J Cryptol (2024)37:31
https://doi.org/10.1007/s00145-024-09511-8

Journal of
CRYPTOLOGY

Research Article

Lattice-Based Polynomial Commitments: Towards Asymptotic and Concrete Efficiency*

Giacomo Fenzi · Hossein Moghaddas
EPFL, Vaud, Switzerland
giacomo.fenzi@epfl.ch
hossein.moghaddas@epfl.ch

Ngoc Khanh Nguyen
King's College London, London, UK
ngoc_khanh.nguyen@kcl.ac.uk

Communicated by Benoit Libert and Carla Rafols Salvador.

Received 27 December 2023 / Revised 10 May 2024 / Accepted 5 June 2024
Online publication 10 July 2024

Abstract. Polynomial commitments schemes are a powerful tool that enables one party to commit to a polynomial p of degree d, and prove that the committed function evaluates to a certain value z at a specified point u, i.e. $p(u) = z$, without revealing any additional information about the polynomial. Recently, polynomial commitments have been extensively used as a cryptographic building block to transform polynomial interactive oracle proofs (PIOPs) into efficient succinct arguments. In this paper, we propose a lattice-based polynomial commitment that achieves succinct proof size and verification time in the degree d of the polynomial. Extractability of our scheme holds in the random oracle model under a natural ring version of the BASIS assumption introduced by Wee and Wu (EUROCRYPT 2023). Unlike recent constructions of polynomial commitments by Albrecht et al. (CRYPTO 2022), and by Wee and Wu, we do not require any expensive preprocessing steps, which makes our scheme particularly attractive as an ingredient of a PIOP compiler for succinct arguments. We further instantiate our polynomial commitment, together with the Marlin PIOP (EUROCRYPT 2020), to obtain a publicly-verifiable trusted-setup succinct argument for Rank-1 Constraint System (R1CS). Performance-wise, we achieve 17MB proof size for 2^{20} constraints, which is 15X smaller than currently the only publicly-verifiable lattice-based SNARK proposed by Albrecht et al.

Keywords. Lattices, Polynomial commitments, Succinct arguments, Zero-knowledge, Random oracle model.

*This paper was reviewed by Michael Klooss and an anonymous reviewer.

1. Introduction

Due to the significant progress in building quantum computers by various industry leaders, e.g. IBM and Google, there has been a tremendous amount of interest in post-quantum cryptography. This is highly evidenced by the NIST PQC Competition for standardising quantum-safe key encapsulation mechanisms and signatures, where the vast majority of the selected algorithms are based on algebraic lattices. Indeed, not only do the lattice-based constructions offer relatively small key and signature sizes [28,42,48], but they are also renowned for their very fast implementation [71,78]. Consequently, lattices seem to be a natural candidate to build more complex quantum-safe primitives, such as non-interactive zero-knowledge proofs (NIZKs).

The last several years have seen enormous progress in constructing practically efficient NIZKs for lattice relations [12,44,68] which can produce proofs of size a few dozen kilobytes. This has led to rather compact and practical constructions of privacy-preserving primitives, such as ring signatures [67], blind signatures [1] and anonymous credentials [27,56]. Unfortunately, the aforementioned protocols suffer the following limitations—both the proof size and verification time are linear in the length of the witness. Hence, for proving more complex statements, efficient NIZKs with succinct proof size and verification complexity are desired, i.e. zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs).

Polynomial commitment schemes [58] have been getting more and more spotlight in the SNARKs community. The main reason is that, in combination with Polynomial Interactive Oracle Proofs (PIOPs) [31,37], this cryptographic primitive can be used to obtain succinct arguments with concrete efficiency (see e.g. [23,51,79]). In a polynomial commitment scheme, one can commit to any polynomial $f := \sum_{i=0}^{d} f_i \mathsf{X}^i$ of bounded degree d over a ring R, and then later prove that f evaluated at some public point $u \in R$ is equal to a public image $z \in R$, i.e.

$$f(u) = z \ . \tag{1}$$

In the context of PIOPs, we require both the proof π and the verification time to be succinct (i.e. polylogarithmic in the degree d), even if the evaluation point is chosen adaptively by a verifier. Further, to obtain a SNARK, we need π to be a proof of *knowledge*; thus we call such a polynomial commitment *extractable*.

Recently, various lattice-based polynomial commitments [3,13,35,76,83] were introduced[1], mainly as a direct application of functional commitments [64] over standard cyclotomic rings $R := \mathbb{Z}_q[X]/(X^N + 1)$ where N is a power-of-two. Indeed, (1) can be seen as a degree-one multivariate polynomial

$$\begin{bmatrix} 1 & u & u^2 & \cdots & u^d \end{bmatrix} \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_d \end{bmatrix} = z \ . \tag{2}$$

[1] We excluded generic constructions which simply commit to a polynomial and use a general-purpose SNARK to prove correctness of the evaluation.

Unfortunately, the aforementioned constructions suffer several limitations when applied in the context of PIOPs. Firstly, succinct verification requires a preprocessing step, meaning that the evaluation point u must be known when public parameters are generated and cannot be chosen adaptively. Further, only [3,13] offer extractable polynomial commitments which unfortunately suffer from the following limitations: (i) they rely on a knowledge assumption, which now seems to be at least "morally" broken [82], (ii) message space can only consist of short vectors, and (iii) they only support linear functions with short coefficients. This makes proving relations as in (2) cumbersome for large degrees d. Even though one of the issues was circumvented by a promising recent work from Wee and Wu [83], which allows committing to vectors of arbitrarily large coefficients, their knowledge soundness analysis is left for future work. Therefore, constructing extractable polynomial commitments with succinct verification from lattices still remains an open problem.

1.1. *Our Contributions*

In this work we propose a lattice-based PIOP-friendly polynomial commitment scheme. Concretely, our construction supports committing to arbitrary polynomials $f \in R[\mathsf{X}]$ of bounded degree d over R, and proving evaluations for any point $u \in R$ with no preprocessing necessary. Extractability holds in the random oracle model via the Fiat–Shamir transformation [46] under a variant of the BASIS assumption defined recently by Wee and Wu [83], which we call $\mathsf{PowerBASIS}$.

At the core of our construction lie two split-and-fold interactive protocols for proving polynomial evaluations. The first one, which brings resemblance to lattice Bulletproofs [5,8,26], enjoys proof size and verification complexity polylogarithmic in the degree d. Unfortunately, due to certain restrictions on the challenge space, which are inherited from the aforementioned works, the protocol achieves only $1/\mathsf{poly}(\lambda)$ knowledge soundness error. Even though soundness can be amplified via parallel repetition [9] for the interactive protocol, this is not necessarily the case in the non-interactive setting when applying the Fiat–Shamir transformation, as discussed in [10]. To this end, we propose the second protocol, which achieves negligible soundness error in one-shot at the cost of *quasi*-polylogarithmic $d^{O(1/\log\log d)}$ proof size and verification runtime. Furthermore, the non-interactive version of the scheme can be proven secure in the random oracle using the framework by Attema et al. [10]. Last but not least, we show how to upgrade the evaluation proof to achieve zero-knowledge using the standard Fiat–Shamir-with-aborts paradigm [29,65,66]. As a downside, our constructions suffer (i) from having a trusted setup, with the common reference string (crs) size being quadratic in the degree d, and (ii) the committing time is also $O(d^2)$. We summarise the efficiency of both schemes in Table 1.

As a direct application, we combine our polynomial commitment scheme, which includes batch evaluation proofs, with the Marlin Polynomial IOP [37] to obtain a trusted-setup (zero-knowledge) succinct non-interactive arguments of knowledge for Rank-1 Constraint System (R1CS). Practically, for $\approx 2^{20}$ constraints our construction achieves proofs of size 17MB, which is around 15X smaller than the only concretely instantiated lattice-based proof system with succinct verification by Albrecht et al. [3]. Moreover, we obtain a square-root improvement over [3] in terms of the prover runtime. In comparison

Table 1. Efficiency overview of our polynomial commitment scheme.

scheme	Commit time	Prover time	Verifier time	crs size	Commitment size	Asymptotic proof size	Commitment size	Concrete proof size
Construction 1 (Section 5.2)	$O(d^2)$	$O(d)$	$O(\log d)$	$O(d^2)$	$O(1)$	$O(\log d)$	480 KB	105 MB
Construction 2 (Section 5.3)	$O(d^2)$	$O(d)$	$d^{O(1/\log\log d)}$	$O(d^2)$	$O(1)$	$d^{O(1/\log\log d)}$	209 KB	3 MB

In this setting, we commit to polynomials of degree at most d over the ring $R{:=}\mathbb{Z}_q[X]/(X^N+1)$. We count the runtime (resp. sizes) in the number of ring operations (resp. elements), which take time (resp. size) polylog(d) each. For clarity, we ignore the terms related to the security parameter λ. When computing concrete proof sizes, we set $\lambda = 128$ and $d = 2^{20}$. We also include the Fiat–Shamir loss of $Q = 2^{64}$ random oracle queries.

Table 2. Comparison of lattice-based publicly verifiable proof systems for NP relations of size ℓ with sublinear communication complexity .

Scheme	Assumptions	TP	NI	Time		Size		Concrete proof size
				Prover	Verifier	crs	Proof	
[14]	(M-)SIS, RO	✓	✓	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\sqrt{\ell})$	–
[26]	(M-)SIS, RO	✓	✓	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\ell^{\varepsilon})$	–
Lattice Bulletproofs [5, 8,26]	M-SIS	✓	✗	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\log \ell)$	–
[32]	(M)-SIS, RO	✓	✓	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\log \ell)$	–
[75]	M-SIS, RO	✓	✓	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\sqrt{\ell})$	6 MB
Labrador [20]	M-SIS, RO	✓	✓	$O(\ell)$	$O(\ell)$	$O(1)$	$O(\log \ell)$	49KB
[3]	Knowledge k-M-SIS	✗	✓	$O(\ell^4 \log \ell)$	$O(\log \ell)$	$O(\ell^2)$	$O(\log \ell)$	261 MB
This work	PowerBASIS, RO	✗	✓	$O(\ell^2)$	$\ell^{O(1/\log\log \ell)}$	$O(\ell^2)$	$\ell^{O(1/\log\log \ell)}$	17 MB

We count the runtime (resp. sizes) in the number of ring operations (resp. elements), which take time (resp. size) polylog(ℓ) each, and we ignore the terms related polynomially in the security parameter λ. We exclude the preprocessing step from the verifier runtime. Here $0 < \varepsilon < 1$ is a constant. The "TP" column specifies whether the scheme has transparent setup, and "NI" means whether the protocol can be made non-interactive with negligible soundness error. The concrete proof sizes correspond to proving R1CS with $\ell = 2^{20}$ as reported in the respective works.

with other lattice-based arguments which admit linear verification time, our scheme produces comparable proofs to the recent "square-root" protocol by Nguyen and Seiler [75] for bigger R1CS instances, such as 2^{30} constraints, but still more than two orders of magnitude larger than the current state-of-the-art by Beullens and Seiler [20]. We refer to Table 2 for full comparison and Sect. 6 for more details on sizes.

1.2. *Technical Overview*

We provide a brief overview of our techniques. 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 $\mathcal{R}_q{:=}\mathbb{Z}_q[X]/(X^N+1)$. Let $\mathcal{R}_q^{\times}$ be the set of invertible elements in $\mathcal{R}_q$. For a base $\delta \geq 2$ and $n \geq 1$, 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 it is clear from the context. Further, for a fixed matrix $\mathbf{T} \in \mathcal{R}_q^{n \times k}$ and matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, we denote by $\mathbf{S} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{T})$ sampling $\mathbf{S} \in \mathcal{R}_q^{m \times k}$ from the discrete Gaussian distribution with Gaussian parameter $\sigma > 0$ conditioned on $\mathbf{AS} = \mathbf{T}$ over $\mathcal{R}_q$.

1.2.1. *BASIS Commitment Scheme*

Until lately, lattice-based commitment schemes were split into two disjoint classes: Hashed-Message Commitments [2] and Unbounded-Message Commitments [15]. The former one has the property that the sizes of commitments are almost independent of the sizes of the committed values, and thus the commitments are *compressing*. This comes at the cost of the restricted message space being only vectors of small norm. On the other hand, the main characteristic of the latter class is the unbounded message space, but the commitment size is linear in the size of the message.

Recently, Wee and Wu [83] proposed the first lattice-based commitment scheme which is compressing, and simultaneously supports arbitrarily large messages over $\mathcal{R}_q$. The downside of the construction is a requirement on having a trusted setup, which was not necessary in prior works, as well as the quadratic committing time in the message length. In the following, we describe the main intuition behind the construction by Wee and Wu. To this end, we recall the BASIS assumption[2], which lies at the core of the binding property of the commitment.

BASIS *assumption*. As in the (Module-)SIS problem [61], the adversary's final goal is to find a non-zero vector $\mathbf{s}$ of small norm such that $\mathbf{As} = \mathbf{0}$ for a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$. However, in the BASIS setting the adversary is given more information. Namely, let $(\mathbf{B}, \mathsf{aux}) \leftarrow \mathsf{Samp}(\mathbf{A})$ be an efficient algorithm, which given matrix $\mathbf{A}$ as input, outputs another matrix $\mathbf{B} \in \mathcal{R}_q^{n' \times m'}$ along with some auxiliary information aux. Then, in addition to the challenge matrix $\mathbf{A}$, the adversary is given a tuple $(\mathbf{B}, \mathsf{aux}, \mathbf{T})$, where $\mathbf{T}$ is a trapdoor[3] for $\mathbf{B}$. In particular, $\mathbf{T}$ can be used to efficiently emulate sampling from $\mathbf{B}_\sigma^{-1}(\mathbf{t})$ for any image $\mathbf{t} \in \mathcal{R}_q^{n'}$ under certain conditions on the parameter $\sigma > 0$.

Note that hardness of the BASIS assumption heavily depends on the Samp algorithm. For instance, if $\mathsf{Samp}(\mathbf{A})$ is an identity function and simply outputs $\mathbf{B} := \mathbf{A}$, then using the trapdoor $\mathbf{T}$ we can find a short non-zero solution to $\mathbf{A}$ by sampling $\mathbf{s} \leftarrow \mathbf{B}_\sigma^{-1}(\mathbf{0})$. In this paper, we consider the following three instantiations of the Samp algorithm:

- $\mathsf{StructBASIS}$: The sampling algorithm $\mathsf{Samp}(\mathbf{A})$ first generates a row $\mathbf{a}^\intercal \leftarrow \mathcal{R}_q^\ell$ and sets

$$\mathbf{A}^\star := \begin{bmatrix} \mathbf{a}^\intercal \\ \mathbf{A} \end{bmatrix} \in \mathcal{R}_q^{(n+1) \times \ell} \quad . \tag{3}$$

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

[3] In [83], the trapdoor $\mathbf{T}$ is generated by sampling $\mathbf{T} \leftarrow \mathbf{B}_\sigma^{-1}(\mathbf{G})$. Since the matrix $\mathbf{T} \in \mathcal{R}_q^{m' \times n'\tilde{q}}$ is short and $\mathbf{BT} = \mathbf{G}$, it can be used in Micciancio-Peikert trapdoor sampling [72] to efficiently generate preimages under $\mathbf{B}$.

Next, it samples square matrices $\mathbf{W}_1, \ldots, \mathbf{W}_\ell \in \mathcal{R}_q^{(n+1)\times(n+1)}$ and outputs

$$\mathbf{B}_\ell := \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A}^\star & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & \mathbf{W}_\ell\mathbf{A}^\star & -\mathbf{G}_{n+1} \end{array}\right] \quad \text{and} \quad \mathsf{aux} := (\mathbf{W}_1, \ldots, \mathbf{W}_\ell) \ .$$

■ PowerBASIS: $\mathsf{Samp}(\mathbf{A})$ generates a row $\mathbf{a}^\intercal \leftarrow \mathcal{R}_q^\ell$ and sets $\mathbf{A}^\star$ as in (3). Then, it samples a single square matrix $\mathbf{W} \leftarrow \mathcal{R}_q^{(n+1)\times(n+1)}$ and outputs

$$\mathbf{B}_\ell := \left[\begin{array}{ccc|c} \mathbf{W}^0\mathbf{A}^\star & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & \mathbf{W}^{\ell-1}\mathbf{A}^\star & -\mathbf{G}_{n+1} \end{array}\right] \quad \text{and} \quad \mathsf{aux} := \mathbf{W} \ . \tag{4}$$

■ PRISIS[4]: $\mathsf{Samp}(\mathbf{A})$ samples a row $\mathbf{a}^\intercal \leftarrow \mathcal{R}_q^\ell$ and sets $\mathbf{A}^\star$ as in (3). Then, it samples a uniformly random polynomial $w \leftarrow \mathcal{R}_q$ and outputs

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

Observe that the only difference between these variants is how the square matrices $\mathbf{W}_1, \ldots, \mathbf{W}_\ell$ are generated. For StructBASIS they are picked independently and uniformly at random, while for PowerBASIS (resp. PRISIS) each matrix $\mathbf{W}_i$ is defined as $\mathbf{W}_i := \mathbf{W}^{i-1}$ for $i \in [\ell]$, where $\mathbf{W} \leftarrow \mathcal{R}_q^{(n+1)\times(n+1)}$ (resp. $\mathbf{W} := w \cdot \mathbf{I}_{n+1}$ for $w \leftarrow \mathcal{R}_q$). Not to mention the fact that the functional commitment from [83] can be built on top of all three BASIS instantiations [5].

In this work, we analyse hardness of the three newly introduced assumptions for $\ell = 2$. Concretely, we prove that under a certain parameter selection

$$\mathsf{StructBASIS} \xleftrightarrow{\text{Lemma 3.6}} \mathsf{PowerBASIS} \quad \text{and} \quad \mathsf{PRISIS} \xrightarrow{\text{Lemma 3.7}} \mathsf{MSIS} \ .$$

Unfortunately, the techniques do not translate well for larger values of ℓ, as we argue in Sect. 3.2. Therefore, hardness of the BASIS assumption for $\ell > 2$ is left as an open problem.

[4] The name stands for Power-Ring-BASIS.

[5] A reader familiar with the work of [83] can notice a difference between StructBASIS and the original $\mathsf{BASIS}_{\mathsf{struct}}$ from [83, Assumption 3.3]. Namely, the latter one directly sets the matrix $\mathbf{A}^\star := \mathbf{A}$ without appending an additional row $\mathbf{a}^\intercal$ at the top (as in $\mathsf{BASIS}_{\mathsf{rand}}$ [83, Assumption 3.3]). Note that it is possible to build a commitment scheme based on such a variant, as described in [83, Section 4], but this would increase the commitment, as well the opening sizes, by a factor of $n\tilde{q}$. Hence, for efficiency we consider the modified version of $\mathsf{BASIS}_{\mathsf{struct}}$ as presented here.

Commitment construction. We describe a commitment scheme based on the PowerBASIS assumption. Trivial modifications can be made in order to make the scheme secure under the StructBASIS or PRISIS assumptions.

Consider a message space of arbitrary vectors in $\mathcal{R}_q^{d+1}$ of length $d+1$. The setup algorithm generates a (pseudo-)random matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$, along with a uniformly random invertible matrix $\mathbf{W} \in \mathcal{R}_q^{n\times n}$. Further, it computes a trapdoor $\mathbf{T}$ for the matrix

$$\mathbf{B} := \left[\begin{array}{ccc|c} \mathbf{W}^0\mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right]. \tag{5}$$

Then, the common reference string is $\mathsf{crs} := (\mathbf{A}, \mathbf{W}, \mathbf{T})$.

In order to commit to a vector $\mathbf{f} = (f_0, f_1, \ldots, f_d) \in \mathcal{R}_q^{d+1}$, one uses the trapdoor $\mathbf{T}$ to sample short $\mathbf{s}_0, \ldots, \mathbf{s}_d \in \mathcal{R}_q^m$ and $\hat{\mathbf{t}} \in \mathcal{R}_q^{n\tilde{q}}$ as follows:

$$\begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \leftarrow \mathbf{B}_\sigma^{-1}\left(\begin{bmatrix} -f_0\mathbf{W}^0\mathbf{e}_1 \\ -f_1\mathbf{W}^1\mathbf{e}_1 \\ \vdots \\ -f_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}\right)$$

where $\mathbf{e}_1 := (1, 0, \ldots, 0)^\mathsf{T} \in \mathcal{R}_q^n$. The commitment becomes $\mathbf{t} := \mathbf{G}\hat{\mathbf{t}}$, and the opening consists of $(\mathbf{s}_i)_{i\in[0,d]}$. The opening algorithm, given the common reference string crs, commitment $\mathbf{t} \in \mathcal{R}_q^n$ and openings $(\mathbf{s}_i)_{i\in[0,d]}$ as input, checks whether for all $i = 0, 1, \ldots, d$:

$$\mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \quad \text{and} \quad \|\mathbf{s}_i\| \le \beta$$

for some norm parameter $\beta > 0$.

Security properties. In this paper, we consider the notion of *relaxed binding* [12]. Namely, we say that a relaxed opening for a commitment $\mathbf{t}$ consists of (i) a vector of openings $\mathbf{s} = (\mathbf{s}_0, \ldots, \mathbf{s}_d)$, (ii) a message $\mathbf{f} = (f_0, \ldots, f_d) \in \mathcal{R}_q^{d+1}$, and (iii) a vector of relaxation factors $\mathbf{c} := (c_0, \ldots, c_d) \in \mathcal{R}_q^{d+1}$, which together satisfy:

$$\mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t}, \quad \|c_i \cdot \mathbf{s}_i\| \le \beta, \quad \|c_i\|_1 \le \kappa \quad \text{and} \quad c_i \in \mathcal{R}_q^\times$$

for $i = 0, 1, \ldots, d$ and some $\kappa \ge 1$. In particular, vectors $\mathbf{s}_i$ do not need to be short.

Now, we show that the commitment scheme is binding w.r.t. relaxed openings under the PowerBASIS assumption. Indeed, let $\mathcal{B}$ be the following adversary for the PowerBASIS security game, which is given as input a tuple $(\mathbf{A}, \mathbf{B}, \mathbf{W}, \mathbf{T})$ from the challenger, where $\mathbf{B}$ is defined as in (4) for $\ell = d+1$, and $\mathbf{A}^\star$ is constructed as in (3). First, $\mathcal{B}$ aborts if $\mathbf{W}$ is not invertible[6]. Otherwise, $\mathcal{B}$ passes $\mathsf{crs} := (\mathbf{A}^\star, \mathbf{W}, \mathbf{T})$ to the adversary $\mathcal{A}$ against the relaxed

[6] Unlike in PowerBASIS, the commitment construction requires that matrix $\mathbf{W}$ is invertible. However, by carefully choosing parameters q and N, one can argue that the probability of $\mathbf{W} \leftarrow \mathcal{R}_q^{n\times n}$ not being invertible is negligible (c.f. [29, Appendix C.3] and [45, Appendix C]).

binding game. Suppose $\mathcal{A}$ comes up with two relaxed openings $(\mathbf{s}, \mathbf{f}, \mathbf{c})$ and $(\mathbf{s}', \mathbf{f}', \mathbf{c}')$ for the same commitment $\mathbf{t}$ and $\mathbf{f} \neq \mathbf{f}'$. Thus, for some index i we have $f_i \neq f_i'$. Then, by definition of relaxed openings we have

$$\mathbf{A}^{\star}(\mathbf{s}_i - \mathbf{s}_i') + (f_i - f_i')\mathbf{e}_1 = \mathbf{0} \ .$$

Since $f_i - f_i' \neq 0$, we must have $\bar{\mathbf{s}}_i := \mathbf{s}_i - \mathbf{s}_i' \neq 0$. Hence by definition of $\mathbf{A}^{\star}$, $\bar{\mathbf{s}}_i$ is a non-zero solution for the matrix $\mathbf{A}$, but not necessarily a short one. To conclude the proof, note that $c_i c_i' \bar{\mathbf{s}}_i$ is still a non-zero vector, due to the invertibility property of c_i, c_i', and at the same time:

$$\|c_i c_i' \bar{\mathbf{s}}_i\| \leq \|c_i'(c_i \mathbf{s}_i)\| + \|c_i(c_i' \mathbf{s}_i')\| \leq 2\kappa\beta \ . \tag{6}$$

Thus, $c_i c_i' \bar{\mathbf{s}}_i$ is a valid solution for the PowerBASIS problem.

Finally, the statistical hiding property is directly inherited from the original construction of the BASIS commitment by Wee and Wu [83].

1.2.2. *Framework for Proving Polynomial Evaluations*

We use the construction above to build our polynomial commitment scheme. Namely, given a polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ of degree at most d over $\mathcal{R}_q$, we commit to f by committing to its coefficient vector $\mathbf{f} = (f_0, f_1, \ldots, f_d) \in \mathcal{R}_q^{d+1}$, as described in Sect. 1.2.1, to obtain a commitment $\mathbf{t} \in \mathcal{R}_q^n$ along with a short opening $(\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_d)$, where each $\mathbf{s}_i \in \mathcal{R}_q^m$.

An essential property of polynomial commitments is being able to prove that the committed polynomial was evaluated correctly, i.e. $f(u) = z$ for public u and z in $\mathcal{R}_q$. In the setting of our commitment scheme, we are interested in the following ternary relation[7]:

$$\mathsf{R}_{d,\beta} := \left\{ \big((\mathbf{A}, \mathbf{W}, \mathbf{T}), (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_{0 \leq i \leq d}) \big) \ \middle| \ \begin{array}{c} \forall 0 \leq i \leq d, \mathbf{A}\mathbf{s}_i + f_i \mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \wedge \|\mathbf{s}_i\| \leq \beta \\ \wedge \ f(u) = z \end{array} \right\} . \tag{7}$$

The key ingredient for proving such relations efficiently will be the compressed Σ-protocol in Fig. 1, which we will use recursively.

We take inspiration from a common split-and-fold technique used by prior works, e.g. FRI [17] and DARK [31]. Concretely, take $k \in \mathbb{N}$ and suppose $d + 1 = k^h$ for some $h \in \mathbb{N}$. Let us write the polynomial $f(\mathsf{X}) = \sum_{i=0}^{d} f_i \mathsf{X}^i$ as

$$f(\mathsf{X}) = \sum_{t=1}^{k} f_t(\mathsf{X}^k)\mathsf{X}^{t-1}, \quad \text{where } f_t(\mathsf{X}) := \sum_{i=0}^{\frac{d+1}{k}-1} f_{ki+t-1}\mathsf{X}^i \quad \text{for } t = 1, 2, \ldots, k \ .$$

[7] We use the standard notation that the first entry corresponds to the common reference string, the second one is the statement, and the last one is the witness. Also, $\mathbf{T}$ is not going to be used by the prover, nor by the verifier.

Σ-Protocol for $\mathsf{R}_{d,\beta}$

$$\begin{array}{lcl}
\underline{\text{Prover } \mathcal{P}(\mathsf{crs}, (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_{0\le i\le d}))} & & \underline{\text{Verifier } \mathcal{V}(\mathsf{crs}, (\mathbf{t}, u, z))} \\
f(\mathsf{X}) = \sum_{t=1}^{k} f_t(\mathsf{X}^k)\mathsf{X}^{t-1} & & \\
z_t = f_t(u^k) \text{ for } t = 1, \ldots k & \xrightarrow{z_1, \ldots, z_k} & \\
 & \xleftarrow{\alpha_1, \ldots, \alpha_k} & (\alpha_1, \ldots, \alpha_k) \leftarrow \mathcal{C} \subseteq \mathcal{R}_q^k \\
g(\mathsf{X}) = \sum_{t=1}^{k} \alpha_t f_t(\mathsf{X}) & & \\
\mathbf{z}_i = \sum_{t=1}^{k} \alpha_t \mathbf{s}_{ki+t-1} \text{ for } i = 0, \ldots, d' & \xrightarrow{g, (\mathbf{z}_i)_{i\in[0,d']}} & \\
 & & \text{Check:} \\
 & & \sum_{t=1}^{k} z_t u^{t-1} = z \\
 & & \sum_{t=1}^{k} \alpha_t z_t = g(u^k) \\
 & & \text{For } i = 0, 1, \ldots, d' : \\
 & & \quad \mathbf{A}\mathbf{z}_i + g_i\mathbf{e}_1 = (\mathbf{W}^k)^{-i} \left(\sum_{i=1}^{k} \alpha_i \mathbf{W}^{-(i-1)} \right) \mathbf{t} \\
 & & \quad \|\mathbf{z}_i\| \le \mathrm{w}\,\beta
\end{array}$$

Fig. 1. Compressed Σ-protocol for the relation $\mathsf{R}_{d,\beta}$ from (7). Here, $\mathsf{crs} = (\mathbf{A}, \mathbf{W}, \mathbf{T})$ is the common reference string for our polynomial commitment scheme and $d + 1 = k^h$. We denote $d' := (d + 1)/k - 1$ to be degree of the polynomial g, and $\mathrm{w} := \max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\boldsymbol{\alpha}\|_1$.

Then, we want to prove that $f(u) = \sum_{t=1}^{k} f_t(u^k)u^{t-1} = z$. To this end, we let the prover send these partial evaluations $z_t := f_t(u^k)$ for $t \in [k]$, and the verifier manually checks whether

$$\sum_{t=1}^{k} z_t u^{t-1} = z \ . \tag{8}$$

Further, the verifier returns a challenge $\boldsymbol{\alpha} := (\alpha_1, \ldots, \alpha_k)$ from a challenge space $\mathcal{C} \subseteq \mathcal{R}_q^k$. We denote $\mathrm{w} := \max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\boldsymbol{\alpha}\|_1$. Later we will discuss concrete instantiations for $\mathcal{C}$.

Now, consider the folded polynomial $g(\mathsf{X}) = \sum_{t=1}^{k} \alpha_t f_t(\mathsf{X})$ which is of degree at most $d' := (d + 1)/k - 1 = k^{h-1} - 1$. The crucial observation here is that using the structure of the $\mathsf{PowerBASIS}$ commitment[8] from Sect. 1.2.1 we get for every $i = 0, 1, \ldots, d'$:

$$\begin{aligned}
(\mathbf{W}^k)^{-i} \left(\sum_{t=1}^{k} \alpha_t \mathbf{W}^{-(t-1)} \right) \mathbf{t} &= \sum_{t=1}^{k} \alpha_t \mathbf{W}^{-(ki+t-1)} \mathbf{t} \\
&= \mathbf{A} \left(\sum_{t=1}^{k} \alpha_i \mathbf{s}_{ki+t-1} \right) + \left(\sum_{t=1}^{k} \alpha_i f_{ki+t-1} \right) \mathbf{e}_1 \\
&= \mathbf{A}\mathbf{z}_i + g_i \mathbf{e}_1
\end{aligned}$$

[8] We note that a similar result could be obtained using PRISIS.

where $\mathbf{z}_i{:=}\sum_{t=1}^{k} \alpha_t \mathbf{s}_{ki+t-1}$ satisfies $\|\mathbf{z}_i\| \leq \beta'{:=}\mathrm{w}\beta$. In other words, $\mathbf{t}'{:=}(\sum_{t=1}^{k} \alpha_t \mathbf{W}^{-(t-1)}) \cdot \mathbf{t}$, which can be computed by the verifier in time $O(k)$, is a commitment to the polynomial g with the opening $(\mathbf{z}_j)_{j\in[0,d']}$ w.r.t. the new common reference string $\mathsf{crs}'{:=}(\mathbf{A}, \mathbf{W}^k, \mathbf{T})$. Further, by definition of g:

$$g(u^k) = \sum_{t=1}^{k} \alpha_t f_t(u^k) = \sum_{t=1}^{k} \alpha_t z_t \ .$$

Thus, we can conclude that:

$$\left((\mathbf{A}, \mathbf{W}^k, \mathbf{T}), \left(\sum_{t=1}^{k} \alpha_t \mathbf{W}^{-(t-1)}\mathbf{t}, u^k, \sum_{t=1}^{k} \alpha_t z_t\right), \big(g, (\mathbf{z}_i)_{i\in[0,d']}\big)\right) \in \mathsf{R}_{d',\mathrm{w}\beta} \ . \tag{9}$$

In our Σ-protocol, the prover directly outputs $\big(g, (\mathbf{z}_i)_{j\in[0,d']}\big)$ to the verifier, who checks Eqs. (9) and (8). To achieve succinct proofs and verification, we let the prover recursively run the Σ-protocol on the new instance tuple (9) until the degree of the folded polynomial is zero[9]. Overall, the protocol has $2h + 1$ rounds and the last prover message is a pair of the form $(g, \mathbf{z}) \in \mathcal{R}_q \times \mathcal{R}_q^m$, where $\|\mathbf{z}\| \leq \beta'{:=}\mathrm{w}^h\beta$. Performance-wise (excluding the $\mathsf{poly}(\lambda)$ factors), the prover sends $O(hk)$ elements in $\mathcal{R}_q$, while the verifier makes in total $O(hk)$ operations in $\mathcal{R}_q$.

We now focus on knowledge soundness. As common in the lattice setting, we aim to extract a witness with respect to the relaxed relation:

$$\tilde{\mathsf{R}}_{d,\beta,\kappa}{:=}\Big\{\big((\mathbf{A}, \mathbf{W}, \mathbf{T}), (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_{0\leq i\leq d}, (c_i)_{0\leq i\leq d})\big) \ \Bigg| \ \begin{array}{c} \forall 0 \leq i \leq d, \mathbf{A}\mathbf{s}_i + f_i \mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \\ \wedge \|c_i \cdot \mathbf{s}_i\| \leq \beta \wedge \|c_i\|_1 \leq \kappa \\ \wedge c_i \in \mathcal{R}_q^\times \wedge f(u) = z \end{array} \Bigg\} \ .$$

In other words, the witness is now a *relaxed opening* for the commitment $\mathbf{t}$. Note that the relation is still meaningful as long as the commitment scheme is binding w.r.t. relaxed openings.

The knowledge extraction strategy for $\tilde{\mathsf{R}}_{\beta,\kappa}$ will strongly depend on the instantiation of the challenge space $\mathcal{C}$. In this work, we consider two variants described below.

Construction 1: Monomial protocol. As the name suggests, we will make use of certain invertibility properties of the set of signed monomials in $\mathcal{R}_q$, following the approach from lattice Bulletproofs [5,8,26]. Namely, we set $(k, h) = (2, \log(d + 1))$ and define the challenge space

$$\mathcal{C}{:=}\Big\{(1, X^i) : i \in \mathbb{Z}\Big\} \subseteq \mathcal{R}_q^k \ .$$

[9] For concrete efficiency, it might be more beneficial to apply the protocol recursively until the degree of the folded polynomial is *sufficiently* small, instead of going down to zero.

By construction, $\mathrm{w} = 2$ and $|\mathcal{C}| = 2N$. Now, we show that for the challenge space $\mathcal{C}$ above, the Σ-protocol in Fig. 1 is special sound w.r.t. the relaxed relation $\tilde{\mathsf{R}}$. The methodology can then be extended to show that our recursive protocol is $(2, \ldots, 2)$-special sound. Thus, the general parallel repetition results [9], as well as security of the Fiat–Shamir transformation in the random oracle model [10] would directly apply here.

To this end, suppose we are given two transcripts

$$\mathsf{tr}_j := ((z_1, z_2), (1, \alpha_j), (g_j, (\mathbf{z}_{j,i})_{i\in[0,d']})) \quad \text{for } j = 0, 1$$

with the same first message (z_1, z_2) and two distinct challenges $(1, \alpha_0) \neq (1, \alpha_1)$ in $\mathcal{C}$ such that

$$\begin{cases} \left((\mathbf{A}, \mathbf{W}^2, \mathbf{T}), \left((\mathbf{I}_n + \alpha_j \mathbf{W}^{-1})\mathbf{t}, u^2, z_1 + \alpha_j z_2\right), \left(g_j, (\mathbf{z}_{j,i})_{i\in[0,d']}\right)\right) \in \mathsf{R}_{d',\beta'} \\ z_1 + u z_2 = z \end{cases}$$

where $\beta' := \mathrm{w}\beta = 2\beta$. Observing that $\alpha_0 - \alpha_1 \in \mathcal{R}_q^\times$, we define for $i = 0, 1, \ldots, d' := (d-1)/2$

$$\bar{f}_{2i+1} := \frac{g_{0,i} - g_{1,i}}{\alpha_0 - \alpha_1}, \quad \bar{f}_{2i} := \frac{\alpha_1 g_{0,i} - \alpha_0 g_{1,i}}{\alpha_1 - \alpha_0} \tag{10}$$

and similarly

$$\bar{\mathbf{s}}_{2i+1} := \frac{\mathbf{z}_{0,i} - \mathbf{z}_{1,i}}{\alpha_0 - \alpha_1}, \quad \bar{\mathbf{s}}_{2i} := \frac{\alpha_1 \mathbf{z}_{0,i} - \alpha_0 \mathbf{z}_{1,i}}{\alpha_1 - \alpha_0} \ .$$

Denote $\mathbf{2} := (2, \ldots, 2) \in \mathcal{R}_q^{d+1}$. We claim that

$$\left((\mathbf{A}, \mathbf{W}, \mathbf{T}), (\mathbf{t}, u, z), \left(\bar{f}, (\bar{\mathbf{s}}_i)_{i\in[0,d]}, \mathbf{2}\right)\right) \in \tilde{\mathsf{R}}_{d,2N\beta',2} \ .$$

Let us start with proving correctness of the relaxed opening. By careful inspection:

$$\begin{aligned} \mathbf{A}\bar{\mathbf{s}}_{2i+1} + \bar{f}_{2i+1}\mathbf{e}_1 &= \frac{1}{\alpha_0 - \alpha_1}\left((\mathbf{A}\mathbf{z}_{0,i} + g_{0,i}\mathbf{e}_1) - (\mathbf{A}\mathbf{z}_{1,i} + g_{1,i}\mathbf{e}_1)\right) \\ &= \frac{\mathbf{W}^{-2i}}{\alpha_0 - \alpha_1}\left((\mathbf{I}_n + \alpha_0\mathbf{W}^{-1})\mathbf{t} - (\mathbf{I}_n + \alpha_1\mathbf{W}^{-1})\mathbf{t}\right) \\ &= \mathbf{W}^{-(2i+1)}\mathbf{t} \end{aligned}$$

and similarly $\mathbf{A}\bar{\mathbf{s}}_{2i} + \bar{f}_{2i}\mathbf{e}_1 = \mathbf{W}^{-2i}\mathbf{t}$. As for shortness, we use the result from [19] which says that $\|\frac{2}{\alpha_0-\alpha_1}\|_\infty = 1$ for any distinct $\alpha_0, \alpha_1 \in \{X^i : i \in \mathbb{Z}\}$. Thus, for any $i \in [0, d']$ we have

$$\|2 \cdot \bar{\mathbf{s}}_{2i+1}\| \leq \left\|\frac{2}{\alpha_0 - \alpha_1} \cdot (\mathbf{z}_{0,i} - \mathbf{z}_{1,i})\right\| \leq \left\|\frac{2}{\alpha_0 - \alpha_1}\right\|_1 \cdot \left\|\mathbf{z}_{0,i} - \mathbf{z}_{1,i}\right\| \leq 2N\beta'$$

and similarly

$$\|2 \cdot \bar{\mathbf{s}}_{2i}\| \leq \left\|\frac{2}{\alpha_1 - \alpha_0} \cdot (\alpha_1\mathbf{z}_{0,i} - \alpha_0\mathbf{z}_{1,i})\right\| \leq \left\|\frac{2}{\alpha_1 - \alpha_0}\right\|_1 \cdot \left\|\alpha_1\mathbf{z}_{0,i} - \alpha_0\mathbf{z}_{1,i}\right\| \leq 2N\beta'.$$

Finally, we need to prove that the extracted polynomial $\bar{f}$ satisfies $\bar{f}(u) = z$. From the verification equations we know that $g_0(u^2) = z_1 + \alpha_0 z_2$ and $g_1(u^2) = z_1 + \alpha_1 z_2$. Hence,

$$\begin{aligned}\bar{f}(u) &= \sum_{i=0}^{d'} \bar{f}_{2i} u^{2i} + \sum_{i=0}^{d'} \bar{f}_{2i+1} u^{2i+1} \\ &= \sum_{i=0}^{d'} \frac{\alpha_1 g_{0,i} - \alpha_0 g_{1,i}}{\alpha_1 - \alpha_0} \cdot u^{2i} + \sum_{i=0}^{d'} \frac{g_{0,i} - g_{1,i}}{\alpha_0 - \alpha_1} \cdot u^{2i+1} \\ &= \frac{\alpha_1 g_0(u^2) - \alpha_0 g_1(u^2)}{\alpha_1 - \alpha_0} + \frac{g_0(u^2) - g_1(u^2)}{\alpha_0 - \alpha_1} \cdot u \\ &= z_1 + u z_2 \\ &= z\end{aligned}$$

which concludes the proof of the claim.

