equivalent to finding a non-trivial vector of norm smaller than β in the lattice $\Lambda := \Lambda^{\perp}(\mathbf{A})$. We make use of the Block-Korkine-Zolatorev algorithm (BZK)[SE94; CN11], which makes itself use of an algorithm for the shortest vector problem (SVP) in lattices of dimension b (we refer to this b as the block size). Using the best known algorithm to solve SVP with no memory constraints, due to Becker et al. [BDGL16], BKZ run with block size b on the mN-dimensional lattice Λ terminates in time $8mN \cdot 2^{0.292b+16.4}$. We would like to compute the minimum block size such that BKZ outputs a feasible solution. Recall that BKZ outputs a vector with norm $\delta_{\mathsf{rhf}}^{mN} \cdot \det(\Lambda)^{\frac{1}{mN}}$, where δ_{rhf} is the root Hermite factor defined as $\delta_{\mathsf{rhf}} := \left(\frac{b(\pi b)^{1/b}}{2\pi e}\right)^{\frac{1}{2(b-1)}}$. In our usual parameter ranges, $\mathbf{A}$ will be of

[11] Observe that the resulting protocol is not zero-knowledge.

Table 4: Parameters and concrete sizes for the polynomial commitment described in Theorem 5.9. δ and norms in log form. Here Q refers to number of adversarial queries to the random oracle allowed (in log form).

k	ℓ	d	λ	Q	n	m	N	δ	$\log q$	β	$\|\pi\|$
1	1	2^{20}	128	64	162	2329	64	22	276	258	36.5 MB
1	8	2^{30}	128	64	583	9559	32	35	508	405	767 MB

full rank with overwhelming probability (cf. Lemma 2.10) and thus $\det(\Lambda) = q^{nN}$. We use the formula from Micciancio and Regev [MR09]:

$$\delta_{\mathsf{rhf}}^{mN} \cdot q^{\frac{nN}{mN}} \geq 2^{2\sqrt{nN \log q \log \delta}} \quad ,$$

with equality when $mN = \sqrt{nN \log q / \log \delta}$. Given a bound $\beta < q$, once can then solve for δ_{rhf}, and obtain b accordingly, which leads us to estimate the time for BKZ to solve MSIS. Since Lemma 4.2 incurs in a multiplicative reduction loss of $d+1$, in estimating parameters we target an higher security level of $\lambda + \log d$ for the underlying MSIS instance.

Deterministic preimage sampling. The instantiations that we are concerned with in this section are not concerned with zero-knowledge. Thus, we can replace the $\mathsf{SamplePre}$ procedure in the commit phase to make us of simpler *deterministic* sampling. More formally, given a matrix $\mathbf{B}$ and a corresponding trapdoor $\mathbf{T}$, one can compute a preimage of a target vector $\mathbf{t}$ as $\mathbf{v} := \mathbf{T} \cdot \mathbf{G}^{-1}(\mathbf{t})$. It can then be easily verified that $\mathbf{B} \cdot \mathbf{v} = \mathbf{B} \cdot \mathbf{T} \cdot \mathbf{G}^{-1}(\mathbf{t}) = \mathbf{G} \cdot \mathbf{G}^{-1}(\mathbf{t}) = \mathbf{t}$. This leads to shorter commitments and openings. First, recall that $s_1(\mathbf{T}_j) \leq \sigma_0 \cdot \sqrt{m'\, n'\, N}$. Then, the norm bound of the resulting opening can be estimated as $\beta \leq s_1(\mathbf{T}_j) \cdot \delta \cdot \sqrt{2\, n \cdot \tilde{q} \cdot N} = \sigma_0 \cdot \sqrt{m'\, n'\, N} \cdot \delta \cdot N \sqrt{2\, n \cdot \tilde{q}}$.

Parameters. We performed parameter selection by means of an exhaustive grid search. The results are summarised in Table 4.

As one can see, the resulting parameters are rather large, and thus we cannot claim that our scheme achieves concrete efficiency. Part of the inefficiencies of the scheme lie in the *starting* norm that results from the [MP12] sampling procedure. As one can see, for reasonable choices of parameter the large β forces $\log q$ to be rather large (even without accounting for the further blow-up that extraction introduces). Further, each round of the protocol involves sending 2^k openings, each of which consists of m short elements of $\mathcal{R}_q$, a cost that practically adds up very quickly.

