

# Concretely Efficient Lattice-based Polynomial Commitment from Standard Assumptions

Intak Hwang , Jinyeong Seo , and Yongsoo Song 

Seoul National University

{intak.hwang, jinyeong.seo, y.song}@snu.ac.kr

**Abstract.** Polynomial commitment is a crucial cryptographic primitive in constructing zkSNARKs. Most practical constructions to date are either vulnerable against quantum adversaries or lack homomorphic properties, which are essential for recursive proof composition and proof batching. Recently, lattice-based constructions have drawn attention for their potential to achieve all the desirable properties, though they often suffer from concrete inefficiency or rely on newly introduced assumptions requiring further cryptanalysis.

In this paper, we propose a novel construction of a polynomial commitment scheme based on standard lattice-based assumptions. Our scheme achieves a square-root proof size and verification complexity, ensuring concrete efficiency in proof size, proof generation, and verification. Additionally, it features a transparent setup and publicly verifiability.

When compared with Brakedown (CRYPTO 2023), a recent code-based construction, our scheme offers comparable performance across all metrics. Furthermore, its proof size is approximately 4.1 times smaller than SLAP (EUROCRYPT 2024), a recent lattice-based construction.

**Keywords:** Polynomial Commitment, Lattice, Zero-Knowledge

## 1 Introduction

Polynomial commitment (PC) is a cryptographic primitive that allows a prover to commit to a polynomial  $f \in \mathbb{Z}_p[X]$  by publishing a short commitment. Later, given two public values  $x$  and  $y$ , the prover can generate a proof  $\pi$  convincing a verifier that  $y = f(x)$ . A polynomial commitment scheme is said to be evaluation binding if it cannot be proven to evaluate to any other value  $y' \neq y$  for  $x$ , evaluation hiding if the proof does not reveal any information on the coefficients of  $f$ , and extractable if it provides a proof of knowledge. It has been a core building block for many cryptographic protocols, such as verifiable secret sharing [38, 41], or composing zero-knowledge arguments for membership or range relations [14].

The most prominent use case of polynomial commitments is in constructing zero-knowledge non-interactive succinct arguments of knowledge (zkSNARKs), as recent zkSNARKs are based on polynomial interactive oracle proofs [18] combined with polynomial commitment schemes. Consequently, constructing efficient polynomial commitment schemes in terms of prover complexity, verifier

complexity, and proof size has been a significant focus of recent research. To date, the most efficient polynomial commitment schemes [9, 16, 25, 26, 30] are either based on discrete logarithm assumptions or error-correcting codes. The discrete logarithm-based constructions [16, 26, 30] are known for their compact proof sizes and rich features, such as proof batching or recursive SNARK compositions [12, 17, 28], due to their homomorphic properties, but they are insecure against quantum adversaries. In contrast, code-based constructions [9, 25] possess post-quantum security but lack the homomorphic properties that enable efficient proof composition.

Recently, lattice-based constructions [2, 3, 21, 23] have drawn attention for possessing all these desirable properties for SNARKs. These studies primarily focus on asymptotically succinct proof size and verification complexity, achieving polylogarithmic scale. However, some of these constructions have issues with extractability, a crucial requirement for SNARKs. For example, the extractability of [2, 21, 23] relies on newly introduced assumptions, called the k-R-ISIS assumption, which has been heuristically broken in [39]. Additionally, all these constructions require a trusted setup<sup>1</sup> and lack concrete benchmark results to demonstrate their practicality.

### 1.1 Our Contribution

In this paper, we propose a new lattice-based polynomial commitment scheme that is practical in terms of proof size and efficient in proof generation and verification. Additionally, our polynomial commitment scheme utilizes a transparent setup and is publicly verifiable. As mentioned above, there have been several research efforts on lattice-based polynomial commitments, but their concrete efficiency has not been thoroughly discussed. Hence, our aim is to provide a lattice-based polynomial commitment scheme that is concretely efficient, addressing practical concerns that arise in actual implementations.

**Modified Ajtai Commitment Scheme.** For the basic building block, we employ the Ajtai commitment scheme [1]. There has been a line of research [7, 11, 15, 36] which focused on constructing SNARKs directly from Ajtai commitment schemes. This is due to its intrinsic compressing property, where commitment size is almost independent of the input size, akin to the Pedersen commitment. Following their approaches, we also utilize the Ajtai commitment scheme to commit coefficients of the input polynomial to reduce asymptotic proof size. We note that the security of the Ajtai commitment scheme relies on standard assumptions: the MSIS and MLWE problems. However, several practical issues arise when we use Ajtai commitment for constructing polynomial commitments: we can only commit small messages for the binding property, and we need to perform rejection sampling [31] to achieve zero-knowledgeness. These two issues can be critical in the context of polynomial commitment schemes. Firstly, due to soundness, the modulus  $p$  of polynomial coefficients is typically set to exponentially large values, such as 255 bits in size. This incurs parameter blow-ups if

<sup>1</sup> In this context, we only consider the extractable versions of [2, 21].

we directly commit these large coefficients. Secondly, multiple polynomials are committed during the instantiation of PIOP, leading to degraded prover performance since the rejection rate grows exponentially with respect to the number of committed polynomials.