An almost identical strategy can be applied to our recursive protocol when given a general $(2, \ldots, 2)$-tree of transcripts [8]. In this case, we can extract a relaxed opening $(\bar{f}, (\bar{\mathbf{s}}_i)_{i\in[0,d]}, \mathbf{2^h})$ to the commitment $\mathbf{t}$ which satisfies

$$\left((\mathbf{A}, \mathbf{W}, \mathbf{T}), (\mathbf{t}, u, z), \left(\bar{f}, (\bar{\mathbf{s}}_i)_{i\in[0,d]}, \mathbf{2^h}\right)\right) \in \tilde{\mathsf{R}}_{d,(2N)^h\beta',2^h}$$

where $\beta' := 2^h\beta$ and $\mathbf{2^h} := (2^h, \ldots, 2^h)$. In terms of performance, the communication complexity and the verifier runtime (in terms of operations in $\mathcal{R}_q$) are $O(\log d)$.

Using the knowledge soundness result from [8], we deduce that the soundness error for our protocol is $h/|\mathcal{C}| = h/(2N)$. Since $N = \mathsf{poly}(\lambda)$, we only manage to obtain an inverse-polynomial soundness error. Even though this can be further reduced via parallel repetition in the interactive case [9], such amplification does not combine with the Fiat–Shamir transformation [10]. Our second construction circumvents this issue by achieving negligible soundness error in one-shot.

Construction 2: Large sampling set protocol. In this scenario, we define the challenge space as

$$\mathcal{C} := \{(\alpha_1, \ldots, \alpha_k) : \forall i \in [k], \|\alpha_i\|_\infty \leq \beta_{\mathcal{C}}\}$$

for some suitable parameter $\beta_{\mathcal{C}} \geq 1$. Hence, by construction $\mathrm{w} \leq k\beta_{\mathcal{C}}N$.

One could naively adapt the strategy from Construction 1 to prove knowledge soundness of the Σ-protocol as follows. To begin with, we aim to extract k accepting transcripts with k pairwise distinct challenges $\boldsymbol{\alpha}_j \in \mathcal{C}$ for $j = 1, \ldots, k$. Further, we compute the extracted polynomial f by inverting the $k \times k$ matrix $\mathbf{C}$, where the j-th row corresponds to the challenge $\boldsymbol{\alpha}_j$ in the j-th transcript. Unfortunately, this approach contains a few critical issues. Firstly, it is unclear whether the matrix $\mathbf{C}$ is invertible. But even if it is, the resulting polynomial f may contain large coefficients, or in the context of relaxed openings, there might be no sufficiently short element $v \in \mathcal{R}_q$ such that $v \cdot f_i$ is short for all coefficients f_i.

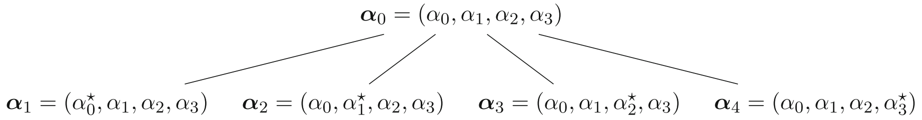
$\boldsymbol{\alpha}_0 = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)$
$\boldsymbol{\alpha}_1 = (\alpha_0^\star, \alpha_1, \alpha_2, \alpha_3)$
$\boldsymbol{\alpha}_2 = (\alpha_0, \alpha_1^\star, \alpha_2, \alpha_3)$
$\boldsymbol{\alpha}_3 = (\alpha_0, \alpha_1, \alpha_2^\star, \alpha_3)$
$\boldsymbol{\alpha}_4 = (\alpha_0, \alpha_1, \alpha_2, \alpha_3^\star)$

Fig. 2. Visualisation of the notion of coordinate-wise special soundness (CWSS) for $k = 4$ coordinates. Here, $\alpha_i^\star \neq \alpha_i$ for all $i \in [4]$.

We propose an alternative approach which relies on a notion, called *coordinate-wise special soundness*[10] (CWSS). As in special soundness, it says that given $k + 1$ valid transcripts $\mathsf{tr}_j = (\mathsf{a}, \boldsymbol{\alpha}_j, \mathsf{z}_j)$ for $j = 0, 1, \ldots, d$, such that $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_k \in \mathcal{C}$ satisfy a certain relation, then one can extract the witness. The relation is defined as follows: for every $j \in [k]$, vectors $\boldsymbol{\alpha}_0 = (\alpha_{0,1}, \ldots, \alpha_{0,k})$ and $\boldsymbol{\alpha}_j = (\alpha_{j,1}, \ldots, \alpha_{j,k})$ differ *exactly* in the j-th coordinate, i.e. $\forall i \in [k]\backslash\{j\}, \alpha_{j,i} = \alpha_{0,i}$ and $\alpha_{j,j} \neq \alpha_{0,j}$ (see Fig. 2 for visualisation). We prove that for multi-round protocols, CWSS implies knowledge soundness both in the interactive and non-interactive setting where the Fiat–Shamir transformation is applied.

In the following, we show that our Σ-protocol satisfies CWSS. Suppose we are given $k + 1$ valid transcripts

$$\mathsf{tr}_j := \big((z_1, \ldots, z_k), \boldsymbol{\alpha}_j = (\alpha_{j,1}, \ldots, \alpha_{j,k}), (g_j, (\mathbf{z}_{j,i})_{i \in [0,d']})\big) \quad \text{for } j = 0, 1, \ldots, k \ .$$

Let us fix $j \in [k]$ and consider the transcripts tr_0 and tr_j. From the verification equations we have for $i = 0, \ldots, d'$:

$$\mathbf{A}\mathbf{z}_{0,i} + g_{0,i}\mathbf{e}_1 = \mathbf{W}^{-ki}\left(\sum_{t=1}^{k} \alpha_{0,t}\mathbf{W}^{-(t-1)}\right)\mathbf{t}$$

$$\mathbf{A}\mathbf{z}_{j,i} + g_{j,i}\mathbf{e}_1 = \mathbf{W}^{-ki}\left(\sum_{t=1}^{k} \alpha_{j,t}\mathbf{W}^{-(t-1)}\right)\mathbf{t}.$$

Since $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_j$ are the same in all coordinates apart from the j-th one, by subtracting the two equations we obtain

$$\mathbf{A}(\mathbf{z}_{0,i} - \mathbf{z}_{j,i}) + (g_{0,i} - g_{j,i})\mathbf{e}_1 = (\alpha_{0,j} - \alpha_{j,j})\mathbf{W}^{-(ki+j-1)}\mathbf{t} \ .$$

Now, by choosing parameters $q, N, \beta_{\mathcal{C}}$ appropriately, and using the result by Lyubashevsky and Seiler that short elements in $\mathcal{R}_q$ are invertible [70], we deduce that $\alpha_{0,j} - \alpha_{j,j} \in \mathcal{R}_q^\times$ and thus can define the extracted openings

$$\bar{\mathbf{s}}_{ki+j-1} := \frac{\mathbf{z}_{0,i} - \mathbf{z}_{j,i}}{\alpha_{0,j} - \alpha_{j,j}} \quad \text{and} \quad \bar{f}_{ki+j-1} := \frac{g_{0,i} - g_{j,i}}{\alpha_{0,j} - \alpha_{j,j}}$$

[10] As far as we are aware, this strategy was first introduced by Baum et al. [14] in the context of amortised lattice-based zero-knowledge proofs.

and the partial vector of relaxation factors $\mathbf{c}_j:=(\alpha_{0,j}-\alpha_{j,j},\ldots,\alpha_{0,j}-\alpha_{j,j})\in\mathcal{R}_q^{d'+1}$. Then, by construction we have $\mathbf{A}\bar{\mathbf{s}}_{ki+j-1}+\bar{f}_{ki+j-1}\mathbf{e}_1=\mathbf{W}^{-(ki+j-1)}\mathbf{t}$, and further

$$\|(\alpha_{0,j}-\alpha_{j,j})\cdot\bar{\mathbf{s}}_{ki+j-1}\|\leq 2\mathrm{w}\beta \quad\text{and}\quad \|\alpha_{0,j}-\alpha_{j,j}\|\leq 2\beta_{\mathcal{C}}N\ .$$

From the other verification checks we similarly conclude that $\sum_{i=0}^{d'}\bar{f}_{ki+j-1}u^{ki}=z_j$.

Eventually, by running the argument above for $j=1,2,\ldots,k$, we reconstruct a polynomial $f\in\mathcal{R}_q^{\leq d}[\mathsf{X}]$, along with $(\mathbf{s}_i)_{i\in[0,d]}$, and the vector $\mathbf{c}:=(\mathbf{c}_1,\ldots,\mathbf{c}_k)$ of relaxation factors so that

$$\left((\mathbf{A},\mathbf{W},\mathbf{T}),(\mathbf{t},u,z)\,,\left(\bar{f},(\bar{\mathbf{s}}_i)_{i\in[0,d]},\mathbf{c}\right)\right)\in\tilde{\mathsf{R}}_{d,2\mathrm{w}\beta,2\beta_{\mathcal{C}}N}\ .$$

In terms of security, we show that the knowledge soundness error of our Σ-protocol is bounded by $k/(2\beta_{\mathcal{C}}+1)^N$, where $(2\beta_{\mathcal{C}}+1)^N$ is the number of all possible choices for a single coordinate in $\mathcal{C}$. Consequently, by picking $k,\beta_{\mathcal{C}}\geq 1$ and $N=\mathsf{poly}(\lambda)$ appropriately, we achieve negligible soundness error in one-shot.

This strategy can be further applied in our recursive protocol. That is, analogously as for special soundness, we first generalise the notion of coordinate-wise special soundness in the multi-round setting, and then prove that our protocol satisfies CWSS as above. By following the methodology from [8,10], we obtain the knowledge soundness error equal to $hk/(2\beta_{\mathcal{C}}+1)^N$, while the knowledge extractor runs the prover expected $(k+1)^h$ times, and outputs a relaxed opening $(\bar{f},(\bar{\mathbf{s}}_i)_{i\in[0,d]},\mathbf{c})$ such that

$$\left((\mathbf{A},\mathbf{W},\mathbf{T}),(\mathbf{t},u,z)\,,\left(\bar{f},(\bar{\mathbf{s}}_i)_{i\in[0,d]},\mathbf{c}\right)\right)\in\tilde{\mathsf{R}}_{d,\gamma,\xi}$$

where $\gamma:=(2^h(2\beta_{\mathcal{C}}N)^{2^h-h-1}\mathrm{w}^h)\cdot\beta$ and $\xi:=2\beta_{\mathcal{C}}(2\beta_{\mathcal{C}}N)^{2^h-2}N$. We highlight that the norm blow-up is much larger here than in the monomial case due to certain technical differences[11]. As a result, we cannot pick $k=2$ and $h=O(\log d)$ since then one would require $\log q=O(d)$ for relaxed binding to hold (c.f. Eq. (6)); thus making the proof size and verifier time polynomial in d. Instead, we instantiate the protocol by choosing $k=O\left(d^{\frac{1}{\log\log d}}\right)$ and $h=O(\log\log d)$. In this case, $\log q=\mathsf{polylog}(d)$, and the proof size and verifier complexity, in terms of operations over $\mathcal{R}_q$, become $O(d^{\frac{1}{\log\log d}}\log\log d)=d^{O(1/\log\log d)}$.

1.2.3. *Polynomial Commitments over Finite Fields*

Until now, we were focusing on polynomial commitments over the ring $\mathcal{R}_q:=\mathbb{Z}_q[X]/(X^N+1)$. Here, we sketch how to obtain a polynomial commitment over a *finite field*, which is required by Polynomial IOPs [31,37] to compile into succinct arguments. The key ingredient, which allows us to do that is the ability to commit to *arbitrarily* large elements in $\mathcal{R}_q$.

[11] Roughly speaking, in Construction 1 we managed to keep the norm growth smaller due to the fact that the relaxation factors $\mathbf{2^h}$ are independent of the extracted transcripts, which is not the case for the relaxation factors $\mathbf{c}$ in Construction 2. We refer to Sect. 5.3 for more details.

Let $l \geq 1$ be a divisor of N. It is a well-known fact [70] that if $q \equiv 2N/l + 1 \pmod{4N/l + 1}$, then there exists a ring isomorphism φ from $\mathbb{F}^{N/l}$ to $\mathcal{R}_q$, where $\mathbb{F}$ is a finite field of size q^l. Thus, we define a map $\varphi_{\mathbb{F}} : \mathbb{F} \rightarrow \mathcal{R}_q$ as $x \mapsto \varphi(x, 0, \dots, 0)$, and denote the image of $\varphi_{\mathbb{F}}$ as $\mathcal{S}_q$. We will make use of the fact that $\mathcal{S}_q$ is an ideal of $\mathcal{R}_q$.

Suppose we want to commit to a polynomial $F \in \mathbb{F}^{\leq d}[\mathsf{X}]$ and prove that $F(x) = y$ for $x, y \in \mathbb{F}$. Using the homomorphic property of $\varphi_{\mathbb{F}}$, it is easy to see that this is equivalent to proving $f(u) = z$ over $\mathcal{R}_q$, where $f[\mathsf{X}] := \sum_{i=0}^{d} \varphi_{\mathbb{F}}(F_i)\mathsf{X}^i \in \mathcal{S}_q[\mathsf{X}]$, $u = \varphi_{\mathbb{F}}(x) \in \mathcal{S}_q$ and $z = \varphi_{\mathbb{F}}(y) \in \mathcal{S}_q$. Therefore, we commit to the polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ and prove evaluation of u at the point z as before.

What we need to take care of is proving that all coefficients of f indeed lie in $\mathcal{S}_q$. This allows us to extract the polynomial $\bar{F} \in \mathbb{F}[\mathsf{X}]$ by taking the inverse of $\varphi_{\mathbb{F}}$ coefficient-wise. Looking at our underlying Σ protocol in Fig. 1, the additional proof comes without any change on the prover's side, while the verifier also checks whether $g \in \mathcal{S}_q[\mathsf{X}]$, which is the case since $\mathcal{S}_q$ is an ideal. To see why this modification is sufficient, consider the extraction strategy in Eq. (10). Since now $g_{0,i}, g_{1,i} \in \mathcal{S}_q$, we again use the fact that $\mathcal{S}_q$ is an ideal and conclude that $\bar{f}_{2i+1} = (g_{0,i} - g_{1,i})/(\alpha_0 - \alpha_1)$ also lies in $\mathcal{S}_q$. Identical reasoning follows for both Construction 1 and 2.

1.3. *Related Works*

The first lattice-based interactive proof with sublinear communication complexity for arithmetic ℓ-gate circuit satisfiability was formally proposed by Baum et al. [14], where the authors achieve $O(\sqrt{\ell})$ size proofs.

The construction was later generalised by Bootle et al. [26] who define so-called "levelled commitments" and give $O(\ell^{1/k})$ size proofs for proving knowledge of a commitment opening with $k = O(1)$ levels. The main drawback of the scheme is that the modulus for the proof system increases exponentially in k and thus considering more than 2-3 levels seems impractical. Recently, Nguyen and Seiler [75] combined the square-root approach from [14] with the CRT-packing technique from [44] to obtain a practically efficient square-root NIZK, with 6MB proofs for circuits of size $\ell = 2^{20}$.

Bootle et al. [26] also proposed the first lattice adaptation of the Bulletproofs protocol [22,30] over polynomial rings $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^N + 1)$ which offers polylog(ℓ) proof sizes. This approach was later improved independently by Attema et al. [8] and Albrecht and Lai [5] in terms of tighter soundness analysis, and also generalised to a more abstract setting by Bootle et al. [24]. While the *split-and-fold* strategy from Bulletproofs is very attractive in the discrete logarithm setting and keeps asymptotic efficiency in the lattice scenario, it does not mix well with the shortness condition required in lattice-based cryptography. Consequently, this leads to a concrete blow-up of the parameters as well as the proof size. Roughly speaking, for the knowledge soundness argument it must be possible to invert the folding in the extraction such that the extracted solution vector is still short. To this end, one needs a challenge space of the underlying compressed Σ-protocol to have a property that (a scaled) inverse of a difference of any two distinct challenges is still short - such sets are called *subtractive*. Hence, Bootle et al. [26] picked the challenge space to consist of monomial challenges $\mathcal{C} := \{X^i : i \in \mathbb{Z}\} \subseteq \mathcal{R}_q$, which is indeed subtractive as shown in [19]. Since the Σ-protocol is 3-special sound, norm of the extracted solution vector grows by a factor of $O(N^3)$ for *every* level of folding. Then, the parameters must be chosen such that Module-SIS is hard with respect to the norm of the extracted solution vector, resulting in the need for a huge modulus q. Note that a similar issue occurs in our Construction 1 (c.f. Sect. 5.2). However, since our underlying compressed Σ-protocol is only 2-special sound, norm of the extracted vector grows by only a factor of $O(N)$ for each folding level (but at the price of having a trusted setup).

In addition to the norm growth of the extracted witness, the restriction on the challenges has a negative impact on the soundness error. Indeed, since the challenge space $\mathcal{C}$ in [26] has size $2N$, the soundness error becomes only $1/\mathsf{poly}(\lambda)$. Furthermore, it was proven by Albrecht and Lai [5] that *all* subtractive set over $\mathcal{R}_q$ have size $O(N)$. This becomes problematic especially in the non-interactive setting due to the result by Attema et al. [10], who showed that the Fiat–Shamir transformation of a parallel repetition of special sound protocols *does not* necessarily decrease the soundness error. A promising solution to circumvent this limitation was recently proposed by Bünz and Fisch [32], who suggested a new knowledge extraction strategy, i.e. the notion of *almost special soundness*, which does not require subtractive sets. Instead, the challenges are picked from the exponential-sized set of integers $[0, 2^{\lambda-1})$. Unfortunately, the former issue with the norm growth for each folding level is still present in [32].

Recently, Beullens and Seiler [20] showed that by combining a split-and-fold approach with algebraic techniques introduced in linear-sized lattice-based NIZKs [68], it is possible to achieve negligible soundness error whilst controlling the norm growth. This is evidenced with impressive 50KB proofs for circuits of size $\ell = 2^{20}$.

Major downside of all the aforementioned works is a linear verification time, which can be the main efficiency bottleneck when proving satisfiability of large circuits. Until now, the only lattice-based publicly verifiable succinct argument of knowledge with efficient verification (excluding the preprocessing step) was proposed by Albrecht et al. [3]. The construction is obtained as a direct application of functional commitments [64] and soundness holds under a knowledge assumption. However, similar to our scheme, a trusted setup is required, and more importantly, the prover algorithm runs in time $O(\ell^4 \log \ell)$ which makes it unappealing to implement in practice. Even more worryingly, the underlying assumption has been recently (at least morally) broken [82].

Prior to [3], all lattice-based zk-SNARKs were in the designated-verifier setting [49, 55,81] . The constructions use the Linear-PCP compiler [21] to transform into succinct arguments. Notably, the most recent work by Steinfeld et al. [81] achieves proofs of size 6KB for $\ell = 2^{20}$ constraints at the cost of very large crs (in the order of tens of gigabytes).

Naturally, there is a line of research focusing on the security of lattice-based zero-knowledge proofs against *quantum adversaries* [41,59,60]. Particularly, Lai et al. [60] show that any multi-round protocol, which satisfies special soundness and *collapsing*, is knowledge sound in the post-quantum setting. As a special case, they demonstrate that the lattice Bulletproofs protocol [26] is knowledge sound against quantum provers. Since our constructions not only satisfy (coordinate-wise) special soundness but also follow the split-and-fold strategy from [26], we believe that the general result from [60] can be adapted to our setting.

Interestingly, lattice assumptions are not only used to build lattice-based commitments, but also to construct non-interactive arguments in the standard model, i.e. without the random oracle. For instance, there is a line of works [33,53,54] which focuses on instantiating the Fiat–Shamir transformation with a correlation intractable hash function [34], that itself can be built from the Learning with Errors (LWE) problem [53]. Following this template, Choudhuri, Jain and Jin [38] built a SNARG for languages in P only based on the LWE problem with polynomial modulus. Moreover, the LWE assumption can be used to construct non-interactive succinct (and batched) arguments without the Fiat–Shamir transformation, but via somewhere extractable hash functions [40,57]. We believe that naturally, due to relying on more assumptions, constructions based on the random oracle model should perform much better in terms of concrete efficiency.

1.4. *Concurrent and Subsequent Works*

Recently, Bootle et al. [25] and Cini et al. [39] independently proposed variants of the lattice Bulletproofs protocol that achieve polylogarithmic verification time. The former work proposes a new delegation algorithm inspired from [62], which requires an additional pre-processing step. The latter one introduces more (power-like) structure on the Ajtai commitment [2] which allows for fast verification, at the cost of relying on a new assumption called Vanishing-SIS (vSIS). We note that there is a close similarity between vSIS and the PRISIS, and we leave the concrete relationship between the two for the future work. Nevertheless, the aforementioned work inherit the issue from the original construction [26] that the soundness error is non-negligible and parallel repetitions are required.

Fisch et al. [47] recently presented a polynomial commitment scheme, as an application of their linear functional commitment. Following the work of [3], the construction relies on the knowledge k-M-ISIS assumption, which appears to be morally invalidated in [82].

As a subsequent work, Albrecht et al. [4] proposed a new polynomial commitment scheme with polylogarithmic communication and verification complexity *under standard assumptions*. To this end, the authors construct a new commitment scheme that combines our $\mathsf{PowerBASIS}$ construction together with the Merkle tree paradigm. Consequently, the committing runtime becomes quasilinear in the length of the message, while the size crs shrinks to only polylogarithmic. The binding property of the commitment relies on a "multi-instance" version of the PRISIS assumption. Finally, using the exact strategy from Lemma 3.7, security of the aforementioned assumption is further reduced to Module-SIS.

2. Preliminaries

Notation. 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 $[a, b]{:=}\{a, a+1, \ldots, b\}$, $[a]{:=}[1, a]$. For $q \in \mathbb{N}$ write $\mathbb{Z}_q$ for the integers modulo q. We denote vectors with lowercase boldface (i.e. $\mathbf{u}$, $\mathbf{v}$) and matrices with uppercase boldface (i.e. $\mathbf{A}$, $\mathbf{B}$). For a vector $\mathbf{x}$ we write x_i or $\mathbf{x}[i]$ for its i-th entry.

Norms. We define the ℓ_p norm on $\mathbb{C}^n$ as $\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$ for $p < \infty$ and $\|\mathbf{x}\|_\infty := \max_i |x_i|$. 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.

Linear algebra. We let $\mathbf{e}_i$ be the vector with 1 in its i-th entry, 0 everywhere else. For $\mathbf{B} \in \mathbb{R}^{n\times m}$ we let $s_1(\mathbf{B}) = \sup\{\|\mathbf{Bv}\| : \mathbf{v} \in \mathbb{R}^m \wedge \|\mathbf{v}\| = 1\}$ be the **spectral norm** of $\mathbf{B}$. We also denote by $\tilde{\mathbf{B}}$ the Gram-Schmidt orthonormalization of $\mathbf{B}$. The Gram-Schmidt norm of $\mathbf{B}$ is defined as

$$\|\tilde{\mathbf{B}}\| := \max_{i\in[m]} \|\tilde{\mathbf{b}}_i\|$$

where $\tilde{\mathbf{b}}_i$ is the i-th column of $\tilde{\mathbf{B}}$.

For a ring R, we define $\mathsf{GL}(n, R)$ to be the group of $n \times n$ invertible matrices over R.

2.1. *Lattices*

A subset $\Lambda \subseteq \mathbb{R}^m$ is a lattice if the following conditions hold:

- $\mathbf{0} \in \Lambda$, and for $\mathbf{x}, \mathbf{y} \in \Lambda$, $\mathbf{x} + \mathbf{y} \in \Lambda$.
- For every $\mathbf{x} \in \Lambda$, there exists $\epsilon > 0$ such that $\{\mathbf{y} \in \mathbb{R}^m : \|\mathbf{x} - \mathbf{y}\| < \epsilon\} \cap \Lambda = \{\mathbf{x}\}$.

We say $\mathbf{B} \in \mathbb{R}^{m\times k}$ is a basis for Λ if its columns are linearly independent and $\Lambda = \mathcal{L}(\mathbf{B}){:=}\{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^k\}$. If $k = m$ then we say that Λ is full-rank. 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 $\Lambda^* = \{\mathbf{w} \in \mathrm{Span}(\Lambda) : \langle \Lambda, \mathbf{w}\rangle \subseteq \mathbb{Z}\}$. If $\Lambda \subseteq \mathbb{Z}^m$, we call it an integral lattice. For I an ideal of $\mathbb{R}^m$, we let $I \cdot \Lambda = \{i \cdot \mathbf{x} : i \in I, \mathbf{x} \in \Lambda\}$, which is also a lattice. For a lattice Λ we denote

$$\lambda_1(\Lambda) := \min_{0\neq\mathbf{x}\in\Lambda} \|\mathbf{x}\| \quad \text{and} \quad \lambda_1^\infty(\Lambda) := \min_{0\neq\mathbf{x}\in\Lambda} \|\mathbf{x}\|_\infty \ .$$

For $\mathbf{t} \in \mathrm{Span}(\Lambda)$, we also define the shifted lattice $\mathbf{t} + \Lambda{:=}\{\mathbf{t} + \mathbf{x} : \mathbf{x} \in \Lambda\}$. We also consider q-ary lattices, namely those with $q\mathbb{Z} \subseteq \Lambda$. For an arbitrary $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ we define the full rank q-ary lattice

$$\begin{aligned}\Lambda^\perp(\mathbf{A}) &= \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = 0 \pmod q\}\\ \Lambda(\mathbf{A}) &= \{\mathbf{z} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{Az} = \mathbf{s} \pmod q\}\end{aligned}$$

For any $\mathbf{u} \in \mathbb{Z}_q^n$ such that there exists $\mathbf{x}$ with $\mathbf{Ax} = \mathbf{u}$, we define $\Lambda_{\mathbf{u}}^\perp(\mathbf{A}){:=}\{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{u} \pmod q\} = \Lambda^\perp(\mathbf{A}) + \mathbf{x}$.

2.2. *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 $\mathcal{K}$. 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 $\mathcal{R}_q$.

We recall the following well-known inequality, which allows to bound norms on products in the ring $\mathcal{R}$.

Lemma 2.1. *Let $u, v \in R$. Then $\|uv\| \leq \|u\|_1 \cdot \|v\|$.*

Proof. Let $u := u_0 + u_1 X + \cdots + u_{N-1} X^{N-1} \in \mathcal{R}$. Then, by the triangle inequality we get

$$\|uv\| \leq \sum_{i=0}^{N-1} \|u_i v \cdot X^i\| = \sum_{i=0}^{N-1} \|u_i v\| = \sum_{i=0}^{N-1} |u_i| \cdot \|v\| = \|u\|_1 \cdot \|v\| \ .$$

□

Coefficient embedding . For $x \in \mathcal{K}$, we can consider the additive group isomorphism

$$\begin{aligned} \mathsf{vec} : \mathcal{K} &\to \mathbb{Q}^N \\ a_0 + a_1 X + \cdots + a_{N-1} X^{N-1} &\mapsto (a_0, \ldots, a_{N-1})^\top \end{aligned}$$

and we refer this as the coefficient embedding of $\mathcal{K}$. Note that, for $f, g \in \mathcal{K}$, $\langle f, g \rangle = \langle \mathsf{vec}(f), \mathsf{vec}(g) \rangle$ and thus $\|\mathsf{vec}(f)\| = \|f\|$. Furthermore, vec restricts to an isomorphism between $\mathcal{R}_q \cong \mathbb{Z}_q^N$ and $\mathcal{R} \cong \mathbb{Z}^N$. We also extend this to a mapping $\mathcal{K}^m \to \mathbb{Q}^{mN}$ by applying it component-wise. For $f \in \mathcal{K}$, we let

$$\mathsf{rot}(f) := (\mathsf{vec}(f), \mathsf{vec}(X \cdot f), \ldots, \mathsf{vec}(X^{N-1} \cdot f)) \in \mathbb{Q}^{N \times N} \ ,$$

noting that $\mathsf{rot}(f)\mathsf{vec}(g) := \mathsf{vec}(fg)$ and $\mathsf{rot}(f)\mathsf{rot}(g) = \mathsf{rot}(fg)$. We extend this to matrices $\mathbf{B} \in \mathcal{K}^{m \times n}$ by writing

$$\mathsf{rot}(\mathbf{B}) := \begin{bmatrix} \mathsf{rot}(b_{1,1}) & \ldots & \mathsf{rot}(b_{1,n}) \\ \vdots & \ddots & \vdots \\ \mathsf{rot}(b_{m,1}) & \ldots & \mathsf{rot}(b_{m,n}) \end{bmatrix} \in \mathbb{Q}^{mN \times nN} \ .$$

Module lattices. For $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{x} \in \mathcal{R}_q^m$, $\mathbf{u} = \mathbf{A}\mathbf{x}$, define

$$\begin{aligned} \Lambda^\perp(\mathbf{A}) &:= \{\mathbf{z} \in \mathcal{R}^m : \mathbf{A}\mathbf{z} = \mathbf{0} \bmod q\} \\ \Lambda_\mathbf{u}^\perp(\mathbf{A}) &:= \{\mathbf{z} \in \mathcal{R}^m : \mathbf{A}\mathbf{z} = \mathbf{u} \bmod q\} = \Lambda^\perp(\mathbf{A}) + \mathbf{x} \ . \end{aligned}$$

Then, $\Lambda^\perp(\mathbf{A}) = \mathsf{vec}^{-1}(\Lambda^\perp(\mathsf{rot}(\mathbf{A})))$ and $\Lambda_\mathbf{u}^\perp(\mathbf{A}) = \mathsf{vec}^{-1}(\Lambda_{\mathsf{vec}(\mathbf{u})}^\perp(\mathsf{rot}(\mathbf{A})))$.

Spectral norm. Let $s_1(\mathbf{R}) := \sup\{\|\mathbf{R}\mathbf{v}\| : \mathbf{v} \in \mathcal{K}^w \wedge \|\mathbf{v}\| = 1\}$ be the spectral norm of $\mathbf{R} \in \mathcal{R}^{m\times w}$. Clearly, $s_1(\mathsf{rot}(\mathbf{R})) = s_1(\mathbf{R})$, where the spectral norm of the left-hand side is over $\mathbb{R}$. Here, we recall a simple bound.

Lemma 2.2. *Let* $\mathbf{R} \in \mathcal{R}_q^{m\times t}$. *Then* $s_1(\mathbf{R}) \leq \sqrt{N} \cdot \|\mathbf{R}\|$.

Proof. Let $\mathbf{r}_1, \ldots, \mathbf{r}_m$ be the rows of $\mathbf{R}$. Note that by the Cauchy–Schwarz inequality, for any $\mathbf{u}$ with $\|\mathbf{u}\| = 1$ we have that

$$\|\langle \mathbf{r}_i, \mathbf{u}\rangle\|^2 \leq \left(\sum_{j\in[t]} \|r_{i,j}s_j\|\right)^2 \leq N\left(\sum_{j\in[t]} \|r_{i,j}\| \cdot \|s_j\|\right)^2 \leq N\|\mathbf{r}_i\|^2 \cdot \|\mathbf{u}\|^2 \leq N\|\mathbf{r}_i\|^2 \ .$$

Thus, $\|\mathbf{R}\mathbf{u}\|^2 \leq N\|\mathbf{R}\|^2$ which concludes the proof. □

In this work we will work with $q \equiv 5 \pmod 8$. In this setting, the probability that a uniformly random matrix is full-rank is overwhelming.

Lemma 2.3. (Appendix C.3 of [29]). *Let* $q \equiv 5 \pmod 8$ *be prime,* $N = O(\lambda)$ *and* $m \geq n \geq 1$. *Then, for a uniformly random matrix* $\mathbf{A} \leftarrow \mathcal{R}_q^{n\times m}$, *the probability that* $\mathbf{A}$ *is not full-rank is* $\mathsf{negl}(\lambda)$.

2.3. *Discrete Gaussian Distributions*

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

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

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

Smoothing parameter. The smoothing parameter $\eta_\epsilon(\Lambda)$ of a lattice is the smallest $s > 0$ such that $\rho_{1/s}(\Lambda^*) \leq 1 + \epsilon$. Below we recall the standard upper-bounds on the smoothing parameter [50,73].

Lemma 2.4. *Let* $\Lambda \subseteq \mathbb{R}^m$ *be a lattice, and let* $\epsilon > 0$. *Then,*

$$\eta_\epsilon(\Lambda) \leq \frac{1}{\lambda_1^\infty(\Lambda^*)} \cdot \sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}}$$

and in fact, for every basis $\mathbf{B}$ *of* Λ,

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

We also recall the bound from [50, Lemma 5.3] and [83, Lemma 2.5] for the block-diagonal matrices. Here, we consider the ring setting which can be easily adapted from the aforementioned results.

Lemma 2.5. *Let $\ell, \delta > 1$ and suppose q is prime and $m \geq 2n \log_\delta q$. Then, there exists a negligible function ε such that for all $\mathbf{A}_2, \ldots, \mathbf{A}_\ell \in \mathcal{R}_q^{n\times m}$:*

$$\Pr\left[\eta_\varepsilon(\Lambda^\perp(\mathsf{diag}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_\ell)) \leq \delta \cdot \log(\ell m N) : \mathbf{A}_1 \leftarrow \mathcal{R}_q^{n\times m}\right] \geq 1 - q^{-nN} \ .$$

Further, we recall the regularity lemma from [69].

Lemma 2.6. (Regularity Lemma). *Let $q \equiv 5 \pmod 8$ be a prime, $N = \mathsf{poly}(\lambda)$ and k, n be positive integers such that $\mathsf{poly}(\lambda) \geq m \geq n$. Take $\mathfrak{s} > 2N \cdot q^{n/m + 2/(Nm)}$. Then, the following distributions are statistically close:*

$$\left\{(\mathbf{A}, \mathbf{A}\mathbf{x}) \middle| \begin{array}{c} \mathbf{A} \leftarrow \mathcal{R}_q^{n\times m} \\ \mathbf{x} \leftarrow \mathcal{D}_{\mathfrak{s}}^{mN} \end{array}\right\} \ \textit{and} \ \left\{(\mathbf{A}, \mathbf{u}) \middle| \begin{array}{c} \mathbf{A} \leftarrow \mathcal{R}_q^{n\times m} \\ \mathbf{u} \leftarrow \mathcal{R}_q^{n} \end{array}\right\} \ .$$

This is slightly modified from the original result in [69, Corollary 7.5] and [29, Lemma 4.2] in a sense that $\mathbf{A}$ might not be full-rank. However, Lemma 2.3 makes sure the event happens with negligible probability.

Tail bounds. When sampling over a sufficiently wide discrete Gaussian distribution, a small portion of the probability mass will be in the tail of the distribution, and thus with overwhelming probability the sampled lattice elements will have short norm. The following lemma from [73] formalises this intuition.

Lemma 2.7. *For any $0 < \epsilon < 1$, lattice $\Lambda \subseteq \mathbb{R}^m$, center $\mathbf{c} \in Span(\Lambda)$ and $\sigma > \eta_\epsilon(\Lambda)$,*

$$\Pr\left[\|\mathbf{z}\| \geq \sigma \cdot \sqrt{m} : \mathbf{z} \leftarrow \mathcal{D}_{\sigma,\Lambda,\mathbf{c}}\right] \leq \frac{1+\epsilon}{1-\epsilon} 2^{-m} \ .$$

We also recall the tail bounds for the regular discrete Gaussian distribution over integers [66].

Lemma 2.8. *Let $\mathbf{z} \leftarrow D_{\mathfrak{s}}^m$. Then* $\Pr\left[\|\mathbf{z}\| > t \cdot \mathfrak{s}\sqrt{\frac{m}{2\pi}}\right] < \left(te^{\frac{1-t^2}{2}}\right)^m$.

By setting $t = \sqrt{2\pi}$, the right-hand side can be upper-bounded by 2^{-2m}.

Preimage sampling for module lattices. Let $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ be a matrix over $\mathcal{R}_q$ and take any $\mathbf{u} \in \mathcal{R}_q^n$. We write $\mathbf{s} \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{u})$ to denote sampling $\mathbf{s} \leftarrow \mathcal{D}_\sigma^{mN}$ conditioned on $\mathbf{A}\mathbf{s} = \mathbf{u}$. Assuming there is some $\mathbf{x} \in \mathcal{R}_q^m$ which satisfies $\mathbf{A}\mathbf{x} = \mathbf{u}$, this is the same as sampling $\mathbf{s} \leftarrow \mathcal{D}_{\sigma,\mathbf{x},\Lambda^\perp(\mathbf{A})}$.

We will need the following lemma from [83, Lemma 2.7] for proving hiding property of the commitment scheme.

Lemma 2.9. *Let $n, m, q > 0$. Take any matrices $\mathbf{A} \in \mathcal{R}_q^{n\times m}, \mathbf{B} \in \mathcal{R}_q^{n\times \ell}$ where $\ell = \mathsf{poly}(n, \log q)$. Suppose the columns of $\mathbf{A}$ generate $\mathcal{R}_q$ and let $\mathbf{C}{:=}[\mathbf{A} \mid \mathbf{B}]$. Then, for every target vector $\mathbf{t} \in \mathcal{R}_q^n$ and any $\sigma \geq \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ for some $\epsilon = \mathsf{negl}(\lambda)$, the following distributions are statistically close:*

$$\left\{\mathbf{v} \middle| \mathbf{v} \leftarrow \mathbf{C}_\sigma^{-1}(\mathbf{t})\right\} \text{ and } \left\{\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix} \middle| \mathbf{v}_2 \leftarrow \mathcal{D}_\sigma^{\ell N}, \mathbf{v}_1 \leftarrow \mathbf{A}_\sigma^{-1}(\mathbf{t} - \mathbf{B}\mathbf{v}_2)\right\}.$$

Module-SIS. We recall the standard lattice-based Module-SIS assumption [61]

Definition 2.10. (Module-SIS). Let $q = q(\lambda)$, $n = n(\lambda)$, $m = m(\lambda)$, $\beta = \beta(\lambda)$ and $N = N(\lambda)$. We say that the $\mathsf{MSIS}_{n,m,N,q,\beta}$ assumption holds if for any PPT adversary $\mathcal{A}$, the following holds:

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

2.4. *NTRU Lattices*

As defined before, let N be a power of two, q a positive integer and $h \in \mathcal{R}_q$. The NTRU lattice associated to h is defined as

$$\Lambda_h{:=}\{(u, v) \in \mathcal{R}^2 : u + vh = 0 \bmod q\} \ .$$

Recall that there is an efficient algorithm $\mathsf{NTRU.TrapGen}$ [43,48,52,80], which given modulus q, the ring dimension N and the parameter $\mathfrak{s}$, outputs $h \in \mathcal{R}_q$ and a short basis of Λ_h. Below, we assume that $X^N + 1$ splits into two factors modulo q and we apply the main result of Stehlé and Steinfeld [80].

Lemma 2.11. (NTRU Trapdoor). *Let $q = \omega(N)$ be a prime such that $q \equiv 5 \pmod 8$. Take $\epsilon \in (0, 1/3)$ and $\mathfrak{s} \geq \max(\sqrt{N \ln(8Nq)} \cdot q^{1/2+\epsilon}, \omega(N^{3/2} \ln^{3/2} N))$. Then, there is a PPT algorithm $\mathsf{NTRU.TrapGen}(q, N, \mathfrak{s})$ which with an overwhelming probability outputs $h \in \mathcal{R}_q$ and a basis $\mathbf{T}_{\mathsf{NTRU}}$ of Λ_h such that $\|\tilde{\mathbf{T}}_{\mathsf{NTRU}}\| \leq N\mathfrak{s}$. Further, the statistical distance between the distribution of h and uniform over $\mathcal{R}_q^\times$ is at most $2^{10N} q^{-\lfloor \epsilon N \rfloor}$.*

2.5. *Gadget Trapdoors*

In this section, we recall the notion of gadget trapdoors as in [72], reformulate them for the module setting and state the key results on efficient sampling preimages using trapdoors.

We say that a matrix $\mathbf{G} \in \mathcal{R}_q^{n\times t}$ is primitive if its columns generate $\mathcal{R}_q^n$, i.e. if $\mathbf{G} \cdot \mathcal{R}^t = \mathcal{R}_q^n$. Note that if $\mathbf{G}$ is primitive, then $\mathsf{rot}(\mathbf{G})$ also is w.r.t. $\mathbb{Z}_q^{nN}$(i.e. $\mathsf{rot}(\mathbf{G})\mathbb{Z}^{tN} = \mathbb{Z}_q^{nN}$). We also recall the notion of a gadget trapdoor.

Definition 2.12. Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{H} \in \mathcal{R}_q^{n \times n}$, $\mathbf{G} \in \mathcal{R}_q^{n \times t}$ with $t \geq n$ and $\mathbf{H}$ invertible over $\mathcal{R}_q$. A $\mathbf{G}$-trapdoor for $\mathbf{A}$ with tag $\mathbf{H}$ is a matrix $\mathbf{R} \in \mathcal{R}_q^{m \times t}$ with $\mathbf{AR} = \mathbf{HG}$. The quality of a trapdoor is $s_1(\mathbf{R})$.