Acknowledgments

Oleksandra Lapiha was supported by the EPSRC and the UK Government as part of the Centre for Doctoral Training in Cyber Security for the Everyday at Royal Holloway, University of London (EP/S021817/1). Ngoc Khanh Nguyen was supported by the Protocol Labs RFP-013: Cryptonet network grant.

References

[ABD16] Martin R. Albrecht, Shi Bai, and Léo Ducas. "A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes". In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Heidelberg, Aug. 2016, pp. 153–178. DOI: 10.1007/978-3-662-53018-4_6.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO 2021, Part II.* Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 549–579. DOI: 10.1007/978-3-030-84245-1_19.

[ACLMT22] 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: *CRYPTO 2022, Part II.* Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 102–132. DOI: 10.1007/978-3-031-15979-4_4.

[AF22] Thomas Attema and Serge Fehr. "Parallel Repetition of $(k_1, \ldots, k_\mu)$-Special-Sound Multi-round Interactive Proofs". In: *CRYPTO 2022, Part I.* Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13507. LNCS. Springer, Heidelberg, Aug. 2022, pp. 415–443. DOI: 10.1007/978-3-031-15802-5_15.

[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-round Interactive Proofs". In: *TCC 2022, Part I.* Ed. by Eike Kiltz and Vinod Vaikuntanathan. Vol. 13747. LNCS. Springer, Heidelberg, Nov. 2022, pp. 113–142. DOI: 10.1007/978-3-031-22318-1_5.

[AHIV22] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. *Ligero: Lightweight Sublinear Arguments Without a Trusted Setup.* Cryptology ePrint Archive, Report 2022/1608. https://eprint.iacr.org/2022/1608. 2022.

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO 2021, Part II.* Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 519–548. DOI: 10.1007/978-3-030-84245-1_18.

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO 2020, Part II.* Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 470–499. DOI: 10.1007/978-3-030-56880-1_17.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *28th ACM STOC.* ACM Press, May 1996, pp. 99–108. DOI: 10.1145/237814.237838.

[BBBPWM18] 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.* IEEE Computer Society Press, May 2018, pp. 315–334. DOI: 10.1109/SP.2018.00020.

[BBCdGL18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *CRYPTO 2018, Part II.* Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. LNCS. Springer, Heidelberg, Aug. 2018, pp. 669–699. DOI: 10.1007/978-3-319-96881-0_23.

[BBHR18a] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed-Solomon Interactive Oracle Proofs of Proximity". In: *ICALP 2018.* Ed. by Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella. Vol. 107. LIPIcs. Schloss Dagstuhl, July 2018, 14:1–14:17. DOI: 10.4230/LIPIcs.ICALP.2018.14.

[BBHR18b] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. *Scalable, transparent, and post-quantum secure computational integrity.* Cryptology ePrint Archive, Report 2018/046. https://eprint.iacr.org/2018/046. 2018.

[BCCGP16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *EUROCRYPT 2016, Part II.* Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 327–357. DOI: `10.1007/978-3-662-49896-5_12`.

[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. `https://eprint.iacr.org/2022/1365`. 2022.

[BCKLN14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. "Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures". In: *ASIACRYPT 2014, Part I.* Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8873. LNCS. Springer, Heidelberg, Dec. 2014, pp. 551–572. DOI: `10.1007/978-3-662-45611-8_29`.

[BCRSVW19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent Succinct Arguments for R1CS". In: *EUROCRYPT 2019, Part I.* Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11476. LNCS. Springer, Heidelberg, May 2019, pp. 103–128. DOI: `10.1007/978-3-030-17653-2_4`.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs". In: *TCC 2016-B, Part II.* Ed. by Martin Hirt and Adam D. Smith. Vol. 9986. LNCS. Springer, Heidelberg, 2016, pp. 31–60. DOI: `10.1007/978-3-662-53644-5_2`.

[BCS21] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. "Sumcheck Arguments and Their Applications". In: *CRYPTO 2021, Part I.* Ed. by Tal Malkin and Chris Peikert. Vol. 12825. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 742–773. DOI: `10.1007/978-3-030-84242-0_26`.

[BCS23] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. "Lattice-Based Succinct Arguments for NP with Polylogarithmic-Time Verification". In: *Advances in Cryptology - CRYPTO 2023.* Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14082. Lecture Notes in Computer Science. Springer, 2023, pp. 227–251. DOI: `10.1007/978-3-031-38545-2\_8`. URL: `https://doi.org/10.1007/978-3-031-38545-2\_8`.