To address the first issue, we propose a new encoding method inspired by [19, 20], which maps vectors of elements in  $\mathbb{Z}_p$  into small normed polynomial ring elements while preserving the homomorphic property. For example, our method can encode 255-bit-sized messages into polynomials with 16-bit-sized coefficients. As a result, the new encoding method effectively resolves overhead from the large modulus  $p$ . For the second issue, we newly devise a randomized encoding technique to achieve the simulatability of Ajtai-based proof systems. Kim et al. [27] achieves simulatability of BDLOP-based proof systems [4, 8, 22] without rejection sampling by leveraging the properties of the discrete Gaussian distribution. Inspired by their approach, we develop a new technique that randomizes encoded messages so they follow discrete Gaussian distributions over cosets of a lattice. As a consequence, we successfully eliminate rejection sampling from our protocol.

**Polynomial Evaluation Protocol.** Based on our improvements in Ajtai-based proof systems, we design a protocol for proving polynomial evaluation. We adapt the methodology from [13, 14], which constructs polynomial commitment schemes based on the Pedersen commitment scheme, achieving proof sizes that are square-root in the polynomial degree. However, directly converting these techniques to our context is not straightforward due to the inherent differences in the proof techniques between the Pedersen commitment scheme and the Ajtai commitment scheme.

To resolve the issues, we first redesign the knowledge extractor based on the batched sigma protocol in [7], as the knowledge extractor in [13, 14] relies on the invertibility of the Vandermonde matrix over  $\mathbb{Z}_p$ , which is not compatible with the Ajtai commitment scheme. Second, we adapt the blinding technique from [13, 14] into the Ajtai commitment scheme. To achieve the evaluation hiding property, which ensures the proof does not reveal information other than evaluation results, the previous construction uses random blinders from  $\mathbb{Z}_p$  to hide committed polynomial coefficients. We adapt this by sampling random blinders from the discrete Gaussian distribution, extending our randomized encoding technique. As a result, we successfully construct a polynomial commitment scheme that provides square-root proof size and verification complexity in terms of polynomial degree, along with extractability and evaluation hiding properties. Additionally, we note that our construction is publicly verifiable and uses a transparent setup.

**PoC Implementation.** We implement our polynomial commitment scheme at a proof-of-concept level to demonstrate its concrete performance. Since our protocol requires sampling from discrete Gaussian distributions over cosets of the integer lattice, we use the convolution sampler introduced in [35]. Additionally, we utilize Residue Number System (RNS) representation and Number Theoretic Transform (NTT) to instantiate efficient polynomial ring arithmetic. Finally, we

use the optimization technique from [5] to further reduce the proof size. The benchmark results indicate that our polynomial commitment scheme offers comparable performance in terms of proof generation, verification, and proof size when compared to Brakedown [25], a recent code-based polynomial commitment scheme, which also yields square-root proof size. Furthermore, in comparison to SLAP [3], a recent lattice-based polynomial commitment scheme, our construction achieves approximately 4.1 times smaller proof size for a polynomial degree of  $2^{20}$ .

## 1.2 Related Works

The most relevant work to ours is [29], which constructs a polynomial commitment scheme based on BDLOP, following the methodology from [13, 14]. However, due to the usage of the BDLOP commitment scheme, which does not have a compressing property, its proof size is not sublinear in the input polynomial degree. Additionally, they utilize rejection sampling to achieve zero-knowledgeness. Recently, there has been a line of research [2, 21, 23, 40], which constructs lattice-based polynomial commitment schemes from newly introduced assumptions. However, as discussed earlier, their extractability requires further analysis, and their setup phase is not transparent. Meanwhile, there is another construction [3] that provides an efficient polynomial commitment scheme based on standard assumptions, but it also requires a trusted setup, and concrete performance for the prover and verifier were not provided.