When not specified, we set the tag $\mathbf{H}{:=}\mathbf{I}$. In fact, all the theorems in this section can be generalised with a tag.

In this work, we consider one particular primitive matrix that naturally represents δ-base decomposition which we call the gadget matrix.

Definition 2.13. (Gadget Matrix). Let $\delta \geq 2$. We set $\tilde{q}{:=}\lfloor \log_\delta q \rfloor + 1$, and $\mathbf{g}^\top = [1, \delta, \ldots, \delta^{\tilde{q}-1}] \in \mathcal{R}_q^{1 \times \tilde{q}}$ and $\mathbf{G}_n{:=}\mathbf{I}_n \otimes \mathbf{g}^\top \in \mathcal{R}_q^{n \times n\tilde{q}}$. When the dimension are clear from context we simply write $\mathbf{G}$. Write $\mathbf{G}_n^{-1} : \mathcal{R}_q^{n \times t} \to \mathcal{R}_q^{n\tilde{q} \times t}$ 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 $\mathbf{G}_1^{-1}$.

[72, Lemma 5.3] says that having a $\mathbf{G}$-trapdoor for some matrix $\mathbf{A}$ enables to translate any nice basis of $\mathbf{G}$'s induced lattice into one for $\mathbf{A}$'s, whose shortness is proportional to the quality of the trapdoor.

Lemma 2.14. *Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{G} \in \mathcal{R}_q^{n \times t}$ be the gadget matrix with decomposition base δ, and suppose there exists a $\mathbf{G}$-trapdoor $\mathbf{R}$ for $\mathbf{A}$. Then, there is a basis $\mathbf{S_A}$ of $\Lambda^\perp(\mathbf{A})$ which satisfies $\left\|\tilde{\mathbf{S}}_\mathbf{A}\right\| \leq (s_1(\mathbf{R}) + 1)\sqrt{\delta^2 + 1}$. In particular, if $\|\mathbf{R}\| \leq \beta$ then for $\epsilon = \mathsf{negl}(\lambda)$:*

$$\eta_\epsilon(\Lambda^\perp(\mathbf{A})) \leq \beta\delta \cdot \omega(\sqrt{N \log(mN)}) \ .$$

We now give crucial properties about the trapdoor generation from [72].

Lemma 2.15. (Trapdoor Generation). *Let $q \equiv 5 \pmod 8$ be a prime, $N, n > 0, t = n\tilde{q}$ and $\mathbf{G}_n \in \mathcal{R}_q^{n \times t}$ be the gadget matrix. Take $m > t + n$. 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 $\mathbf{AR} = \mathbf{G}_n$ and $\|\mathbf{R}\| \leq \mathfrak{s}\sqrt{2t(m-t)N}$ where $\mathfrak{s} > 2N \cdot q^{\frac{n}{m-t} + \frac{2}{N(m-t)}}$. Moreover, $\mathbf{A}$ is statistically close to a uniformly random matrix in $\mathcal{R}_q^{n \times m}$.*

Proof. Let $m' = m - t$. Consider the following algorithm [72, Alg 1]:

1. Sample $\bar{\mathbf{A}} \leftarrow \mathcal{R}_q^{n \times m'}$.
2. Sample a matrix $\bar{\mathbf{R}} \leftarrow \mathcal{D}_\mathfrak{s}^{m'N \times tN}$ from a discrete Gaussian distribution.
3. Return $\mathbf{A}{:=}[\bar{\mathbf{A}} | \mathbf{G}_n - \bar{\mathbf{A}}\bar{\mathbf{R}}]$ and $\mathbf{R}{:=} \begin{bmatrix} \bar{\mathbf{R}} \\ \mathbf{I}_t \end{bmatrix}$

First, $\mathbf{AR} = \mathbf{G}$ as desired and $\|\mathbf{R}\| \leq \sqrt{t(\mathfrak{s}^2 m'N + 1)} \leq \mathfrak{s}\sqrt{2t(m-t)N}$ with an overwhelming probability by Lemma 2.8 for $t = \sqrt{2\pi}$. To argue pseudorandomness, we apply Lemma 2.6 and the hybrid argument to get that $\bar{\mathbf{A}}\bar{\mathbf{R}}$ is statistically close to uniform over $\mathcal{R}_q^{n \times t}$, and thus so is $\mathbf{A}$. □

The next lemma states that given a short **G**-trapdoor matrix **R** for **A**, one can efficiently sample preimages of **A** according to the discrete Gaussian distribution.

Lemma 2.16. (Preimage Sampling). *Let $N, n, m > 0$ and $t = n\tilde{q}$. 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 $\mathbf{s} \in \mathcal{R}_q^m$ such that $\mathbf{A}\mathbf{s} = \mathbf{v}$. Further, if $\sigma \geq \delta s_1(\mathbf{R}) \cdot \omega(\sqrt{\log nN})$, then the statistical distance between the following distributions is negligible:*

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

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* $\mathsf{SamplePre}(\mathbf{A}, \mathbf{R}, \mathbf{v}_i, \sigma)$.

Subtractive sets for monomials. We recall the following widely-used result from [19], which says that the (scaled) inverse of two distinct monomials in $\mathcal{R}$ has coefficients in $\{-1, 0, 1\}$.

Lemma 2.17. *Let $\mathcal{C} := \{X^i : i \in \mathbb{Z}\} \subseteq \mathcal{R}$. Then, for any two distinct $x, y \in \mathcal{C}$, we have* $\|\frac{2}{x-y}\|_\infty = 1$.

Short elements are invertible. For $\kappa > 0$, we define $S_\kappa := \{x \in \mathcal{R}_q : \|x\|_\infty \leq \kappa\}$ to be the set of ring elements in $\mathcal{R}_q$ with infinity norm at most κ. We recall the following invertibility result by Lyubashevsky and Seiler [70].

Lemma 2.18. *Let $1 \leq l < N$ be a power-of-two and suppose $q \equiv 2N/l + 1 \pmod{4N/l}$. Then, every non-zero element in S_κ is invertible over $\mathcal{R}_q$ as long as $\kappa < \sqrt{l/N} \cdot q^{l/N}$.*

We will use this lemma for $q \equiv 5 \pmod 8$ and thus $\ell = N/2$.

Rejection sampling. A crucial component in proving the *zero-knowledge* property of lattice-based (non-interactive) arguments is a rejection sampling procedure [66]. We recall the generalised version introduced recently by Boschini et al. [29] (specifically, [29, Lemma 3.1] for $t := \alpha/\pi$) for discrete Gaussian over arbitrary lattices (here we omit the case for ellipsoidal Gaussians).

Lemma 2.19. (Rejection Sampling [29]). *Take any $\alpha, T > 0$ and $\varepsilon \leq 1/2$. Let $\Lambda \subseteq \mathcal{R}^m$ be a lattice over $\mathcal{R}$ and $\sigma \geq \max(\alpha T, \eta_\varepsilon(\Lambda))$ be a parameter. Let $h : \mathcal{R}^m \times \mathcal{R}^m \rightarrow [0, 1]$ be a probability distribution which returns $(\mathbf{u}, \mathbf{v})$ where the vector $\mathbf{v}$ satisfies $\|\mathbf{v}\| \leq T$. Further, define $M := \exp(\frac{\pi}{\alpha^2} + 1)$ and $\epsilon := 2\frac{1+\varepsilon}{1-\varepsilon}\exp(-\alpha^2 \cdot \frac{\pi-1}{\pi^2})$. Then, the statistical distance between distributions* $\mathsf{RejSamp}$ *and* SimRS *defined in Fig. 3 is at*

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

Fig. 3. Rejection sampling [29].

most $\frac{\epsilon}{2M} + \frac{2\varepsilon}{M}$. *Moreover, the probability that* RejSamp *outputs something is at least* $\frac{1-\epsilon}{M}\left(1 - \frac{4\varepsilon}{(1+\varepsilon)^2}\right)$.

2.6. *Commitment Scheme*

We recall the notion of a commitment scheme, which is a crucial component of various proof systems. As folklore in lattice-based cryptography, we introduce the slack space, which has a role in the binding property.

Definition 2.20. 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 [12] $c \in \mathcal{S}$ and outputs a bit indicating whether C is a valid commitment to m under crs.

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 2.21. (Completeness). We say that 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 $m \in \mathcal{M}$:

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

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

Definition 2.22. (Relaxed Binding). A commitment scheme $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ satisfies relaxed binding if for every PPT adversary $\boldsymbol{\mathcal{A}}$:

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

Definition 2.23. (Hiding). A commitment scheme $\mathsf{CM} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ satisfies hiding if for every (stateful) PPT adversary $\boldsymbol{\mathcal{A}}$:

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

2.7. *Polynomial Commitment Scheme*

We also recall the notion of polynomial commitment schemes [58]. Polynomial commitment schemes extend commitments with the ability to prove evaluations of the committed polynomial.

Definition 2.24. 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:

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

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

  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.

Additionally, we require that the evaluations procedure satisfy some additional properties that we detail next. For simplicity, we give these definitions for non-interactive polynomial commitments, the interactive variant follows similarly.

Definition 2.25. (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 $u \in R$:

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

Definition 2.26. (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 κ if for all stateful PPT adversaries $\mathcal{P}^*$, there exists an expected PPT extractor $\mathcal{E}$ such that

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

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.

2.8. *Interactive Proofs*

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 $(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in R$. 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.

We recall a few basic properties of interactive proof systems: completeness and knowledge soundness.

Definition 2.27. (Completeness). A proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ for the relation R has statistical completeness with correctness error ϵ if for all adversaries $\boldsymbol{\mathcal{A}}$,

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

Definition 2.28. (Knowledge Soundness). A proof system $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ for the relation R is knowledge sound with knowledge error κ if there exists an expected PPT extractor $\mathcal{E}$ such that for any stateful PPT adversary $\mathcal{P}^*$:

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

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.

2.9. *Coordinate-Wise Special Soundness*

We generalise the notion of *special soundness* in the following way. Let S be a set and $\ell, k \in \mathbb{N}$. 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:

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

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 boil down to checking whether two elements are distinct. Further, we can define the set

$$\mathsf{SS}(S, \ell, k){:=}\left\{ \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \subseteq (S^\ell)^K : \begin{array}{c} \exists e \in [K], \forall i \in [\ell], \\ \exists J = \{j_1, \ldots, j_{k-1}\} \subseteq [K] \setminus \{e\}, \\ \forall j \in J, \mathbf{x}_e \equiv_i \mathbf{x}_j \end{array} \right\} ,$$

where $K{:=}\ell(k-1)+1$. To develop an intuition of the meaning of $\mathsf{SS}(S, \ell, k)$, consider a set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \in \mathsf{SS}(S, \ell, k)$. There is a "central" vector $\mathbf{x}_e \in X$ such that for each coordinate of $\mathbf{x}_e$, there are $k - 1$ other vectors in X that differ from $\mathbf{x}_e$ only in that coordinate. In other words, for each coordinate, there are k vectors in X that differ from each other only in that coordinate, and $\mathbf{x}_e$ is always one of them. As a simple example,

$$\{(2, 0, 0), (0, 1, 0), (0, 0, 0), (0, 0, 5), (0, 0, 4), (0, 2, 0), (3, 0, 0)\} \in \mathsf{SS}(\mathbb{Z}_7, 3, 3)$$

– the "central" vector $(0, 0, 0)$ differs in, and only in, each coordinate from two other vectors in the set. Note that $\mathsf{SS}(S, 1, k)$ simply contains k-sets of distinct elements in S.

We are ready to define the notion of coordinate-wise special soundness. We start with the case for Σ-protocols.

Definition 2.29. (CWSS for Σ-protocols). Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be public-coin three-round interactive proof system for relation R, and suppose the challenge space of $\mathcal{V}$ is $\mathcal{C} = S^\ell$. We say that Π is ℓ-coordinate-wise k-special sound if there exists a polynomial time algorithm that on input an index $\mathbb{i}$, statement $\mathbb{x}$ and $\ell(k-1)+1$ accepting transcripts $(a, \mathbf{c}_i, z_i)_{i \in [\ell(k-1)+1]}$, with $\{\mathbf{c}_1, \ldots, \mathbf{c}_{\ell(k-1)+1}\} \in \mathsf{SS}(S, \ell, k)$ and common first message a, outputs a witness $\mathbb{w} \in \mathsf{R}(\mathbb{i}, \mathbb{x})$.

Clearly, we obtain the standard k-special soundness property if $\ell = 1$. Next, we extend this notion to multi-round protocols via a tree of transcripts. For simplicity, we assume that in each round the verifier picks challenge uniformly at random from the same challenge space S^ℓ, which will be the case for most of our protocols.

Definition 2.30. (CWSS for Multi-Round Protocols). Let $\Pi = (\mathsf{Setup}, \mathcal{P}, \mathcal{V})$ be 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 $K = (\ell(k-1)+1)^\mu$ 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 $\ell(k-1)+1$ children corresponding to $\ell(k-1)+1$ distinct challenges which, as a set of vectors, lie in $\mathsf{SS}(S, \ell, k)$. Every transcript corresponds to exactly one path from the root to a leaf node.

We say that Π is ℓ-coordinate-wise k-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 $\mathbb{w} \in \mathsf{R}(\mathbb{i}, \mathbb{x})$.

In this paper, we only focus on ℓ-coordinate-wise 2-special sound protocols, which we will call ℓ-coordinate-wise special sound.

We prove in Sect. 7 that coordinate-wise special soundness implies knowledge soundness in the interactive setting.

Lemma 2.31. *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 k-special sound and $(\ell(k-1))^\mu = \mathsf{poly}(\lambda)$, then it is knowledge sound with knowledge error $\mu\ell(k-1)/|S|$.*

The resulting knowledge extractor runs the malicious prover $(\ell(k-1)+1)^\mu$ times in expectation. Hence, in order to keep the knowledge extractor expected PPT, we need $(\ell(k-1))^\mu = \mathsf{poly}(\lambda)$.

The result can be easily extended to the case, where in each i-th round the challenges from the verifier are picked from S^{ℓ_i} for $\ell_i > 0$. Then, the knowledge error becomes $(\ell_1 + \ldots + \ell_\mu)(k-1)/|S|$ and the extractor runs the malicious prover at most $\prod_{i=1}^{\mu}(\ell_i(k-1)+1)$ times.

Finally, using the exact methodology as in [10], in Sect. 8 we show that coordinate-wise special soundness implies (adaptive) knowledge soundness of the Fiat–Shamir transformed protocol in the random oracle model.

Lemma 2.32. (Informal). *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* k*-special sound and* $(\ell(k-1))^{\mu} = \mathsf{poly}(\lambda)$, *then the Fiat–Shamir transformation of* Π *is knowledge sound in the random oracle model with knowledge error* $(Q+1)\mu\ell(k-1)/|S|$, *where* Q *is the number of random oracle queries made by an adversary.*

3. Power-BASIS Assumption

Our construction of the polynomial commitment will rely on a new lattice-based assumption $\mathsf{PowerBASIS}$ which is a special case of the BASIS assumption[13] introduced by Wee and Wu [83]. We begin by adapting the latter assumption to the ring setting. Recall that $\mathbf{G}_n$ is a gadget matrix with base δ as in Definition 2.13 and N is the ring dimension of $\mathcal{R}$. We fix the prime modulus $q \equiv 5 \pmod 8$ and set $\tilde{q} := \lfloor \log_\delta q \rfloor + 1$.

Definition 3.1. (BASIS). Let $q, n, m, n', m', \ell, N, \sigma, \beta$ be 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 $\mathsf{BASIS}_{n,m,n',m',N,q,\ell,\sigma,\beta}$ assumption holds w.r.t. Samp if for any PPT adversary $\mathcal{A}$:

$$\Pr\left[\begin{array}{c} \mathbf{A}\mathbf{s} = \mathbf{0} \\ 0 < \|\mathbf{s}\| \le \beta \end{array} \middle| \begin{array}{c} \mathbf{A} \leftarrow \mathcal{R}_q^{n\times m}, (\mathbf{B}, \mathsf{aux}) \leftarrow \mathsf{Samp}(\mathbf{A}) \\ \mathbf{T} \leftarrow \mathbf{B}_{\sigma}^{-1}(\mathbf{G}_{n'}) \\ \mathbf{s} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{B}, \mathbf{T}, \mathsf{aux}) \end{array}\right] \le \mathsf{negl}(\lambda) \ .$$

Intuitively, the BASIS assumption says that it is hard to find a short solution for $\mathbf{A}$, even when given a trapdoor for a matrix $\mathbf{B}$ related to $\mathbf{A}$. The trapdoor allows the adversary to sample preimages of $\mathbf{B}$, and thus it is easy to break the assumption if $\mathbf{B}$ contains too much information about $\mathbf{A}$, e.g. when $\mathbf{B} = \mathbf{A}$.

Furthermore, we provide three concrete instantiations of the sampling algorithm Samp.

Definition 3.2. (BASIS Instantiations). We consider three concrete instantiations of the BASIS assumption:

- $\mathsf{StructBASIS}_{n,m,N,q,\ell,\sigma,\beta}$: The sampling algorithm $\mathsf{Samp}(\mathbf{A})$ first generates a row $\mathbf{a}^{\mathsf{T}} \leftarrow \mathcal{R}_q^m$ and sets

$$\mathbf{A}^{\star} := \begin{bmatrix} \mathbf{a}^{\mathsf{T}} \\ \mathbf{A} \end{bmatrix} \in \mathcal{R}_q^{(n+1)\times m} \ . \tag{11}$$

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

Further, it samples $\mathbf{W}_i \leftarrow \mathsf{GL}(n+1, \mathcal{R}_q)$ for all $i \in [\ell]$, and outputs

$$\mathbf{B}_\ell := \left[\begin{array}{ccc|c} \mathbf{W}_1\mathbf{A}^\star & & & -\mathbf{G}_{n+1} \\ & \ddots & & \vdots \\ & & \mathbf{W}_\ell\mathbf{A}^\star & -\mathbf{G}_{n+1} \end{array}\right] \quad \text{and} \quad \mathsf{aux} := (\mathbf{W}_1, \dots, \mathbf{W}_\ell) \ .$$

- $\mathsf{PowerBASIS}_{n,m,N,q,\ell,\sigma,\beta}$: Here, $\mathsf{Samp}(\mathbf{A})$ generates a row $\mathbf{a}^\intercal \leftarrow \mathcal{R}_q^\ell$ and sets $\mathbf{A}^\star$ as in (11). Then, it samples $\mathbf{W} \leftarrow \mathsf{GL}(n+1, \mathcal{R}_q)$, and outputs

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

- $\mathsf{PRISIS}_{n,m,N,q,\ell,\sigma,\beta}$: $\mathsf{Samp}(\mathbf{A})$ samples a row $\mathbf{a}^\intercal \leftarrow \mathcal{R}_q^\ell$ and sets $\mathbf{A}^\star$ as in (11). Then, it samples $w \leftarrow \mathsf{GL}(1, \mathcal{R}_q)$, and outputs

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

Informally, the StructBASIS variant corresponds to the structured version of the BASIS assumption used to build functional commitments [83]. PowerBASIS is the special case, where instead of picking ℓ uniformly random invertible matrices $\mathbf{W}_i$, one takes a single invertible matrix, and sets $\mathbf{W}_i := \mathbf{W}^{i-1}$ for $i \in [\ell]$. Finally, PRISIS is the instance where each $\mathbf{W}_i := w^{i-1}\mathbf{I}_{n+1}$ for $i \in [\ell]$ and $w \in \mathcal{R}_q$ is an invertible element.

Intuitively, StructBASIS seems to be the hardest variant to break out of the three since it carries the least structure. Then, PowerBASIS should be an easier problem due to the very specific relation between matrices $\mathbf{W}_i$. Finally, PRISIS carries a lot of structure, since it introduces commutativity between the matrices $\mathbf{W}_i$ and $\mathbf{A}^\star$, i.e. $w^{i-1}\mathbf{A}^\star = \mathbf{A}^\star(w^{i-1} \cdot \mathbf{I}_m)$, which can somehow be useful for the adversary to break the assumption.

Remark 3.3. To simplify reductions in the paper, we explicitly require the matrices $\mathbf{W}_i$ to be invertible (unlike in [83]). Note that this condition can be dropped by arguing that, depending on the parameters q and N, with overwhelming probability a uniformly random matrix $\mathbf{W}$ is invertible over $\mathcal{R}_q$ (cf. Lemma 2.3).

Remark 3.4. In the PowerBASIS assumption, an adversary is given a trapdoor to the structured matrix $\mathbf{B}_\ell$. The most intuitive and standard way to attack lattice problems is to perform certain linear operations. In one way or another, here we would end up with obtaining (many) short solutions $\mathbf{s}_1, \dots, \mathbf{s}_\ell$ such that: $\mathbf{W}^0\mathbf{A}^\star\mathbf{s}_1 + \dots + \mathbf{W}^{\ell-1}\mathbf{A}^\star\mathbf{s}_\ell = 0$. The intuition why this should not help to find a short non-zero solution to $\mathbf{A}^\star$ (or $\mathbf{A}$) is two-fold. Firstly, since $\mathbf{W}$ is an invertible matrix, the adversary is not able to use any commutative properties, e.g. $\mathbf{W}\mathbf{A}^\star = \mathbf{A}^\star\mathbf{W}$, thus limiting the adversary's algebraic capabilities to find short solutions to $\mathbf{A}^\star$. Even if this is the case (as in PRISIS), the matrix $\mathbf{W}$ does contain arbitrarily large $\mathcal{R}_q$ elements. Hence, obtaining a short non-zero solution to $\mathbf{A}^\star$, which involves W, seems unlikely.

Falsifiable version of PowerBASIS. Note that the challenger in the PowerBASIS game from Sect. 3 is not efficient since it needs to sample a random trapdoor $\mathbf{T}$ according to a discrete Gaussian distribution. In order to make the assumption falsifiable, one could let the challenger sample efficiently using the SamplePre algorithm, e.g. as in the Setup algorithm of Fig. 4. Further, for efficiency we can ensure that the sampled matrix $\mathbf{A}$ from $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m)$ is *computationally* indistinguishable from random[14].

3.1. *Hardness of BASIS for Low Dimensions*

We analyse the relationship between the three newly introduced instantiations for the dimension $\ell = 2$. To this end, we analyse the following technical lemma which will be used in all our results of this section. Intuitively, it says that if one can find a short solution to a specific linear equation, then one can also build a BASIS trapdoor.

Lemma 3.5. *Let $n, m, N > 0$ and $\alpha \geq 1$. Denote $t = n\tilde{q}$. Then, there exists an efficient deterministic algorithm, that given as input a matrix $\mathbf{A}^\star \in \mathcal{R}_q^{n\times m}$, invertible $\mathbf{W}_1, \mathbf{W}_2, \mathbf{H} \in \mathsf{GL}(n, \mathcal{R}_q)$ and two matrices $\mathbf{T}_1, \mathbf{T}_2 \in \mathcal{R}_q^{m\times t}$, which satisfy $\|(\mathbf{T}_1, \mathbf{T}_2)\| \leq \alpha$ for $i = 1, 2$ and*

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

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:

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

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

Proof. Define the following matrices:

$$\begin{aligned} \mathbf{S}_{1,3} &:= \mathbf{G}^{-1}(\mathbf{W}_1\mathbf{A}^\star\mathbf{T}_1 - \mathbf{H}\mathbf{G}_n) = \mathbf{G}^{-1}(\mathbf{W}_2\mathbf{A}^\star\mathbf{T}_2) \\ \mathbf{S}_{2,3} &:= \mathbf{G}^{-1}(-\mathbf{W}_1\mathbf{A}^\star\mathbf{T}_2 - \mathbf{H}\mathbf{G}_n) = \mathbf{G}^{-1}(-\mathbf{W}_1\mathbf{A}^\star\mathbf{T}_1). \end{aligned}$$

Then, by construction we get:

$$\begin{bmatrix} \mathbf{W}_1\mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{W}_2\mathbf{A}^\star & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{T}_1 & -\mathbf{T}_1 \\ \mathbf{T}_2 & -\mathbf{T}_2 \\ \mathbf{S}_{1,3} & \mathbf{S}_{2,3} \end{bmatrix} = \begin{bmatrix} \mathbf{H}\mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}\mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} \end{bmatrix}.$$

[14] Concretely, in the proof of Lemma 2.15 we would rely on the argument that $\bar{\mathbf{A}}\bar{\mathbf{R}}$ is pseudorandom based on Module-LWE [61] rather than Lemma 2.6.

By setting

$$\mathbf{S} := \begin{bmatrix} \mathbf{T}_1 & -\mathbf{T}_1 \\ \mathbf{T}_2 & -\mathbf{T}_2 \\ \mathbf{S}_{1,3} & \mathbf{S}_{2,3} \end{bmatrix} \quad \text{and} \quad \mathbf{H}^* := \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} ,$$

we observe that $\mathbf{S}$ is a $\mathbf{G}_{2n}$-trapdoor for $\mathbf{B}$ with a tag $\mathbf{H}^*$ and $\|\mathbf{S}\|^2 \leq 2\alpha^2 + 2t^2 N$, which concludes the proof. □

Our first result says that $\mathsf{StructBASIS}$ and $\mathsf{PowerBASIS}$ are equivalent for the dimension $\ell = 2$.

Lemma 3.6. ($\mathsf{StructBASIS} \iff \mathsf{PowerBASIS}$). *Let $n, N, \beta \geq 1$ and $t := (n+1)\tilde{q}$. Suppose $m > t + n$ and $\mathfrak{s} > 2N \cdot q^{\frac{n+1}{m-t} + \frac{2}{N(m-t)}}$. If σ_0, σ_1 satisfy the following inequalities:*

$$\sigma_0 \geq \delta \mathfrak{s} N \cdot \omega(\sqrt{t(m-t)\log(mN)}), \quad \sigma_1 \geq \delta \sqrt{2tN(\sigma_0^2 m' + t)} N \cdot \omega(\sqrt{\log nN}),$$

where $m' = 2m + t$, then the following statements are true:

1. $\mathsf{StructBASIS}_{n,m,N,q,2,\sigma_0,\beta}$ *assumption holds under the* $\mathsf{PowerBASIS}_{n,m,N,q,2,\sigma_1,\beta}$ *assumption.*
2. $\mathsf{PowerBASIS}_{n,m,N,q,2,\sigma_0,\beta}$ *assumption holds under the* $\mathsf{StructBASIS}_{n,m,N,q,2,\sigma_1,\beta}$ *assumption.*

Proof. We only show the first statement since the other direction follows identically. Let $\mathcal{A}$ be a PPT adversary for the $\mathsf{StructBASIS}_{n,m,N,q,2,\sigma,\beta}$ problem and suppose it wins with probability ϵ. We provide a PPT algorithm $\mathcal{B}$ for solving $\mathsf{PowerBASIS}_{n,m,N,q,2,\sigma,\beta}$ which does the following. First, $\mathcal{B}$ is given a tuple $(\mathbf{A}, \mathbf{B}, \mathbf{T}, \mathbf{W})$ where

$$\mathbf{B} := \begin{bmatrix} \mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{W}\mathbf{A}^\star & -\mathbf{G} \end{bmatrix} \quad \text{and} \quad \mathbf{T} := \begin{bmatrix} \mathbf{T}_{1,1} & \mathbf{T}_{1,2} \\ \mathbf{T}_{2,1} & \mathbf{T}_{2,2} \\ \mathbf{T}_{3,1} & \mathbf{T}_{3,2} \end{bmatrix} .$$

First, we claim that the following probability is negligible:

$$\epsilon_{\mathsf{smooth}} := \Pr\left[\sigma_0 < \eta_\epsilon(\Lambda^\perp(\mathbf{B})) \middle| \mathbf{A}^\star \leftarrow \mathcal{R}_q^{(n+1)\times m} \right] .$$

Indeed, note that by Lemma 2.15 we obtain:

$$\Pr\left[\sigma_0 < \eta_\epsilon(\Lambda^\perp(\mathbf{B})) \middle| (\mathbf{A}^\star, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n+1, m) \right] \geq \epsilon_{\mathsf{smooth}} - \mathsf{negl}(\lambda) .$$

If $(\mathbf{A}^\star, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n+1, m)$ then the following matrix $\mathbf{R}^*$ is a $\mathbf{G}_{2n}$-trapdoor for $\mathbf{B}$ with a tag $\mathbf{H}^*$, where:

$$\mathbf{R}^* := \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{H}^* := \begin{bmatrix} \mathbf{I}_{n+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{bmatrix} .$$

Moreover, $\|\mathbf{R}^*\| \leq 2\mathfrak{s}\sqrt{t(m-t)N}$ with an overwhelming probability (cf. Lemma 2.7). If this is the case then by assumption $\sigma_0 \geq \delta \cdot \|\mathbf{R}^*\| \cdot \omega(\sqrt{t(m-t)\log(mN)})$. Then, by combining Lemma 2.14 with Lemma 2.2, we obtain

$$\mathsf{negl}(\lambda) = \Pr\left[\sigma_0 < \eta_\epsilon(\Lambda^\perp(\mathbf{B})) \middle| (\mathbf{A}^\star, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n+1, m)\right] \geq \epsilon_{\mathsf{smooth}} - \mathsf{negl}(\lambda)$$

and thus $\sigma_0 \geq \eta_\epsilon(\Lambda^\perp(\mathbf{B}))$ with an overwhelming probability, where $\mathbf{B}$ is the matrix received by $\mathcal{B}$. Thus, we can apply Lemma 2.7 to deduce that with an overwhelming probability[15]

$$\left\| \begin{bmatrix} \mathbf{T}_{1,1} \\ \mathbf{T}_{1,2} \end{bmatrix} \right\| \leq \alpha := \sigma_0 \sqrt{m' t N} .$$

Further, by simple calculation we can deduce that

$$\mathbf{A}^\star \mathbf{T}_{1,1} - \mathbf{W}\mathbf{A}^\star \mathbf{T}_{1,2} = \mathbf{G} .$$

The reduction $\mathcal{B}$ now samples a uniformly random $\mathbf{W}_1 \leftarrow \mathsf{GL}(n+1, \mathcal{R}_q)$ and defines $\mathbf{W}_2 := \mathbf{W}_1\mathbf{W}$. Thus

$$\mathbf{W}_1\mathbf{A}^\star \mathbf{T}_{1,1} - \mathbf{W}_2\mathbf{A}^\star \mathbf{T}_{1,2} = \mathbf{W}_1\mathbf{G} .$$

By applying Lemma 3.5, $\mathcal{B}$ can obtain a $\mathbf{G}_{2(n+1)}$-trapdoor $\mathbf{S}$ for

$$\mathbf{B}' := \begin{bmatrix} \mathbf{W}_1\mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{W}_2\mathbf{A}^\star & -\mathbf{G} \end{bmatrix}$$

with the tag $\mathbf{H}^* := \mathbf{I}_2 \otimes \mathbf{W}_1$ where $\|\mathbf{S}\| \leq \sqrt{2(\alpha^2 + t^2 N)} \leq \sqrt{2tN(\sigma_0^2 m' + t)}$. Then, the algorithm $\mathcal{B}$ runs $\mathbf{T}' \leftarrow \mathsf{SamplePre}(\mathbf{B}', \mathbf{S}, \mathbf{G}_{2(n+1)}, \sigma_1)$. Finally, $\mathcal{B}$ sends $(\mathbf{A}, \mathbf{B}', \mathbf{T}', \mathsf{aux}' := (\mathbf{W}_1, \mathbf{W}_2))$ to $\mathcal{A}$ and returns what $\mathcal{A}$ outputs.

To argue correctness of the reduction, first note that aux' and $\mathbf{B}'$ are correctly generated. Further, by assumption we have $\sigma_1 \geq \delta\|\mathbf{S}\| \cdot \omega(\sqrt{N \log nN})$ and thus by Lemma 2.16, the distribution of $\mathsf{SamplePre}(\mathbf{B}', \mathbf{S}, \mathbf{G}_{2(n+1)}, \sigma_1)$ is statistically close to $\mathbf{B}'^{-1}_{\sigma_1}(\mathbf{G}_{2(n+1)})$. Consequently, $\mathcal{A}$ outputs a valid answer to $\mathcal{B}$ with probability $\epsilon - \mathsf{negl}(\lambda)$. Finally, a valid solution for $\mathsf{StructBASIS}$ implies a valid solution for $\mathsf{PowerBASIS}$, which concludes the proof. □

The next result focuses on the PRISIS variant. It turns out that the commutative property of the assumption allows to reduce to standard assumptions.

[15] We note that the bound is not tight.

Lemma 3.7. *(*$\mathsf{PRISIS} \implies \mathsf{MSIS}$*). Let $n > 0, m \geq n$ and denote $t = (n+1)\tilde{q}$. Let $q = \omega(N)$. Take $\epsilon \in (0, 1/3)$ and $\mathfrak{s} \geq \max(\sqrt{N \ln(8Nq)} \cdot q^{1/2+\epsilon}, \omega(N^{3/2} \ln^{3/2} N))$ such that $2^{10N} q^{-\lfloor \epsilon N \rfloor}$ is negligible. Let*

$$\sigma \geq \delta \sqrt{tN \cdot (N^2 \mathfrak{s}^2 m + 2t)} \cdot \omega(\sqrt{N \log nN}).$$

Then, $\mathsf{PRISIS}_{n,m,N,q,2,\sigma,\beta}$ is hard under the $\mathsf{MSIS}_{n,m,N,q,\beta}$ assumption.

Proof. Suppose there is a PPT algorithm $\mathcal{A}$ which wins $\mathsf{PRISIS}_{n,m,N,q,2,\sigma,\beta}$ with probability ϵ. We revisit the PRISIS security game and introduce a single game hop. The purpose of the hybrid argument will be to plug in the NTRU trapdoor inside the auxiliary information w. We define ε_i to be the probability that $\mathcal{A}$ wins $\mathsf{Game}\ i$.

Game 1: This is the standard PRISIS security game. To recall, the challenger samples $\mathbf{a} \leftarrow \mathcal{R}_q^m$, $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$ and sets $\mathbf{A}^\star$ as in (11). Then, it generates an invertible element $w \leftarrow \mathcal{R}_q^\times$ and computes the matrix:

$$\mathbf{B} := \begin{bmatrix} \mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{W}\mathbf{A}^\star & -\mathbf{G} \end{bmatrix} .$$

where $\mathbf{W} := w \cdot \mathbf{I}_{n+1}$. Then, it samples $\mathbf{T} \leftarrow \mathbf{B}_{\sigma_1}^{-1}(\mathbf{G}_{2(n+1)})$ and outputs $(\mathbf{A}, \mathbf{B}, \mathbf{T}, w)$ to the adversary $\mathcal{A}$. By definition, $\varepsilon_1 = \epsilon$.

Game 2: In this game, we obtain w by running $(w, \mathbf{T}_{\mathsf{NTRU}}) \leftarrow \mathsf{NTRU.TrapGen}(q, N, \mathfrak{s})$ algorithm. By Lemma 2.11, $\varepsilon_2 \geq \varepsilon_1 - 2^{10N} q^{-\lfloor \varepsilon N \rfloor}$.

Suppose there is an adversary which wins Game_2. We now show how to build a PRISIS trapdoor $\mathbf{T}$ given the Module-SIS matrix $\mathbf{A}$ and the NTRU trapdoor $\mathbf{T}_{\mathsf{NTRU}}$. To this end, we will show how to find short matrices $\mathbf{S}_1, \mathbf{S}_2$ such that:

$$\mathbf{A}^\star \mathbf{S}_1 - w \mathbf{A}^\star \mathbf{S}_2 = \mathbf{G} .$$

Let $\mathbf{g}_i$ be the i-th column of $\mathbf{G}$. Assuming that $\mathbf{A}^\star$ is full-rank (cf. Lemma 2.3) and using linear algebra, we can find a (possibly large) vector $\mathbf{t}$ such that $\mathbf{A}^\star \mathbf{t} = \mathbf{g}_i$. Now, using the NTRU trapdoor $\mathbf{T}_{\mathsf{NTRU}}$ (such that $\|\tilde{\mathbf{T}}_{\mathsf{NTRU}}\| \leq N\mathfrak{s}$ by Lemma 2.11) and the nearest plane algorithm [63], we can find vectors $(\mathbf{s}_{1,i}, \mathbf{s}_{2,i}) \in \mathcal{R}_q^m \times \mathcal{R}_q^m$ such that:

$$\mathbf{s}_{1,i} - w\mathbf{s}_{2,i} = \mathbf{t} \text{ and } \|(\mathbf{s}_{1,i}, \mathbf{s}_{2,i})\| \leq N\mathfrak{s}\sqrt{mN/2}.$$

Therefore

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

Thus, we obtain the matrices $\mathbf{S}_1, \mathbf{S}_2$ by concatenation where

$$\left\| \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \end{bmatrix} \right\| \leq \alpha := N\mathfrak{s}\sqrt{mtN/2} .$$

Consequently, by Lemma 3.5, we can build a $\mathbf{G}_{2(n+1)}$-trapdoor $\mathbf{S}$ for $\mathbf{B}$ such that

$$\|\mathbf{S}\| \leq \sqrt{2(\alpha^2 + t^2 N)} = \sqrt{tN \cdot (N^2 \mathfrak{s}^2 m + 2t)} \ .$$

Hence, the reduction $\mathcal{B}$ can construct the trapdoor $\mathbf{S}$ as above and then randomise the trapdoor for $\mathbf{B}$ by running $\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{S}, \mathbf{G}_{2(n+1)}, \sigma)$. Finally it sends the tuple to $\mathcal{A}$ and returns what it outputs. By Lemma 2.16, $\mathcal{B}$ wins the Module-SIS game with probability at least $\varepsilon_2 - \mathsf{negl}(\lambda)$, which concludes the proof. □

3.2. *Higher Dimensions*

One could hope that the techniques to analyse hardness of the BASIS assumption can be translated to higher dimensions. This could be promising especially for the PRISIS assumption, which we managed to reduce to standard lattice assumptions for the $\ell = 2$ case. Unfortunately, the reduction falls flat when considering higher dimensions.

We showcase this for $\ell = 3$. Following the approach for the smaller dimension, the goal is to find short matrices $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ such that

$$\begin{aligned} \mathbf{A}^\star \mathbf{S}_1 - w\mathbf{A}^\star \mathbf{S}_2 &= \mathbf{Z}_1 \\ \mathbf{A}^\star \mathbf{S}_2 - w\mathbf{A}^\star \mathbf{S}_3 &= \mathbf{Z}_2 \end{aligned} \tag{12}$$

for any $\mathbf{Z}_1, \mathbf{Z}_2$ given the NTRU trapdoor for w. If this is possible, we could set $\mathbf{Z}_1 = \mathbf{G}$ and $\mathbf{Z}_2 = \mathbf{0}$ which would give us:

$$\begin{aligned} \mathbf{A}^\star \mathbf{S}_1 - w\mathbf{A}^\star \mathbf{S}_2 &= \mathbf{G} \\ w\mathbf{A}^\star \mathbf{S}_2 - w^2\mathbf{A}^\star \mathbf{S}_3 &= \mathbf{0}. \end{aligned}$$

Set $\mathbf{S}_4 := \mathbf{G}^{-1}(\mathbf{A}^\star \mathbf{S}_1 - \mathbf{G})$. Then, we have:

$$\begin{bmatrix} \mathbf{A}^\star & \mathbf{0} & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & w\mathbf{A}^\star & \mathbf{0} & -\mathbf{G} \\ \mathbf{0} & \mathbf{0} & w^2\mathbf{A}^\star & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \mathbf{S}_3 \\ \mathbf{S}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{G} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \ .$$

We proceed similarly for

$$(\mathbf{Z}_1, \mathbf{Z}_2) = (-\mathbf{G}, w^{-1}\mathbf{G}) \quad \text{and} \quad (\mathbf{Z}_1, \mathbf{Z}_2) = (\mathbf{0}, -w^{-1}\mathbf{G}) \ .$$

Thus, we managed to build a $\mathbf{G}_{3(n+1)}$-trapdoor for $\mathbf{B}$. What is left to do is to produce short $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ which satisfy (12). To this end, consider the q-ary lattice

$$\Lambda = \left\{ (s_1, s_2, s_3) : \begin{bmatrix} 1 & -w & 0 \\ 0 & w & -w^2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \mathbf{0} \bmod q \right\} \ .$$

Suppose we can build a short basis for Λ given the NTRU trapdoor for w. Let $\mathbf{z}_{1,i}, \mathbf{z}_{2,i}$ be the i-th column of $\mathbf{Z}_1$ and $\mathbf{Z}_2$. Now, assuming that $\mathbf{A}^\star$ is full-rank, we can find (possibly large) $\mathbf{t}_1$ and $\mathbf{t}_2$ such that $\mathbf{A}^\star \mathbf{t}_j = \mathbf{z}_{j,i}$ for $j = 1, 2$. Now, using the short basis for Λ, we can sample short vectors $\mathbf{s}_{1,i}, \mathbf{s}_{2,i}, \mathbf{s}_{3,i}$ such that:

$$\begin{aligned} \mathbf{s}_{1,i} - w\mathbf{s}_{2,i} &= \mathbf{t}_1 \\ \mathbf{s}_{2,i} - w\mathbf{s}_{3,i} &= \mathbf{t}_2. \end{aligned}$$

Hence,

$$\begin{aligned} \mathbf{A}^\star\mathbf{s}_{1,i} - w\mathbf{A}^\star\mathbf{s}_{2,i} &= \mathbf{A}^\star(\mathbf{s}_{1,i} - w\mathbf{s}_{2,i}) = \mathbf{A}^\star\mathbf{t}_1 = \mathbf{z}_{1,i} \\ \mathbf{A}^\star\mathbf{s}_{2,i} - w\mathbf{A}^\star\mathbf{s}_{3,i} &= \mathbf{A}^\star(\mathbf{s}_{2,i} - w\mathbf{s}_{3,i}) = \mathbf{A}^\star\mathbf{t}_2 = \mathbf{z}_{2,i}. \end{aligned}$$

Therefore, we obtain the matrices $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ by concatenation.