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving". In: *27th SODA.* Ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: `10.1137/1.9781611974331.ch2`.

[BF22] Benedikt Bünz and Ben Fisch. *Schwartz-Zippel for multilinear polynomials mod N.* Cryptology ePrint Archive, Report 2022/458. `https://eprint.iacr.org/2022/458`. 2022.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. "Checking computations in polylogarithmic time". In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing.* STOC '91. 1991, pp. 21–32.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers". In: *EUROCRYPT 2020, Part I.* Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. LNCS. Springer, Heidelberg, May 2020, pp. 677–706. DOI: `10.1007/978-3-030-45721-1_24`.

[BHRRS21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. "Time- and Space-Efficient Arguments from Groups of Unknown Order". In: *CRYPTO 2021, Part IV.* Ed. by Tal Malkin and Chris Peikert. Vol. 12828. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 123–152. DOI: `10.1007/978-3-030-84259-8_5`.

[BJRW23] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "On the Hardness of Module Learning with Errors with Short Distributions". In: *Journal of Cryptology* 36.1 (Jan. 2023), p. 1. DOI: `10.1007/s00145-022-09441-3`.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *CRYPTO 2020, Part II.* Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 441–469. DOI: 10.1007/978-3-030-56880-1_16.

[BLNS23] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Alessandro Sorniotti. "A Framework for Practical Anonymous Credentials from Lattices". In: *CRYPTO (2).* Vol. 14082. Lecture Notes in Computer Science. Springer, 2023, pp. 384–417.

[BMMTV21] Benedikt Bünz, Mary Maller, Pratyush Mishra, Nirvan Tyagi, and Psi Vesely. "Proofs for Inner Pairing Products and Applications". In: *ASIACRYPT 2021, Part III.* Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13092. LNCS. Springer, Heidelberg, Dec. 2021, pp. 65–97. DOI: 10.1007/978-3-030-92078-4_3.

[BS23] Ward Beullens and Gregor Seiler. "LaBRADOR: Compact Proofs for R1CS from Module-SIS". In: *CRYPTO (5).* Vol. 14085. Lecture Notes in Computer Science. Springer, 2023, pp. 518–548.

[BTT22] Cecilia Boschini, Akira Takahashi, and Mehdi Tibouchi. "MuSig-L: Lattice-Based Multi-signature with Single-Round Online Phase". In: *CRYPTO 2022, Part II.* Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 276–305. DOI: 10.1007/978-3-031-15979-4_10.

[Bab85] László Babai. "Trading Group Theory for Randomness". In: *17th ACM STOC.* ACM Press, May 1985, pp. 421–429. DOI: 10.1145/22145.22192.

[CHMMVW20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. "Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS". In: *EUROCRYPT 2020, Part I.* Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. LNCS. Springer, Heidelberg, May 2020, pp. 738–768. DOI: 10.1007/978-3-030-45721-1_26.

[CI20] Anne Canteaut and Yuval Ishai, eds. *EUROCRYPT 2020, Part I.* Vol. 12105. LNCS. Springer, Heidelberg, May 2020.

[CJL16] Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. *An Algorithm for NTRU Problems and Cryptanalysis of the GGH Multilinear Map without a Low Level Encoding of Zero.* Cryptology ePrint Archive, Report 2016/139. https://eprint.iacr.org/2016/139. 2016.

[CLM23] Valerio Cini, Russell W. F. Lai, and Giulio Malavolta. "Lattice-Based Succinct Arguments from Vanishing Polynomials". In: *Advances in Cryptology – CRYPTO 2023.* Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 72–105.

[CN11] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *ASIACRYPT 2011.* Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Heidelberg, Dec. 2011, pp. 1–20. DOI: 10.1007/978-3-642-25385-0_1.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. "Fractal: Post-quantum and Transparent Recursive Proofs from Holography". In: *EUROCRYPT 2020, Part I.* Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. LNCS. Springer, Heidelberg, May 2020, pp. 769–793. DOI: 10.1007/978-3-030-45721-1_27.

[CP23] Leo de Castro and Chris Peikert. "Functional Commitments for All Functions, with Transparent Setup and from SIS". In: *EUROCRYPT 2023, Part III.* Ed. by Carmit Hazay and Martijn Stam. Vol. 14006. LNCS. Springer, Heidelberg, Apr. 2023, pp. 287–320. DOI: 10.1007/978-3-031-30620-4_10.