On the other hand, another series of work [7, 11, 15, 36] has focused on directly building SNARKs from the Ajtai commitment scheme. To address the issues related to small messages, [7] employed an encoding method for the finite field  $GF(p^k)$  when  $p$  is small, [36] used base representation to decompose inputs into small elements, and [11] utilized non-adjacent form. Concerning the asymptotic proof size, the SNARK constructions in [7, 36] provide a proof size that scales with the square root of the input size. The leveled Ajtai commitment scheme in [15] further reduces the asymptotic proof size by generalizing the approach in [7]. Meanwhile, the proof size in [11] and the Bulletproof variant of [15] provide a polylogarithmic scale. To achieve zero-knowledgeness, [7, 15, 36] utilize rejection sampling, but the zero-knowledgeness of [11] and the Bulletproof variant of [15] were not precisely described in the literature. Regarding the concrete instantiation, [36] and [11] provide concrete proof sizes that can be practically deployed, but the performance of the prover and verifier was not provided.

## 2 Preliminaries

### 2.1 Notation

For a positive integer  $q$ , we use  $\mathbb{Z} \cap (-q/2, q/2]$  as a representative set of  $\mathbb{Z}_q$ , and denote by  $[a]_q$  the reduction of  $a$  modulo  $q$ . Vectors over  $\mathbb{Z}$  or  $\mathbb{Z}_q$  are denoted with regular lowercase letters and arrows, such as  $\vec{v}$ , and matrices over  $\mathbb{Z}$  or  $\mathbb{Z}_q$

are represented by regular uppercase letters. We regard all vectors as column vectors, and we use the symbol  $\parallel$  for the concatenation of two vectors.

Let  $d$  be a power of two. We denote by  $R = \mathbb{Z}[X]/(X^d + 1)$  the ring of integers of the  $2d$ -th cyclotomic field and  $R_q = \mathbb{Z}_q[X]/(X^d + 1)$  the residue ring of  $R$  modulo  $q$ . For polynomials in  $R$  or  $R_q$ , we use bold lowercase letters to denote them e.g.  $\mathbf{f}$ . We often regard them as  $d$ -dimensional vectors over  $\mathbb{Z}$  or  $\mathbb{Z}_q$  with components corresponding to coefficients. Vectors over  $R$  or  $R_q$  are denoted with bold lowercase letters and arrows, such as  $\vec{\mathbf{f}}$ , and matrices over  $R$  or  $R_q$  are represented by bold uppercase letters.

For a vector  $\vec{v} = (v_0, \dots, v_{n-1}) \in \mathbb{Z}^n$ , the  $\ell^p$  ( $p \geq 1$ ) and  $\ell^\infty$  norms are defined as follows:

$$\|v\|_p := \sqrt[p]{\sum_{i=0}^{n-1} |v_i|^p}, \quad \|v\|_\infty := \max_{0 \leq i < n} |v_i|$$

For a polynomial  $\mathbf{f}$  or a vector of polynomials  $\vec{\mathbf{f}}$ ,  $\|\mathbf{f}\|_p$  and  $\|\vec{\mathbf{f}}\|_p$  are calculated by regarding them as coefficient vectors. For a matrix  $A \in \mathbb{R}^{n \times n}$ , we denote the matrix norm of  $A$  by  $\|A\|_2 := \max_{0 \neq \vec{x} \in \mathbb{R}^n} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2}$ .

## 2.2 Probability Distributions

We denote sampling  $x$  from the distribution  $\mathcal{D}$  by  $x \leftarrow \mathcal{D}$ . For distributions  $\mathcal{D}_1$  and  $\mathcal{D}_2$  over a countable set  $S$  (e.g.  $\mathbb{Z}^n$ ), the statistical distance of  $\mathcal{D}_1$  and  $\mathcal{D}_2$  is defined as  $\frac{1}{2} \cdot \sum_{x \in S} |\mathcal{D}_1(x) - \mathcal{D}_2(x)| \in [0, 1]$ . We denote the uniform distribution over  $S$  by  $\mathcal{U}(S)$  when  $S$  is finite.