Unfortunately, we are only aware of the following two bases of Λ:

$$\begin{bmatrix} w^2 & w & 1 \\ q & 0 & 0 \\ 0 & q & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} u^2 & uv & v^2 \\ \bar{u}^2 & \bar{u}\bar{v} & \bar{v}^2 \\ \bar{u}u & \bar{u}v & \bar{v}v \end{bmatrix} ,$$

where $\mathbf{T}_{\mathsf{NTRU}} := ((u, v), (\bar{u}, \bar{v}))$ is the short NTRU basis. Since $\|u\|, \|v\| \approx \sqrt{q}$, the latter basis cannot have short coefficients. We leave further analysis of this approach for future work.

4. Power-BASIS Commitment Scheme

In this section we define a compressing commitment scheme which stems from the vector commitment construction of Wee and Wu [83]. We inherit a crucial property from the aforementioned work that we support committing to arbitrarily large ring elements. Let $\ell := d + 1$ be the length of the committed vectors over $\mathcal{R}_q$. Thus, the message space is $\mathcal{M} := \mathcal{R}_q^{d+1}$. We let γ, β_s be the parameters controlling the norm of various vectors. Further, we define the slack space as the vector of short polynomials:

$$\mathcal{S} := \{(c_0, \ldots, c_d) : \forall i \in [0, d], c_i \in \mathcal{R}_q^\times \wedge \|c_i\|_1 \leq \beta_s\} \ .$$

Informally, we say that a slack is a single element $c \in \mathcal{R}_q$ if $(c, \ldots, c) \in \mathcal{S}$. Finally, we define $t = n\tilde{q}$ and $\mathbf{G} := \mathbf{G}_n \in \mathcal{R}_q^{n \times t}$.

We now give intuition on the construction, and provide a formal description in Fig. 4. The setup algorithm uses the $\mathsf{TrapGen}$ and $\mathsf{SamplePre}$ algorithms defined in Sect. 2.5. Namely, it first generates the two matrices $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m)$ along with a uniformly random invertible $\mathbf{W} \leftarrow \mathsf{GL}(n, \mathcal{R}_q)$. Then, $\mathbf{AR} = \mathbf{G}$, where $\|\mathbf{R}\| \leq \mathfrak{s}\sqrt{2t(m-t)N}$ and $\mathfrak{s} > 2N \cdot q^{\frac{n}{m-t} + \frac{2}{N(m-t)}}$ (c.f. Lemma 2.15). Further, it com-

PowerBASIS **Commitment Scheme**

```
Setup(1^λ)
  1. Sample $(\mathbf{A}, \mathbf{R}) \leftarrow \mathsf{TrapGen}(n, m)$.
  2. Sample $\mathbf{W} \leftarrow \mathsf{GL}(n, \mathcal{R}_q)$
  3. Let $\mathbf{R}_i := \mathbf{R}\mathbf{G}^{-1}(\mathbf{W}^{-i}\mathbf{G})$ for $i \in [0, d]$.
  4. Set
     $\mathbf{B} := \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \quad \tilde{\mathbf{R}} := \left[\begin{array}{ccc} \mathbf{R}_0 & & \\ & \ddots & \\ & & \mathbf{R}_d \\ \hline & \mathbf{0} & \end{array}\right].$
  5. Sample $\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \tilde{\mathbf{R}}, \mathbf{G}_{n(d+1)}, \sigma_0)$.
  6. Return $\mathsf{crs} := (\mathbf{A}, \mathbf{W}, \mathbf{T})$.

Commit($\mathsf{crs}, \mathbf{f} \in \mathcal{R}_q^{d+1}$)
  1. Parse $\mathbf{f} := (f_0, f_1, \ldots, f_d)$
  2. Set $\mathbf{u} := \begin{bmatrix} -f_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -f_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}$
  3. Sample $\begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{T}, \mathbf{u}, \sigma_1)$.
  4. Set $\mathbf{t} := \mathbf{G}\hat{\mathbf{t}}$.
  5. Return $(C := \mathbf{t}, \mathsf{st} := (\mathbf{s}_i)_{i \in [0,d]})$.

Open($\mathsf{crs}, C, \mathbf{f} \in \mathcal{R}_q^{d+1}, \mathsf{st}, \mathbf{c} \in \mathcal{R}_q^{d+1}$)
  1. Parse $\mathbf{f} := (f_0, f_1, \ldots, f_d)$ and $\mathbf{c} := (c_0, \ldots, c_d)$.
  2. Parse $C := \mathbf{t} \in \mathcal{R}_q^n$ and $\mathsf{st} := (\mathbf{s}_i)_{d \in [0,d]}$.
  3. Return 1 if and only if for all $i \in [0, d]$,
     • $\mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t}$.
     • $\|c_i\mathbf{s}_i\| \leq \gamma$.
```

Fig. 4. PowerBASIS commitment scheme for arbitrary messages in the message space $\mathcal{M} = \mathcal{R}_q^{d+1}$ with the slack space $\mathcal{S} := \{(c_0, \ldots, c_d) : \forall i \in [0, d], c_i \in \mathcal{R}_q^\times \wedge \|c_i\|_\infty \leq \beta_s\}$. Here, $\mathbf{G} \in \mathcal{R}_q^{n \times n\tilde{q}}$ is the gadget matrix of height n.

putes $\mathbf{R}_i := \mathbf{R}\mathbf{G}^{-1}(\mathbf{W}^{-i}\mathbf{G})$ for $i = 0, 1, \ldots, d$. Note that

$$\mathbf{W}^i\mathbf{A}\mathbf{R}_i = \mathbf{W}^i\mathbf{A}\mathbf{R}\mathbf{G}^{-1}(\mathbf{W}^{-i}\mathbf{G}) = \mathbf{W}^i\mathbf{G}\mathbf{G}^{-1}(\mathbf{W}^{-i}\mathbf{G}) = \mathbf{G}$$

and thus $\mathbf{R}_i$ is a $\mathbf{G}$-trapdoor for $\mathbf{W}^i\mathbf{A}$ and by Lemma 2.2:

$$\|\mathbf{R}_i\| \leq \|\mathbf{R}\| \cdot N\sqrt{nt} \leq \mathfrak{s}Nt\sqrt{2n(m-t)N}.$$

Then, the algorithm computes the $\mathsf{PowerBASIS}$ matrix along with its trapdoor:

$$\mathbf{B}:=\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \quad \tilde{\mathbf{R}}:=\left[\begin{array}{ccc} \mathbf{R}_0 & & \\ & \ddots & \\ & & \mathbf{R}_d \\ \hline & \mathbf{0} & \end{array}\right] . \tag{13}$$

Indeed, one can check that $\mathbf{B}\tilde{\mathbf{R}} = \mathbf{G}_{n(d+1)}$ and $\|\tilde{\mathbf{R}}\| \leq \mathfrak{s}Nt\sqrt{2(d+1)n(m-t)N}$. Finally, the setup algorithm re-randomises the trapdoor $\tilde{\mathbf{R}}$ by running

$$\mathbf{T} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \tilde{\mathbf{R}}, \mathbf{G}_{n(d+1)}, \sigma_0) \ ,$$

and thus $\mathbf{BT} = \mathbf{G}_{n(d+1)}$. Finally, the public parameters $\mathsf{crs}:=(\mathbf{A}, \mathbf{W}, \mathbf{T})$ are returned.

Suppose we want to commit to a vector $(f_0, f_1, \ldots, f_d)$ of length $d+1$. To this end, we use crs to compute

$$\begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \begin{bmatrix} -f_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -f_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}, \mathbf{T}, \sigma_1 \right) .$$

By definition, this means that $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_d \in \mathcal{R}_q^m$ and $\mathbf{t}:=\mathbf{G}\hat{\mathbf{t}}$ satisfy:

$$\mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \quad \text{for } i = 0, 1, \ldots, d \ . \tag{14}$$

The commitment and the decommitment state are $C:=\mathbf{t}$ and $\mathsf{st}:=(\mathbf{s}_i)_{i\in[0,d]}$.

Finally, the opening function takes the public parameters crs, the commitment $\mathbf{t}$, a message vector $\mathbf{f}:=(f_0, \ldots, f_d)$, the decommitment state $(\mathbf{s}_i)_{i\in[0,d]}$ and a relaxation factor $(c_0, \ldots, c_d) \in \mathcal{S}$, and accepts if and only if (14) holds and $\|c_i\mathbf{s}_i\| \leq \gamma$ for all $i = 0, 1, \ldots, d$.

4.1. *Security Analysis*

In the following, we show that the $\mathsf{PowerBASIS}$ commitment scheme satisfies completeness, relaxed binding and hiding. As before, we assume $q \equiv 5 \pmod 8$ is a prime.

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

$$\sigma_0 \geq \delta\mathfrak{s}Nt\omega(\sqrt{2(d+1)n(m-t)N\log t'N}) \quad \textit{and} \quad \sigma_1 \geq \delta\sigma_0 N \cdot \omega(\sqrt{m'n'}\log t'N) \ .$$

If $\gamma \geq \sigma_1\sqrt{m'N}$ then the $\mathsf{PowerBASIS}$ commitment scheme satisfies completeness.

Proof. In the discussion above, we already showed that Eq. (14) is true. We will show that $\|\mathbf{s}_i\| \leq \gamma$ for all i, and thus we can pick the global relaxation to be $(1, \ldots, 1) \in \mathcal{S}$.

First, note that the matrix $\tilde{\mathbf{R}} \in \mathcal{R}_q^{m' \times n'}$ satisfies $\|\tilde{\mathbf{R}}\| \leq \mathfrak{s} N t \sqrt{2(d+1)n(m-t)N}$ with high probability by Lemma 2.8. Hence $\sigma_0 \geq \delta \|\tilde{\mathbf{R}}\| \cdot \omega(\sqrt{N \log t'N})$ for $t' = \max(n', m')$ and thus we can apply both Lemma 2.16 and Lemma 2.7 to deduce that with an overwhelming probability $\|\mathbf{T}\| \leq \sigma_0 \sqrt{m'n'N}$. Similarly, we have $\sigma_1 \geq \delta \|\mathbf{T}\| \cdot \omega(\sqrt{N \log t'N})$ and thus $\|\mathbf{s}_i\| \leq \sigma_1 \sqrt{m'N} \leq \gamma$ with an overwhelming probability for all $i = 0, 1, \ldots, d$, which concludes the proof. □

Based on the parameters above, we would require $\sigma_0 = \tilde{O}(\sqrt{d})$ and $\sigma_1 = \tilde{O}(d^{3/2})$, ignoring the polynomial factors related to the security parameter.

Lemma 4.2. (Relaxed Binding). *Let* $t = n\tilde{q}$, $m > t + n$ *and* $n' = n\tilde{q}(d+1)$. *Take* $\mathfrak{s} > 2N \cdot q^{\frac{n}{m-t} + \frac{2}{N(m-t)}}$. *If* $\sigma_0 \geq \delta \mathfrak{s} N t \omega(\sqrt{2(d+1)n(m-t)N \log n'N})$ *then under the* $\mathsf{PowerBASIS}_{n-1,m,N,q,d+1,\sigma_0,2\beta_s\gamma}$ *assumption,* $\mathsf{PowerBASIS}$ *commitment scheme satisfies relaxed binding.*

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

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

Game 1: Here, we swap the $\mathsf{SamplePre}$ algorithm with sampling truly from a discrete Gaussian distribution. Since $\sigma_0 \geq \delta \mathfrak{s} N t \omega(\sqrt{2(d+1)n(m-t)N \log n'N})$, 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 2.6, we deduce that $\varepsilon_2 \geq \varepsilon_1 - \mathsf{negl}(\lambda)$.

We claim that $\varepsilon_2 = \mathsf{negl}(\lambda)$ under the $\mathsf{PowerBASIS}$ assumption. First, by definition of the $\mathsf{PowerBASIS}$ assumption, our goal is to extract a short non-zero solution for the matrix $\mathbf{A}^*$, where

$$\mathbf{A} := \begin{bmatrix} \mathbf{a}^\top \\ \hline \mathbf{A}^* \end{bmatrix} .$$

Denote the tuple $\mathcal{A}$ outputs as:

$$\mathbf{t}, (\mathbf{f}, (\mathbf{v}_0 \ldots, \mathbf{v}_d), (c_0, \ldots, c_d)), (\mathbf{f}', (\mathbf{v}'_0 \ldots, \mathbf{v}'_d), (c'_0, \ldots, c'_d)).$$

By definition, whenever $\mathcal{A}$ wins, it must be that openings are valid and $\mathbf{f} \neq \mathbf{f}'$, which implies there is at least an index j with $f_j \neq f'_j$. Thus, by subtracting the verification equations, we have that

$$\mathbf{A}(\mathbf{v}_j - \mathbf{v}'_j) = \begin{bmatrix} f'_j - f_j \\ 0 \\ \vdots \\ 0 \end{bmatrix} .$$

Since $f'_j - f_j \neq 0$, this implies that $\bar{\mathbf{v}}:=(\mathbf{v}_j - \mathbf{v}'_j) \neq \mathbf{0}$. Consequently, $\mathbf{A}^*\bar{\mathbf{v}} = \mathbf{0}$. Now, $\bar{\mathbf{v}}$ might not be short. Hence, we consider $c_j c'_j \bar{\mathbf{v}}$ instead. Clearly, this is still a non-zero solution for $\mathbf{A}^*$ since c_j, c'_j are invertible. Further,

$$\|c_j c'_j \bar{\mathbf{v}}\| \leq \|c'_j(c_j\mathbf{v})\| + \|c_j(c'_j\mathbf{v}')\| \leq 2\beta_s\gamma \ .$$

Therefore, $c_j c'_j \bar{\mathbf{v}}$ is a valid solution to $\mathsf{PowerBASIS}$. □

Lemma 4.3. (Hiding). *Suppose $n, N \geq 1$ and denote $t:=n\tilde{q}$. Let $m > t+n$, $m':=m(d+1)+n\tilde{q}$, $n':=n\tilde{q}(d+1)$ and $t':=\max(n',m')$. Take*

$$\sigma_0 \geq \delta\mathfrak{s}Nt\omega(\sqrt{2(d+1)n(m-t)N\log t'N}),$$
$$\sigma_1 \geq \delta \cdot \max\left(\log((d+1)mN), \sigma_0 N \cdot \omega(\sqrt{m'n'\log t'N})\right).$$

Then, the $\mathsf{PowerBASIS}$ *commitment scheme satisfies hiding.*

Proof. Take an unbounded adversary $\mathcal{A}$ which wins the hiding game with probability ϵ. We prove the statement via a sequence of games, where in each game we change the algorithm of Commit. Let ϵ_i be the advantage of the adversary against $\mathsf{Game}\ i$.

Game 1: This is the original hiding game where Commit is defined in Fig. 4. For the purpose of the proof, we assume Commit does not output st. Then, by definition $\epsilon_1 = \epsilon$.

Game 2: In this game, Commit (inefficiently) samples

$$\begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \leftarrow \mathbf{B}_{\sigma_1}^{-1}\left(\begin{bmatrix} -f_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -f_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}\right)$$

and outputs $\mathbf{t}:=\mathbf{G}\hat{\mathbf{t}}$. By our assumption on σ_0, σ_1 we can argue similarly as in Lemma 4.1 to deduce that $|\epsilon_2 - \epsilon_1| = \mathsf{negl}(\lambda)$.

Game 3: Here we make use of the fact that $\mathbf{B}:=[\mathbf{E} \mid \mathbf{F}]$ where

$$\mathbf{E}:=\begin{bmatrix} \mathbf{A} & & \\ & \ddots & \\ & & \mathbf{W}^d\mathbf{A} \end{bmatrix} \quad \text{and} \quad \mathbf{F}:=\begin{bmatrix} -\mathbf{G} \\ \vdots \\ -\mathbf{G} \end{bmatrix}$$

Concretely, the Commit algorithm first samples $\hat{\mathbf{t}} \leftarrow \mathcal{D}_{\sigma_1}^{tN}$, sets

$$\begin{bmatrix} \mathbf{t} \\ \vdots \\ \mathbf{t} \end{bmatrix} :=\mathbf{F}\hat{\mathbf{t}}$$

and then generates

$$\begin{bmatrix}\mathbf{s}_1\\ \vdots \\ \mathbf{s}_d\end{bmatrix} \leftarrow \mathbf{E}_{\sigma_1}^{-1}\left(\begin{bmatrix}-f_0\mathbf{W}^0\mathbf{e}_1\\ \vdots \\ -f_d\mathbf{W}^d\mathbf{e}_1\end{bmatrix} - \begin{bmatrix}\mathbf{t}\\ \vdots \\ \mathbf{t}\end{bmatrix}\right).$$

Finally, the algorithm outputs $\mathbf{t}$.

By Lemma 2.5, there is a negligible function ε such that $\sigma_1 \geq \eta_\varepsilon(\Lambda^\perp(\mathbf{E}))$. Further, by Lemma 2.3 the matrix $\mathbf{A}$ is full-rank (and so is $\mathbf{E}$) with an overwhelming probability. Hence, we can apply Lemma 2.9 to conclude $|\epsilon_3 - \epsilon_2| = \mathsf{negl}(\lambda)$.

Game 4: The Commit algorithm simply samples $\hat{\mathbf{t}} \leftarrow \mathcal{D}_{\sigma_1}^{tN}$ and outputs $\mathbf{t}:=\mathbf{G}\hat{\mathbf{t}}$. Clearly, there is no difference between the outputs of Game 3 and 4, thus $\epsilon_4 = \epsilon_3$.

Finally, the output of Commit in Game 4 does not depend on the challenge messages m_0, m_1 from $\mathcal{A}$. Hence, we get that $\epsilon_4 = 1/2$. By the hybrid argument we obtain $\epsilon = 1/2 + \mathsf{negl}(\lambda)$, which concludes the proof. □

Efficiency. The main bottleneck of the Commit algorithm is the trapdoor sampling procedure, which asymptotically takes $O(d^2)$ operations over $\mathcal{R}_q$. On the other hand, the opening algorithm makes $O(d)$ operations in $\mathcal{R}_q$.

Remark 4.4. Wee and Wu [83] proposed an alternative approach, which allows for linear-time commitment generation. This comes at the cost of (i) losing the hiding property, and (ii) the message space inherently must only contain short vectors. Since both properties are important in our polynomial commitment scheme, we do not describe the more efficient method in this work and refer to [83, Remark 4.12] for more details.

5. Efficient Proofs of Polynomial Evaluation

In this section we illustrate how to prove evaluations of a polynomial that is committed using the $\mathsf{PowerBASIS}$ commitment scheme from Fig. 4. We start by presenting a general framework for proving polynomial evaluations in Sect. 5.1, and then we describe two distinct instantiations in Sects. 5.2 and 5.3. For clarity, we give an overview of frequently used parameters in Table 3. We implicitly assume that lattice dimension parameters, such as n, m, N, are $\mathsf{poly}(\lambda)$.

5.1. *Framework for Proving Evaluations*

The main intuition can be described as follows. We design a relation that captures statements of the form: “the commitment $\mathbf{t}$ has an opening $f \in \mathcal{R}_q^{d+1}$ (with respect to a given crs) such that $f(u) = v$, where $f \in \mathcal{R}_q^{\leq d}[\mathsf{X}]$ is now interpreted as polynomial”. The core observation is that there exists a Σ-protocol that interactively reduces an instance of that relation to a related one, in which the size of the committed polynomial is decreased. This new relation is with respect to a *different* common reference string, that can be *efficiently computed* from the previous one. We then exploit this recursion to shrink to a commitment with a constant-size opening.

Table 3. Overview of parameters and notation .

Parameter	Explanation
q	proof system modulus
N	degree of the cyclotomic ring $\mathcal{R}:=\mathbb{Z}[X]/(X^N+1)$
l	power-of-two such that $q \equiv 2N/l+1 \pmod{4N/l}$
d	degree of the committed polynomial $f \in \mathcal{R}_q[\mathsf{X}]$
n	height of the matrix $\mathbf{A}$
m	width of the matrix $\mathbf{A}$
δ	decomposition base of the gadget matrix $\mathbf{G}$
$\tilde{q}$	$\lfloor \log_\delta q \rfloor + 1$
n'	$n\tilde{q}(d+1)$
m'	$m\tilde{q}(d+1)+n\tilde{q}$
t'	$\max(n', m')$
k	folding factor of the folding protocol
h	$2h+1$ is the number of rounds
β	initial norm of the witness openings
w	L_1 norm of elements in the challenge space $\mathcal{C}$
$\beta_\mathcal{C}$	L_∞ of elements in $\mathcal{C}$ (used in Sect. 5.3)
β_h	norm of the opening vectors sent in the last round
β_s	infinity norm of the extracted relaxation factors
γ	extracted norm

We formalise this discussion by introducing the opening relation below

$$\mathsf{R}_{d,\beta}:=\left\{((\mathbf{A},\mathbf{W}),(\mathbf{t},u,z),(f,(\mathbf{s}_i)_i)) \middle| \begin{array}{c} f(u)=z \\ \forall i \in [0,d], \mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \\ \wedge \|\mathbf{s}_i\| \le \beta \end{array} \right\} . \quad (15)$$

We describe the Σ-protocol, upon which our main evaluation protocol is built, in Fig. 5. Roughly speaking, the prover divides the initial polynomial f of degree at most d into k polynomials $g_1, \ldots, g_k$ of degree at most $d':=(d+1)/k-1$ by writing

$$f(\mathsf{X}):=\sum_{t\in[k]} \mathsf{X}^{t-1} g_t(\mathsf{X}^k) . \quad (16)$$

Then, it "commits" to the partial polynomials by providing their evaluations at the point u, say $z_i:=g_i(u^k)$. Thus, by construction

$$z=f(u)=\sum_{t\in[k]} u^{t-1} g_t(u^k) = \sum_{t=1}^{k} z_t u^{t-1} . \quad (17)$$

Next, the verifier outputs a challenge $(\alpha_1,\ldots,\alpha_k) \leftarrow \mathcal{C} \subseteq \mathcal{R}_q^k$. Note that by considering the folded polynomial $g=\sum_{t=1}^{k} \alpha_t g_t$ of degree at most d', we obtain a new polynomial

Σ-Protocol for $\mathsf{R}_{d,\beta}$

Prover **Verifier**

$\sum_{t\in[k]} \mathsf{X}^{t-1} g_t(\mathsf{X}^k) =: f(\mathsf{X})$

$z_t := g_t(u^k)$ for $t \in [k]$

$$\xrightarrow{(z_t)_{t\in[k]}}$$

$\boldsymbol{\alpha} \leftarrow \mathcal{C} \subseteq \mathcal{R}_q^k$

$$\xleftarrow{\boldsymbol{\alpha}}$$

$g := \sum_{t\in[k]} \alpha_t g_t$

$\mathbf{z}_i := \sum_{t\in[k]} \alpha_t \mathbf{s}_{t,i}$ for $i \in [0, d']$

$$\xrightarrow{g, (\mathbf{z}_i)_{i\in[0,d']}}$$

$\beta' := \mathrm{w}\,\beta$

$\mathbf{t}' := \left(\sum_{t\in[k]} \alpha_t \mathbf{W}^{-(t-1)}\right) \cdot \mathbf{t}$

$\mathbb{i}' := (\mathbf{A}, \mathbf{W}^k)$

$\mathbb{x}' := \left(\mathbf{t}', u^k, \sum_{t\in[k]} \alpha_t z_t\right)$

$\mathbb{w}' := (g, (\mathbf{z}_i)_{i\in[0,d']})$

Check:

$z = \sum_{t\in[k]} u^{t-1} z_t$

$(\mathbb{i}', \mathbb{x}', \mathbb{w}') \in \mathsf{R}_{d',\beta'}$

Fig. 5. The Σ-protocol $\Sigma[d, k, \mathcal{C}, \beta]$ for relation $\mathsf{R}_{d,\beta}$ in Equation (15). Here, $d' := (d+1)/k - 1$, $\mathrm{w} := \max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\boldsymbol{\alpha}\|_1$ and $\mathbf{s}_{t,i} := \mathbf{s}_{ki+t-1}$ for $i \in [0, d']$ and $t \in [k]$.

evaluation statement about g:

$$g(u^k) = \sum_{t=1}^{k} \alpha_t z_t \ . \tag{18}$$

The main strength of the $\mathsf{PowerBASIS}$ commitment from Fig. 4 is that the commitment (resp. openings) to g can be efficiently computed from the commitment $\mathbf{t}$ (resp. openings $\mathbf{s}_i$) of f given $\alpha_1, \ldots, \alpha_k$ in time $O(k)$. This is the key idea for achieving succinct verification. Hence, the prover outputs the polynomial g in the clear, along with its opening vectors. The verifier eventually checks correctness of the openings with respect to the message g, as well as (17) and (18).

We first prove that this protocol transforms an instance of $\mathsf{R}_{d,\beta}$ into a smaller one of $\mathsf{R}_{d',\beta'}$.

Lemma 5.1. (Completeness for Σ). *Let* $\Pi{:=}\Sigma[d, k, \mathcal{C}, \beta]$ *as in Fig. 5. Then,* Π *is an interactive protocol with perfect completeness for* $\mathsf{R}_{d,\beta}$.

Proof. Let $(\mathbb{i}, \mathbb{x}, \mathbb{w}) = ((\mathbf{A}, \mathbf{W}), (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_{i\in[0,d]})) \in \mathsf{R}_{d,\beta}$. Since $f(u) = z$, the first verification check always succeeds by Equation (17). We are left to show that the new instance is valid. First, $g(u^k) = \sum_{t\in[k]} \alpha_t g_t(u^k) = \sum_{t\in[k]} \alpha_t z_t$. Further, recall that for $i \in [0, d']$ and $t \in [k]$ we have

$$\mathbf{s}_{t,i} = \mathbf{s}_{ki+t-1} \quad \text{and} \quad g_{t,i} = f_{ki+t-1} \ ,$$

where $g_{t,i}$ is the i-th coefficient of the polynomial g_t. Hence, the i-th coefficient of g satisfies $g_i = \sum_{t\in[k]} \alpha_t g_{t,i} = \sum_{t\in[k]} \alpha_t f_{ki+t-1}$. Therefore,

$$\begin{aligned}
\mathbf{A}\mathbf{z}_i + g_i\mathbf{e}_1 &= \mathbf{A}\left(\sum_{t\in[k]} \alpha_t \mathbf{s}_{t,i}\right) + \left(\sum_{t\in[k]} \alpha_i f_{ki+t-1}\right) \cdot \mathbf{e}_1 \\
&= \sum_{t\in[k]} \alpha_t \left(\mathbf{A}\mathbf{s}_{ki+t-1} + f_{ki+t-1}\mathbf{e}_1\right) \\
&= \sum_{t\in[k]} \alpha_t \left(\mathbf{W}^{-(ki+t-1)}\mathbf{t}\right) \\
&= \left(\sum_{t\in[k]} \alpha_t \mathbf{W}^{-(ki+t-1)}\right) \cdot \mathbf{t} \\
&= (\mathbf{W}^k)^{-i} \left(\sum_{t\in[k]} \alpha_t \mathbf{W}^{-(t-1)}\right) \cdot \mathbf{t}.
\end{aligned}$$

Finally, by Lemma 2.1 for $\boldsymbol{\alpha} \in \mathcal{C}$, $\|\mathbf{z}_i\| \leq \sum_{t\in[k]} \left\|\alpha_t \mathbf{s}_{t,i}\right\| \leq \sum_{t\in[k]} \|\alpha_t\|_1 \cdot \beta \leq \mathrm{w}\beta$ where $\mathrm{w}{:=}\max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\alpha\|_1$. This shows that the new instance is in $\mathsf{R}_{d',\beta'}$, and thus the verifier accepts. □

We now apply the Σ-protocol recursively h times, reducing the final opening size to $(d + 1)/k^h$, while increasing the final norm for verification by a factor w^h.

Construction 5.2. Let k, h be integers, and let $\mathcal{C} \subseteq \mathcal{R}_q^k$. We let $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]{:=} (\mathcal{P}, \mathcal{V})$ be the protocol that we describe in Fig. 6.

Completeness of the protocol is easily shown by applying Lemma 5.1 h times.

Lemma 5.3. (Completeness for Eval). *Let* $\Pi{:=}\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$. *Then,* Π *is an interactive protocol with perfect completeness for* $\mathsf{R}_{d,\beta}$.

Proof. Denote by $(\mathbb{i}_r, \mathbb{x}_r, \mathbb{w}_r){:=}((\mathbf{A}, \mathbf{W}_r), (\mathbf{t}_r, u_r, z_r), (f_r, (\mathbf{s}_{r,i})_{i\in[d_r]}))$ for $r \in [h]$. By Lemma 5.1, $(\mathbb{i}_r, \mathbb{x}_r, \mathbb{w}_r) \in \mathsf{R}_{d_r,\beta_r}$ implies $(\mathbb{i}_{r+1}, \mathbb{x}_{r+1}, \mathbb{w}_{r+1}) \in \mathsf{R}_{d_{r+1},\beta_{r+1}}$ with

Interactive Protocol for $\mathsf{R}_{d,\beta}$

```
$\mathcal{P}((\mathbf{A},\mathbf{W}),(\mathbf{t},u_0,z_0),(f_0,(\mathbf{s}_{0,i})_{i\in[0,d]}))$
1. Set $d_0 := d$.
2. For $r \in [h]$:
    (a) Set $d_r := (d_{r-1}+1)/k - 1$.
    (b) Write $f_{r-1}(\mathsf{X}) := \sum_{t\in[k]} \mathsf{X}^{t-1} f_{r-1,t}(\mathsf{X}^k)$ for $f_{r-1,1},\dots f_{r-1,k} \in \mathcal{R}_q^{\leq d_r}[\mathsf{X}]$.
    (c) Set $z_{r-1,t} := f_{r-1,t}(u_{r-1}^k)$ for $t \in [k]$.
    (d) Send $(z_{r-1,t})_{t\in[k]}$ to the verifier.
    (e) Receive $\boldsymbol{\alpha}_r$ from the verifier.
    (f) Compute $f_r := \sum_{t\in[k]} \alpha_{r,t} f_{r-1,t}$.
    (g) Compute $\mathbf{s}_{r,i} := \sum_{t\in[k]} \alpha_{r,t}\mathbf{s}_{r-1,ki+t-1}$ for $i \in [0,d_r]$.
    (h) Compute $u_r := u_{r-1}^k$.
3. Send $(f_h,(\mathbf{s}_{h,i})_{i\in[0,d_h]})$ to the verifier.

$\mathcal{V}((\mathbf{A},\mathbf{W}_0),(\mathbf{t}_0,u_0,z_0))$
1. $\beta_0 := \beta$.
2. For $r \in [h]$:
    (a) Receive $(z_{r-1,t})_{t\in[k]}$ from the prover.
    (b) Check $z_{r-1} = \sum_{t\in[k]} u_{r-1}^{t-1} z_{r-1,t}$.
    (c) Sample $\boldsymbol{\alpha}_r \leftarrow \mathcal{C}$ and send it to the prover.
    (d) Set $\mathbf{W}_r := \mathbf{W}_{r-1}^k$.
    (e) Set $\mathbf{t}_r := \left(\sum_{t\in[k]} \alpha_{r,t}\mathbf{W}_{r-1}^{-(t-1)}\right)\cdot \mathbf{t}_{r-1}$.
    (f) Set $\beta_r := \mathrm{w}\cdot\beta_{r-1}$.
    (g) Set $u_r := u_{r-1}^k$.
    (h) Set $z_r := \sum_{t\in[k]} \alpha_{r,t} z_{r-1,t}$.
3. Receive $(f_h,(\mathbf{s}_{h,i})_{i\in[0,d_h]})$ from the prover.
4. Check:
    (a) $f_h(u_h) = z_h$.
    (b) $\mathbf{A}\mathbf{s}_{h,i} + f_{h,i}\mathbf{e}_1 = \mathbf{W}_h^{-i}\mathbf{t}_h$ for $i \in [0,d_h]$.
    (c) $\|\mathbf{s}_{h,i}\| \leq \beta_h$ for $i \in [0,d_h]$.
```

Fig. 6. The protocol $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ for $\mathsf{R}_{d,\beta}$. As before, we denote $\mathrm{w}:= \max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\boldsymbol{\alpha}\|_1$.

probability 1. Since $(\mathbb{i}_0, \mathbb{x}_0, \mathbb{w}_0) \in \mathsf{R}_{d,\beta_0}$, then $(\mathbb{i}_h, \mathbb{x}_h, \mathbb{w}_h) \in \mathsf{R}_{d_h,\beta_h}$, and thus the verifier final checks accept. □

Remark 5.4. The protocol that we have described has folding factor k constant across every round of interaction. In fact, we can gain more flexibility by allowing each round to use a different folding factor. This can be beneficial, for example, to obtain a constant polynomial in the last round of the protocol when the original degree is not a h-power.

We analyse the communication complexity of $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ in the next lemma.

Lemma 5.5. (Efficiency for Eval). *The total communication complexity of* $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ *(in bits) can be bounded by*

$$h \cdot (kN\lceil \log q\rceil + \lceil \log |\mathcal{C}|\rceil) + \frac{d+1}{k^h} N\lceil \log q\rceil + \left(\frac{d+1}{k^h} + 1\right) mN\lceil \log(2w^h\beta)\rceil\ .$$

Further, the prover makes $O(md)$ operations in $\mathcal{R}_q$ while the verifier makes $O\left((n+m)^2 (hk + d/k^h)\right)$ operations in $\mathcal{R}_q$.

Proof. In each round the prover sends k elements of $\mathcal{R}_q$ to the verifier, and the verifier sends 1 element of $\mathcal{C}$. In the final round, the prover sends a polynomial with $d_h = (d+1)/k^h$ coefficients, and $d_h + 1$ opening vectors, each of which has norm at most β_h.

We turn to the prover complexity and first consider Step 2. Every r-th round out of $[h]$, the prover makes $O(mkd_r) = O(md_{r-1})$ operations in $\mathcal{R}_q$. Since $d_0 = O(d)$ and in general $d_r = O(d/k^r)$, the total runtime of the prover can be bounded by

$$O\left(\sum_{r=0}^{h-1} md_r\right) = O\left(m \sum_{r=0}^{h-1} d/k^r\right) = O\left(md \cdot \frac{1-1/k^h}{1-1/k}\right) = O(md) \ .$$

We move to the verifier analysis. In Step 2, for every round $r \in [h]$, the verifier makes at most $O(kn^2)$ operations. Hence, the total cost of Step 2 is $O(hkn^2)$. The rest of the algorithm takes $O(d_h(nm + n^2))$ steps. Thus, the total runtime can be bounded by $O\left((n+m)^2(hk + d/k^h)\right)$ ring operations. □

Next, we provide two instantiations of the protocol in Fig. 6 which will differ in the selection of the challenge space $\mathcal{C}$. This has direct impact on the knowledge extraction strategy.

5.2. *Monomial Protocol*

In the following, we describe a so-called *monomial* variant of the protocol, where the name comes from the description of the challenge space $\mathcal{C}$. Fix $k{:=}2$, and $\mathcal{C}{:=}\{1\} \times \{X^i : i \in \mathbb{Z}\}$. Note that by definition $\mathrm{w} = 2$, and $\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathcal{C}$ with $\boldsymbol{\alpha} \neq \boldsymbol{\alpha}'$ implies that $\alpha_2 - \alpha_2' \in \mathcal{R}_q^\times$. In this section, we also assume that $2 \in \mathcal{R}_q^\times$ (which can be enforced if $\gcd(2, q) = 1$).

We aim to show that $\Pi{:=}\mathsf{Eval}[d, 2, h, \mathcal{C}, \beta]$ is 2-special sound. In fact, we will not be able to show this *exactly*, as the extraction will introduce some slack. Rather we show that Π is special sound for the *relaxed* opening relation that we describe next:

$$\tilde{\mathsf{R}}_{d,c,\gamma}{:=}\left\{ \left((\mathbf{A}, \mathbf{W}), (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_{i\in[0,d]})\right) \middle| \begin{array}{c} \forall i \in [0,d], \mathbf{A}\mathbf{s}_i + f_i \mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \wedge \\ \wedge\, c \in \mathcal{R}_q^\times \wedge\ \|c \cdot \mathbf{s}_i\| \leq \gamma \\ \wedge\ f(u) = z \end{array} \right\} . \tag{19}$$

We will directly show that Eval is special sound, which also implies special soundness of the Σ-protocol by noting that the two protocols are equivalent when $h = 1$. To argue soundness we will first prove that there exists an extractor that is able to extract witnesses of the higher layer of the transcript tree from the children.

Lemma 5.6. (Special Soundness for Σ). *Let $c \in \mathcal{R}_q^\times$, and let $\mathbb{i} = (\mathbf{A}, \mathbf{W})$, $\mathbb{x} = (\mathbf{t}, u, z)$. There exists an algorithm that, given two accepting transcripts tr_j of the fol-*

lowing form

$$\mathsf{tr}_j := \Big((z_1, z_2), \boldsymbol{\alpha}_j := (1, \alpha_j) \in \mathcal{C}, \mathbb{w}'_j := (g_j, (\mathbf{z}_{j,i})_i)\Big) \quad \textit{for } j = 0, 1$$

where $\alpha_0 \neq \alpha_1$*, outputs* $\mathbb{w} := (\bar{f}, (\bar{\mathbf{s}}_i)_i)$*. Furthermore, let* d', $\mathbb{i}'$, $\mathbb{x}'_0$, $\mathbb{x}'_1$ *be obtained as in Fig. 5. If, for* $i \in \{0, 1\}$, $(\mathbb{i}', \mathbb{x}'_i, \mathbb{w}'_i), \in \tilde{\mathsf{R}}_{d',c,\beta}$*, and* $z = z_1 + u z_2$*, then* $(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \tilde{\mathsf{R}}_{d,2c,\gamma}$ *where* $\gamma := 2N\beta$*.*

Remark 5.7. We highlight that the g_j contained $\mathbb{w}'_j$ from the statement above are *different* from polynomials g_1, g_2 defined in Fig. 5.

Proof. Consider the following algorithm:

```
$\underline{\mathcal{E}(\mathsf{tr}_0, \mathsf{tr}_1)}$:
  1. Set $\bar{\mathbf{s}}_{2i} := \frac{\alpha_1 \mathbf{z}_{0,i} - \alpha_0 \mathbf{z}_{1,i}}{\alpha_1 - \alpha_0}$, $\bar{\mathbf{s}}_{2i+1} := \frac{\mathbf{z}_{0,i} - \mathbf{z}_{1,i}}{\alpha_0 - \alpha_1}$ for $i \in [0, (d-1)/2]$.
  2. Set $\bar{f}_1 := \frac{\alpha_1 g_0 - \alpha_0 g_1}{\alpha_1 - \alpha_0}$, $\bar{f}_2 := \frac{g_0 - g_1}{\alpha_0 - \alpha_1}$.
  3. Set $\bar{f} := \bar{f}_1(\mathsf{X}^2) + \mathsf{X}\bar{f}_2(\mathsf{X}^2)$.
  4. Return $\bar{f}, (\bar{\mathbf{s}}_i)_{i \in [0,d]}$.
```

Let now $(\bar{f}, (\bar{\mathbf{s}}_i)_i) \leftarrow \mathcal{E}(\mathsf{tr})$. Note that

$$\begin{aligned}\mathbf{A}\bar{\mathbf{s}}_{2i} + \bar{f}_{2i}\mathbf{e}_1 &= \mathbf{W}^{-2i}\mathbf{t} \\ \mathbf{A}\bar{\mathbf{s}}_{2i+1} + \bar{f}_{2i+1}\mathbf{e}_1 &= \mathbf{W}^{-(2i+1)}\mathbf{t} \ .\end{aligned}$$

Now, we have that:

$$\begin{aligned}\bar{f}(u) &= \bar{f}_1(u^2) + u\bar{f}_2(u^2) \\ &= \frac{\alpha_1 g_0(u^2) - \alpha_0 g_1(u^2)}{\alpha_1 - \alpha_0} + u\frac{g_0(u^2) - g_1(u^2)}{\alpha_0 - \alpha_1} \\ &= z_1 + u z_2 \\ &= z \ .\end{aligned}$$

Finally, we set $c^* := 2c$. First, note that $c^* \in \mathcal{R}_q^\times$ since $2 \in \mathcal{R}_q^\times$. Now, for $i \in [0, d']$, we have:

$$\begin{aligned}\left\|c^* \cdot \bar{\mathbf{s}}_{2i}\right\| &= \left\|\frac{2}{\alpha_1 - \alpha_0} \cdot c \cdot (\alpha_1 \mathbf{z}_{0,i} - \alpha_0 \mathbf{z}_{1,i})\right\| \\ &\leq \left\|\frac{2}{\alpha_1 - \alpha_0}\right\|_\infty \left\|c(\alpha_1 \mathbf{z}_{0,i} - \alpha_0 \mathbf{z}_{1,i})\right\|_1 \\ &= \left\|c(\alpha_1 \mathbf{z}_{0,i} - \alpha_0 \mathbf{z}_{1,i})\right\|_1 \\ &\leq \sqrt{N}(c\left\|\alpha_1 \mathbf{z}_{0,i}\right\| + \left\|c\alpha_0 \mathbf{z}_{1,i}\right\|) \\ &\leq N(\|\alpha_1\| \cdot \left\|c\mathbf{z}_{0,i}\right\| + \|\alpha_0\| \cdot \left\|c\mathbf{z}_{1,i}\right\|) \\ &\leq 2N\beta = \gamma\end{aligned}$$

where the second equality follows by Lemma 2.17 and the last inequality by $\|\alpha\| = 1$ for $(1, \alpha) \in \mathcal{C}$. Similarly, $\|c^* \cdot \bar{\mathbf{s}}_{2i+1}\| \leq \gamma$. □

Using this extractor, we show that Π is $(2, \ldots, 2)$-special sound. The new extractor will start from the leaves of the tree of transcripts, applying the extractor described in Lemma 5.6 to obtain witnesses [16] for the upper layer.