[DS22] Yevgeniy Dodis and Thomas Shrimpton, eds. *CRYPTO 2022, Part II.* Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022.

[DW21] Léo Ducas and Wessel P. J. van Woerden. “NTRU Fatigue: How Stretched is Overstretched?” In: *ASIACRYPT 2021, Part IV*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Heidelberg, Dec. 2021, pp. 3–32. DOI: 10.1007/978-3-030-92068-5_1.

[FLV23] Ben Fisch, Zeyu Liu, and Psi Vesely. “Orbweaver: Succinct Linear Functional Commitments from Lattices”. In: *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya. Cham: Springer Nature Switzerland, 2023, pp. 106–131.

[FN23] Giacomo Fenzi and Ngoc Khanh Nguyen. *Lattice-Based Polynomial Commitments: Towards Asymptotic and Concrete Efficiency*. Cryptology ePrint Archive, Paper 2023/xxx. https://eprint.iacr.org/2023/xxx. 2023. URL: https://eprint.iacr.org/2023/xxx.

[GHL22] Craig Gentry, Shai Halevi, and Vadim Lyubashevsky. “Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties”. In: *EUROCRYPT 2022, Part I*. Ed. by Orr Dunkelman and Stefan Dziembowski. Vol. 13275. LNCS. Springer, Heidelberg, 2022, pp. 458–487. DOI: 10.1007/978-3-031-06944-4_16.

[GLSTW21] Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S. Wahby. *Brakedown: Linear-time and post-quantum SNARKs for R1CS*. Cryptology ePrint Archive, Report 2021/1043. https://eprint.iacr.org/2021/1043. 2021.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. “The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)”. In: *17th ACM STOC*. ACM Press, May 1985, pp. 291–304. DOI: 10.1145/22145.22178.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. “Trapdoors for hard lattices and new cryptographic constructions”. In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 197–206. DOI: 10.1145/1374376.1374407.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. “NTRU: A Ring-Based Public Key Cryptosystem”. In: *Third Algorithmic Number Theory Symposium (ANTS)*. Vol. 1423. LNCS. Springer, Heidelberg, June 1998, pp. 267–288.

[HS23] Carmit Hazay and Martijn Stam, eds. *EUROCRYPT 2023, Part III*. Vol. 14006. LNCS. Springer, Heidelberg, Apr. 2023.

[KF17] Paul Kirchner and Pierre-Alain Fouque. “Revisiting Lattice Attacks on Overstretched NTRU Parameters”. In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, 2017, pp. 3–26. DOI: 10.1007/978-3-319-56620-7_1.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. “Constant-Size Commitments to Polynomials and Their Applications”. In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Heidelberg, Dec. 2010, pp. 177–194. DOI: 10.1007/978-3-642-17373-8_11.

[Kil92] Joe Kilian. “A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract)”. In: *24th ACM STOC*. ACM Press, May 1992, pp. 723–732. DOI: 10.1145/129712.129782.

[LD08] Richard E. Ladner and Cynthia Dwork, eds. *40th ACM STOC*. ACM Press, May 2008.

[LLL82] Arjen Lenstra, Hendrik Lenstra Jr., and Laszlo Lovasz. “Factoring polynomials with rational coefficients”. In: *Mathematische Annalen* 261 (1982), pp. 513–534.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. “Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General”. In: *CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3.

[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments". In: *PKC 2021, Part I.* Ed. by Juan Garay. Vol. 12710. LNCS. Springer, Heidelberg, May 2021, pp. 215–241. DOI: 10.1007/978-3-030-75245-3_9.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT 2010.* Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "A Toolkit for Ring-LWE Cryptography". In: *EUROCRYPT 2013.* Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 35–54. DOI: 10.1007/978-3-642-38348-9_3.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptogr.* 75.3 (2015), pp. 565–599.

[LS18] Vadim Lyubashevsky and Gregor Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs". In: *EUROCRYPT 2018, Part I.* Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10820. LNCS. Springer, Heidelberg, 2018, pp. 204–224. DOI: 10.1007/978-3-319-78381-9_8.

[Lee21] Jonathan Lee. "Dory: Efficient, Transparent Arguments for Generalised Inner Products and Polynomial Commitments". In: *TCC 2021, Part II.* Ed. by Kobbi Nissim and Brent Waters. Vol. 13043. LNCS. Springer, Heidelberg, Nov. 2021, pp. 1–34. DOI: 10.1007/978-3-030-90453-1_1.