We define the  $n$ -dimensional spherical Gaussian function  $\rho : \mathbb{R}^n \rightarrow (0, 1]$  as  $\rho(\vec{x}) := \exp(-\pi \cdot \vec{x}^\top \vec{x})$ . In general, for a positive definite matrix  $\Sigma \in \mathbb{R}^{n \times n}$ , we define the elliptical Gaussian function  $\rho_{\sqrt{\Sigma}} : \mathbb{R}^n \rightarrow (0, 1]$  as  $\rho_{\sqrt{\Sigma}}(\vec{x}) := \exp(-\pi \cdot \vec{x}^\top \Sigma^{-1} \vec{x})$ . Let  $\Lambda \subseteq \mathbb{R}^n$  be a lattice and  $\vec{c} \in \mathbb{R}^n$ . The discrete Gaussian distribution  $\mathcal{D}_{\vec{c}+\Lambda, \sqrt{\Sigma}}$  is defined as a distribution over the coset  $\vec{c} + \Lambda$ , whose probability mass function is  $\mathcal{D}_{\vec{c}+\Lambda, \sqrt{\Sigma}}(\vec{x}) = \rho_{\sqrt{\Sigma}}(\vec{x}) / \rho_{\sqrt{\Sigma}}(\vec{c} + \Lambda)$  for  $\vec{x} \in \vec{c} + \Lambda$  where  $\rho_{\sqrt{\Sigma}}(\vec{c} + \Lambda) := \sum_{\vec{v} \in \vec{c} + \Lambda} \rho_{\sqrt{\Sigma}}(\vec{v}) < \infty$ . When  $\Sigma = \sigma^2 \cdot I_n$  for  $\sigma > 0$  where  $I_n$  is the  $n$ -dimensional identity matrix, then we substitute  $\sqrt{\Sigma}$  by  $\sigma$  in the subscript and refer to  $\sigma$  as the width parameter. For a polynomial  $\mathbf{x}$ , we denote by  $\mathbf{x} \leftarrow \mathcal{D}_{\vec{c}+\Lambda, \sqrt{\Sigma}}$  if we sample its coefficient vector from  $\mathcal{D}_{\vec{c}+\Lambda, \sqrt{\Sigma}}$ .

## 2.3 Useful Lemmas

**Definition 1 (Def. 3.1 [34]).** For an  $n$ -dimensional lattice  $\Lambda$  and positive real  $\varepsilon > 0$ , the smoothing parameter  $\eta_\varepsilon(\Lambda)$  is the smallest  $s$  such that  $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \varepsilon$ .

**Definition 2 (Def. 2.3 [37]).** Let  $\Sigma$  be a positive-definite matrix. We say that  $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$  if  $\eta_\varepsilon(\sqrt{\Sigma}^{-1} \cdot \Lambda) \leq 1$ .

**Lemma 1 (Lem. 3.3 [34]).** For any  $n$ -dimensional lattice  $\Lambda$  and  $\varepsilon > 0$ ,

$$\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n(\Lambda)$$

where  $\lambda_n(\Lambda)$  is the smallest real number  $r > 0$  such that  $\dim(\text{span}(\Lambda \cap \mathcal{B}(r))) = n$  and  $\mathcal{B}(r)$  is the  $n$ -dimensional ball with radius  $r$  centered at the origin.

**Lemma 2 (Lem. 5 [27]).** For a positive-definite matrix  $\Sigma$ ,  $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$  holds if  $\|\Sigma^{-1}\|_2 \leq \eta_\varepsilon(\Lambda)^{-2}$ .

**Lemma 3 (Lem. 2.4 [37]).** Let  $\Lambda \subseteq \mathbb{R}^n$  be any  $n$ -dimensional lattice. For any  $0 < \varepsilon < 1$ ,  $\Sigma > 0$  such that  $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$ , and any  $\vec{c} \in \mathbb{R}^n$ ,

$$\rho_{\sqrt{\Sigma}}(\vec{c} + \Lambda) \in \left[ \frac{1-\varepsilon}{1+\varepsilon}, 1 \right] \cdot \rho_{\sqrt{\Sigma}}(\Lambda)$$

**Lemma 4 (Lem. 2.4 [6]).** Let  $\Lambda \subseteq \mathbb{R}^n$  be any  $n$ -dimensional lattice. For any  $0 < \varepsilon < 1/3$ ,  $\sigma \geq \eta_\varepsilon(\Lambda)$ , and any  $\vec{c} \in \mathbb{R}^n$ ,

$$\Pr[\|\vec{x}\|_\infty > 5\sigma \mid \vec{x} \leftarrow \mathcal{D}_{\vec{c}+\Lambda, \sigma}] \leq n \cdot 2^{-111}$$

**Lemma 5 (Lem. 4.4 [34]).** Let  $\Lambda \subseteq \mathbb{R}^n$  be any  $n$ -dimensional lattice. For any  $0 < \varepsilon < 1/3$ ,  $\sigma \geq \eta_\varepsilon(\Lambda)$ , and any  $\vec{c} \in \mathbb{R}^n$ ,

$$\Pr[\|\vec{x}\|_2 > \sigma\sqrt{n} \mid \vec{x} \leftarrow \mathcal{D}_{\vec{c}+\Lambda, \sigma}] \leq 2^{-n+1}$$