Lemma 5.8. (Special Soundness for Eval). *Let* $\mathcal{C}{:=}\{1\} \times \{X^i : i \in \mathbb{Z}\}$ *and let* $\Pi{:=}\mathsf{Eval}[d, 2, h, \mathcal{C}, \beta]$ *be as in Construction 5.2. Set* $\gamma{:=}(2N)^h \cdot \beta_h$. *Then* Π *is a special sound proof system for* $\tilde{\mathsf{R}}_{d,2^h,\gamma}$.

Proof. Let tr be a tree of transcripts, which we index as follows.

- $\alpha_{(r,j)}$ for $(r, j) \in [h] \times [2^r]$ is the j-th challenge in the r-th layer of the transcript.
- $(z_{(r,j),1}, z_{(r,j),2})$ for $(r, j) \in [0, h-1] \times [2^r]$ is the j-th response in the r-th layer of the transcript.
- $(\bar{f}_{(h,j)}, (\bar{\mathbf{s}}_{(h,j),i})_i)$ for $j \in [2^h]$ is the final message sent by the prover.

We introduce the following notation as in the verifier algorithm:

- $d_0{:=}d$, $d_r{:=}d_{r-1}/2$ for $r \in [h]$
- $\mathbf{W}_0{:=}\mathbf{W}$, $\mathbf{W}_r{:=}\mathbf{W}_{r-1}^2$ for $r \in [h]$.
- $\mathbf{t}_{(0,1)}{:=}\mathbf{t}, \mathbf{t}_{(r,2j-1)}{:=}(1+\alpha_{(r,2j-1)}\mathbf{W}_{r-1}^{-1})\mathbf{t}_{(r-1,j)}, \mathbf{t}_{(r,2j)}{:=}(1+\alpha_{(r,2j)}\mathbf{W}_{r-1}^{-1})\mathbf{t}_{(r-1,j)}$ for $(r, j) \in [h] \times [2^r]$.
- $\beta_0{:=}\beta$, $\beta_r{:=}2N \cdot \beta_{r-1}$ for $r \in [h]$.
- $u_0{:=}u$, $u_r{:=}u_{r-1}^2$ for $r \in [h]$.
- $z_{(r,2j-1)}{:=}z_{(r-1,j),1} + \alpha_{(r,2j-1)}z_{(r-1,j),2}$, $z_{(r,2j)}{:=}z_{(r-1,j),1} + \alpha_{(r,2j)}z_{(r-1,j),2}$ for $(r, j) \in [h] \times [2^r - 1]$.

Denote with $\mathcal{E}^{(1)}$ the extractor of Lemma 5.6.

```
$\underline{\mathcal{E}(\mathsf{tr})}$:
  1. Set $d_0{:=}d$, $d_r{:=}d_{r-1}/2$ for $r \in [h]$.
  2. For $r{:=}h, \ldots, 1$:
     (a) Set, for $j \in [2^{r-1}]$,
         $\mathsf{tr}_{(r-1,j)}{:=}\left((z_{(r-1,j),1}, z_{(r-1,j),2}), \begin{matrix} \alpha_{(r,2j-1)}, (\bar{f}_{(r,2j-1)}, (\bar{\mathbf{s}}_{(r,2j-1),i})_i) \\ \alpha_{(r,2j)}, (\bar{f}_{(r,2j)}, (\bar{\mathbf{s}}_{(r,2j),i})_i) \end{matrix}\right)$.
     (b) Compute $\bar{f}_{(r-1,j)}, (\bar{\mathbf{s}}_{(r-1,j),i})_{i\in[0,d_{r-1}]} \leftarrow \mathcal{E}^{(1)}(\mathsf{tr}_{(r-1,j)})$ for $j \in [2^{r-1}]$
  3. Return $\bar{f}_{(0,1)}, (\bar{\mathbf{s}}_{(0,1),i})_{i\in[d]}$.
```

We prove that this extractor yields a valid witness by induction on r. First note that, by the verifier checks, for $(r, j) \in [h] \times [2^r]$

$$z_{(r-1,j)} = z_{(r-1,j),1} + u_{r-1}z_{(r-1,j),2} \ .$$

[16] We also implicitly collect the corresponding relaxation factors, which are *the same* across the same layer.

Table 4. Parameters for the interactive polynomial commitment scheme obtained from Fig. 4 and running the ℓ-parallel repetition of $\mathsf{Eval}[d, 2, h, \mathcal{C}, \beta]$ for proofs of evaluation .

Parameters	Instantiation
m	$\geq n(1+\tilde{q})$
l	$N/2$
δ	$q^{1/O(1)}$
$\mathfrak{s}$	$> 2Nq^{\frac{n}{m-n\tilde{q}}+\frac{2}{N(m-n\tilde{q})}}$
σ_0	$\geq \delta\mathfrak{s}Nn\tilde{q}\cdot\omega(\sqrt{2(d+1)n(m-n\tilde{q})N\log t'N})$
σ_1	$\geq \delta\sigma_0 N\cdot\omega(\sqrt{m'n'\log t'N})$
β	$\geq \sigma_1\sqrt{m'N}$
k	2
$\mathcal{C}$	$\{1\}\times\{X^i : i\in\mathbb{Z}\}$
w	2
β_h	$\mathrm{w}^h\cdot\beta$
γ	$(2N)^h\cdot\beta_h$
β_s	2^h
Soundness	$\left(\frac{h}{2N}\right)^\ell$
Commitment size	$nN\log q$
Communication complexity	$\ell\cdot\begin{pmatrix} h(2N\log q+\log N+1)+ \\ \frac{d+1}{2^h}(N\log q+mN\log\beta_h)\end{pmatrix}$
Prover time	$O(\ell\cdot md)$
Verifier time	$O(\ell\cdot(n+m)^2\cdot(2h+d/2^h))$

We compute the prover and verifier runtime in terms of operations in $\mathcal{R}_q$.

Write $\mathbb{i}_{(r,j)}:=(\mathbf{A},\mathbf{W}_r)$, $\mathbb{x}_{(r,j)}:=(\mathbf{t}_{(r,j)},u_{(r,j)},z_{(r,j)})$, $\mathbb{w}_{(r,j)}:=(\bar{f}_{(r,j)},(\bar{\mathbf{s}}_{(r,j),i})_i)$ for $(r,j)\in[h]\times[2^r]$. For $r=h$, since the transcripts are accepting, $(\mathbb{i}_{(h,j)},\mathbb{x}_{(h,j)},\mathbb{w}_{(h,j)})\in\mathsf{R}_{d_h,\beta_h}=\tilde{\mathsf{R}}_{d_h,1,\beta_h}$ for $j\in[2^h]$. Thus, by Lemma 5.6, $(\mathbb{i}_{(h-1,j)},\mathbb{x}_{(h-1,j)},\mathbb{w}_{(h-1,j)})\in\tilde{\mathsf{R}}_{d_{h-1},2,2N\beta_h}$.

We can continue with the induction, and this yields that for the extracted witness $\mathbb{w}_{(0,1)}:=(\bar{f}_{(0,1)},(\bar{\mathbf{s}}_{(0,1),i})_{i\in[d]})$ we have that:

$$(\mathbb{i}_{(0,1)},\mathbb{x}_{(0,1)},\mathbb{w}_{(0,1)})\in\tilde{\mathsf{R}}_{d,2^h,(2N)^h\beta_h}\ .$$

Setting $\gamma:=(2N)^h\beta_h$, and noting that $2^h\in\mathcal{R}_q^\times$, this concludes our proof. □

We can use Eval to construct a polynomial commitment scheme. We detail the construction in Theorem 5.9 and summarise the parameters and efficiency features in Table 4.

Theorem 5.9. *Let* $\mathsf{PC}=(\mathsf{Setup},\mathsf{Commit},\mathsf{Open},\mathcal{P}^\ell,\mathcal{V}^\ell)$ *where* Setup, Commit, Open *are as in Fig. 4 and* $\mathcal{P}^\ell,\mathcal{V}^\ell$ *are the* ℓ*-parallel repetitions of the prover and verifier of* Eval. *Then* PC *is an interactive polynomial commitment scheme with the efficiency properties and parameters shown in Table 4. In particular, when* $h=O(\log d)$ *and* $\ell>\frac{\lambda}{\log N+1-\log h}$ *we obtain an interactive polynomial commitment scheme with negligible knowledge soundness error, polylogarithmic communication complexity, and polylogarithmic verifier time.*

Proof. Completeness and relaxed binding follow from Lemmata 4.1 and 4.2. Perfect evaluation completeness follows from Lemma 5.1. For evaluation knowledge soundness, we apply [9, Theorem 4] to Lemma 5.8. Communication complexity follows from Lemma 5.5. Additionally, claims about the prover and verifier runtime hold by Lemma 5.5 and the fact that both $\log q$ and N are polynomial in λ. □

5.3. *Large Sampling Set*

We present a second instantiation which allows us to obtain negligible knowledge soundness error *without parallel repetition*, using coordinate-wise special soundness (c.f. Sect. 2.9) and a large challenge space. We let $t, k \in \mathbb{N}$. Fix also $\beta_{\mathcal{C}} > 0$. Recall that $S_\kappa := \{\alpha \in \mathcal{R}_q : \|\alpha\|_\infty \leq \kappa\}$. We define the challenge space and the slack space as

$$\mathcal{C} := S_{\beta_{\mathcal{C}}}^k \quad \text{and} \quad \mathcal{S}_t := \left\{ \prod_{i \in [t]} \alpha_i - \alpha_i' : \alpha_i, \alpha_i' \in S_{\beta_{\mathcal{C}}}, \alpha_i \neq \alpha_i' \right\} .$$

Note that $|\mathcal{C}| = (2\beta_{\mathcal{C}} + 1)^{kN}$ and $\mathrm{w} \leq \beta_{\mathcal{C}} kN$. We also let $\beta_{s,t} := \max_{c \in \mathcal{S}_t} \|c\|_\infty$. Note that, for $c \in \mathcal{S}_t$,

$$\|c\|_\infty \leq \left\| \prod_i (\alpha_i - \alpha_i') \right\|_\infty \leq \left\|\alpha_1 - \alpha_1'\right\|_\infty \cdot \prod_{i \neq 1} \left\|\alpha_i - \alpha_i'\right\|_1 \leq 2\beta_{\mathcal{C}} \cdot (2\beta_{\mathcal{C}} N)^{t-1} ,$$

and thus $\|c\|_1 \leq (2\beta_{\mathcal{C}} N)^t$.

We show a simple invertibility result that will be useful in the proof of soundness.

Lemma 5.10. *Let* $1 \leq l < N$ *be a power of two, and suppose that* $q \equiv 2N/l + 1 \pmod{4N/l}$. *If* $2\beta_{\mathcal{C}} < \sqrt{l/N} q^{l/N}$, *then for any* $t \geq 1$, $\mathcal{S}_t \subseteq \mathcal{R}_q^\times$.

Proof. Let $\alpha \neq \alpha' \in S_{\beta_{\mathcal{C}}}$. Then, $\alpha - \alpha' \neq 0$, and $\left\|\alpha - \alpha'\right\|_\infty \leq 2\beta_{\mathcal{C}}$. Thus, by Lemma 2.18, $\alpha - \alpha' \in \mathcal{R}_q^\times$. Elements of $\mathcal{S}_t$ are products of elements of that form, and since the product of invertible elements is itself invertible, the result follows. □

We will assume thereafter that we are in the regime in which Lemma 2.18 holds (as in Table 3).

We again aim to show that $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ is knowledge sound. As before, we define an opening relation, which will differ from Equation (19) in that the relaxation factors will not be the same across openings, but rather will be included as part of the witness. This will reflect the fact that the extracted opening will have different slack derived from the challenges.

$$\tilde{\mathsf{R}}_{d,\beta,t} := \left\{ ((\mathbf{A},\mathbf{W}),(\mathbf{t},u,z),(f,(\mathbf{s}_i)_i,(c_i)_i)) \middle| \begin{array}{c} \forall i \in [0,d], \mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \wedge \\ \wedge c_i \in \mathcal{S}_t \wedge \|c_i \cdot \mathbf{s}_i\| \leq \beta \\ \wedge\ f(u) = z \end{array} \right\} . \tag{20}$$

As before, to argue that the protocol is knowledge sound, we will first show an extractor to be used to move between layers of the transcript tree. In this case however, we will argue using coordinate-wise special soundness instead of special soundness.

Lemma 5.11. (Coordinate-Wise Special Soundness for Σ). *Let $c \in \mathcal{R}_q^\times$, and let $\mathbb{i} = (\mathbf{A}, \mathbf{W})$, $\mathbb{x} = (\mathbf{t}, u, z)$. There exists an algorithm that, given $k+1$ transcripts $(\mathsf{tr}_j)_{j\in[0,k]}$ of the following form:*

$$\mathsf{tr}_j := \begin{pmatrix} (z_1,\ldots,z_k) \\ \boldsymbol{\alpha}_j \\ (g_j,(\mathbf{s}_{j,i})_{i\in[0,d']}) \end{pmatrix} \text{ with } (\boldsymbol{\alpha}_j)_j \in \mathsf{SS}(S_{\beta_C},2,k) \ ,$$

and slack $(c_{j,i})_{j,i}$ outputs $\mathbb{w} := (\bar{f}, (\bar{\mathbf{s}}_i)_i, (\bar{c}_i)_i)$. Furthermore, let $\mathbb{i}'$, $(\mathbb{x}'_j)_{j\in[k]}$ be obtained as in Fig. 5 (where $\mathbb{x}'_j$ is obtained from the j-th leaf of the transcript) and $\mathbb{w}'_j := (g_j, (s_{j,i})_i, (c_{j,i})_i)$. If, for $i \in [0,k]$, $(\mathbb{i}', \mathbb{x}'_i, \mathbb{w}'_i), \in \tilde{\mathsf{R}}_{d',\beta,t}$, and $z = \sum_{t\in[k]} u^{t-1} z_t$, then $(\mathbb{i}, \mathbb{x}, \mathbb{w}) \in \tilde{\mathsf{R}}_{d,\gamma,2t+1}$ where $\gamma := 2\beta$ if $t = 0$ and $\gamma := 2N\beta_{s,t}\beta$ otherwise.

Proof. Assume, without loss of generality, that the transcripts are arranged so that, for $j \in [k]$, $\boldsymbol{\alpha}_0 \equiv_j \boldsymbol{\alpha}_j$. We thus can write $\boldsymbol{\alpha}_0 = (\alpha_1, \ldots, \alpha_k)$ and $\boldsymbol{\alpha}_j := (\alpha_1, \ldots, \alpha'_j, \ldots \alpha_k)$ with $\alpha_j \neq \alpha'_j$. Consider the extractor

```
$\underline{\mathcal{E}(\mathsf{tr} = (\mathsf{tr}_0, \ldots, \mathsf{tr}_k), (\tilde{c}_{j,i})_{j,i})}$:
  1. For $j \in [k]$:
     (a) Set $\bar{f}_j := \frac{g_0 - g_j}{\alpha_j - \alpha'_j}$.
     (b) For $i \in [0, d']$:
         i. Set $\bar{\mathbf{s}}_{ki+j-1} := \frac{\mathbf{z}_{0,i} - \mathbf{z}_{j,i}}{\alpha_j - \alpha'_j}$.
         ii. Set $\bar{c}_{ki+j-1} := (\alpha_j - \alpha'_j) c_{0,i} c_{j,i}$.
  2. Set $\bar{f} := \sum_{j\in[k]} \mathsf{X}^{j-1} \bar{f}_j(\mathsf{X}^k)$.
  3. Return $(\bar{f}, (\bar{\mathbf{s}}_i)_{i\in[0,d]}), (\bar{c}_i)_{i\in[0,d]}$.
```

Since the transcript is accepting, for $j \in [0,k]$, $i \in [0, d']$ we have that

$$\mathbf{A}\mathbf{z}_{j,i} + g_{j,i}\mathbf{e}_1 = (\mathbf{W}^k)^{-i} \left(\sum_{t\in[k]} \alpha_{j,t} \mathbf{W}^{t-1} \right) \mathbf{t} \ .$$

Subtracting the equation for $j = 0$ from the equation for $j \in [k]$ yields that, for $i \in [0, d']$:

$$\mathbf{A}\left(\frac{\mathbf{z}_{0,i} - \mathbf{z}_{j,i}}{\alpha_j - \alpha'_j}\right) + \left(\frac{g_{0,i} - g_{j,i}}{\alpha_j - \alpha'_j}\right)\mathbf{e}_1 = \mathbf{W}^{-(ki+j-1)}\mathbf{t} \ .$$

To show that the extracted $\bar{f}$ evaluates to z at u, note that:

$$\begin{aligned}\bar{f}(u) &= \sum_{j\in[k]} u^{j-1}\bar{f}_j(u^k)\\ &= \sum_{j\in[k]} u^{j-1}\frac{g_0(u^k) - g_j(u^k)}{\alpha_j - \alpha'_j}\\ &= \sum_{j\in[k]} u^{j-1}\frac{\sum_{t\in[k]}(\alpha_{0,t} - \alpha_{j,t})z_t}{\alpha_j - \alpha'_j}\\ &= \sum_{j\in[k]} u^{j-1}z_j = z \ .\end{aligned}$$

where in the third equality we have used that the verifier check accepts, and for the fourth $\sum_{t\in[k]}(\alpha_{0,t} - \alpha_{j,t})z_t = (\alpha_j - \alpha'_j)z_j$. We argue that the extracted $\bar{\mathbf{s}}_i$ are (relaxed) short.

$$\begin{aligned}\left\|\bar{c}_{ki+j-1} \cdot \bar{\mathbf{s}}_{ki+j-1}\right\| &= \left\|(\alpha_j - \alpha'_j)c_{0,i}c_{j,i}\frac{\mathbf{z}_{0,i} - \mathbf{z}_{j,i}}{\alpha_j - \alpha'_j}\right\|\\ &= \left\|c_{0,i}c_{j,i}(\mathbf{z}_{0,i} - \mathbf{z}_{j,i})\right\|\\ &\le \left\|c_{j,i}c_{0,i}\mathbf{z}_{0,i}\right\| + \left\|c_{0,i}c_{j,i}\mathbf{z}_{j,i}\right\|\\ &\le \sqrt{N}\beta_{s,t}(\left\|c_{0,i}\mathbf{z}_{0,i}\right\|_1 + \left\|c_{j,i}\mathbf{z}_{j,i}\right\|_1)\\ &\le 2N\beta_{s,t}\beta = \gamma \ .\end{aligned}$$

If $t = 0$, then the slacks must have been 1, and thus $\left\|\bar{c}_{ki+j-1}\bar{\mathbf{s}}_{ki+j-1}\right\| \le \left\|\mathbf{z}_{0,i} - \mathbf{z}_{j,i}\right\| \le 2\beta$ as desired. Finally, what is left to show is that the new slack is in the prescribed slack space. This is easy to show as the previous two slacks are a product of t differences of challenges, that we then multiply with a new difference, leading to a product of $2t + 1$ differences of challenges. Lemma 5.10 guarantees that this new slack is invertible as long as $\beta_{\mathcal{C}}$ is small enough. □

We then use this extractor recursively to show that Eval is coordinate-wise special sound.

Lemma 5.12. (Coordinate-Wise Special Soundness for Eval). *Let $k, h \in \mathbb{N}$, $\beta_{\mathcal{C}} > 0$. Let $\Pi{:=}\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ be as in Construction 5.2. Then, Π is a k-coordinate-wise special sound proof system for the relation $\tilde{\mathsf{R}}_{d,\gamma,t}$ where*

$$\begin{aligned}\gamma &{:=} 2^h \cdot (2\beta_{\mathcal{C}}N)^{2^h-h-1} \cdot \beta_h\\ t &{:=} 2^h - 1 \ .\end{aligned}$$

Proof. We index the transcript as in Lemma 5.8. Denote by $\mathcal{E}^{(1)}$ the extractor of Lemma 5.11. Consider the new extractor

```
$\underline{\mathcal{E}(\mathsf{tr})}$:
  1. Set $\bar{c}_{(h,j)} = 1$ for $j \in [(k+1)^h]$.
  2. For $r := h, \dots, 1$:
     (a) Set for $j \in [(k+1)^{r-1}]$:
         $$\mathsf{tr}_{(r-1,j)} := \begin{pmatrix} & (\boldsymbol{\alpha}_{(r,(j-1)(k+1)+1)}, (\bar{f}_{(r,(j-1)(k+1)+1)}, (\bar{\mathbf{s}}_{(r,(j-1)(k+1)+1),i})_i)) \\ (z_{(r-1,j),t})_{t\in[k]} & \vdots \\ & (\boldsymbol{\alpha}_{(r,j(k+1))}, (\bar{f}_{(r,j(k+1))}, (\bar{\mathbf{s}}_{(r,j(k+1)),i})_i)) \end{pmatrix} .$$
     (b) Compute $(\bar{f}_{(r-1,j)}, (\bar{\mathbf{s}}_{(r-1,j),i})_i, (\bar{c}_{(r-1,j),i})_i) \leftarrow \mathcal{E}^{(1)}(\mathsf{tr}_{(r-1,j)}, (\bar{c}_{(r,(j-1)(k+1)+t),i})_{t,i})$.
  3. Return $\bar{f}_{(0,1)}, (\bar{\mathbf{s}}_{(0,1),t}), (\bar{c}_{(0,1),t})_t$.
```

We argue that the extractor yields a valid witness inductively. We again note that for $(r, j) \in [h] \times [(k+1)^r]$, since the transcripts are accepting,

$$z_{(r-1,j)} = \sum_{t\in[k]} u_{r-1}^{k-1} z_{(r-1,j),t} \quad .$$

Write $\mathbb{i}_{(r,j)} := (\mathbf{A}, \mathbf{W}_r)$, $\mathbb{x}_{(r,j)} := (\mathbf{t}_{(r,j)}, u_r, (z_{(r,j),i})_i)$ and $\mathbb{w}_{(r,j)} := (\bar{f}_{(r,j)}, (\bar{\mathbf{s}}_{(r,j),i})_i, (\bar{c}_{(r,j),i})_i)$. Since the leaves are accepting (and the relaxed relation is equivalent to the exact one when the relaxation factors are one), $(\mathbb{i}_{(h,j)}, \mathbb{x}_{(h,j)}, \mathbb{w}_{(h,j)}) \in \tilde{\mathsf{R}}_{d_h,\beta_h,0}$. Thus, Lemma 5.11 (in the case $t = 0$) implies that $(\mathbb{i}_{(h-1,j)}, \mathbb{x}_{(h-1,j)}, \mathbb{w}_{(h-1,j)}) \in \tilde{\mathsf{R}}_{d_{h-1},2\beta_h,1}$. Now, we define the recurrence relations:

$$t_r := \begin{cases} 1 & \text{if } r = 1 \\ 2t_{r-1} + 1 & \text{otherwise} \end{cases} \text{ and } \gamma_r := \begin{cases} 2\beta & \text{if } r = 1 \\ 2N\beta_{s,t_{r-1}}\gamma_{r-1} & \text{otherwise} \end{cases} .$$

Lemma 5.11 implies exactly that, if $(\mathbb{i}_{(r,j)}, \mathbb{x}_{(r,j)}, \mathbb{w}_{(r,j)}) \in \tilde{\mathsf{R}}_{d_{r-i},\gamma_r,t_r}$, then the extracted witness $(\mathbb{i}_{(r+1,j)}, \mathbb{x}_{(r+1,j)}, \mathbb{w}_{(r+1,j)}) \in \tilde{\mathsf{R}}_{d_{k-r-1},\gamma_{r+1},t_{r+1}}$. Unfolding the recurrence relations, we note that $t_r = 2^r - 1$ and

$$\begin{aligned} \gamma_r &= 2^r N^{r-1} \left(\prod_{i=1}^{r-1} \beta_{s,t_i} \right) \beta_h \\ &\leq 2^r N^{r-1} \left(\prod_{i=1}^{r-1} 2\beta_{\mathcal{C}} (2\beta_{\mathcal{C}} N)^{2^i - 2} \right) \beta_h \\ &= 2^r N^{r-1} (2\beta_{\mathcal{C}})^{r-1} (2\beta_{\mathcal{C}} N)^{\sum_{i=1}^{r-1} 2^i - 2} \cdot \beta_h \\ &= 2^j N^{r-1} (2\beta_{\mathcal{C}})^{r-1} (2\beta_{\mathcal{C}} N)^{2^r - 2r} \cdot \beta_h \end{aligned}$$

Table 5. Parameters for the polynomial commitment scheme obtained from Fig. 4 and the Fiat–Shamir transform of $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ for proofs of evaluation.

Parameters	Instantiation
m	$\geq n(1+\tilde{q}) + \omega(\log \lambda)$
l	$N/2$
δ	$q^{1/O(1)}$
$\mathfrak{s}$	$> 2Nq^{\frac{n}{m-n\tilde{q}} + \frac{2}{N(m-n\tilde{q})}}$
σ_0	$\geq \delta \mathfrak{s} N n \tilde{q} \cdot \omega(\sqrt{2(d+1)n(m-n\tilde{q})N \log t'N})$
σ_1	$\geq \delta \sigma_0 N \cdot \omega(\sqrt{m'n' \log t'N})$
β	$\geq \sigma_1 \sqrt{m'N}$
$\mathcal{C}$	$S^k_{\beta_\mathcal{C}}$
$\beta_\mathcal{C}$	$< \frac{1}{2}\sqrt{l/N} q^{l/N}$
w	$kN\beta_\mathcal{C}$
β_h	$\mathrm{w}^h \cdot \beta$
γ	$2^h \cdot (2\beta_\mathcal{C} N)^{2^h-h-1} \cdot \beta_h$
β_s	$(2\beta_\mathcal{C} N)^{2^h-1}$
Soundness	$\frac{(Q+1)\cdot hk}{(2\beta_\mathcal{C}+1)^N}$
Commitment size	$nN \log q$
Proof size	$h(kN \log q) + \frac{d+1}{k^h}(N \log q + mN \log \beta_h)$
Prover time	$O(md)$
Verifier time	$O((n+m)^2 \cdot (hk + d/k^h))$

We let Q be an upper bound on the number of queries an adversary can make to the random oracle.

$$= 2^r (2\beta_\mathcal{C} N)^{2^r-r-1} \cdot \beta_h$$

Taking this to its natural conclusion:

$$(\mathbb{i}_{(0,1)}, \mathbb{x}_{(0,1)}, \mathbb{w}_{(0,1)}) \in \tilde{\mathsf{R}}_{d,\gamma_h,t_h} \quad ,$$

and setting $\gamma := \gamma_h$, $t := t_h$ implies the result. □

Again, we can use Eval to construct a polynomial commitment scheme.

Theorem 5.13. *Let* $\mathsf{PC} = (\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval}, \mathsf{Verify})$ *where* Setup, Commit, Open *are as in Fig. 4 and* Eval, Verify *are obtained by applying the Fiat–Shamir transform to* $\mathsf{Eval}[d, k, h, \mathcal{C}, \beta]$ *when* $k^h = \mathsf{poly}(\lambda)[d]$. *Then,* PC *is an polynomial commitment scheme with the efficiency properties and parameters shown in Table 5.*

Proof. Completeness and relaxed binding follow from Lemmata 4.1 and 4.2. Perfect evaluation completeness follows from Lemma 5.1. Communication complexity and runtimes follow from Lemma 5.5. Knowledge soundness follows from Lemma 2.31 and Lemma 5.12, noting that when $k^h = \mathsf{poly}(\lambda)[d]$ and thus the extractor runs in expected polynomial time. □

At this point, one might be tempted to instantiate the scheme in Theorem 5.13 with $h = O(\log d)$ and $k = O(1)$ to obtain a protocol with logarithmic communication complexity as in Theorem 5.9 and small soundness error. This unfortunately does not succeed, as the extracted norm in this case grows $\exp(d)$ and thus $\log q \geq \mathrm{poly}(d)$. The resulting protocol will communicate logarithmically many elements of $\mathcal{R}_q$, but the overall communication complexity will thus be polynomial in d. Thus, h must be at most $O(\log\log d)$. In fact, let $0 < \epsilon < 1$ be a constant and set $h = 1/\epsilon = O(1)$, $k = d^{\epsilon}$. It is easy to see from Table 5 that then the communication complexity will be $O(d^{\epsilon})$ elements of $\mathcal{R}_q$ and we can set $\log q = \mathrm{polylog}(d)$ to obtain overall sublinear communication complexity. Accordingly, the verifier time will also be sublinear. In fact, we can further improve on this. Set now $h \approx \log\log d$, and $k \approx d^{1/\log\log d}$. It can be easily verified that in this case we obtain

$$\log q = O\left(\frac{\log^2 d}{\log\log d}\right),$$

and in terms of communication complexity: $O((\log\log d) \cdot d^{1/\log\log d})$ elements of $\mathcal{R}_q$ or $\mathrm{polylog}(d) \cdot d^{1/\log\log d}$ bits (similarly for the verifier complexity). As such, we can conclude that Theorem 5.13 gives rise to a *quasi-polylogarithmic* non-interactive polynomial commitment scheme from lattice assumptions.

5.4. *Batching Evaluations*

5.4.1. *Multiple Evaluations at a Single Point*

We show a simple approach to amortise the cost of proving evaluations of multiple evaluations at a single point. More concretely, we have a list of (committed) polynomials $f_1, \ldots, f_r$ and want to show that $f_i(u) = z_i$. First we define the corresponding relation, namely:

$$\mathsf{R}^r_{d,\beta} := \left\{ (\mathbf{A}, \mathbf{W}), ((\mathbf{t}_j)_j, u, (z_j)_j), ((f_j)_j, (\mathbf{s}_{j,i})_{j,i}) \;\middle|\; \begin{array}{c} \forall j \in [r], \\ ((\mathbf{A}, \mathbf{W}), (\mathbf{t}_j, u, z_j), (f_j, (\mathbf{s}_{j,i})_i)) \in \mathsf{R}_{d,\beta} \end{array} \right\}.$$

The intuition of the protocol that we design is to take a random linear combinations of the polynomials $f_1, \ldots, f_r$, and prove that its evaluation at u is equal to the linear combination of the claimed evaluations. The protocol that we describe in Fig. 7 takes this idea and combines it with one round of Fig. 5, which is useful for better concrete efficiency.

Lemma 5.14. (Completeness for $\mathsf{multiEval}$). *Let* $\Pi := \mathsf{multiEval}[d, r, k, \mathcal{C}, \beta]$ *be the protocol in Fig. 7. Then,* Π *is a* Σ*-protocol with perfect completeness for* $\mathsf{R}^r_{d,\beta}$.

Proving Multiple Evaluations at a Single Point

Prover **Verifier**

$$\sum_{t\in[k]} \mathsf{X}^{t-1} g_{\iota,t}(\mathsf{X}^k) =: f_\iota(\mathsf{X}) \text{ for } \iota \in [r]$$

$$z_{\iota,t} := g_{\iota,t}(u^k) \text{ for } (\iota,t) \in [r]\times[k]$$

$$\xrightarrow{(z_{\iota,t})_{(\iota,t)\in[r]\times[k]}}$$

$$\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_r) \leftarrow \mathcal{C} := S_{\beta_{\mathcal{C}}}^{rk}$$

$$\xleftarrow{\boldsymbol{\alpha}}$$

$$g := \sum_{(\iota,t)\in[r]\times[k]} \alpha_{\iota,t} g_{\iota,t}$$

$$\mathbf{z}_i := \sum_{(\iota,t)\in[r]\times[k]} \alpha_{\iota,t}\mathbf{s}_{\iota,t,i} \text{ for } i \in [0,d']$$

$$\xrightarrow{g, (\mathbf{z}_i)_{i\in[0,d']}}$$

$$\beta' := \mathrm{w}\,\beta$$

$$\mathbf{t}' := \left(\sum_{(\iota,t)\in[r]\times[k]} \alpha_{\iota,t}\mathbf{W}^{-(t-1)}\cdot\mathbf{t}_\iota\right)$$

$$\mathbb{i}' := (\mathbf{A}, \mathbf{W}^k)$$

$$\mathbb{x}' := \left(\mathbf{t}', u^k, \sum_{(\iota,t)\in[r]\times[k]} \alpha_{\iota,t} z_{\iota,t}\right)$$

$$\mathbb{w}' := (g, (\mathbf{z}_i)_{i\in[0,d']})$$

Check:

$$z_\iota = \sum_{t\in[k]} u^{t-1} z_{\iota,t} \text{ for } \iota \in [r]$$

$$(\mathbb{i}', \mathbb{x}', \mathbb{w}') \in \mathsf{R}_{d',\beta'}$$

Fig. 7. The protocol $\mathsf{multiEval}[d, r, k, \mathcal{C}, \beta]$ for proving evaluations of r polynomials at a single point. In the above $\mathrm{w} := \max_{\alpha\in\mathcal{C}} \|\alpha\|_1$. As before, we define $d' := (d+1)/k - 1$ and $\mathbf{s}_{\iota,t,i} := \mathbf{s}_{\iota,ki+t-1}$ for $\iota \in [r]$.

Proof. It is easy to see that $g(u^k) = \sum_{\iota,t} \alpha_{\iota,t} g_{\iota,t}(u^k) = \sum \alpha_{\iota,t} z_{\iota,t}$. Also, for $i \in [0, d']$,

$$\begin{aligned}
\mathbf{A}\mathbf{z}_i + g_i\mathbf{e}_1 &= \sum_{\iota,t\in[r]\times[k]} \alpha_{\iota,t}\left(\mathbf{A}\mathbf{s}_{\iota,t,i} + g_{\iota,t,i}\mathbf{e}_1\right)\\
&= \sum_{\iota,t\in[r]\times[k]} \alpha_{\iota,t}\left(\mathbf{A}\mathbf{s}_{\iota,ki+t-1} + g_{\iota,ki+t-1}\mathbf{e}_1\right)\\
&= \sum_{\iota,t\in[r]\times[k]} \alpha_{\iota,t}\mathbf{W}^{-(ki+t-1)}\mathbf{t}_\iota.
\end{aligned}$$

Finally, $\|\mathbf{z}_i\| = \left\|\sum_{\iota,t} \alpha_{\iota,t}\mathbf{s}_{\iota,t,i}\right\| \le \mathrm{w}\beta = \beta'$ as desired. □

As before, we define a relaxed opening relation (we use the definition of $\tilde{\mathsf{R}}$ from Equation (20)):

$$\tilde{\mathsf{R}}^{r}_{d,\beta,t}:=\left\{\begin{pmatrix}(\mathbf{A},\mathbf{W}),\\((\mathbf{t}_\iota)_\iota,u,(z_\iota)_\iota),\\((f_\iota)_\iota,(\mathbf{s}_{\iota,i})_{\iota,i},(c_{\iota,i})_{\iota,i})\end{pmatrix}\;\middle|\;\begin{array}{c}\forall\iota\in[r],\\((\mathbf{A},\mathbf{W}),(\mathbf{t}_\iota,u,z_j),(f_\iota,(\mathbf{s}_{\iota,i})_i),(c_{\iota,i})_i))\in\tilde{\mathsf{R}}_{d,\beta,t}\end{array}\right\}.$$

We now prove coordinate-wise special soundness for the set $\mathcal{C}:=S^{rk}_{\beta_\mathcal{C}}\subseteq\mathcal{R}^{rk}_q$, where each element has rk coordinates. Then, it is easy to show (e.g. using the composition results as in [20, Section 3]) that composing $\mathsf{multiEval}$ with Eval yields a knowledge sound protocol for this relaxed relation.

Lemma 5.15. (Coordinate-Wise Special Soundness for $\mathsf{multiEval}$). *Let* $\Pi:=\mathsf{multiEval}[d,r,k,\mathcal{C},\beta]$ *be the protocol in Fig. 7. Let* $\mathbb{i}:=(\mathbf{A},\mathbf{W})$, $\mathbb{x}:=((\mathbf{t}_\iota)_\iota,u,(z_\iota)_\iota)$. *There exists an algorithm that, given* $rk+1$ *transcripts* $(\mathsf{tr}_j)_{j\in[0,rk]}$ *of the following form:*

$$\mathsf{tr}_j:=\begin{pmatrix}(z_{\iota,t})_{\iota,t}\\\boldsymbol{\alpha}_j\\(g_j,(\mathbf{z}_{j,i})_{i\in[0,d]})\end{pmatrix}\text{ with }(\boldsymbol{\alpha}_j)_j\in\mathsf{SS}(S_{\beta_\mathcal{C}},rk)\;,$$

and relaxation factors $(c_{j,i})_{j,i}$, *outputs* $\mathbb{w}:=((\bar{f}_\iota)_\iota,(\bar{\mathbf{s}}_{\iota,i})_{\iota,i},(\bar{c}_{\iota,i})_{\iota,i})$. *Now, set* $\mathbb{i}':=(\mathbf{A},\mathbf{W}^k)$, $\mathbb{x}_j:=(\sum_{\iota,t}\alpha_{j,\iota,t}\mathbf{t}_\iota,u^k,\sum_{\iota,t}\alpha_{j,\iota,t}z_{\iota,t})$, $\mathbb{w}_j:=(g_j,(\mathbf{z}_{j,i})_i,(c_{j,i})_i)$. *If for* $j\in[0,r]$, $(\mathbb{i}',\mathbb{x}_j,\mathbb{w}_j)\in\tilde{\mathsf{R}}_{d,\beta,t}$, *and* $z_\iota=\sum_{t\in[k]}u^{t-1}z_{\iota,t}$ *for* $\iota\in[r]$, *then* $(\mathbb{i},\mathbb{x},\mathbb{w})\in\tilde{\mathsf{R}}^{r}_{d,\gamma,t'}$ *where* $\gamma:=2N\beta_{s,t}\beta$, $t':=2t+1$.

Proof. Again, assume without loss of generality that $\boldsymbol{\alpha}_0\equiv_j\boldsymbol{\alpha}_j$ for $j\in[rk]$. Now, reindex $\boldsymbol{\alpha}_1\ldots,\boldsymbol{\alpha}_{rk}$ into a $r\times k$ matrix $\boldsymbol{\alpha}_{1,1},\ldots,\boldsymbol{\alpha}_{r,k}$. We write $\boldsymbol{\alpha}_0=(\alpha^*_{1,1},\ldots,\alpha^*_{r,k})$ and thus assume that $\boldsymbol{\alpha}_{v,w}=(\alpha^*_{1,1},\ldots,\alpha'_{v,w},\ldots,\alpha^*_{r,k})$ with $\alpha'_{v,w}\neq\alpha^*_{v,w}$. We also reindex $(g_j)_j,(\mathbf{z}_{j,i})$ accordingly so that $g_{v,w}$ corresponds the $\boldsymbol{\alpha}_{v,w}$ challenge (note that we skip the 0-th challenge $\boldsymbol{\alpha}_0$).