[Lyu09] Vadim Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures". In: *ASIACRYPT 2009.* Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 598–616. DOI: 10.1007/978-3-642-10366-7_35.

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012.* Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 700–718. DOI: 10.1007/978-3-642-29011-4_41.

[MP21] Tal Malkin and Chris Peikert, eds. *CRYPTO 2021, Part II.* Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021.

[MR07] Daniele Micciancio and Oded Regev. "Worst-Case to Average-Case Reductions Based on Gaussian Measures". In: *SIAM Journal on Computing* 37 (1 2007), pp. 267–302.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: *Post-Quantum Cryptography.* Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 147–191. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_5. URL: https://doi.org/10.1007/978-3-540-88702-7_5.

[MR20] Daniele Micciancio and Thomas Ristenpart, eds. *CRYPTO 2020, Part II.* Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020.

[Mat09] Mitsuru Matsui, ed. *ASIACRYPT 2009.* Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009.

[Mer90] Ralph C. Merkle. "A Certified Digital Signature". In: *CRYPTO'89.* Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 218–238. DOI: 10.1007/0-387-34805-0_21.

[Mic94] Silvio Micali. "CS Proofs (Extended Abstracts)". In: *35th FOCS.* IEEE Computer Society Press, Nov. 1994, pp. 436–453. DOI: 10.1109/SFCS.1994.365746.

[NS22] Ngoc Khanh Nguyen and Gregor Seiler. "Practical Sublinear Proofs for R1CS from Lattices". In: *CRYPTO 2022, Part II.* Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. LNCS. Springer, Heidelberg, Aug. 2022, pp. 133–162. DOI: 10.1007/978-3-031-15979-4_5.

[PPS21] Chris Peikert, Zachary Pepin, and Chad Sharp. "Vector and Functional Commitments from Lattices". In: *TCC 2021, Part III*. Ed. by Kobbi Nissim and Brent Waters. Vol. 13044. LNCS. Springer, Heidelberg, Nov. 2021, pp. 480–511. DOI: `10.1007/978-3-030-90456-2_16`.

[PR06] Chris Peikert and Alon Rosen. "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices". In: *TCC 2006*. Ed. by Shai Halevi and Tal Rabin. Vol. 3876. LNCS. Springer, Heidelberg, Mar. 2006, pp. 145–166. DOI: `10.1007/11681878_8`.

[PS21] Alice Pellet-Mary and Damien Stehlé. "On the Hardness of the NTRU Problem". In: *ASIACRYPT 2021, Part I*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13090. LNCS. Springer, Heidelberg, Dec. 2021, pp. 3–35. DOI: `10.1007/978-3-030-92062-3_1`.

[PW08] Chris Peikert and Brent Waters. "Lossy trapdoor functions and their applications". In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 187–196. DOI: `10.1145/1374376.1374406`.

[Pei07] Chris Peikert. "Limits on the Hardness of Lattice Problems in ell _p Norms". In: *Computational Complexity Conference*. IEEE Computer Society, 2007, pp. 333–346.

[SE94] Claus-Peter Schnorr and M. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Math. Program.* 66 (1994), pp. 181–199.

[SS13] Damien Stehlé and Ron Steinfeld. *Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over Ideal Lattices*. Cryptology ePrint Archive, Report 2013/004. `https://eprint.iacr.org/2013/004`. 2013.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. "Efficient Public Key Encryption Based on Ideal Lattices". In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 617–635. DOI: `10.1007/978-3-642-10366-7_36`.

[Sto] *17th ACM STOC*. ACM Press, May 1985.

[WTsTW17] Riad S. Wahby, Ioanna Tzialla, abhi shelat, Justin Thaler, and Michael Walfish. *Doubly-efficient zkSNARKs without trusted setup*. Cryptology ePrint Archive, Report 2017/1132. `https://eprint.iacr.org/2017/1132`. 2017.

[WW23a] Hoeteck Wee and David J. Wu. "Lattice-Based Functional Commitments: Fast Verification and Cryptanalysis". In: *ASIACRYPT 2023*. Springer-Verlag, 2023.

[WW23b] Hoeteck Wee and David J. Wu. "Succinct Vector, Polynomial, and Functional Commitments from Lattices". In: *EUROCRYPT 2023, Part III*. Ed. by Carmit Hazay and Martijn Stam. Vol. 14006. LNCS. Springer, Heidelberg, Apr. 2023, pp. 385–416. DOI: `10.1007/978-3-031-30620-4_13`.