**Lemma 6 (Fact 2.1 [37]).** Let  $\Sigma_1, \Sigma_2 > 0$  be positive-definite matrices. Define  $\Sigma_0^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1} > 0$  and  $\Sigma_3 = \Sigma_1 + \Sigma_2 > 0$ . Let  $\vec{x}, \vec{c}_1, \vec{c}_2 \in \mathbb{R}^n$  be arbitrary, and let  $\vec{c}_0 \in \mathbb{R}^n$  be such that  $\Sigma_0^{-1}\vec{c}_0 = \Sigma_1^{-1}\vec{c}_1 + \Sigma_2^{-1}\vec{c}_2$ . Then,

$$\rho_{\sqrt{\Sigma_1}}(\vec{x} - \vec{c}_1) \cdot \rho_{\sqrt{\Sigma_2}}(\vec{x} - \vec{c}_2) = \rho_{\sqrt{\Sigma_0}}(\vec{x} - \vec{c}_0) \cdot \rho_{\sqrt{\Sigma_3}}(\vec{c}_1 - \vec{c}_2)$$

**Lemma 7 (Simplified Thm. 1 [37]).** Let  $\Sigma_1, \Sigma_2 > 0$  be positive-definite matrices. Let  $\Lambda_1, \Lambda_2 \subseteq \mathbb{R}^n$  be  $n$ -dimensional lattices such that  $\Lambda_2 \subseteq \Lambda_1$ ,  $\sqrt{\Sigma_1} \geq \eta_\varepsilon(\Lambda_1)$ , and  $\sqrt{(\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}} \geq \eta_\varepsilon(\Lambda_2)$  for some  $0 \leq \varepsilon < 1/2$ . Then, for any  $\vec{c}_1, \vec{c}_2 \in \mathbb{R}^n$ , the distribution  $D_{\vec{c}_1+\Lambda_1, \sqrt{\Sigma_1}} + D_{\vec{c}_2+\Lambda_2, \sqrt{\Sigma_2}}$  is within statistical distance  $8\varepsilon$  of  $D_{\vec{c}_1+\vec{c}_2+\Lambda_1, \sqrt{\Sigma_1+\Sigma_2}}$ .

**Lemma 8 (Lem. 3.1 [10]).** Let  $d$  be a power of two, and let  $0 \leq i, j < 2d$  such that  $i \neq j$ . Then,  $2(X^i - X^j)^{-1}$  is an element of  $R$  such that

$$\|2(X^i - X^j)^{-1}\|_\infty \leq 1,$$

where the inverse of  $(X^i - X^j)$  is taken over the field  $\mathbb{Q}[X]/(X^d + 1)$ .

## 2.4 Module SIS/LWE

**Definition 3.** Let  $\mu, \ell$  be positive integers, and  $0 < \beta < q$ . Then, the goal of the Module-SIS (MSIS) problem is to find, for a given matrix  $\mathbf{A} \leftarrow \mathcal{U}(R_q^{\mu \times \ell})$ ,  $\vec{\mathbf{x}} \in R_q^{\mu + \ell}$  such that  $[\mathbf{I}_\mu | \mathbf{A}] \vec{\mathbf{x}} = \mathbf{0} \pmod{q}$  and  $\|\vec{\mathbf{x}}\|_2 < \beta$ . We say that a PPT adversary  $\mathcal{A}$  has advantages  $\varepsilon$  in solving  $\text{MSIS}_{R, \mu, q, \beta}$  if

$$\Pr [\|\vec{\mathbf{x}}\|_2 < \beta \wedge [\mathbf{I}_\mu | \mathbf{A}] \vec{\mathbf{x}} = \mathbf{0} \pmod{q} \mid \mathbf{A} \leftarrow \mathcal{U}(R_q^{\mu \times \ell}); \vec{\mathbf{x}} \leftarrow \mathcal{A}(\mathbf{A})] \geq \varepsilon.$$

**Definition 4.** Let  $\nu, \ell$  be positive integer, and  $\chi$  be a distribution over  $R^{\nu + \ell}$ . Then, the goal of the Module-LWE (MLWE) problem is to distinguish  $(\mathbf{A}, \vec{\mathbf{u}})$  from  $(\mathbf{A}, [\mathbf{A} | \mathbf{I}_\ell] \vec{\mathbf{r}})$  for  $\mathbf{A} \leftarrow \mathcal{U}(R_q^{\ell \times \nu})$ ,  $\vec{\mathbf{u}} \leftarrow \mathcal{U}(R_q^\ell)$ , and  $\vec{\mathbf{r}} \leftarrow \chi$ . We say that a PPT adversary  $\mathcal{A}$  has advantages  $\varepsilon$  in solving  $\text{MLWE}_{R, \nu, q, \chi}$  if