With these conventions, we let the extractor be the following.

```
$\underline{\mathcal{E}(\mathsf{tr})}$:
  1. For $\iota\in[r], t\in[k]$:
     (a) Let $\bar{f}_{\iota,t}:=\frac{g_0-g_{\iota,t}}{\alpha^*_{\iota,t}-\alpha'_{\iota,t}}$.
     (b) Let $\bar{\mathbf{s}}_{\iota,ki+t-1}:=\frac{\mathbf{z}_{0,i}-\mathbf{z}_{\iota,t,i}}{\alpha^*_{\iota,t}-\alpha'_{\iota,t}}$ for $i\in[0,d']$.
     (c) Let $\bar{c}_{l,ki+t-1}:=(\alpha^*_{\iota,t}-\alpha'_{\iota,t})c_{0,i}c_{\iota,t,i}$ for $i\in[0,d']$.
  2. Set $\bar{f}_\iota:=\sum_{t\in[k]}\mathsf{X}^{t-1}f_{\iota,t}$ for $\iota\in[r]$.
  3. Return $(\bar{f}_\iota)_\iota,((\bar{\mathbf{s}}_{\iota,i})_i)_\iota,((\bar{c}_{\iota,i})_i)_\iota$.
```

First note that by assumption, $g_0(u^k) = \sum_{\iota,t} \alpha^*_{\iota,t} z_{\iota,t}$ and $g_{v,w}(u^k) = \alpha'_{v,w} z_{v,w} + \sum_{(\iota,t)\neq(v,w)} \alpha^*_{\iota,t} z_{\iota,t}$. Thus, $\bar{f}_{v,w}(u^k) = \frac{g_0 - g_{v,w}}{\alpha^*_{v,w} - \alpha'_{v,w}}(u^k) = z_{v,w}$. Thus, for $\iota \in [r]$:

$$\bar{f}_\iota(u) = \sum_{t\in[k]} u^{t-1} \bar{f}_{\iota,t}(u^k) = \sum_{t\in[k]} u^{t-1} \frac{g_0 - g_{\iota,t}}{\alpha_{\iota,t} - \alpha'_{\iota,t}}(u^k) = \sum_{t\in[k]} u^{t-1} z_{\iota,t} = z_\iota \ .$$

Now, also by assumption:

$$\begin{aligned} \mathbf{A}\mathbf{z}_{0,i} + g_{0,i}\mathbf{e}_1 &= \mathbf{W}^{-i}\left(\sum_{(\iota,t)} \alpha^*_{\iota,t}\mathbf{t}_\iota\right) \\ \mathbf{A}\mathbf{z}_{v,w,i} + g_{v,w,i}\mathbf{e}_1 &= \mathbf{W}^{-i}\left(\alpha'_{v,w}\mathbf{t}_v + \sum_{(\iota,t)\neq(v,w)} \alpha^*_{\iota,t}\mathbf{t}_\iota\right) \\ &\Downarrow \\ \mathbf{A}\left(\frac{\mathbf{z}_{0,i} - \mathbf{z}_{v,w,i}}{\alpha^*_{v,w} - \alpha'_{v,w}}\right) + \left(\frac{g_{0,i} - g_{v,w,i}}{\alpha^*_{v,w} - \alpha'_{v,w}}\right) \cdot \mathbf{e}_1 &= \mathbf{W}^{-(ki+w-1)}\mathbf{t}_v \\ &\Downarrow \\ \mathbf{A}\bar{\mathbf{s}}_{v,ki+w-1} + \bar{f}_{v,ki+w-1}\mathbf{e}_1 &= \mathbf{W}^{-(ki+w-1)}\mathbf{t}_v \ . \end{aligned}$$

Finally, note that $\left\|\bar{c}_{\iota,i}\bar{\mathbf{s}}_{\iota,i}\right\| \leq 2N\beta_{s,t}\beta$ by exactly the same reasoning as in Lemma 5.11. □

5.4.2. *Multiple Evaluations at Distinct Points*

Next, we consider the dual problem, namely amortising proving many statements of the form $f_\iota(u_\iota) = z_\iota$ for $\iota \in [r]$ where $u_1, \ldots, u_r$ can be potentially distinct. Looking at Lemma 5.5, a large part of the communication complexity is represented by the last round, where the prover has to send openings $\mathbf{s}_0, \ldots, \mathbf{s}_{d_h}$. We amortise this by taking a random linear combination of these openings. As before, for concrete efficiency reasons, we integrate this within a round of compression.

The relation that we consider is the following:

$$\mathsf{R}^r_{d,\beta} := \left\{ \left(\begin{array}{c} (\mathbf{A}, \mathbf{W}), \\ (\mathbf{t}_\iota, u_\iota, z_\iota)_\iota \\ (f_\iota, \mathbf{s}_{\iota,i})_{\iota,i} \end{array} \right) \middle| \begin{array}{c} \forall \iota \in [r] \\ ((\mathbf{A}, \mathbf{W}), (\mathbf{t}_\iota, u_\iota, z_\iota), (f_\iota, \mathbf{s}_{\iota,i})_{\iota,i}) \in \mathsf{R}_{d,\beta} \end{array} \right\} \ .$$

The protocol is then described in Fig. 8. Now, we show evalMulti has perfect completeness..

Lemma 5.16. (Completeness for evalMulti). *Let* Π:=evalMulti$[d, r, k, \mathcal{C}, \beta]$. *Then* Π *is a* Σ*-protocol with perfect completeness for* $\mathsf{R}^r_{d,\beta}$.

Proving Multiple Evaluations at Distinct Points

Prover **Verifier**

$\sum_{t\in[k]} \mathsf{X}^{t-1} h_{\iota,t}(\mathsf{X}^k) := f_\iota(\mathsf{X})$ for $l \in [r]$

$z_{\iota,t} := h_{\iota,t}(u_\iota^k)$

$$\xrightarrow{(z_{\iota,t})_{\iota,t}}$$

$\boldsymbol{\alpha} \leftarrow \mathcal{C} := S_{\beta_\mathcal{C}}^{rk}$

$$\xleftarrow{\boldsymbol{\alpha}}$$

$g_\iota := \sum_{t\in[k]} \alpha_{\iota,t} h_{\iota,t}$ for $\iota \in [r]$

$\mathbf{z}_i := \sum_{\iota,t\in[r]\times[k]} \alpha_{\iota,t} \mathbf{s}_{\iota,t,i}$ for $i \in [d']$

$$\xrightarrow{(g_\iota)_\iota, (\mathbf{z}_i)_i}$$

Check:

$z_\iota = \sum_{t\in[k]} u_\iota^{t-1} z_{\iota,t}$ for $\iota \in [r]$

$g_\iota(u_\iota^k) = \sum_{t\in[k]} \alpha_{\iota,t} z_{\iota,t}$ for $\iota \in [r]$

$\mathbf{A}\mathbf{z}_i + \left(\sum_{\iota\in[r]} g_{\iota,i}\right) \mathbf{e}_1 = \mathbf{W}^{-ki} \left(\sum_{\iota,t} \alpha_{\iota,t} \mathbf{W}^{-(t-1)} \mathbf{t}_\iota\right)$

$\|\mathbf{z}_i\| \leq \mathrm{w}\,\beta$ for $i \in [0, d']$

Fig. 8. The protocol $\mathsf{evalMulti}[d, r, k, \mathcal{C}, \beta]$ for proving evaluations of multiple polynomials at multiple points. In the above $\mathrm{w}:=\max_{\boldsymbol{\alpha}\in\mathcal{C}} \|\boldsymbol{\alpha}\|_1$ and $d':=(d+1)/k-1$.

Proof. For the first verifier check,

$$z_\iota = f_\iota(u_\iota) = \sum_{t\in[k]} u_\iota^{t-1} h_{\iota,t}(u_\iota^k) = \sum_{t\in[k]} u_\iota^{t-1} z_{\iota,t} \enspace .$$

Next, we check that g_ι evaluates to the correct value.

$$g_\iota(u_\iota^k) = \sum_{t\in[k]} \alpha_{\iota,t} h_{\iota,t}(u_\iota^k) = \sum_{t\in[k]} \alpha_{\iota,t} z_{\iota,t} \enspace .$$

Checking validity of the openings is similarly straightforward:

$$\begin{aligned}
\mathbf{A}\mathbf{z}_i + \left(\sum_\iota g_{\iota,i}\right) \mathbf{e}_1 &= \mathbf{A}\left(\sum_{\iota,t} \alpha_{\iota,t} \mathbf{s}_{\iota,t,i}\right) + \left(\sum_{\iota,t} \alpha_{\iota,t} h_{\iota,t,i}\right) \mathbf{e}_1 \\
&= \sum_{\iota,t} \alpha_{\iota,t} \left(\mathbf{A}\mathbf{s}_{\iota,t,i} + h_{\iota,t,i}\mathbf{e}_1\right) \\
&= \sum_{\iota,t} \alpha_{\iota,t} \left(\mathbf{A}\mathbf{s}_{\iota,ki+t-1} + f_{\iota,ki+t-1}\mathbf{e}_1\right)
\end{aligned}$$

$$= \sum_{\iota,t} \alpha_{\iota,t}\left(\mathbf{W}^{-(ki+t-1)}\mathbf{t}_\iota\right)$$
$$= (\mathbf{W}^k)^{-i}\cdot\left(\sum_{\iota,t}\alpha_{\iota,t}\mathbf{W}^{-(t-1)}\mathbf{t}_\iota\right)\ .$$

Finally, $\|\mathbf{z}_i\| = \left\|\sum_{\iota,t}\alpha_{\iota,t}\mathbf{s}_{\iota,t,i}\right\| \le \mathrm{w}\beta$. □

For knowledge soundness, we again define a relaxed opening relation, namely:

$$\tilde{\mathsf{R}}^r_{d,\beta} := \left\{ \left(\begin{matrix} (\mathbf{A},\mathbf{W}), \\ (\mathbf{t}_\iota, u_\iota, z_\iota)_\iota \\ (f_\iota, \mathbf{s}_{\iota,i}, c_{\iota,i})_{\iota,i} \end{matrix} \right) \middle| \begin{matrix} \forall \iota \in [r] \\ ((\mathbf{A},\mathbf{W}), (\mathbf{t}_\iota, u_\iota, z_\iota), (f_\iota, \mathbf{s}_{\iota,i}, c_{\iota,i})_{\iota,i}) \in \tilde{\mathsf{R}}_{d,\beta,1} \end{matrix} \right\}\ .$$

Lemma 5.17. (Coordinate-Wise Special Soundness for $\mathsf{multiEval}$). *Let* $\Pi := \mathsf{multiEval}[d, r, k, \mathcal{C}, \beta]$ *be the protocol in Fig. 7. Then,* Π *is a* rk*-coordinate-wise knowledge sound proof system for* $\tilde{\mathsf{R}}^r_{d,2\beta}$.

Proof. For $j \in [0, rk]$, consider transcripts of the following form:

$$\mathsf{tr}_j := \left(\begin{matrix} (z_{\iota,t})_{\iota,t} \\ \boldsymbol{\alpha}_j \\ ((g_{j,\iota})_\iota, (\mathbf{z}_{j,i})_i) \end{matrix} \right) \text{ with } (\boldsymbol{\alpha}_j)_j \in \mathsf{SS}(S_{\beta_\mathcal{C}}, rk)\ ,$$

and again assume, without loss of generality, that the transcripts are arranged so that, for $j \in [r]$, $\boldsymbol{\alpha}_0 \equiv_j \boldsymbol{\alpha}_j$. Reindex and arrange the challenges as in the proof of Lemma 5.15.

Consider the following extractor:

```
$\mathcal{E}(\mathsf{tr}_0, \ldots, \mathsf{tr}_{rk})$:
  1. For $\iota \in [r]$, $t \in [k]$:
     (a) Set $\bar{f}_{\iota,t} := \frac{g_0 - g_{\iota,t}}{\alpha^*_{\iota,t} - \alpha'_{\iota,t}}$.
     (b) Set $\bar{\mathbf{s}}_{\iota,ki+t-1} := \frac{\mathbf{z}_{0,i} - \mathbf{z}_{\iota,t,i}}{\alpha^*_{\iota,t} - \alpha'_{\iota,t}}$ for $i \in [0, d']$.
     (c) Set $\bar{c}_{\iota,ki+t-1} := \alpha^*_{\iota,t} - \alpha'_{\iota,t}$ for $i \in [0, d']$.
  2. Set $\bar{f}_\iota := \sum_{t\in[k]} \mathsf{X}^{t-1}\bar{f}_{\iota,t}$ for $\iota \in [r]$.
  3. Return $(\bar{f}_\iota)_\iota, (\bar{\mathbf{s}}_{\iota,i})_{\iota,i}, (\bar{c}_{\iota,i})_{\iota,i}$.
```

Since the transcripts are accepting, we have that $z_\iota = \sum_{t\in[k]} u_\iota^{t-1} z_{\iota,t}$ for $\iota \in [r]$. Also, $g_{0,\iota}(u_\iota^k) = \sum_{t\in[k]} \alpha^*_{\iota,t} z_{\iota,t}$ and $g_{v,w,\iota}(u_\iota^k) = \alpha'_{v,w} z_{v,w} + \sum_{t\neq w} \alpha^*_{\iota,t} z_{\iota,t}$. Thus, $\frac{g_{0,\iota} - g_{v,w}}{\alpha^*_{v,w} - \alpha'_{v,w}}(u_\iota^k) = z_{v,w}$. Now,

$$\bar{f}_\iota(u_\iota) = \sum_{t\in[k]} u_\iota^{t-1}\bar{f}_{\iota,t}(u_\iota^k) = \sum_{t\in[k]} u_\iota^{t-1}\frac{g_0 - g_{\iota,t}}{\alpha^*_{\iota,t} - \alpha'_{\iota,t}}(u_\iota^k) = \sum_{t\in[k]} u_\iota^{t-1} z_{\iota,t} = z_\iota\ .$$

We also have that

$$\begin{aligned}
\mathbf{A}\mathbf{z}_{0,i} + \left(\sum_{\iota} g_{0,\iota,i}\right)\mathbf{e}_1 &= \mathbf{W}^{-ki}\left(\sum_{\iota,t}\alpha^*_{\iota,t}\mathbf{W}^{-(t-1)}\mathbf{t}_\iota\right)\\
\mathbf{A}\mathbf{z}_{v,w,i} + \left(\sum_{\iota} g_{v,w,\iota,i}\right)\mathbf{e}_1 &= \mathbf{W}^{-ki}\left(\alpha'_{v,w}\mathbf{W}^{-(w-1)}\mathbf{t}_v + \sum_{\iota,t\neq(v,w)}\alpha^*_{\iota,t}\mathbf{W}^{-(t-1)}\mathbf{t}_\iota\right)\\
&\Downarrow\\
\mathbf{A}\left(\frac{\mathbf{z}_{0,i}-\mathbf{z}_{v,w,i}}{\alpha^*_{v,w}-\alpha'_{v,w}}\right) + \bar{f}_{v,w,i}\mathbf{e}_1 &= \mathbf{W}^{-ki}\left(\mathbf{W}^{-(w-1)}\mathbf{t}_v\right)\\
&\Downarrow\\
\mathbf{A}\bar{\mathbf{s}}_{v,ki+w-1} + \bar{f}_{v,ki+w-1}\mathbf{e}_1 &= \mathbf{W}^{-(ki+w-1)}\mathbf{t}_v \ .
\end{aligned}$$

Finally, $\left\|\bar{c}_{\iota,ki+t-1}\bar{\mathbf{s}}_{\iota,ki+t-1}\right\| \leq \left\|\mathbf{z}_{0,i}\right\| + \left\|\mathbf{z}_{\iota,t,i}\right\| \leq 2\beta$ as desired. □

We can combine these two newly presented protocols with Eval to obtain a protocol for multiple evaluations. Let $u_1, \ldots, u_r \in \mathcal{R}_q$, and suppose we want to show that $f_{\iota,m}(u_\iota) = z_{\iota,m}$ for $\iota \in [r], m \in [r_\iota]$ for committed polynomials $(f_{\iota,m})_{\iota,m}$. Write $\mathrm{w}_s := \max_{\boldsymbol{\alpha}\leftarrow S^s_{\beta_C}} \|\boldsymbol{\alpha}\|_1$. The combined protocol runs (in parallel) $\mathsf{multiEval}[d, r_\iota, k, S^{r_\iota\cdot k}_{\beta_C}, \beta]$ with input $(f_{\iota,m})_{m\in[r_i]}$ for $\iota \in [r]$. This outputs r claims, which we handle by running $\mathsf{Eval}[d/k, k, S^k_{\beta_C}, \mathrm{w}_{r_\iota k}\cdot\beta]$ r-times into parallel. Finally, we run a single instance of $\mathsf{multiEval}[d/k^{h+1}, r, k, S^{rk}_{\beta_C}, (\max_\iota \mathrm{w}_{r_\iota k})\cdot \mathrm{w}^h_k\beta]$. The final complexity of this protocol is summarised in Table 6.

5.5. *Honest-Verifier Zero-Knowledge*

We provide a linear-sized Σ-protocol for the relation $\mathsf{R}_{d,\beta}$ (c.f. Equation (15)) which satisfies honest-verifier zero-knowledge. Combined with the recursive methodology described above, we can achieve zero-knowledge succinct proofs of polynomial evaluation. The strategy can identically be applied when proving knowledge of multiple polynomials at the same query point, which brings resemblance to [14].

Recall that we want to prove knowledge of the polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ of degree at most d, and the openings $(\mathbf{s}_i)_{i\in[0,d]}$ such that $f(u) = z$ and $\mathbf{A}\mathbf{s}_i + f_i\mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t}$ and $\|\mathbf{s}_i\| \leq \beta$ for $i = 0, 1, \ldots, d$. In addition to the public matrices ($\mathbf{A} \in \mathcal{R}_q^{n\times m}, \mathbf{W} \in \mathcal{R}_q^{n\times n}$), this time the index $\mathbb{i}$ contains a short basis $\mathbf{T}$ such that $\mathbf{B}\mathbf{T} = \mathbf{G}_{n(d+1)}$ where [17]

$$\mathbf{B} := \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G}\\ & \ddots & & \vdots\\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G}\end{array}\right] \quad \text{and} \quad \|\mathbf{T}\| \leq \beta_T \ . \tag{21}$$

[17] See Lemma 4.1 on how to obtain the bound on $\|\mathbf{T}\|$. For presentation, we assume the bound β_T is known.

Table 6. Parameters and complexity of the multi-evaluation protocol .

Parameters	Instantiation
m	$\geq n(1+\tilde{q}) + \omega(\log \lambda)$
δ	$q^{1/O(1)}$
$\mathfrak{s}$	$> 2Nq^{\frac{n}{m-n\tilde{q}} + \frac{2}{N(m-n\tilde{q})}}$
σ_0	$\geq \delta \mathfrak{s} N n\tilde{q} \cdot \omega(\sqrt{2(d+1)n(m-n\tilde{q})N \log t'N})$
σ_1	$\geq \delta \sigma_0 N \cdot \omega(\sqrt{m'n' \log t'N})$
β	$\geq \sigma_1 \sqrt{m'N}$
$\beta_{\mathcal{C}}$	$< \frac{1}{2}\sqrt{l/N} q^{l/N}$
w_s	$sN\beta_{\mathcal{C}}$
β_h	$\left(\max_\iota \mathrm{w}_{r_\iota k}\right) \mathrm{w}_k^h \mathrm{w}_{rk} \cdot \beta$
γ	$2^{h+2} \cdot (2\beta_{\mathcal{C}} N)^{2^{h+2}-h-3} \cdot \beta_h$
β_s	$(2\beta_{\mathcal{C}} N)^{2^{h+2}-1}$
Soundness	$(Q+1) \cdot \left(\frac{(\max_\iota r_\iota + h + r)k}{(2\beta_{\mathcal{C}}+1)^N} \right)$
Commitment size	$nN \log q \cdot \sum_\iota r_\iota$
Proof size	$\left(\sum_\iota r_\iota k N \log q\right) + r(h+1) \cdot (kN \log q) + \frac{d+1}{k^{h+2}} (rN \log q + mN \log \beta_h)$

This is the case when generating the PowerBASIS commitment in Sect. 4 since the public parameters are indeed of the form $\mathsf{crs} := (\mathbf{A}, \mathbf{W}, \mathbf{T})$.

We present the protocol in Fig. 9. The strategy follows the Fiat–Shamir with Aborts paradigm [65] using the generalised rejection sampling from [29]. That is, the prover starts by sampling uniformly random masking vector $\mathbf{g} := (g_0, \ldots, g_d) \leftarrow \mathcal{R}_q^{d+1}$, which corresponds to coefficients of a uniformly random polynomial $g \in \mathcal{R}_q[\mathsf{X}]$ of degree at most d. Then, the prover runs the PowerBASIS commitment algorithm for $\mathbf{g}$ (c.f. Fig. 4). Namely, it samples

$$\begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix} \leftarrow \mathsf{SamplePre}(\mathbf{B}, \mathbf{u}, \mathbf{T}, \sigma), \quad \text{where } \mathbf{u} := \begin{bmatrix} -g_0 \mathbf{W}^0 \mathbf{e}_1 \\ \vdots \\ -g_d \mathbf{W}^d \mathbf{e}_1 \end{bmatrix} ,$$

and sets $\mathbf{t}_y := \mathbf{G}\hat{\mathbf{t}}_y$. The first message sent by the prover is $(\mathbf{t}_y, v)$ where $v := \sum_{i=0}^{d} g_i u^i$ is the evaluation of g at the point u. Then, the verifier picks a challenge α from the challenge space $\mathcal{C} := S_{\beta_{\mathcal{C}}}$ of short polynomials of infinity norm at most $\beta_{\mathcal{C}}$.

Next, given a challenge $\alpha \leftarrow \mathcal{C}$ from the verifier, the prover computes

$$\mathbf{z}_i := \mathbf{y}_i + \alpha \mathbf{s}_i \quad \text{and} \quad h_i := g_i + \alpha f_i \quad \text{for } i = 0, 1, \ldots, d \ ,$$

and outputs $(\mathbf{z}_i, h_i)$ after performing the rejection sampling procedure. Note that the distribution of $\mathbf{z}_i$ can be written alternatively as:

$$\begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_d \\ \hat{\mathbf{t}}_z \end{bmatrix} = \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \tag{22}$$

where

$$\begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \begin{bmatrix} -g_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -g_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}, \mathbf{T}, \sigma \right) \tag{23}$$

and $\hat{\mathbf{t}} = \mathbf{G}^{-1}(\mathbf{t})$. Hence, this vector comes from a shifted discrete Gaussian distribution (over a coset of $\Lambda^{\perp}(\mathbf{B})$), where the norm of the shifted vector can be bounded by:

$$\left\| \alpha \begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \right\| \leq \beta_{\mathcal{C}} N \cdot \sqrt{(d+1)\beta^2 + n\tilde{q}N} \ . \tag{24}$$

This interpretation will be useful when analysing the rejection sampling algorithm.

Finally, the verifier checks whether

$$\begin{aligned} &\mathbf{A}\mathbf{z}_i + \mathbf{h}_i\mathbf{e}_1 = \mathbf{W}^{-i}(\mathbf{t}_y + \alpha\mathbf{t}) \quad \text{for } i = 0, 1, \ldots, d \\ &\|z_i\| \leq \beta_z \quad \text{for } i = 0, 1, \ldots, d \\ &\sum_{i=0}^{d} h_i u^i = v + \alpha z. \end{aligned}$$

In the following, we give a brief reasoning about completeness, special soundness and honest-verifier zero-knowledge.

Completeness. By careful inspection, we can deduce from the third verification check:

$$\sum_{i=0}^{d} h_i u^i = \sum_{i=0}^{d} g_i u^i + \alpha \sum_{i=0}^{d} f_i u^i = v + \alpha z \ ,$$

and from the second verification check:

$$\mathbf{A}\mathbf{z}_i + \mathbf{h}_i\mathbf{e}_1 = \mathbf{A}\mathbf{y}_i + \mathbf{g}_i\mathbf{e}_1 + \alpha(\mathbf{A}\mathbf{s}_i + \mathbf{f}_i\mathbf{e}_1) = \mathbf{W}^{-i}\mathbf{t}_y + \alpha\mathbf{W}^{-i}\mathbf{t} = \mathbf{W}^{-i}(\mathbf{t}_y + \alpha\mathbf{t}).$$

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

```
Prover                                                      Verifier
$\mathbf{s} := (\mathbf{s}_0, \ldots, \mathbf{s}_d, \hat{\mathbf{t}})$ where $\hat{\mathbf{t}} := \mathbf{G}^{-1}(\mathbf{t})$
$\mathbf{g} := (g_0, \ldots, g_d) \leftarrow \mathcal{R}_q^{d+1}$
$v := g_0 + g_1 u + \ldots + g_d u^d$
Sample $\begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix}$ as in (23)
$\mathbf{t}_y := \mathbf{G}\hat{\mathbf{t}}_y$                      $\xrightarrow{\mathbf{t}_y, v}$
                                                            $\alpha \leftarrow \mathcal{C} := S_{\beta_\mathcal{C}}$
                                      $\xleftarrow{\alpha}$
$\hat{\mathbf{t}}_z := \hat{\mathbf{t}}_y + \alpha\hat{\mathbf{t}}$
for $i = 0, 1, \ldots, d$ :
  $\mathbf{z}_i := \mathbf{y}_i + \alpha\mathbf{s}_i$
  $h_i := g_i + \alpha f_i$
$\mathbf{z} := (\mathbf{z}_1, \ldots, \mathbf{z}_d, \hat{\mathbf{t}}_z)$
$\rho \leftarrow [0, 1)$
if $\rho > \min\left(\frac{\mathcal{D}_\sigma^{m'N}(\mathbf{z})}{M \cdot \mathcal{D}_{\sigma,\alpha\mathbf{s}}^{m'N}(\mathbf{z})}, 1\right)$ :
  $\mathbf{z} := \perp$
                                      $\xrightarrow{(z_i, h_i)_{i \in [0,d]}}$
                                                            Check:
                                                            $\sum_{i \in [0,d]} h_i u^{i-1} = v + \alpha z$
                                                            $\forall i, \mathbf{A}\mathbf{z}_i + \mathbf{h}_i\mathbf{e}_1 = \mathbf{W}^{-i}(\mathbf{t}_y + \alpha\mathbf{t})$
                                                            $\forall i, \|\mathbf{z}_i\| \leq \beta_z$
```

Fig. 9. The honest-verifier zero-knowledge Σ-protocol for $\mathsf{R}_{d,\beta}$. Here, $m' := (d+1)m + n\tilde{q}$ is the width of the matrix $\mathbf{B}$ in (21).

What we have left to show is shortness of $\mathbf{z}_i$. Take the standard deviation

$$\sigma \geq \max\left(O(\sqrt{\lambda}) \cdot \beta_\mathcal{C} N \cdot \sqrt{(d+1)\beta^2 + n\tilde{q}N}, \beta_T \cdot \omega(\sqrt{N \log tN})\right) \tag{25}$$

where $t = \max(n, m)$. By Lemma 2.16, we can swap the $\mathsf{SamplePre}$ algorithm with truly sampling from a discrete Gaussian. Further, since σ is larger than the shifted vector in (24) by a factor of $O(\sqrt{\lambda})$, using rejection sampling (c.f. Lemma 2.19) we enforce the distribution of $(\mathbf{z}_0, \ldots, \mathbf{z}_d, \hat{\mathbf{t}}_z)$ from (22) to be from a discrete Gaussian on $\Lambda_\mathbf{u}^\perp(\mathbf{B})$

where

$$\mathbf{u}:=\begin{bmatrix} -(g_0+\alpha f_0)\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -(g_d+\alpha f_d)\mathbf{W}^d\mathbf{e}_1 \end{bmatrix} .$$

Thus, by Lemma 2.8, we can set $\beta_z:=\sigma\sqrt{(d+1)mN+n\tilde{q}N}$. The correctness error becomes $\approx 1/M$.

Special soundness. Given two valid transcripts $(\mathbf{t}_y, v, \alpha, (z_i, h_i))$, $(\mathbf{t}_y, v, \alpha', (z_i', h_i'))$ with distinct challenges $\alpha, \alpha' \in \mathcal{C}$, we can define

$$\bar{\mathbf{s}}_i:=\frac{\mathbf{z}_i-\mathbf{z}_i'}{\alpha-\alpha'} \quad \text{and} \quad \bar{f}_i:=\frac{h_i-h_i'}{\alpha-\alpha'} \quad \text{for } i=0,1,\ldots,d \ .$$

Note that $\|\alpha-\alpha'\|_\infty \leq 2\beta_{\mathcal{C}}$. If $\beta_{\mathcal{C}}$ is chosen according to Lemma 2.18 then we deduce that the difference is invertible over $\mathcal{R}_q$. Further, by construction

$$\bar{f}(u)=\sum_{i=0}^{d}\bar{f}_i u^i=\frac{1}{\alpha-\alpha'}\sum_{i=0}^{d}(h_i-h_i')u^i=\frac{\alpha z-\alpha' z}{\alpha-\alpha'}=z \ .$$

Furthermore, for $i=0,1,\ldots,d$ we have $\|(\alpha-\alpha')\mathbf{s}_i\|\leq 2\beta_z$ and

$$\begin{aligned}\mathbf{A}\bar{\mathbf{s}}_i+\bar{f}_i\mathbf{e}_1 &= \frac{1}{\alpha-\alpha'}\left(\mathbf{A}\mathbf{z}_i+h_i\mathbf{e}_1-(\mathbf{A}\mathbf{z}_i'+h_i'\mathbf{e}_1)\right)\\ &=\frac{1}{\alpha-\alpha'}\left(\alpha\mathbf{W}^{-i}\mathbf{t}-\alpha'\mathbf{W}^{-i}\mathbf{t}\right)=\mathbf{W}^{-i}\mathbf{t} \ .\end{aligned}$$

Thus, $(\bar{\mathbf{s}}_0,\ldots,\bar{\mathbf{s}}_d)$ along with the message $(\bar{f}_0,\ldots,\bar{f}_d)$ is a relaxed opening for the PowerBASIS commitment $\mathbf{t}$ with the relaxation factor $\alpha-\alpha'$. Hence, we can extract the witness for the relaxed relation $\tilde{\mathsf{R}}_{d,2\beta_z,1}$ in (20).

Honest-verifier zero-knowledge. We show how to simulate the transcripts when the verifier behaves honestly. To this end, we prove the following lemma which is almost analogous to [29, Lemma B.8].

Lemma 5.18. (Honest-Verifier Zero-Knowledge). *Let σ be chosen as in* (25) *where* $t=\max(n,m)$. *Then, the output distributions of $\mathcal{T}$ and $\mathcal{S}$ in Fig. 10 are statistically indistinguishable.*

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

- Hyb_0 is identical to $\mathcal{T}$ as in Fig. 10.
- Hyb_1 is identical to Hyb_0, but now we define $\hat{\mathbf{t}}_z:=\hat{\mathbf{t}}_y+\alpha\hat{\mathbf{t}}$, where $\hat{\mathbf{t}}:=\mathbf{G}^{-1}(\mathbf{t})$, and compute $\mathbf{t}_y:=\mathbf{G}\hat{\mathbf{t}}_z-\alpha\mathbf{t}$. By construction, the output distribution of Hyb_1 is identical

```
$\mathcal{T}((\mathbf{A},\mathbf{W},\mathbf{T}),(\mathbf{s}_0,\ldots,\mathbf{s}_d),(f_0,\ldots,f_d),\mathbf{t},u,z)$
$\mathbf{s} := (\mathbf{s}_0,\ldots,\mathbf{s}_d,\hat{\mathbf{t}} := \mathbf{G}^{-1}(\mathbf{t}))$
$\mathbf{g} := (g_0,\ldots,g_d) \leftarrow \mathcal{R}_q^{d+1}$
$v := g(u)$
$\begin{bmatrix}\mathbf{y}_0\\ \vdots\\ \mathbf{y}_d\\ \hat{\mathbf{t}}_y\end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c}\mathbf{A} & & & -\mathbf{G}\\ & \ddots & & \vdots\\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G}\end{array}\right], \begin{bmatrix}-g_0\mathbf{W}^0\mathbf{e}_1\\ \vdots\\ -g_d\mathbf{W}^d\mathbf{e}_1\end{bmatrix}, \mathbf{T}, \sigma\right)$
$\mathbf{t}_y := \mathbf{G}\hat{\mathbf{t}}_y$
$\alpha \leftarrow \mathcal{C}$
for $i = 0,1,\ldots,d$ :
    $\mathbf{z}_i := \mathbf{y}_i + \alpha\mathbf{s}_i$
    $h_i := g_i + \alpha f_i$
$\mathbf{z} := (\mathbf{z}_0,\ldots,\mathbf{z}_d,\hat{\mathbf{t}}_y + \alpha\hat{\mathbf{t}})$
$\rho \leftarrow [0,1)$
if $\rho > \min\left(\frac{\mathcal{D}_\sigma^{m'N}(\mathbf{z})}{M\cdot\mathcal{D}_{\sigma,\alpha\mathbf{s}}^{m'N}(\mathbf{z})}, 1\right)$:
    $\mathbf{z} := \perp$
return $(\mathbf{t}_y, v, \alpha, (h_i,\mathbf{z}_i)_{i\in[0,d]})$

$\mathcal{S}((\mathbf{A},\mathbf{W},\mathbf{T}),\mathbf{t},u,z)$
$\mathbf{h} := (h_0,\ldots,h_d) \leftarrow \mathcal{R}_q^{d+1}$
$\begin{bmatrix}\mathbf{z}_0\\ \vdots\\ \mathbf{z}_d\\ \hat{\mathbf{t}}_z\end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c}\mathbf{A} & & & -\mathbf{G}\\ & \ddots & & \vdots\\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G}\end{array}\right], \begin{bmatrix}-h_0\mathbf{W}^0\mathbf{e}_1\\ \vdots\\ -h_d\mathbf{W}^d\mathbf{e}_1\end{bmatrix}, \mathbf{T}, \sigma\right)$
$\alpha \leftarrow \mathcal{C}$
$\mathbf{t}_y := \mathbf{G}\hat{\mathbf{t}}_z - \alpha\mathbf{t}$
$v := h(u) - \alpha z$
$\rho \leftarrow [0,1)$
if $\rho > 1/M$:
    $\mathbf{z} := \perp$
return $(\mathbf{t}_y, v, \alpha, (h_i,\mathbf{z}_i)_{i\in[0,d]})$
```

Fig. 10. Simulating the transcripts from the Σ-protocol described in Fig. 10.

to Hyb_0 and

$$\begin{bmatrix}\mathbf{z}_0\\ \vdots\\ \mathbf{z}_d\\ \hat{\mathbf{t}}_z\end{bmatrix} = \begin{bmatrix}\mathbf{y}_0\\ \vdots\\ \mathbf{y}_d\\ \hat{\mathbf{t}}_y\end{bmatrix} + \alpha\begin{bmatrix}\mathbf{s}_0\\ \vdots\\ \mathbf{s}_d\\ \hat{\mathbf{t}}\end{bmatrix} \quad \text{where} \quad \begin{bmatrix}\mathbf{y}_0\\ \vdots\\ \mathbf{y}_d\\ \hat{\mathbf{t}}_y\end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c}\mathbf{A} & & & -\mathbf{G}\\ & \ddots & & \vdots\\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G}\end{array}\right], \begin{bmatrix}-g_0\mathbf{W}^0\mathbf{e}_1\\ \vdots\\ -g_d\mathbf{W}^d\mathbf{e}_1\end{bmatrix}, \mathbf{T}, \sigma\right).$$

- Hyb_2 is identical to Hyb_1, but now we compute

$$\begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_d \\ \hat{\mathbf{t}}_z \end{bmatrix} = \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix} + \alpha \begin{bmatrix} \mathbf{s}_0 \\ \vdots \\ \mathbf{s}_d \\ \hat{\mathbf{t}} \end{bmatrix} \quad \text{where} \quad \begin{bmatrix} \mathbf{y}_0 \\ \vdots \\ \mathbf{y}_d \\ \hat{\mathbf{t}}_y \end{bmatrix}$$
$$\leftarrow \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right]_{\sigma}^{-1} \left(\begin{bmatrix} -g_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -g_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix} \right).$$

  By Lemma 2.16, Hyb_1 and Hyb_2 are statistically close.

- Hyb_3 is identical to Hyb_2, but here we directly sample

$$\begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_d \\ \hat{\mathbf{t}}_z \end{bmatrix} \leftarrow \left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right]_{\sigma}^{-1} \left(\begin{bmatrix} -(g_0 + \alpha f_0)\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -(g_d + \alpha f_d)\mathbf{W}^d\mathbf{e}_1 \end{bmatrix} \right)$$

  and with probability $1-1/M$ we output $\mathbf{z}{:=}\bot$. By the generalised rejection sampling (c.f. Lemma 2.19), Hyb_3 and Hyb_2 are statistically close.

- Hyb_4 is identical to Hyb_3, except now we efficiently sample:

$$\begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_d \\ \hat{\mathbf{t}}_z \end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \begin{bmatrix} -(g_0 + \alpha f_0)\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -(g_d + \alpha f_d)\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}, \mathbf{T}, \sigma \right).$$

  As before, by Lemma 2.16 we deduce that Hyb_4 and Hyb_3 are statistically close.

- Hyb_5 is identical to Hyb_4, except now we define $h_i{:=}g_i + \alpha f_i$ for $i = 0, 1, \ldots, d$. Thus,

$$\begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_d \\ \hat{\mathbf{t}}_z \end{bmatrix} \leftarrow \mathsf{SamplePre}\left(\left[\begin{array}{ccc|c} \mathbf{A} & & & -\mathbf{G} \\ & \ddots & & \vdots \\ & & \mathbf{W}^d\mathbf{A} & -\mathbf{G} \end{array}\right], \begin{bmatrix} -h_0\mathbf{W}^0\mathbf{e}_1 \\ \vdots \\ -h_d\mathbf{W}^d\mathbf{e}_1 \end{bmatrix}, \mathbf{T}, \sigma \right).$$

  Furthermore, we set $v{:=}h(v) - \alpha z$. Clearly, the output distributions of Hyb_5 and Hyb_4 are identical.

- Hyb_6 is identical to Hyb_5, but now we sample each $h_i \leftarrow \mathcal{R}_q$ uniformly at random. Since in Hyb_5 each g_i was sampled uniformly at random from $\mathcal{R}_q$, we conclude that the output distributions of Hyb_6 and Hyb_5 are identical.

Finally, the output distribution of Hyb_6 is identical to the one by $\mathcal{S}$ which ends the proof. □

Remark 5.19. We obtain a succinct zero-knowledge proof of evaluation by modifying the Σ-protocol from Fig. 9, where instead of sending the last message $(z_i, h_i)_{i\in[0,d]}$ in the clear, we prove knowledge of the last message which satisfies the verification equations of the aforementioned protocol using the methodology from Sect. 5.1.

Remark 5.20. Similarly as in Sect. 5.4, we can combine the HVZK protocol with one round of folding to minimise the total round complexity, and thus the extracted norm growth. This yields an almost identical protocol as in [14].

5.6. *Polynomial Commitments over Finite Fields*

So far we showed how to commit and prove evaluations of polynomials over the cyclotomic ring $\mathcal{R}_q$. We now present how to build polynomial commitments over finite fields of specific form. This will be useful when combining with Polynomial IOPs to obtain succinct arguments of knowledge.

Suppose q is a prime which satisfies $q \equiv 2N/l + 1 \pmod{4N/l}$ for some positive divisor l of N. Then by [70, Corollary 1.2], the polynomial $X^N + 1$ factors as:

$$X^N + 1 \equiv \prod_{i=1}^{N/l}(X^l - r_i) \pmod{q}$$

for distinct $r_i \in \mathbb{Z}_q^*$ where all $X^l - r_i$ are irreducible in the ring $\mathbb{Z}_q[X]$. Further, by the Chinese Remainder Theorem, there exists a ring isomorphism $\varphi : \mathbb{F}^{N/l} \to \mathcal{R}_q$ where $\mathbb{F}$ is a finite field of size q^l. Consider the restricted function:

$$\begin{aligned} \varphi_{\mathbb{F}} : \mathbb{F} &\to \mathcal{R}_q \\ x &\mapsto \varphi(x, 0, \ldots, 0). \end{aligned}$$

By construction, the image of $\varphi_{\mathbb{F}}$ can be described as

$$\mathcal{S}_q := \mathsf{Im}(\varphi_{\mathbb{F}}) = \{\varphi(x, 0, \ldots, 0) : x \in \mathbb{F}\}\ .$$

The following simple lemma states that $\mathcal{S}_q$ is an ideal of $\mathcal{R}_q$.

Lemma 5.21. *The set $\mathcal{S}_q \subseteq \mathcal{R}_q$ defined above is an ideal.*

Proof. The fact that $\mathcal{S}_q$ is an additive subgroup of $\mathcal{R}_q$ follows directly from the additively homomorphic properties of φ. Now let $a \in \mathcal{S}_q$, i.e. $\varphi(x, 0, \ldots, 0) = a$ for some $x \in \mathbb{F}$. Further, take arbitrary $\gamma \in \mathcal{R}_q$ and let $(\gamma_1, \ldots, \gamma_{N/l}) := \varphi^{-1}(\gamma)$. Then, by the multiplicative homomorphism of φ we get

$$\gamma \cdot a = \varphi(\gamma_1, \ldots, \gamma_{N/l}) \cdot \varphi(x, 0, \ldots, 0) = \varphi(\gamma_1 x, 0, \ldots, 0) = \varphi_{\mathbb{F}}(\gamma_1 x) \in \mathcal{S}_q\ ,$$

which concludes the proof. □

Suppose we want to commit to a polynomial $F := \sum_{i=0}^{d} F_i \mathsf{X}^i \in \mathbb{F}[\mathsf{X}]$ of degree at most d, and prove evaluation $F(x) = y$ for $x, y \in \mathbb{F}$. By the homomorphic property of $\varphi_{\mathbb{F}}$, this is equivalent to proving $f(u) = z$ over $\mathcal{R}_q$ where

$$\begin{cases} f[\mathsf{X}] = \sum_{i=0}^{d} \varphi_{\mathbb{F}}(F_i)\mathsf{X}^i \in \mathcal{S}_q[\mathsf{X}] \\ u = \varphi_{\mathbb{F}}(x) \in \mathcal{S}_q \\ z = \varphi_{\mathbb{F}}(y) \in \mathcal{S}_q \end{cases} .$$

Hence, we can commit to the polynomial $f \in \mathcal{R}_q[\mathsf{X}]$ and prove evaluation of u at the point z as before. What is new is that we additionally need to prove that coefficients of f indeed lie in $\mathcal{S}_q$. Therefore, we are interested in a stronger relation:

$$\left\{ ((\mathbf{A}, \mathbf{W}), (\mathbf{t}, u, z), (f, (\mathbf{s}_i)_i)) \,\middle|\, \begin{array}{c} f(u) = z \wedge f \in \mathcal{S}_q[\mathsf{X}] \\ \forall i \in [0, d], \mathbf{A}\mathbf{s}_i + f_i \mathbf{e}_1 = \mathbf{W}^{-i}\mathbf{t} \\ \wedge \|\mathbf{s}_i\| \le \beta \end{array} \right\} . \quad (26)$$

We show how to modify the protocol in Fig. 6 to accommodate for this change. Actually, the interaction between the prover and the verifier stays the same but the verifier additionally performs a check whether the final polynomial $f_h \in \mathcal{R}_q[\mathsf{X}]$ sent by the prover has coefficients in $\mathcal{S}_q$.

Completeness follows by induction. We start with the initial polynomial $f_0 := f \in \mathcal{S}_q[\mathsf{X}]$. Then for each $r \in [h]$, the prover computes the polynomial $f_r \in \mathcal{R}_q[\mathsf{X}]$ as a linear combination of "partial terms" of f_{r-1}:

$$f_r := \sum_{t \in [k]} \alpha_{r,t} f_{r-1,t} .$$

If $f_{r-1} \in \mathcal{S}_q[\mathsf{X}]$, then by Lemma 5.21 we deduce that $f_r \in \mathcal{S}_q[\mathsf{X}]$.

To argue (coordinate-wise) special soundness, consider the extractor in the proof of Lemma 5.6. The coefficients of the extracted polynomial f are computed as

$$f_{2i} := \frac{\alpha_1 g_{0,i} - \alpha_0 g_{1,i}}{\alpha_1 - \alpha_0}, \quad f_{2i+1} := \frac{g_{0,i} - g_{1,i}}{\alpha_0 - \alpha_1} \quad \text{for } i \in [0, d/2] .$$

If polynomials g_0 and g_1 have coefficients in $\mathcal{S}_q$, then again by Lemma 5.21 we can deduce that $f \in \mathcal{S}_q[\mathsf{X}]$. Identical argument holds when analysing Lemma 5.11.

Finally, to support honest-verifier zero-knowledge in Fig. 9, we let the prover pick uniformly random elements g_i from $\mathcal{S}_q$ instead of $\mathcal{R}_q$ in order to fully mask the coefficients f_i. Thus, by construction and Lemma 5.21, $h_i = g_i + \alpha f_i \in \mathcal{S}_q$ for all $i = 0, \dots, d$. Hence, the verifier additionally performs the check whether coefficients h_i lie in $\mathcal{S}_q$.

Remark 5.22. This technique can be extended to simultaneously prove N/ℓ polynomial evaluations and it was recently used in [25, Section 7].

6. Concrete Instantiation and Applications to Marlin

As described in Sect. 2, we pick a prime modulus $q \equiv 5 \pmod 8$, and thus $\ell = N/2$.

Hardness of PowerBASIS. In parameter selection, we make a heuristic assumption that PowerBASIS is exactly as hard as MSIS. Hence, one should treat our computed sizes only as intuition on how practical the polynomial commitment is.

In the literature, hardness of the MSIS problems is often analysed identically as the plain SIS since, so far, the best known attacks do not make use of the algebraic structure of the polynomial ring [6]. We follow the methodology from Dilithium [42, Appendix C]. That is, $\mathsf{MSIS}_{n,m,N,q,\beta}$ for matrix $\mathbf{A}$ is equivalent to finding a non-trivial vector of norm smaller than β in the lattice $\Lambda := \Lambda^{\perp}(\mathbf{A})$. In order to find short non-trivial vectors in Λ, we apply the Block-Korkine-Zolotarev algorithm (BKZ) [36,77]. As a subroutine, BKZ uses an algorithm for the shortest vector problem (SVP) in lattices of dimension b, where b is called the block size. If we apply the best known algorithm for solving SVP with no memory constraints by Becker et al. [16], the time required by BKZ to run on the mN-dimensional lattice Λ with block size b is given by $8mN \cdot 2^{0.292b+16.4}$ (one also considers a more *conservative* variant with runtime $2^{0.292b}$). The algorithm outputs a vector of norm $\delta_{\mathsf{rhf}}^{mN} \det(\Lambda)^{\frac{1}{mN}}$ where δ_{rhf} is the root Hermite factor and it is given by

$$\delta_{\mathsf{rhf}} = \left(\frac{b(\pi b)^{1/b}}{2\pi e} \right)^{\frac{1}{2(b-1)}} . \tag{27}$$

For our usual parameter selection, the probability that a random matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ is of full rank is overwhelming (see [45, Appendix C]) and thus $\det(\Lambda) = q^{nN}$. Next, Micciancio and Regev [74] show that

$$\delta_{\mathsf{rhf}}^{mN} \det(\Lambda)^{\frac{1}{mN}} = \delta_{\mathsf{rhf}}^{mN} q^{\frac{nN}{mN}} \geq 2^{2\sqrt{nN \log q \log \delta}}$$

and the equality holds when $mN = \sqrt{nN \log q / \log \delta}$. Hence, given a bound $\beta < q$ we compute δ_{rhf} from the equation $\beta = 2^{2\sqrt{nN \log q \log \delta}}$. Next, we calculate the minimum block size b from Equation (27), and thus we get the total time for BKZ to solve $\mathsf{MSIS}_{n,m,N,q,\beta}$. Hereafter, we will refer to the "aggressive strategy" to set PowerBASIS as the one using the estimate from Becker et al. [16], and to the the "conservative strategy" as the one using $2^{0.292b}$.

Parameters. Using a combination of randomised and exhaustive search, we found parameters for the schemes in Theorem 5.9 and Theorem 5.13. In Table 7 we detail the parameters obtained for the scheme presented in Theorem 5.13 and in Table 8 for that in Theorem 5.9. We also make use of the techniques in [4, Sec 5.5, Sec 6] to further optimise the parameters. Namely, we use the transformation therein to convert our polynomial commitment scheme to one that supports prime order fields, and we use deterministic preimage sampling (since in this section we are not concerned with zero-knowledge). All the tables use the "aggressive strategy" to set parameters. We stress that these parameters are presented to give the reader an indication of the concrete efficiency of the scheme.

Table 7. Parameters and concrete sizes for the polynomial commitment described in Theorem 5.13 .

k	h	d	λ	Q	n	m	N	δ	$\log q$	$2\gamma\beta_s$	β	$\mathfrak{s}$	$\beta_{\mathcal{C}}$	β_h	$\lvert\mathbf{t}\rvert$	$\lvert\pi\rvert$
128	2	2^{14}	80	64	87	1697	64	17	204	203	148	34	1	166	139 KB	2.6 MB
256	3	2^{30}	80	64	139	2919	64	24	322	322	196	49	1	225	350 KB	6.9 MB
128	2	2^{20}	128	64	117	2106	64	20	229	222	163	42	2	182	209 KB	3.4 MB
256	3	2^{30}	128	64	168	3528	64	25	339	338	202	53	2	234	445 KB	8.3 MB

δ, norms and standard deviation given in log form.

Table 8. Parameters and concrete sizes for the interactive polynomial commitment in Theorem 5.9 .

h	d	λ	n	m	N	δ	$\log q$	$2\gamma\beta_s$	β	$\mathfrak{s}$	β_h	t	$\lvert\mathbf{t}\rvert$	$\lvert\mathsf{cc}\rvert$
11	2^{20}	80	17	383	512	22	314	314	170	48	192	13	333 KB	64.1 MB
21	2^{30}	80	27	608	512	38	548	523	249	78	291	15	925 KB	183.7 MB
8	2^{20}	128	3	54	4096	27	320	320	191	65	207	13	480 KB	105.4 MB
20	2^{30}	128	17	408	1024	33	515	515	234	72	274	20	1.07 MB	324.4 MB

δ, norms and standard deviation given in log form.

The commitments have sizes on the order of hundreds of kilobytes, while evaluation proofs are on the order of a few megabytes, and so are larger than desirable in most applications. We also emphasise that the assumption that the hardness of PowerBASIS is as hard as MSIS is an heuristic, and thus, until this heuristic is backed or disproved by sufficient cryptanalysis, the sizes should be considered as an optimistic lower bound.

Applications to Polynomial IOPs. Marlin [37] is a widely deployed preprocessing zkSNARK. As many modern constructions, Marlin is constructed by combining two ingredients:

- a polynomial interactive oracle proof (PIOP) (therein a algebraic holographic proof);
- and a polynomial commitment scheme.

An interactive oracle proof (IOP) is a generalisation of both probabilistically checkable proofs and interactive proofs. Informally, they are interactive protocols between a prover and a verifier, in which the prover sends *oracle messages*, which the verifier is allowed to not read in their entirety. A PIOP is simply an IOP where the prover messages are guaranteed to be (low degree) polynomials. IOPs and PIOPs are *information theoretic object*, and as such inherit a number of efficiency limitations (for example, IOP proof length are required to be at least linear in the size of the instance), but can be compiled using cryptography (see [18]) to obtain arguments that are both asymptotically and concretely efficient. Informally, to compile a PIOP into an interactive argument, the prover can commit to each polynomial oracle using a polynomial commitment scheme, and then prove to the verifier that the evaluations (at points chosen by the verifier) are as claimed. Then, to obtain a NARK, we can apply the Fiat–Shamir transformation to this interactive protocol. We can thus aim to use our polynomial commitment scheme in Theorem 5.13 as an ingredient of Marlin to obtain a zkSNARK for R1CS. Let d denote the size of the R1CS instance that we aim to prove. As detailed in [37, Section

Table 9. Parameters and concrete sizes for Marlin when instantiated with the commitment described in Theorem 5.13 with amortisation as in Table 6.

k	h	d	λ	Q	n	m	N	δ	$\log q$	$2\gamma\beta_s$	β	$\mathfrak{s}$	$\beta_{\mathcal{C}}$	β_h	$\lvert\mathbf{t}\rvert$	$\lvert\pi\rvert$
[32, 32, 48]	1	2^{20}	80	64	138	2691	64	27	324	321	181	50	1	224	6.1 MB	6.6 MB
[32, 128, 128, 192]	2	2^{30}	80	64	224	5376	64	34	517	517	231	66	1	292	15.9 MB	19.6 MB
[32, 32, 48]	1	2^{20}	128	64	186	3627	64	28	343	339	189	54	2	234	8.8 MB	8.6 MB
[32, 128, 128, 192]	2	2^{30}	128	64	271	6504	64	36	562	552	244	74	2	309	20.9 MB	23.6 MB

k in this case reports the folding factors at different rounds of the protocol. δ, norms and standard deviation given in log form. Folding factor varies across rounds as mentioned in Remark 5.4.

9], Marlin after compilation has commitments to 19 total polynomials of degree at most $6d$. The prover has then to produce 19 evaluations proofs for these polynomials, at three distinct points. We can thus apply the techniques in Sect. 5.4 to batch evaluations together and amortise the cost of the last round. In Table 9 we compute parameters for Marlin instantiated using our polynomial commitment scheme and the PIOP therein described. Again, these sizes are meant to give a rough estimate of the concrete efficiency of the scheme, and the same caveats apply as with the polynomial commitment scheme. We also note that Marlin operates over fields with a large multiplicative (or additive) subgroup with smooth order, which imposes an additional requirement on the size of q. Since our moduli are again quite large, this additional requirement is immaterial.

7. Coordinate-Wise Special Soundness Implies Knowledge Soundness

In this section we show that coordinate-wise special soundness implies knowledge soundness for multi-round protocols by extending the techniques presented in [7,8] (cf. Lemma 2.31). We also show that our knowledge extractor is exponentially more efficient than the generic extractor introduced by Attema et al. [11]. The intuition behind this efficiency is that the extractor samples challenges in a certain way that is the most plausible for having a monotone structure. For reference, we will use identical terminology as in [7, Section 6.4]. We always use ℓ to denote the number of coordinates, and we define a challenge space $\mathcal{C}:=S^\ell$ for this and the following section.

7.1. *Σ-Protocols*

We start by considering three-round public coin interactive proofs, i.e. Σ-protocols. Namely, let $\mathcal{A} : \mathcal{C} \to \{0,1\}^*$ be an arbitrary (probabilistic) algorithm, and $V : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$ be the verification function. Then, $\mathcal{A}$ has naturally defined success probability:

$$\epsilon^V(\mathcal{A}) := \Pr_{\mathbf{c} \leftarrow \mathcal{C}}[V(\mathbf{c}, \mathcal{A}(\mathbf{c})) = 1].$$

The standard interpretation is that $\mathcal{A}$ is a malicious prover, which tries to convince the verifier of the underlying Σ-protocol.

Knowledge Extractor for CWSS Σ-Protocols

```
$\underline{\mathcal{E}^{\mathcal{A}}}$
1. $\mathbf{c}_0 := (c_{0,1}, \ldots, c_{0,\ell}) \leftarrow \mathcal{C}$
2. $y_0 \leftarrow \mathcal{A}(\mathbf{c}_0)$
3. If $V(\mathbf{c}_0, y_0) = 0$, then abort
4. For $i = 1, \ldots, \ell$, repeat:
    (a) Sample $c_i^* \leftarrow S \backslash \{c_{0,i}\}$ without replacement
    (b) $\mathbf{c}_i := (c_{0,1}, \ldots, c_{0,i-1}, c_i^*, c_{0,i+1}, \ldots, c_{0,\ell})$
    (c) $y_i \leftarrow \mathcal{A}(\mathbf{c}_i)$
    (d) If $V(\mathbf{c}_i, y_i) = 0$, go to Step 4(a)
   until $k-1$ pairs $(\mathbf{c}_{i,j}, y_{i,j})_{j \in [k-1]}$ s.t. $V(\mathbf{c}_{i,j}, y_{i,j}) = 1$ are collected, or until all $c_i^*$ have
   been tried (in the latter case abort)
5. Return the corresponding $(\mathbf{c}_0, y_0), (\mathbf{c}_{i,j}, y_{i,j})_{i \in [\ell], j \in [k-1]}$
```

Fig. 11. Knowledge extractor for the proof of Lemma 7.1.

The following lemma describes how to extract from CWSS Σ-protocols. The proof methodology is identical to [7, Lemma 6.5].

Lemma 7.1. *Let $k, \ell \in \mathbb{N}$, and S be a finite set of cardinality N. Define $\mathcal{C} := S^\ell$ and take any verification function $V : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$ with the following properties: the algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} : \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most $\ell(k-1)+1$ queries to $\mathcal{A}$ and with probability at least*

$$\epsilon^V(\mathcal{A}) - \frac{\ell(k-1)}{N}$$

outputs $\ell(k-1)+1$ pairs $(\mathbf{c}_0, y_0), \ldots, (\mathbf{c}_{\ell(k-1)}, y_{\ell(k-1)})$ such that $V(\mathbf{c}_i, y_i) = 1$ for all $i \in [0, \ell(k-1)]$ and $\{\mathbf{c}_0, \ldots, \mathbf{c}_{\ell(k-1)}\} \in \mathsf{SS}(S, \ell, k)$.

Proof. The extractor $\mathcal{E}^{\mathcal{A}}$ is defined in Fig. 11. We denote by $\mathbf{C}_0 := (C_{0,1}, \ldots, C_{0,\ell})$ the random variable for the first challenge sampled by $\mathcal{E}$. Also, we denote $\Gamma = V(\mathbf{C}_0, \mathcal{A}(\mathbf{C}_0))$. In particular, $\Pr[\Gamma = 1] = \epsilon^V(\mathcal{A})$.

Let T be the number of $\mathcal{A}$-queries made by $\mathcal{E}$. For $i \in [\ell]$, define T_i to be the number of queries made during the i-th iteration of the loop. By linearity of expectation, we have $\mathbb{E}[T] = 1 + \sum_{i=1}^{\ell} \mathbb{E}[T_i]$. Also, if $\Gamma = 0$ then $T_i = 0$. Moreover, to bound $\mathbb{E}[T_i]$, we reuse the results from [7] and write $\mathbb{E}[T_i] \leq k-1$. Therefore, $\mathbb{E}[T] \leq 1 + \ell(k-1)$.

We now move to the success probability of $\mathcal{E}^{\mathcal{A}}$. Define the random variable $X_i := |\{x \in S : V(\mathbf{C}_i(x), \mathcal{A}(\mathbf{C}_i(x)) = 1\}|$, where $\mathbf{C}_i(x) := (C_{0,1}, \ldots, C_{0,i-1}, x, C_{0,i+1}, \ldots, C_{0,\ell})$. Note that the extractor succeeds with probability $\Pr[\Gamma = 1 \wedge (\wedge_{i=1}^{\ell} X_i \geq k)]$. Now, by the union bound we have

$$\begin{aligned}
\Pr[\Gamma = 1 \wedge (\wedge_{i=1}^{\ell} X_i \geq k)] &= \Pr[\Gamma = 1] - \Pr[\Gamma = 1 \wedge (\vee_{i=1}^{\ell} X_i \leq k-1)] \\
&\geq \Pr[\Gamma = 1] - \sum_{i=1}^{\ell} \Pr[\Gamma = 1 \wedge X_i \leq k-1] \\
&\geq \Pr[\Gamma = 1] - \sum_{i=1}^{\ell} \Pr\left[\Gamma = 1 \mid X_i \leq k-1\right] \cdot \Pr\left[X_i \leq k-1\right] \\
&\geq \Pr[\Gamma = 1] - \frac{\ell(k-1)}{N}.
\end{aligned}$$

The statement follows by recalling that $\Pr[\Gamma = 1] = \epsilon^V(\mathcal{A})$. □

7.2. *Multi-Round Protocols*

Next, we move on to $(2\mu + 1)$-round interactive proofs. To this end, we consider an arbitrary probabilistic algorithm $\mathcal{A} : \mathcal{C} \times \cdots \times \mathcal{C} \to \{0,1\}^*$, and a verification function $V : \mathcal{C} \times \cdots \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Similarly as before, we define

$$\epsilon^V(\mathcal{A}) := \Pr\left[V(\bar{\mathbf{c}}, \mathcal{A}(\bar{\mathbf{c}}))\right],$$

where $\bar{\mathbf{c}} \leftarrow \mathcal{C}^{\mu}$.

Now, the goal of the extractor is, given oracle access to $\mathcal{A}$, to efficiently extract a tree of transcripts, as in Definition 2.30. We will follow the footsteps of [7, Lemma 6.6] and recursively use Lemma 7.1 for the Σ-protocol case.

Lemma 7.2. *Let $k, \ell, \mu \in \mathbb{N}$, and S be a finite set of cardinality N. Define $\mathcal{C} := S^{\ell}$ and take any verification function $V : \mathcal{C} \times \cdots \times \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$ with the following properties: the algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} : \mathcal{C} \times \cdots \times \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most $K := (\ell(k-1)+1)^{\mu}$ queries to $\mathcal{A}$ and with probability at least*

$$\epsilon^V(\mathcal{A}) - \mu \cdot \frac{\ell(k-1)}{N}$$

outputs K pairs $(\mathbf{c}_i, y_i)_{i \in [K]}$ such that $V(\mathbf{c}_i, y_i) = 1$ for all $i \in [K]$ and $(\mathbf{c}_i)_{i \in [K]}$ form a tree of challenges as described in Definition 2.30.

Proof. We prove the statement by induction on $\mu \geq 1$. For $\mu = 1$, we can apply Lemma 7.1. Hence, assume the lemma holds for $\mu = M \geq 1$ and focus on the case $\mu = M + 1$.

For $\mathbf{c} \in \mathcal{C}$, we define $\mathcal{A}_{\mathbf{c}}$ to be the algorithm, which takes input $(\mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(\mu)}) \in \mathcal{C}^{\mu-1}$, and outputs $\mathcal{A}(\mathbf{c}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(\mu)})$. We similarly define a verification function $V_{\mathbf{c}}$ as $V_{\mathbf{c}}(\mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(\mu)}, y) := V(\mathbf{c}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(\mu)}, y)$. By the induction hypothesis, there exists an extractor $\mathcal{E}_{\mu-1}^{\mathcal{A}_{\mathbf{c}}}$, that given oracle access to $\mathcal{A}_{\mathbf{c}}$, outputs a set $\mathcal{Y}$ of $K' := (\ell(k-1)+1)^{\mu-1}$ pairs $(\mathbf{c}_i, y_i) \in \mathcal{C}^{\mu-1} \times \{0,1\}^*$, such that $V_{\mathbf{c}}(\mathbf{c}_i, y_i) = 1$ for all $i \in [K']$ and $(\mathbf{c}_i)_{i \in [K']}$

form a tree of challenge vectors of level $\mu - 1$, with probability at least

$$\epsilon^{V_{\mathbf{c}}}(\mathcal{A}_{\mathbf{c}}) - (\mu - 1) \cdot \frac{\ell(k-1)}{N},$$

and makes at most K' queries to $\mathcal{A}_{\mathbf{c}}$. Now, we define $W : \mathcal{C} \times \{0,1\}^* \to \{0,1\}$ as $W(\mathbf{c}, \mathcal{Y}) = 1$ if and only if $\mathcal{Y}$ satisfies all the properties above. Further, define $\mathcal{B}^{\mathcal{A}} : \mathcal{C} \to \{0,1\}^*$ to be the algorithm, which takes as input $\mathbf{c} \in \mathcal{C}$, and runs $\mathcal{E}_{\mu-1}^{\mathcal{A}_{\mathbf{c}}}$. By Lemma 7.1, there is an extractor $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ that aims to output $\ell(k-1)+1$ pairs $(\mathbf{c}_0^{(1)}, \mathcal{Y}_0), \ldots, (\mathbf{c}_{\ell(k-1)}^{(1)}, \mathcal{Y}_{\ell(k-1)})$ such that $W(\mathbf{c}_i^{(1)}, \mathcal{Y}_i) = 1$ for $i \in [0, \ell(k-1)]$ and $(\mathbf{c}_i^{(1)})_{i \in [0,\ell(k-1)]} \in \mathsf{SS}(S, \ell, k)$. Note that such a set of $\ell(k-1)+1$ trees of challenges is also a tree of challenges of level μ. Thus, we define the extractor $\mathcal{E}^{\mathcal{A}}$ to simply run $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$.

We first discuss the expected number of queries to $\mathcal{A}$ made by $\mathcal{E}$. By Lemma 7.1, $\mathcal{E}_{\mu-1}^{\mathcal{A}_{\mathbf{c}}}$ makes at most $\ell(k-1)+1$ queries to $\mathcal{B}^{\mathcal{A}}$ in expectation. Then, by induction hypothesis, $\mathcal{B}^{\mathcal{A}}$ makes at most K' calls to $\mathcal{A}$ in expectation. Hence, the total expected number of $\mathcal{A}$-queries is at most $(\ell(k-1)+1)K' = (\ell(k-1)+1)^{\mu}$. As for the success probability, we know from Lemma 7.1 and induction hypothesis that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ succeeds with probability at least ϵ' where

$$\begin{aligned} \epsilon' &\geq \epsilon^{W}(\mathcal{B}^{\mathcal{A}}) - \frac{\ell(k-1)}{N} \\ &\geq \mathbb{E}_{\mathbf{c}}\left[\Pr[\mathcal{E}_{\mu-1}^{\mathcal{A}_{\mathbf{c}}} \neq \perp]\right] - \frac{\ell(k-1)}{N} \\ &\geq \mathbb{E}_{\mathbf{c}}\left[\epsilon^{V_{\mathbf{c}}}(\mathcal{A}_{\mathbf{c}}) - (\mu-1)\frac{\ell(k-1)}{N}\right] - \frac{\ell(k-1)}{N} \\ &\geq \epsilon^{V}(\mathcal{A}) - \mu\frac{\ell(k-1)}{N}, \end{aligned}$$

which concludes the proof. □

Finally, Lemma 2.31 follows straightforwardly from Lemma 7.2.

7.3. *Comparison with the Generic Extractor*

The notion of coordinate-wise special soundness is a specific case of general notion of Γ-out-of-$\mathcal{C}$ special soundness introduced by Attema et al. [11]. We refer to their notation and definitions in this section. In their work, a generic knowledge extractor for Γ-out-of-$\mathcal{C}$ special-sound protocols is presented. As they note, as long as the expected runtime of the generic knowledge extractor is polynomial, Γ-out-of-$\mathcal{C}$ special soundness implies knowledge soundness. Although the generic extractor can be useful in many settings, we show that, for a set S and $\ell, k \in \mathbb{N}$, $\ell > 1$, when the generic extractor runs to get a set of accepting challenges $C \in \Gamma \subseteq 2^{\mathcal{C}}$, where challenge set $\mathcal{C}{:=}S^{\ell}$ and

$$\Gamma{:=}\{C : \exists X \in \mathsf{SS}(S, \ell, k), X \subseteq C\} \ ,$$

it is not guaranteed that the generic extractor can output the witness in expected polynomial time. Notice that Γ denotes the monotone structure here.

To that end, let us first recall two crucial definitions from [11]: the set of useful elements and t-value. Then, we prove a lower bound on t-value, which shows the upper bound on the expected runtime of the generic extractor fails to be useful in the proof of knowledge soundness.

Definition 7.3. (Useful Elements, [11]). For a monotone structure $(\Gamma, \mathcal{C})$, we define the following function:

$$U_{\Gamma} : 2^{\mathcal{C}} \rightarrow 2^{\mathcal{C}}, X \mapsto \{c \in \mathcal{C} \setminus X : \exists A \in \Gamma \ s.t. \ X \subset A \wedge A \setminus \{c\} \notin \Gamma\} \ .$$

Definition 7.4. (t-value, [11]). Let $(\Gamma, \mathcal{C})$ be a monotone structure and $S \subseteq \mathcal{C}$. Then

$$t_{\Gamma}(X){:=}\max \left\{ t \in \mathbb{N}_0 : \begin{array}{c} \exists c_1, \ldots, c_t \in \mathcal{C} \ s.t. \\ \forall i, c_i \in U_{\Gamma}(X \cup \{c_1, \ldots, c_{i-1}\}) \end{array} \right\} \ .$$

Further,

$$t_{\Gamma}{:=}t_{\Gamma}(\emptyset) \ .$$

We aim to find a bound for t_{Γ} since Lemma 5 from [11] states that the expected runtime of the generic extractor is upper bounded by $2t_{\Gamma} - 1$. For simplicity, let $k = 2$. We claim that $t_{\Gamma} \geq |S|^{\ell-1} + 1$.

For $d, d' \in S, d \neq d'$, and $\mathbf{v} = (v_2, \ldots, v_{\ell}) \in S^{\ell-1}$, consider the sets

$$\mathcal{A}_d{:=}\{d\} \times S^{\ell-1} \text{ and } \mathcal{B}_{d',\mathbf{v}}{:=}\left\{(d', \mathbf{v})\right\} \ .$$

Although $\mathcal{B}_{d',\mathbf{v}}$ has only one member, it is convenient for our proof to use set notation. Now, notice that t_{Γ} is defined on the longest possible sequence of challenges such that each challenge is in the set of useful elements of all the previous ones. We argue that, for $d, d' \in S, d \neq d'$, and $\mathbf{v} = (v_2, \ldots, v_{\ell}) \in S^{\ell-1}$, the challenge sequence

$$\mathbf{c}_1, \ldots, \mathbf{c}_t, \quad \forall i \in [t-1], \ \mathbf{c}_i \in \mathcal{A}_d, \quad \mathbf{c}_t \in \mathcal{B}_{d',\mathbf{v}},$$

fulfills the mentioned conditions, where $t{:=}|\mathcal{A}_d| + 1$. This implies that $t_{\Gamma} \geq |\mathcal{A}_d| + 1 = |S|^{\ell-1} + 1 \geq \frac{|\mathcal{C}|}{|S|}$. We are left to prove that the specified sequence meets the constraint in the definition of t_{Γ}. First, observe that,

$$\forall d, d' \in S, d \neq d', \forall (d, \mathbf{v}) \in \mathcal{A}_d \implies \mathcal{A}_d \cup \mathcal{B}_{d',\mathbf{v}} \in \Gamma \wedge \mathcal{A}_d \cup \mathcal{B}_{d',\mathbf{v}} \setminus \{(d, \mathbf{v})\} \notin \Gamma \ .$$

Hence, $\mathcal{A}_d \subseteq U_\Gamma(\emptyset)$. Similarly, for any $T \subseteq \mathcal{A}_d$,

$$\forall d, d' \in S, d \neq d', \forall (d, \mathbf{v}) \in \mathcal{A}_d \setminus T \implies \mathcal{A}_d \cup \mathcal{B}_{d',\mathbf{v}} \in \Gamma \wedge \mathcal{A}_d \cup \mathcal{B}_{d',\mathbf{v}} \setminus \{(d, \mathbf{v})\} \notin \Gamma \ .$$

So, $\mathcal{A}_d \setminus T \subseteq U_\Gamma(T)$. Finally, given that $|\mathcal{B}_{d',\mathbf{v}}| = 1$, for any $\mathbf{v} \in S^{\ell-1}$,

$$\forall d, d' \in S, d \neq d', \forall \mathbf{v} \in S^{\ell-1} \implies \mathcal{A}_d \cup \mathcal{B}_{d',\mathbf{v}} \in \Gamma \wedge \mathcal{A}_d \notin \Gamma \ .$$

Therefore, $\mathcal{B}_{d',\mathbf{v}} \subseteq U_\Gamma(\mathcal{A}_d)$. So, a sequence of all members of $\mathcal{A}_d$ (with any order) followed by the only member of $\mathcal{B}_{d',\mathbf{v}}$ is a valid sequence for t_Γ.

In summary, we proved that the expected runtime of the generic extractor can be almost as big as the challenge space for a coordinate-wise special-sound protocol (and thus it could be super-polynomial in the security parameter). Consequently, we cannot argue knowledge soundness for the protocol by leveraging Γ-out-of-$\mathcal{C}$ special soundness and the generic extractor.

8. Knowledge Soundness of a Fiat–Shamir-Transformed Coordinate-Wise Special-Sound Multi-round Protocol

In this section, we show there is an efficient knowledge extractor for the non-interactive protocol obtained by applying Fiat–Shamir transformation on a ℓ-coordinate-wise k-special-sound multi-round protocol.

In the following, we leverage the approach presented by Attema et al. [10]. Namely, we define and analyze an abstract sampling game where the extractor plays the role of a sampler who tries to find "good" entries. In the meantime, we elaborate on how this game relates to knowledge extraction. For reference, we use notation from [10]. Furthermore, we prove a slightly different version of Lemmata 2 and 5 from [10] for our specific reprogramming of the random oracle. As Lemmata 3 and 6 from [10] are independent of how the random oracle gets reprogrammed, we only use them as they are.

8.1. *Analysis of the Abstract Sampling Game*

Figure 12 shows the mentioned sampling game. Similar to [10], the sequence of $j_1, \ldots, j_U \in \{1, \ldots, N\}^\ell$ specifies the function table of the random oracle. Notice that the cardinality of the input space of the random oracle is U. Each entry of M determines what the first message chosen by the deterministic prover would be and if it would be an accepting transcript. For a given sequence of $j_1, \ldots, j_U$, we can extract when the following happens. First, $M(j_1, \ldots, j_U) = (1, i)$ for some $i \in \{1, \ldots, U\}$, and second, by reprogramming j_i to some j'_i (which is different from j_i coordinate-wisely) for enough many times, $M(j_1, \ldots, j'_i, \ldots, j_U) = (1, i)$. In other words, the prover chooses the same first message when given each of these different functional tables of the random oracle, and by coordinate-wise special soundness, it is feasible to extract.

Abstract Sampling Game

Parameters: $\ell, k, N, U \in \mathbb{N}$, and a $U\ell$-dimensional array M with entries in $M(j_1, \ldots, j_U) \in \{0,1\} \times \{1, \ldots, U\}$ for all tuples $j_1, \ldots, j_U \in \{1, \ldots, N\}^\ell$.

- Sample $(j_1, \ldots, j_U) \in \left(\{1, \ldots, N\}^\ell\right)^U$ uniformly at random and set $(v, i) = M(j_1, \ldots, j_U)$.
- If $v = 0$, abort.
- Else, for $1 \leq l \leq \ell$, repeat
  - sample $j'_l \in \{1, \ldots, N\} \setminus \{j_{i,l}\}$ (without replacement),
  - set $j' = (j_{i,1}, \ldots, j_{i,l-1}, j'_l, j_{i,l+1}, \ldots, j_{i,\ell})$
  - compute $(v', i') = M(j_1, \ldots, j_{i-1}, j', j_{i+1}, \ldots, j_U)$,

  until either $k-1$ additional entries equal to $(1, i)$ have been found or until all indices j'_l have been tried.

Fig. 12. Abstract sampling game.

Similar to [10], we define the functions $a_i, a_{i,l} : \left(\{1, \ldots, N\}^\ell\right)^U \to \mathbb{N}_{\geq 0}$ where

$$a_{i,l} : j \mapsto \left|\left\{j' : \left(\forall (i', l') \in [U] \times [\ell] \setminus \{(i,l)\},\, j'_{i',l'} = j_{i',l'}\right) \wedge M(j') = (1, i)\right\}\right| \text{ and} \tag{28}$$

$$a_i : j \mapsto \left|\left\{j' : \left(\forall (i', l') \in [U] \times [\ell], i' \neq i,\, j'_{i',l'} = j_{i',l'}\right) \wedge M(j') = (1, i)\right\}\right| . \tag{29}$$

The value of $a_{i,l}(j)$ shows how many "good" entries there are on a 1-dimensional subarray of M where only $j_{i,l}$ is not fixed. Similarly, $a_i(j)$ determines how many "good" entries there are on a ℓ-dimensional subarray of M where the entire tuple of j_i is not fixed. Having these two functions, in the following lemma, we find two essential properties of this game: the probability of "success" and the expected runtime (i.e., number of samples).

Remark 8.1. We essentially reuse the techniques from [10]. As it is not trivial how one can plug those techniques in, Lemmata 8.2 and 8.3 try to reduce the problem to the problems tackled in [10].

Lemma 8.2. (Abstract Sampling Game). *Consider the game in Fig. 12. Let* $J = (J_1, \ldots, J_U)$ *be uniformly distributed in* $\left(\{1, \ldots, N\}^\ell\right)^U$*, indicating the first entry sampled, and let* $(V, I) = M(J_1, \ldots, J_U)$*. Further, for all* $1 \leq i \leq U$ *and* $1 \leq l \leq \ell$*, let* $A_{i,l} = a_{i,l}(J)$ *and* $A_i = a_i(J)$*. Moreover, let* X *be the number of entries of the form* $(1, i)$ *with* $i = I$ *sampled (including the first one), and let* Λ *be the total number of entries sampled in this game. Then,*

$$\mathbb{E}[\Lambda] \leq 1 + \ell(k-1)P \quad \textit{and}$$

$$\Pr[X = k] \geq \frac{N}{N-k+1}\left(\Pr[V=1] - P \cdot \frac{\ell(k-1)}{N}\right) ,$$

where $P = \sum_{i=1}^{U} \Pr[A_i > 0]$.

Proof (of Lemma 8.2). **Expected Number of Samples.** Let us first derive the upper bound on the expected value of Λ. To this end, let X_l' be the number of sampled entries of the form $(1, i)$ with $i = I$ in the l^{th} iteration of the for loop. Similarly, let Y_l' denote the number of sampled entries of the form (v, i) with $v = 0$ or $i \neq I$, again in the l^{th} iteration. Then $\Lambda = 1 + \sum_{l=1}^{\ell} X_l' + \sum_{l=1}^{\ell} Y_l'$ and for all $1 \leq l \leq \ell$

$$\Pr\left[X_l' = 0 \,\middle|\, V = 0\right] = \Pr\left[Y_l' = 0 \,\middle|\, V = 0\right] = 1 \ .$$

Hence, for all $1 \leq l \leq \ell$, $\mathbb{E}\left[X_l' \,\middle|\, V = 0\right] = \mathbb{E}\left[Y_l' \,\middle|\, V = 0\right] = 0$. Let us consider the expected value $\mathbb{E}\left[Y_l' \,\middle|\, V = 1\right]$ for any $1 \leq l \leq \ell$. Notice that, conditioned on the event $V = 1 \wedge I = i \wedge A_{i,l} = a$ with $a > 0$, Y_l' follows a negative hypergeometric distribution with parameters $N-1$, $a-1$, and $k-1$. Hence, using Lemma 1 from [10],

$$\mathbb{E}\left[Y_l' \,\middle|\, V = 1 \wedge I = i \wedge A_{i,l} = a\right] \leq (k-1)\frac{N-a}{a} ,$$

and thus, using that $\Pr\left[X_l' \leq k-1 | V = 1\right] = 1$,

$$\mathbb{E}\left[X_l' + Y_l' \,\middle|\, V = 1 \wedge I = i \wedge A_{i,l} = a\right] \leq (k-1) + (k-1)\frac{N-a}{a} = (k-1)\frac{N}{a} \ .$$

On the other hand,

$$\Pr\left[V = 1 \wedge I = i \,\middle|\, A_{i,l} = a\right] = \frac{a}{N} ,$$

and thus,

$$\Pr\left[V = 1 \wedge I = i \wedge A_{i,l} = a\right] = \Pr\left[A_{i,l} = a\right]\frac{a}{N} \ . \tag{30}$$

Since $\Pr\left[V = 1 \wedge I = i \wedge A_{i,l} = 0\right] = 0$, we write

$$\begin{aligned}
\Pr[V = 1] \cdot \mathbb{E}\left[X_l' + Y_l' \,\middle|\, V = 1\right] &= \sum_{i=1}^{U}\sum_{a=1}^{N} \Pr\left[V = 1 \wedge I = i \wedge A_{i,l} = a\right] \\
&\qquad \cdot \mathbb{E}\left[X_l' + Y_l' \,\middle|\, V = 1 \wedge I = i \wedge A_{i,l} = a\right] \\
&\leq \sum_{i=1}^{U}\sum_{a=1}^{N} \Pr\left[A_{i,l} = a\right](k-1) \\
&= (k-1)\sum_{i=1}^{U} \Pr\left[A_{i,l} > 0\right] \ .
\end{aligned}$$

Consequently,

$$\begin{aligned}
\mathbb{E}[\Lambda] &= \mathbb{E}\left[1+\sum_{l=1}^{\ell}(X'_l+Y'_l)\right] \\
&= 1+\sum_{l=1}^{\ell}\left(\Pr[V=0]\cdot\mathbb{E}\left[X'_l+Y'_l \mid V=0\right]+\Pr[V=1]\cdot\mathbb{E}\left[X'_l+Y'_l \mid V=1\right]\right) \\
&\le 1+(k-1)\sum_{l=1}^{\ell}\sum_{i=1}^{U}\Pr\left[A_{i,l}>0\right] \\
&\le 1+\ell(k-1)\sum_{i=1}^{U}\Pr[A_i>0] \\
&\le 1+\ell(k-1)P \ ,
\end{aligned}$$

where we used the fact that for all $1 \le l \le \ell$, $\Pr\left[A_{i,l}>0\right] \le \Pr[A_i>0]$. Hence, the claimed upper bound on $\mathbb{E}[\Lambda]$ is proven.

Success Probability. Success happens when for all $1 \le l \le \ell$, we have $X'_l = k-1$. For all $1 \le l \le \ell$, let X_l be the number of sampled entries of the form $(1,i)$ in the l^{th} iteration of for loop and the single sampled entry outside of the loop. Notice that if $V=1$, for all $1 \le l \le \ell$, we have $X_l \ge 1$ even if we do not sample any other entries of the form $(1,i)$ in the for loop. We are interested in finding a lower bound for $\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k\right]$.