$$\begin{aligned} & \left| \Pr [b = 1 \mid \mathbf{A} \leftarrow \mathcal{U}(R_q^{\ell \times \nu}); \vec{\mathbf{r}} \leftarrow \chi; b \leftarrow \mathcal{A}(\mathbf{A}, [\mathbf{A} | \mathbf{I}_\ell] \vec{\mathbf{r}})] \right. \\ & \left. - \Pr [b = 1 \mid (\mathbf{A}, \vec{\mathbf{u}}) \leftarrow \mathcal{U}(R_q^{\ell \times \nu} \times R_q^\ell); b \leftarrow \mathcal{A}(\mathbf{A}, \vec{\mathbf{u}})] \right| \geq \varepsilon. \end{aligned}$$

For spherical discrete Gaussian distributions, we replace them with their width parameters in the MLWE notation for simplicity.

For both problems, the value of  $\ell$  in the hardness estimation is not significant, so we omit it in the parameters for both problems.

## 2.5 Hint-MLWE

The *Hint-MLWE* problem is a variant of MLWE introduced in recent literature [27, 33]. This hardness problem is useful when proving the simulatability of lattice-based proof systems without relying on the rejection sampling technique [31].

**Definition 5 (Hint-MLWE).** Let  $\nu, \ell, n$  be positive integers,  $\chi$  and  $\psi_0, \dots, \psi_{n-1}$  be distributions over  $R^{\nu + \ell}$ , and  $\mathbf{c}_0, \dots, \mathbf{c}_{n-1}$  be elements in  $R$ . The *Hint-MLWE* problem, denoted by  $\text{HintMLWE}_{R, \nu, q, \chi, \psi_0, \dots, \psi_{n-1}}^{\mathbf{c}_0, \dots, \mathbf{c}_{n-1}}$ , asks the adversary  $\mathcal{A}$  to distinguish the following two distributions:

1.  $\left( \mathbf{A}, [\mathbf{A} | \mathbf{I}_\ell] \vec{\mathbf{r}}, \vec{\mathbf{z}}_0, \dots, \vec{\mathbf{z}}_{k-1} \right)$  for  $\mathbf{A} \leftarrow \mathcal{U}(R_q^{\ell \times \nu})$ ,  $\vec{\mathbf{r}} \leftarrow \chi$ ,  $\vec{\mathbf{y}}_i \leftarrow \psi_i$ , and  $\vec{\mathbf{z}}_i = \mathbf{c}_i \cdot \vec{\mathbf{r}} + \vec{\mathbf{y}}_i$  for  $0 \leq i < n$ .
2.  $\left( \mathbf{A}, \vec{\mathbf{u}}, \vec{\mathbf{z}}_0, \dots, \vec{\mathbf{z}}_{n-1} \right)$  for  $\mathbf{A} \leftarrow \mathcal{U}(R_q^{\ell \times \nu})$ ,  $\vec{\mathbf{u}} \leftarrow \mathcal{U}(R_q^\ell)$ ,  $\vec{\mathbf{r}} \leftarrow \chi$ ,  $\vec{\mathbf{y}}_i \leftarrow \psi_i$ , and  $\vec{\mathbf{z}}_i = \mathbf{c}_i \cdot \vec{\mathbf{r}} + \vec{\mathbf{y}}_i$  for  $0 \leq i < n$ .

When  $\chi$  and  $\psi_0, \dots, \psi_{n-1}$  are spherical discrete Gaussian distributions, we simply use their width parameters to denote the *Hint-MLWE* problem for simplicity.

**Theorem 1 (Generalized Thm. 1 [27]).** Let  $\nu, n$  be positive integers and  $\mathbf{c}_0, \dots, \mathbf{c}_{n-1}$  be elements of  $R$ . For  $\sigma_1, \sigma_{2,0}, \dots, \sigma_{2,n-1} > 0$ , let  $\sigma > 0$  be defined as  $1/\sigma^2 = 2(1/\sigma_1^2 + \sum_{i=0}^{n-1} B_i^2/\sigma_{2,i}^2)$ , where  $B_i$ 's are upper bounds for  $\|\mathbf{c}_i\|_1$ 's. If  $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$  for some  $\varepsilon = \text{negl}(\lambda)$ , then there exists an efficient reduction from  $\text{MLWE}_{R, \nu, q, \sigma}$  to  $\text{HintMLWE}_{R, \nu, q, \sigma_1, \sigma_{2,0}, \dots, \sigma_{2,n-1}}^{\mathbf{c}_0, \dots, \mathbf{c}_{n-1}}$ .