For all $1 \le l \le \ell$, $V=0$ implies $X_l = 0$. Therefore, using $k>0$, for all $1 \le l \le \ell$, we write $\Pr[X_l=k] = \Pr[X_l=k \wedge V=1]$ and $\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k\right] = \Pr\left[\bigwedge_{l=1}^{\ell} X_l = k \wedge V=1\right]$. Therefore, we have

$$\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k \,\middle|\, V=1\right] = \frac{\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k\right]}{\Pr[V=1]} \quad \text{and}$$
$$\Pr[X_l=k \mid V=1] = \frac{\Pr[X_l=k]}{\Pr[V=1]} \ . \tag{31}$$

Furthermore, since we sample at most $k-1$ entries of the form $(1,i)$ in each iteration, we can write

$$\begin{aligned}
\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k \,\middle|\, V=1\right] &= \left(1-\Pr\left[\bigvee_{l=1}^{\ell} X_l < k \,\middle|\, V=1\right]\right) \\
&\ge \left(1-\sum_{l=1}^{\ell}\Pr[X_l<k \mid V=1]\right)
\end{aligned}$$

$$= \left(1 - \sum_{l=1}^{\ell} (1 - \Pr[X_l = k \mid V = 1])\right)$$

$$= \left(1 - \sum_{l=1}^{\ell} \left(1 - \frac{\Pr[X_l = k]}{\Pr[V = 1]}\right)\right) \,, \tag{32}$$

where we obtain the first inequality by using a union bound. We need to find a lower bound on $\Pr[X_l = k]$ for all $1 \leq l \leq \ell$. Since we have Equation (30), we can reuse the bound shown by Attema et al. [10]. Hence,

$$\Pr[X_l = k] \geq \frac{N}{N-k+1}\left(\Pr[V = 1] - P_l \cdot \frac{k-1}{N}\right) \,,$$

where $P_l = \sum_{i=1}^{U} \Pr\left[A_{i,l} > 0\right]$. By putting this bound back into Equation (32), we obtain

$$\begin{aligned}
\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k \,\middle|\, V = 1\right] &\geq \left(1 - \sum_{l=1}^{\ell}\left(\frac{P_l \cdot (k-1)}{\Pr[V = 1] \cdot (N-k+1)} - \frac{k-1}{N-k+1}\right)\right) \\
&\geq \left(\frac{N + (\ell-1)(k-1)}{N-k+1} - \frac{\ell \cdot P \cdot (k-1)}{\Pr[V = 1] \cdot (N-k+1)}\right) \\
&\geq \left(\frac{N}{N-k+1} - \frac{\ell \cdot P \cdot (k-1)}{\Pr[V = 1] \cdot (N-k+1)}\right) \\
&\geq \frac{N}{N-k+1}\left(1 - P\frac{\ell(k-1)}{\Pr[V = 1] \cdot N}\right) \,,
\end{aligned}$$

where $P = \sum_{i=1}^{U} \Pr[A_i > 0]$. To get the second inequality, we use that for all $1 \leq l \leq \ell$, $\Pr\left[A_{i,l} > 0\right] \leq \Pr[A_i > 0]$, and consequently, $P_l \leq P$. Also, $(\ell-1)(k-1) \geq 0$ leads us to the third inequality. Using Equation (31), we have

$$\Pr\left[\bigwedge_{l=1}^{\ell} X_l = k\right] \geq \frac{N}{N-k+1}\left(\Pr[V = 1] - P\frac{\ell(k-1)}{N}\right) \,,$$

which completes the proof. □

Lemma 8.2 states bounds that are sufficient for bounding the knowledge error and the runtime of the knowledge extractor in the case of a Fiat–Shamir-transformed Σ-protocol. However, as noted by Attema et al. [10], to show the knowledge extractor of a Fiat–Shamir-transformed multi-round protocol runs in expected polynomial time, we need a refined analysis of expected runtime of the game. The sub-tree knowledge extractor may have an expensive runtime Γ or a cheap runtime γ. We now prove a better bound on runtime for the weighted version of this game which models the cost of sub-tree extractors.

Lemma 8.3. (Abstract Sampling Game - Weighted Version). *Consider the game in Fig. 12, as well a cost function $\Gamma : \left(\{1, \ldots, N\}^{\ell}\right)^U \to \mathbb{R}_{\geq 0}$ and a constant cost $\gamma \in \mathbb{R}_{\geq 0}$. Let $J = (J_1, ..., J_U)$ be uniformly distributed in $\left(\{1, \ldots, N\}^{\ell}\right)^U$, indicating the first entry sampled, and let $(V, I) = M(J)$. Further, for all $1 \leq i \leq U$, let $A_i = a_i(J)$, where the function a_i is as defined in Equation (29).*

We define the cost of sampling an entry $M(j) = (v, i)$ with $i = I$ to be $\Gamma(j)$ and the cost of an entry $M(j) = (v, i)$ with $i \neq I$ to be γ. Let Δ be the total cost of playing this game. Then

$$\mathbb{E}[\Delta] \leq (1 + \ell(k-1)) \cdot \mathbb{E}[\Gamma(J)] + \ell(k-1) \cdot T \cdot \gamma \ ,$$

where $T = \sum_{i=1}^{U} \Pr[I \neq i \wedge A_i > 0] \leq P$.

Proof. Let us break the cost Δ down to Δ_1, Δ_2, and Δ_3, defined as follows. Δ_1 denotes cost of sampling entries of the form $(1, i)$ with $i = I$, and X_l denotes the number of such entries in the l^{th} iteration. Similarly, Δ_2 denotes cost of sampling entries of the form $(0, i)$ with $i = I$, and Y_l denotes the number of such entries in the l^{th} iteration. Finally, Δ_3 denotes cost of (v, i) where $i \neq I$, and Z_l denotes the number of such entries in the l^{th} iteration. We use $\Delta'_{1,l}$ (resp. $\Delta'_{2,l}$) for denoting the part of Δ_1 (resp. Δ_2) that is added during the l^{th} iteration. Clearly, $\Delta = \Delta_1 + \Delta_2 + \Delta_3$.

For $1 \leq i \leq U$ and $1 \leq l \leq \ell$, let us write

$$J_i^* = (J_1, \ldots, J_{i-1}, J_{i+1}, \ldots, J_U) \text{ and } J_{i,l}^{\dagger} = \left(J_{i,1}, \ldots, J_{i,l-1}, J_{i,l+1}, \ldots, J_{i,\ell}\right) \ ,$$

which are respectively uniformly random with support $\{1, \ldots, N\}^{(U-1)\ell}$ and $\{1, \ldots, N\}^{\ell-1}$. Moreover, for all $1 \leq i \leq U$, $1 \leq l \leq \ell$,

$$\begin{aligned} j^* &= (j_1^*, \ldots, j_{i-1}^*, j_{i+1}^*, \ldots, j_U^*) \in \{1, \ldots, N\}^{(U-1)\ell} \ , \text{ and} \\ j^{\dagger} &= (j_1^{\dagger}, \ldots, j_{l-1}^{\dagger}, j_{l+1}^{\dagger}, \ldots, j_{\ell}^{\dagger}) \in \{1, \ldots, N\}^{\ell-1} \ , \end{aligned}$$

let $\Lambda(i, j^*)$ denote the event

$$\Lambda(i, j^*) = [I = i \wedge J_i^* = j^*]$$

and $\Theta(i, j^*, j^{\dagger})$ denote the event

$$\Theta(i, j^*, j^{\dagger}) = [\Lambda(i, j^*) \wedge J_{i,l}^{\dagger} = j^{\dagger}] \ .$$

Notice that conditioned on the event $\Lambda(i, j^*)$, all samples are picked from subarray

$$M\left(j_1^*, \ldots, j_{i-1}^*, \cdot, j_{i+1}^*, \ldots, j_U^*\right) \ ;$$

the first one uniformly at random subject to the index I being i, and the remaining ones (if $V = 1$) uniformly at random (without replacement) for each coordinate. Similarly, conditioned on the event $\Theta(i, j^*, j^\dagger)$, the sampling process follows the same criteria, with samples drawn from subarray

$$M\left(j_1^*, \ldots, j_{i-1}^*, \left(j_1^\dagger, \ldots, j_{l-1}^\dagger, \cdot, j_{l+1}^\dagger, \ldots, j_\ell^\dagger\right), j_{i+1}^*, \ldots, j_U^*\right) .$$

Let us first look into $\mathbb{E}\left[\Delta_1 \mid \Lambda(i, j^*)\right]$. We notice that for all i, and j^* with $\Pr\left[\Lambda(i, j^*)\right] > 0$,

$$\begin{aligned}
\mathbb{E}\left[\Delta_1 \,\middle|\, \Lambda(i, j^*)\right] &= \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Delta_1 \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\
&\quad + \Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Delta_1 \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \\
&= \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Delta_1 \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\
&= \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\sum_l \Delta'_{1,l} \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\
&\quad + \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\
&= \sum_l \left(\Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right]\right) \\
&\quad + \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\
&= \sum_l \mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*)\right] \\
&\quad + \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] . \qquad (33)
\end{aligned}$$

In the above, we use linearity of expectation and $\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] = 0$. Moreover, by conditioning on the value of X_l, we have

$$\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*)\right] = \sum_{x_l=0}^{N-1} \Pr\left[X_l = x_l \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*) \wedge X_l = x_l\right] . \tag{34}$$

Also,

$$\begin{aligned}
&\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*) \wedge X_l = x_l\right] \\
&\quad = \sum_{j^\dagger} \Pr\left[J_{i,l}^\dagger = j^\dagger \,\middle|\, \Lambda(i, j^*) \wedge X_l = x_l\right] \cdot \mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Theta(i, j^*, j^\dagger) \wedge X_l = x_l\right] .
\end{aligned} \tag{35}$$

Let us try to understand $\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Theta(i, j^*, j^\dagger) \wedge X_l = x_l\right]$. The condition means that we are sampling only on coordinate l, the rest of the tuple is fixed on $j^\dagger$, and we sample x_l entries of the form $(1, i)$. In other words, we are looking for a subset of entries of the form $(1, i)$ with size x_l, and also, since J is not fixed, the sampling process is uniform among such entries. Notice that the probability of choosing any of them is x_l times bigger than the probability of choosing the same entry when the size of subset was one. Therefore, the expected total cost is x_l times the expected cost of sampling only one such entry. We can write the expected cost of only one such entry as $\mathbb{E}\left[\Gamma(J) \,\middle|\, \Theta(i, j^*, j^\dagger) \wedge V = 1\right]$. So, we have

$$\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Theta(i, j^*, j^\dagger) \wedge X_l = x_l\right] = \mathbb{E}\left[\Gamma(J) \,\middle|\, \Theta(i, j^*, j^\dagger) \wedge V = 1\right] \cdot x_l \ .$$

Putting this expression back into Equation (35) and Equation (34), we get

$$\mathbb{E}\left[\Delta'_{1,l} \,\middle|\, \Lambda(i, j^*)\right] = \mathbb{E}\left[X_l \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \ . \tag{36}$$

Similarly, for Δ_2, we have

$$\begin{aligned}\mathbb{E}\left[\Delta_2 \,\middle|\, \Lambda(i, j^*)\right] = &\sum_l \mathbb{E}\left[\Delta'_{2,l} \,\middle|\, \Lambda(i, j^*)\right] \\ &+ \Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \quad \text{and}\end{aligned} \tag{37}$$

$$\mathbb{E}\left[\Delta'_{2,l} \,\middle|\, \Lambda(i, j^*)\right] = \mathbb{E}\left[Y_l \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \ . \tag{38}$$

Now, our goal is to upper bound $\mathbb{E}\left[X_l \mid \Lambda(i, j^*)\right]$ and $\mathbb{E}\left[Y_l \mid \Lambda(i, j^*)\right]$. Knowing that $V = 0$ implies $X_l = 0$ and $V = 1$ implies $X_l \leq k$, we write

$$\begin{aligned}\mathbb{E}\left[X_l \,\middle|\, \Lambda(i, j^*)\right] = &\Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[X_l \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \\ &+ \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[X_l \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\ &\leq (k-1) \cdot \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \ .\end{aligned}$$

Hence, and using Equation (33) and Equation (36), we have

$$\begin{aligned}\mathbb{E}\left[\Delta_1 \,\middle|\, \Lambda(i, j^*)\right] \leq &\sum_l (k-1) \cdot \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\ &+ \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \\ &\leq (1 + \ell(k-1)) \cdot \Pr\left[V = 1 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 1\right] \ .\end{aligned} \tag{39}$$

Bounding $\mathbb{E}\left[Y_l \mid \Lambda(i, j^*)\right]$ is more involved and we need to leverage the functions defined in Equation (28) and Equation (29). For the fixed choice of the index $1 \leq i \leq U$ and of $j^* = (j^*_1, \ldots, j^*_{i-1}, j^*_{i+1}, \ldots, j^*_U) \in \{1, \ldots, N\}^{\ell(U-1)}$, and for all $1 \leq l \leq \ell$ and $j^\dagger \in \{1, \ldots, N\}^{\ell-1}$, we define new parameters

$$\begin{aligned}a &:= \left|\left\{j : (v_j, i_j) = M(j^*_1, \ldots, j^*_{i-1}, j, j^*_{i+1}, \ldots, j^*_U) = (1, i)\right\}\right| \ , \\ b &:= \left|\left\{j : (v_j, i_j) = M(j^*_1, \ldots, j^*_{i-1}, j, j^*_{i+1}, \ldots, j^*_U) = (0, i)\right\}\right| \ ,\end{aligned}$$

$$a_{l,j^\dagger} := \left| \left\{ j : (v_j, i_j) = M \begin{pmatrix} j_1^*, \dots, j_{i-1}^*, \\ \begin{pmatrix} j_1^\dagger, \dots, j_{l-1}^\dagger, \\ j, \\ j_{l+1}^\dagger, \dots, j_\ell^\dagger \end{pmatrix} \\ , j_{i+1}^*, \dots, j_U^* \end{pmatrix} = (1, i) \right\} \right| \ , \text{ and}$$

$$b_{l,j^\dagger} := \left| \left\{ j : (v_j, i_j) = M \begin{pmatrix} j_1^*, \dots, j_{i-1}^*, \\ \begin{pmatrix} j_1^\dagger, \dots, j_{l-1}^\dagger, \\ j, \\ j_{l+1}^\dagger, \dots, j_\ell^\dagger \end{pmatrix} \\ , j_{i+1}^*, \dots, j_U^* \end{pmatrix} = (0, i) \right\} \right| \ . \tag{40}$$

Notice that $\Pr\left[V = 1 \mid \Lambda(i, j^*)\right] = \frac{a}{a+b}$ and $\Pr\left[V = 0 \mid \Lambda(i, j^*)\right] = \frac{b}{a+b}$ for all i and j^* with $\Pr\left[\Lambda(i, j^*)\right] > 0$. Observe that if we condition on the event $V = 1 \wedge \Lambda(i, j^*)$ (resp. $V = 1 \wedge \Theta(i, j^*, j^\dagger)$), we implicitly assume that $a > 0$ (resp. $a_{l,j^\dagger} > 0$). Moreover, $\sum_{j^\dagger} a_{l,j^\dagger} = a$ and $\sum_{j^\dagger} b_{l,j^\dagger} = b$. Using the fact that $\mathbb{E}\left[Y_l \mid V = 0 \wedge \Lambda(i, j^*)\right] = 0$, we have

$$\mathbb{E}\left[Y_l \,\middle|\, \Lambda(i, j^*)\right] = \frac{a}{a+b} \cdot \mathbb{E}\left[Y_l \,\middle|\, V = 1 \wedge \Lambda(i, j^*)\right] \ .$$

Conditioned on $V = 1 \wedge \Theta(i, j^*, j^\dagger)$, Y_l follows a negative hypergeometric distribution with parameters $a + b - 1$, $a - 1$, and $k - 1$. We write

$$\begin{aligned} \mathbb{E}\left[Y_l \,\middle|\, V = 1 \wedge \Lambda(i, j^*)\right] &= \sum_{j^\dagger} \Pr\left[J_{i,l}^\dagger = j^\dagger \,\middle|\, V = 1 \wedge \Lambda(i, j^*)\right] \\ &\quad \cdot \mathbb{E}\left[Y_l \,\middle|\, V = 1 \wedge \Theta(i, j^*, j^\dagger)\right] \\ &= \sum_{j^\dagger} \frac{a_{l,j^\dagger}}{a} \cdot \mathbb{E}\left[Y_l \,\middle|\, V = 1 \wedge \Theta(i, j^*, j^\dagger)\right] \\ &\leq \sum_{j^\dagger} \frac{a_{l,j^\dagger}}{a} \cdot (k-1) \frac{b_{l,j^\dagger}}{a_{l,j^\dagger}} \ \text{(by [10, Lemma 1])} \\ &= (k-1) \frac{b}{a} \ . \end{aligned}$$

This implies that

$$\mathbb{E}\left[Y_l \,\middle|\, \Lambda(i, j^*)\right] \leq (k-1) \cdot \Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \ .$$

Using Equation (37) and Equation (38), we have

$$\begin{aligned} \mathbb{E}\left[\Delta_2 \,\middle|\, \Lambda(i, j^*)\right] \leq \sum_l (k-1) \cdot \Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \\ + \Pr\left[V = 0 \,\middle|\, \Lambda(i, j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i, j^*) \wedge V = 0\right] \end{aligned}$$

$$\leq (1+\ell(k-1)) \cdot \Pr\left[V=0 \,\middle|\, \Lambda(i,j^*)\right] \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i,j^*) \wedge V=0\right].$$

Combining with Equation (39), we have

$$\mathbb{E}\left[\Delta_1+\Delta_2 \,\middle|\, \Lambda(i,j^*)\right] \leq (1+\ell(k-1)) \cdot \mathbb{E}\left[\Gamma(J) \,\middle|\, \Lambda(i,j^*)\right] \ .$$

We can remove the condition $\Lambda(i,j^*)$ since this inequality holds for all i and j^* with $\Pr\left[\Lambda(i,j^*)\right] > 0$. Therefore,

$$\mathbb{E}\left[\Delta_1+\Delta_2\right] \leq (1+\ell(k-1)) \cdot \mathbb{E}\left[\Gamma(J)\right] \ .$$

The final step is to show $\mathbb{E}\left[\Delta_3\right] \leq \ell(k-1)T\gamma$, or equivalently, $\mathbb{E}\left[Z\right] \leq \ell(k-1)T$, where $Z=\sum_l Z_l$. Again, we follow the approach we used previously. We fix a choice of i and j^* and set the parameters $a, b, a_{l,j^\dagger}$, and $b_{l,j^\dagger}$ as defined in Equation (40). Consequently, we observe that conditioning on the event $V=1 \wedge \Theta(i,j^*,j^\dagger)$, Z_l follows a negative hypergeometric distribution with parameters $N-b-1$, $a-1$, and $k-1$. Therefore, using the bound in Lemma 1 from [10], we have

$$\begin{aligned}
&\mathbb{E}\left[Z_l \,\middle|\, V=1 \wedge \Lambda(i,j^*)\right]\\
&\quad= \sum_{j^\dagger} \Pr\left[J^\dagger_{i,l}=j^\dagger \,\middle|\, V=1 \wedge \Lambda(i,j^*)\right] \cdot \mathbb{E}\left[Z_l \,\middle|\, V=1 \wedge \Theta(i,j^*,j^\dagger)\right]\\
&\quad= \sum_{j^\dagger} \frac{a_{l,j^\dagger}}{a} \cdot \mathbb{E}\left[Y_l \,\middle|\, V=1 \wedge \Theta(i,j^*,j^\dagger)\right]\\
&\quad\leq \sum_{j^\dagger} \frac{a_{l,j^\dagger}}{a} \cdot (k-1)\frac{N-a_{l,j^\dagger}-b_{l,j^\dagger}}{a_{l,j^\dagger}}\\
&\quad= (k-1)\frac{N-a-b}{a} \ .
\end{aligned}$$

Also, since $\mathbb{E}\left[Z_l \mid V=0 \wedge \Lambda(i,j^*)\right]=0$, we write

$$\mathbb{E}\left[Z_l \,\middle|\, \Lambda(i,j^*)\right] \leq \frac{a}{a+b} \cdot \mathbb{E}\left[Z_l \,\middle|\, V=1 \wedge \Lambda(i,j^*)\right] = (k-1)\frac{N-a-b}{a+b} \ .$$

Using $\Pr\left[I=i \,\middle|\, J^*_i=j^*\right]=\frac{a+b}{N}$, we have

$$\mathbb{E}\left[Z_l \,\middle|\, \Lambda(i,j^*)\right] \leq (k-1) \cdot \left(\frac{\Pr\left[I \neq i \wedge J^*_i=j^*\right]}{\Pr\left[\Lambda(i,j^*)\right]}\right) \ ,$$

and since $Z=\sum_l Z_l$,

$$\mathbb{E}\left[\Delta_3 \,\middle|\, \Lambda(i,j^*)\right] \leq \ell(k-1)\gamma \cdot \left(\frac{\Pr\left[I \neq i \wedge J^*_i=j^*\right]}{\Pr\left[\Lambda(i,j^*)\right]}\right) \ .$$

Extractor $\mathcal{E}$

Parameters: $\ell, k, Q \in \mathbb{N}$
Black-box access to: $\mathcal{A}$

- Run $\mathcal{A}$ to get (I, y_1, v) in the following manner: answer all (distinct) random oracle queries with uniformly random values in $\mathcal{C} := S^{\ell}$. Set $i := I$, let c_i be the response to query i.
- If $v = 0$, abort.
- Else, for $1 \leq l \leq \ell$, repeat
  - sample $c'_{i,l} \in S \setminus \{c_{i,l}\}$ (without replacement),
  - set $c'_i = \left(c_{i,1}, \ldots, c_{i,l-1}, c'_{i,l}, c_{i,l+1}, \ldots, c_{i,\ell}\right)$
  - run $\mathcal{A}$ to get (I', y', v') in the following manner: answer the query to i with c'_i, while answering all other queries consistently if the query was performed by $\mathcal{A}$ already on a previous run, and otherwise, with a fresh random value in $\mathcal{C}$.

  until either $k-1$ additional challenges with $v' = 1$ and $I' = I$ have been found or until all challenges $c'_{i,l} \in S$ have been tried.
- If the former happens for all $1 \leq l \leq \ell$, output the $\ell(k-1)+1$ accepting transcripts $y_1, \ldots, y_{\ell(k-1)+1}$.

Fig. 13. Knowledge Extractor $\mathcal{E}$.

From this point, using the exact same argument by Attema et al. [10, Lemma 5], we have $\mathbb{E}[\varDelta_3] \leq \ell(k-1) \cdot \gamma \cdot T$, and the proof is complete. □

Now, the analysis of the game is complete, and we move forward to knowledge extraction.

8.2. *The Knowledge Extractor*

This section introduces our knowledge extractor for a Fiat–Shamir-transformed ℓ-coordinate-wise k-special-sound $\varSigma$-protocol. One can generalize this extractor for multi-round protocols as done by Attema et al. [10, Section 6]. In the following, we use the notation of Section 4 from [10]. Figure 13 demonstrates our knowledge extractor $\mathcal{E}$. Instead of, for example, answering the query on the first message with a fresh random value in $\mathcal{C}{:=}S^{\ell}$, $\mathcal{E}$ uses new values coordinate by coordinate. Notice that this manner of answering query on the first message is analogous to our abstract sampling game in Fig. 12.

Having Lemmata 8.2 and 8.3 along with [10, Lemmata 3 and 6] at hand, and using the bounds in Sect. 7, we deduce that the knowledge error and the expected runtime of the extractor for a ℓ-coordinate-wise k-special-sound multi-round protocol degrades by a factor of $Q+1$ after applying Fiat–Shamir transformation, and it is independent from the number of rounds.

We note that one can easily generalize this conclusion for a $(\ell_1, \ldots, \ell_\mu)$-coordinate-wise $(k_1, \ldots, k_\mu)$-special-sound $(2\mu+1)$-move protocol and the corresponding Fiat–

Shamir-transformed protocol. We omit the details here because they do not contain any novel aspects.

Acknowledgements

We thank the anonymous reviewers for their useful comments. We thank Martin Albrecht and Sasha Lapiha for discussion on the PowerBASIS assumption. Giacomo Fenzi is partially supported by the Ethereum Foundation. Ngoc Khanh Nguyen is supported by the Protocol Labs RFP-013: Cryptonet network grant.

References

[1] S. Agrawal, E. Kirshanova, D. Stehlé, A. Yadav, Practical, round-optimal lattice-based blind signatures, in *CCS* (ACM, 2022), pp. 39–53

[2] M. Ajtai, Generating hard instances of lattice problems. in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*. STOC '96 (1996), pp. 99–108

[3] M.R. Albrecht, V. Cini, R.W.F. Lai, G. Malavolta, S.A.K. Thyagarajan, Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable—(Extended Abstract), in *CRYPTO (2)*. Lecture Notes in Computer Science, Vol. 13508 (Springer, 2022), pp. 102–132

[4] M.R. Albrecht, G. Fenzi, O. Lapiha, N.K. Nguyen, *SLAP: Succinct Lattice-Based Polynomial Commitments from Standard Assumptions*. Cryptology ePrint Archive, Paper 2023/1469. 2023. URL: https://eprint.iacr.org/2023/1469

[5] M.R. Albrecht, R.W.F. Lai, Subtractive Sets over Cyclotomic Rings—Limits of Schnorr-Like Arguments over Lattices, in *CRYPTO (2)*. Lecture Notes in Computer Science, Vol. 12826 (Springer, 2021), pp. 519–548.

[6] E. Alkim, L. Ducas, T. Pöppelmann, P. Schwabe, Post-quantum Key Exchange—A New Hope, in *USENIX Security Symposium*, (USENIX Association, 2016), pp. 327–343

[7] T. Attema, *Compressed Sigma-protocol theory*. PhD Thesis (2023). URL: https://scholarlypublications.universiteitleiden.nl/access/item%3A3619598/view

[8] T. Attema, R. Cramer, L. Kohl, A compressed Σ-protocol theory for lattices, in *CRYPTO (2)*. Lecture Notes in Computer Science, vol. 12826 (Springer, 2021), pp. 549–579

[9] T. Attema, S. Fehr, Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs, in *CRYPTO (1)*. Lecture Notes in Computer Science, vol. 13507 (Springer, 2022), pp. 415–443

[10] T. Attema, S. Fehr, M. Klooß, Fiat-Shamir Transformation of Multi-Round Interactive Proofs (Extended Version). *J. Cryptol.* **36**(4), 36 (2023)

[11] T. Attema, S. Fehr, N. Resch, *A generalized special-soundness notion and its knowledge extractors*. Cryptology ePrint Archive, Paper 2023/818. 2023. URL: https://eprint.iacr.org/2023/818

[12] T. Attema, V. Lyubashevsky, G. Seiler, Practical product proofs for lattice commitments, in *CRYPTO (2)*. Lecture Notes in Computer Science, vol. 12171 (Springer, 2020), pp. 470–499

[13] D. Balbás, D. Catalano, D. Fiore, R.W.F. Lai, *Functional commitments for circuits from falsifiable assumptions*. Cryptology ePrint Archive, Paper 2022/1365 (2022). URL: https://eprint.iacr.org/2022/1365

[14] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, V. Lyubashevsky, Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits, in *CRYPTO* (2018), pp. 669–699

[15] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, C. Peikert, More efficient commitments from structured lattice assumptions, in *SCN* (2018), pp. 368–385

[16] A. Becker, L. Ducas, N. Gama, T. Laarhoven, New directions in nearest neighbor searching with applications to lattice sieving, in *SODA* (SIAM, 2016), pp. 10–24

[17] E. Ben-Sasson, I. Bentov, Y. Horesh, M. Riabzev, Scalable zero knowledge with no trusted setup, in *Proceedings of the 39th Annual International Cryptology Conference* (CRYPTO '19, 2019), pp. 733–764

[18] E. Ben-Sasson, A. Chiesa, N. Spooner, Interactive oracle proofs. in *Proceedings of the 14th Theory of Cryptography Conference* (TCC '16-B, 2016), pp. 31–60

[19] F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, G. Neven, Better zero-knowledge proofs for lattice encryption and their application to group signatures, in *ASIACRYPT* (2014), pp. 551–572

[20] W. Beullens, G. Seiler, LaBRADOR: compact proofs for R1CS from Module-SIS, in *CRYPTO (5)*. Lecture Notes in Computer Science, vol. 14085 (Springer, 2023), pp. 518–548

[21] N. Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky, O. Paneth, Succinct non-interactive arguments via linear interactive proofs, in *Proceedings of the 10th Theory of Cryptography Conference*. TCC '13 (2013), pp. 315–333

[22] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, C. Petit, Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting, in *EUROCRYPT* (2016), pp. 327–357

[23] J. Bootle, A. Chiesa, Y. Hu, M. Orrù, Gemini: elastic SNARKs for diverse environments, in *Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques* (EUROCRYPT '22, 2022), pp. 427–457

[24] J. Bootle, A. Chiesa, K. Sotiraki, Sumcheck arguments and their applications. in *CRYPTO (1)*. Lecture Notes in Computer Science, vol. 12825 (Springer, 2021), pp. 742–773

[25] J. Bootle, A. Chiesa, K. Sotiraki, Lattice-based succinct arguments for NP with polylogarithmic-time verification, in *Advances in Cryptology - CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya. Lecture Notes in Computer Science, vol. 14082. Full version available at https://eprint.iacr.org/2023/930.pdf (Springer, 2023), pp. 227–251. https://doi.org/10.1007/978-3-031-38545-2_8

[26] J. Bootle, V. Lyubashevsky, N.K. Nguyen, G. Seiler, A non-PCP approach to succinct quantum-safe zero-knowledge, in *CRYPTO (2)*. Lecture Notes in Computer Science, vol. 12171 (Springer, 2020), pp. 441–469

[27] J. Bootle, V. Lyubashevsky, N.K. Nguyen, A. Sorniotti, *A Framework for Practical Anonymous Credentials from Lattices*. To appear at CRYPTO (2023). 2023. URL: https://eprint.iacr.org/2023/560

[28] J.W. Bos et al. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM, in *2018 IEEE European Symposium on Security and Privacy, EuroS&P* (2018), pp. 353–367

[29] C. Boschini, A. Takahashi, M. Tibouchi, MuSig-L: lattice-based multi-signature with single-round online phase. (2022). URL: https://eprint.iacr.org/2022/1036

[30] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, G. Maxwell, Bulletproofs: short proofs for confidential transactions and more, in *IEEE Symposium on Security and Privacy* (2018), pp. 315–334

[31] B. Bünz, B. Fisch, A. Szepieniec, Transparent SNARKs from DARK compilers, in *EUROCRYPT (1)*. Lecture Notes in Computer Science, vol. 12105 (Springer, 2020), pp. 677–706

[32] B. Bünz, B. Fisch, *Multilinear Schwartz-Zippel mod N with Applications to Succinct Arguments*. Cryptology ePrint Archive, Paper 2022/458. 2022. URL: https://eprint.iacr.org/2022/458

[33] R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G.N. Rothblum, R.D. Rothblum, D. Wichs, Fiat-Shamir: from practice to theory, *STOC* (ACM, 2019), pp. 1082–1090

[34] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited. *J. ACM* **51**(4), 557–594 (2004)

[35] L. de Castro, C. Peikert, Functional Commitments for all functions, with transparent setup, in *IACR Cryptol. ePrint Arch* (2022), p. 1368

[36] Y. Chen, P.Q. Nguyen, BKZ 2.0: better lattice security estimates, in *ASIACRYPT*. Lecture Notes in Computer Science, vol. 7073 (Springer, 2011), pp. 1–20

[37] A. Chiesa, Y. Hu, M. Maller, P. Mishra, N. Vesely, N. Ward, Marlin: preprocessing zkSNARKs with universal and updatable SRS, in *Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques*. EUROCRYPT '20 (2020), pp. 738–768

[38] A.R. Choudhuri, A. Jain, Z. Jin, SNARGs for $\mathcal{P}$ from LWE, in *FOCS* (IEEE, 2021), pp. 68–79

[39] V. Cini, R.W.F. Lai, G. Malavolta, Lattice-based succinct arguments from vanishing polynomials, in *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya (Springer Nature Switzerland, Cham, 2023), pp. 72–105

[40] L. Devadas, R. Goyal, Y. Kalai, V. Vaikuntanathan, Rate-1 non-interactive arguments for batch-NP and applications, in *FOCS* (IEEE, 2022), pp. 1057–1068

[41] J. Don, S. Fehr, C. Majenz, The measure-and-reprogram technique 2.0: multi-round Fiat-Shamir and more, in *CRYPTO (3)*. Lecture Notes in Computer Science, vol. 12172 (Springer, 2020), pp. 602–631

[42] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehlé, CRYSTALS-Dilithium: a lattice-based digital signature scheme, in *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2018(1), pp. 238–268 (2018)

[43] L. Ducas, V. Lyubashevsky, T. Prest, Efficient identity-based encryption over NTRU lattices, in *ASIACRYPT* (2014), pp. 22–41

[44] M.F. Esgin, N.K. Nguyen, G. Seiler, Practical Exact proofs from lattices: new techniques to exploit fully-splitting rings, in *ASIACRYPT (2)* (2020), pp. 259–288

[45] M.F. Esgin, R.K. Zhao, R. Steinfeld, J.K. Liu, D. Liu, MatRiCT: efficient, scalable and post-quantum blockchain confidential transactions protocol, in *CCS* (ACM, 2019), pp. 567–584

[46] A. Fiat, A. Shamir, How to prove yourself: practical solutions to identification and signature problems, in *CRYPTO* (1986), pp. 186–194

[47] B. Fisch, Z. Liu, P. Vesely, Orbweaver: succinct linear functional commitments from lattices, in *Advances in Cryptology – CRYPTO 2023*. Ed. by Helena Handschuh and Anna Lysyanskaya (Springer Nature Switzerland, Cham, 2023), pp. 106–131

[48] P.-A. Fouque et al. *Falcon: Fast-Fourier lattice-based compact signatures over NTRU*. Tech. rep. https://falcon-sign.info/falcon.pdf (2020)

[49] R. Gennaro, M. Minelli, A. Nitulescu, M. Orrù, Lattice-based zk-SNARKs from square span programs. in *Proceedings of the 25th ACM Conference on Computer and Communications Security*. CCS '18 (2018), pp. 556–573

[50] C. Gentry, C. Peikert, V. Vaikuntanathan, Trapdoors for hard lattices and new cryptographic constructions, in *STOC* (2008), pp. 197–206

[51] A. Golovnev, J. Lee, S.T.V. Setty, J. Thaler, R.S. Wahby, Brakedown: linear-time and post-quantum SNARKs for R1CS, in *IACR Cryptol. ePrint Arch.* (2021), p. 1043

[52] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J.H. Silverman, W. Whyte, NTRUSIGN: digital signatures using the NTRU lattice, in *CT-RSA* (2003), pp. 122–140

[53] J. Holmgren, A. Lombardi, R.D. Rothblum, Fiat-Shamir via list-recoverable codes (or: parallel repetition of GMW is not zero-knowledge), in *STOC* (ACM, 2021), pp. 750–760

[54] J. Hulett, R. Jawale, D. Khurana, A. Srinivasan, SNARGs for P from Sub-exponential DDH and QR, in *EUROCRYPT (2)*. Lecture Notes in Computer Sciencem, vol. 13276 (Springer, 2022), pp. 520–549

[55] Y. Ishai, H. Su, D. J. Wu, Shorter and faster post-quantum designated-verifier zkSNARKs from lattices, in *CCS* (ACM, 2021), pp. 212–234

[56] C. Jeudy, A. Roux-Langlois, O. Sanders, *Lattice Signature with Efficient Protocols, Application to Anonymous Credentials*. Cryptology ePrint Archive, Paper 2022/509. (2022). URL: https://eprint.iacr.org/2022/509

[57] Y. Kalai, A. Lombardi, V. Vaikuntanathan, D. Wichs, Boosting batch arguments and RAM delegation, in *STOC* (ACM, 2023), pp. 1545–1552

[58] A. Kate, G.M. Zaverucha, I. Goldberg, Constant-size commitments to polynomials and their applications, in *ASIACRYPT*. Lecture Notes in Computer Science, vol. 6477 (Springer, 2010), pp. 177–194

[59] S. Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs, in *CRYPTO (2)*. Lecture Notes in Computer Science, vol. 12826 (Springer, 2021), pp. 580–610

[60] R.W.F. Lai, G. Malavolta, N. Spooner, Quantum rewinding for many-round protocols, in *TCC (1)*. Lecture Notes in Computer Science, vol. 13747 (Springer, 2022), pp. 80–109

[61] A. Langlois, D. Stehlé, Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.* **75**(3), 565–599 (2015)

[62] J. Lee, Dory: efficient, transparent arguments for generalised inner products and polynomial commitments, in *TCC (2)*. Lecture Notes in Computer Science, vol. 13043 (Springer, 2021), pp. 1–34

[63] A. Lenstra, H. Lenstra Jr., L. Lovasz, Factoring polynomials with rational coefficients. *Mathematische Annalen* **261**, 513–534 (1982)

[64] B. Libert, S.C. Ramanna, M. Yung, Functional commitment schemes: from polynomial commitments to pairing-based accumulators from simple assumptions, in *ICALP*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, vol. 55 (2016), pp. 30:1–30:14

[65] V. Lyubashevsky, Fiat-Shamir with Aborts: applications to lattice and factoring-based signatures, in *ASIACRYPT*. (2009), pp. 598–616

[66] V. Lyubashevsky, Lattice signatures without trapdoors, in *EUROCRYPT*. (2012), pp. 738–755

[67] V. Lyubashevsky, N.K. Nguyen, BLOOM: bimodal lattice one-out-of-many proofs and applications, in *ASIACRYPT (4)*. Lecture Notes in Computer Science, vol. 13794 (Springer, 2022), pp. 95–125

[68] V. Lyubashevsky, N.K. Nguyen, M. Plançon, Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general, in *CRYPTO (2)*. Lecture Notes in Computer Science, vol. 13508 (Springer, 2022), pp. 71–101

[69] V. Lyubashevsky, C. Peikert, O. Regev, A toolkit for ring-LWE cryptography, in *EUROCRYPT*. (2013), pp. 35–54

[70] V. Lyubashevsky, G. Seiler, Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs, in *EUROCRYPT (1)* (Springer, 2018), pp. 204–224

[71] V. Lyubashevsky, G. Seiler, NTTRU: truly fast NTRU using NTT, in *IACR Trans. Cryptogr. Hardw. Embed. Syst.* vol. 2019(3) (2019), pp. 180–201

[72] D. Micciancio, C. Peikert, Trapdoors for lattices: simpler, tighter, faster, smaller, in *EUROCRYPT*. (2012), pp. 700–718

[73] D. Micciancio, O. Regev, Worst-case to average-case reductions based on Gaussian measures, in *SIAM Journal on Computing* **37**, 267–302 (2007)

[74] D. Micciancio, O. 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. https://doi.org/10.1007/978-3-540-88702-7_5

[75] N.K. Nguyen, G. Seiler, Practical sublinear proofs for R1CS from lattices, in *CRYPTO (2)*. Lecture Notes in Computer Science. vol. 13508 (Springer, 2022), pp. 133–162

[76] C. Peikert, Z. Pepin, C. Sharp, Vector and functional commitments from lattices, in *TCC (3)*. Lecture Notes in Computer Science, vol. 13044 (Springer, 2021), pp. 480–511

[77] C.-P. Schnorr, M. Euchner, Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Program.* **66**, 181–199 (1994)

[78] G. Seiler, Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography, in *IACR Cryptology ePrint Archive* 2018 (2018). http://eprint.iacr.org/2018/039, p. 39

[79] S. Setty, Spartan: efficient and general-purpose zkSNARKs without trusted setup, in*Proceedings of the 40th Annual International Cryptology Conference*. CRYPTO '20. Referencing Cryptology ePrint Archive, Report 2019/550, revision from 2020.02.28. (2020), pp. 704–737

[80] D. Stehlé, R. Steinfeld, Making NTRUEncrypt and NTRUSign as Secure as Standard Worst-Case Problems over Ideal Lattices, in *IACR Cryptol. ePrint Arch.* (2013), p. 4

[81] R. Steinfeld, A. Sakzad, M.F. Esgin, V. Kuchta, *Private Re-Randomization for Module LWE and Applications to Quasi-Optimal ZK-SNARKs*. Cryptology ePrint Archive, Paper 2022/1690. (2022). URL: https://eprint.iacr.org/2022/1690

[82] H. Wee, D.J. Wu, Lattice-based functional commitments: fast verification and cryptanalysis, in *Advances in Cryptology – ASIACRYPT 2023*. Ed. by Jian Guo and Ron Steinfeld. (Springer Nature, Singapore, 2023), pp. 201–235. ISBN: 978-981-99-8733-7

[83] H. Wee, D.J. Wu, Succinct vector, polynomial, and functional commitments from lattices, *EUROCRYPT (3)*. Lecture Notes in Computer Science, vol. 14006. Full version: https://eprint.iacr.org/2022/1515. (Springer, 2023), pp. 385–416