## 2.6 Commitment Scheme

We recall the notion of a commitment scheme. A commitment scheme consists of three PPT algorithms (**Setup**, **Com**, **Open**).

- **Setup**( $1^\lambda$ )  $\rightarrow$   $\text{ck}$ : Given a security parameter  $\lambda$ , it generates a commitment key  $\text{ck}$ .
- **Com**( $\text{ck}, m$ )  $\rightarrow$   $(c, \delta)$ : Given a message  $m$ , it returns a commitment  $c$  and an opening  $\delta$ .
- **Open**( $\text{ck}, c, m, \delta$ )  $\rightarrow$   $b$ : Given a commitment  $c$ , a message  $m$ , and an opening  $\delta$ , it outputs 0 or 1.

**Definition 6 (Hiding).** A commitment scheme satisfies hiding if for all PPT adversary  $\mathcal{A}$ ,

$$\left| \Pr \left[ b = b' \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ (m_0, m_1) \leftarrow \mathcal{A}(\text{ck}) \\ b \leftarrow \mathcal{U}(\{0, 1\}) \\ (c, \delta) \leftarrow \text{Com}(\text{ck}, m_b) \\ b' \leftarrow \mathcal{A}(c) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}(\lambda).$$

**Definition 7 (Binding).** A commitment scheme satisfies binding if for all PPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \begin{array}{l} \text{Open}(\text{ck}, c, m, \delta) = 1 \wedge \\ \text{Open}(\text{ck}, c, m', \delta') = 1 \wedge \\ m \neq m' \end{array} \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ (c, m, m', \delta, \delta') \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] \leq \text{negl}(\lambda).$$

In some applications of commitment schemes, such as zero-knowledge proof systems, an interactive protocol called proof of opening knowledge is required, which proves the knowledge of opening  $\delta$  for given commitments without revealing the committed messages. Its security is defined as follows, adapted from [27, 32].

**Definition 8.** An interactive protocol  $(\mathcal{P}, \mathcal{V})$  is called a secure proof of opening knowledge protocol for a commitment scheme (**Setup**, **Com**, **Open**) if it satisfies the followings:

- **Completeness.** For all message  $m$ ,

$$\Pr \left[ b = 1 \wedge \text{Open}(\text{ck}, c, m, \delta) = 1 \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ (c, \delta) \leftarrow \text{Com}(\text{ck}, m) \\ (\pi, b) \leftarrow \langle \mathcal{P}(\text{ck}, c, \delta), \mathcal{V}(\text{ck}, c) \rangle \end{array} \right] \geq 1 - \text{negl}(\lambda).$$



- **Soundness.** For all PPT adversaries  $\mathcal{P}^*$ , there exists an expected polynomial time extractor  $\mathcal{E}$  such that

$$\Pr \left[ b = 1 \wedge \text{Open}(\text{ck}, c, m, \delta) = 0 \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ c \leftarrow \mathcal{P}^*(\text{ck}) \\ (\pi, b) \leftarrow \langle \mathcal{P}^*(\text{ck}, c), \mathcal{V}(\text{ck}, c) \rangle \\ (m, \delta) \leftarrow \mathcal{E}^{\mathcal{P}^*}(\text{ck}, c) \end{array} \right] \leq \text{negl}(\lambda).$$

- **Simulatability.** There exists a PPT simulator  $\mathcal{S}$  such that for all PPT adversaries  $\mathcal{A}$ ,

$$\left| \Pr \left[ \mathcal{A}(c, \pi) = 1 \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ m \leftarrow \mathcal{A}(\text{ck}) \\ (c, \pi) \leftarrow \mathcal{S}(\text{ck}) \end{array} \right] - \Pr \left[ \mathcal{A}(c, \pi) = 1 \mid \begin{array}{l} \text{ck} \leftarrow \text{Setup}(1^\lambda) \\ m \leftarrow \mathcal{A}(\text{ck}) \\ (c, \delta) \leftarrow \text{Com}(\text{ck}, m) \\ (\pi, b) \leftarrow \langle \mathcal{P}(\text{ck}, c, \delta), \mathcal{V}(\text{ck}, c) \rangle \end{array} \right] \right| \leq \text{negl}(\lambda).$$

## 2.7 Polynomial Commitment Scheme

We recall the notion of a polynomial commitment scheme, adapted from [26]. A polynomial commitment scheme consists of five PPT algorithms ( $\text{PC.Setup}$ ,  $\text{PC.Com}$ ,  $\text{PC.Open}$ ,  $\text{PC.Eval}$ ,  $\text{PC.Verify}$ ).

- $\text{PC.Setup}(1^\lambda, N) \rightarrow \text{ck}$ : Given a security parameter  $\lambda$  and a polynomial degree upper bound  $N$ , it generates a commitment key  $\text{ck}$ .
- $\text{PC.Com}(\text{ck}, h(X)) \rightarrow (c, \delta)$ : Given a polynomial  $h(X) \in \mathbb{Z}_p[X]$  of degree  $< N$ , it generates a commitment  $c$  and an opening  $\delta$ .
- $\text{PC.Open}(\text{ck}, c, h(X), \delta) \rightarrow b$ : Given a commitment  $c$ , an opening  $\delta$ , and polynomial  $h(X)$ , it outputs 0 or 1.
- $\text{PC.Eval}(x, \delta) \rightarrow (y, \rho)$ : Given an opening  $\delta$  and an evaluation point  $x \in \mathbb{Z}_p$ , it returns an evaluation result  $y$ , and an evaluation proof  $\rho$ .
- $\text{PC.Verify}(\text{ck}, c, x, y, \rho) \rightarrow b$ : Given a commitment  $c$ , an evaluation point  $x$ , an evaluation result  $y$ , and an evaluation proof  $\rho$ , it outputs 0 or 1.

**Definition 9 (Hiding).** A polynomial commitment scheme satisfies hiding if for all PPT adversary  $\mathcal{A}$ ,

$$\left| \Pr \left[ b = b' \mid \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (h_0(X), h_1(X)) \leftarrow \mathcal{A}(\text{ck}) \\ b \leftarrow \mathcal{U}(\{0, 1\}) \\ (c, \delta) \leftarrow \text{PC.Com}(\text{ck}, h_b(X)) \\ b' \leftarrow \mathcal{A}(c) \end{array} \right] - \frac{1}{2} \right| \leq \text{negl}(\lambda).$$

**Definition 10 (Binding).** A polynomial commitment scheme satisfies binding if for all PPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \begin{array}{l} \text{PC.Open}(\text{ck}, c, h(X), \delta) = 1 \wedge \\ \text{PC.Open}(\text{ck}, c, h'(X), \delta') = 1 \wedge \\ h(X) \neq h'(X) \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (c, h(X), h'(X), \delta, \delta') \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] \leq \text{negl}(\lambda).$$

**Definition 11 (Evaluation Binding).** A polynomial commitment scheme satisfies evaluation binding if for all PPT adversary  $\mathcal{A}$ ,

$$\Pr \left[ \begin{array}{l} \text{PC.Verify}(\text{ck}, c, x, y, \rho) = 1 \wedge \\ \text{PC.Verify}(\text{ck}, c, x, y', \rho') = 1 \wedge \\ y \neq y' \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (c, x, y, y', \rho, \rho') \leftarrow \mathcal{A}(\text{ck}) \end{array} \right] \leq \text{negl}(\lambda).$$

**Definition 12 (Evaluation Hiding).** A polynomial commitment scheme satisfies evaluation hiding if there exists a PPT simulator  $\mathcal{S}$  such that for all PPT adversary  $\mathcal{A}$ ,

$$\left| \Pr \left[ \begin{array}{l} \mathcal{A}(c, \rho) = 1 \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (h(X), x) \leftarrow \mathcal{A}(\text{ck}) \\ (c, \rho) \leftarrow \mathcal{S}(\text{ck}, x, h(x)) \end{array} \right] \right. \\ \left. - \Pr \left[ \begin{array}{l} \mathcal{A}(c, \rho) = 1 \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (h(X), x) \leftarrow \mathcal{A}(\text{ck}) \\ (c, \delta) \leftarrow \text{PC.Com}(\text{ck}, h(X)) \\ (y, \rho) \leftarrow \text{PC.Eval}(x, \delta) \end{array} \right] \right| \leq \text{negl}(\lambda).$$

Similar to commitment schemes, some applications of polynomial commitment schemes, such as zk-SNARK, require an interactive protocol called proof of opening knowledge. It proves the knowledge of opening  $\delta$  for given commitments without revealing the committed polynomials. We adapt its definition from [3, 25].

**Definition 13.** An interactive protocol  $(\mathcal{P}, \mathcal{V})$  is called a secure proof of opening knowledge protocol for a polynomial commitment scheme  $(\text{PC.Setup}, \text{PC.Com}, \text{PC.Open}, \text{PC.Eval}, \text{PC.Verify})$  if it satisfies the followings:

- **Completeness.** For every polynomial  $h(X) \in \mathbb{Z}_p[X]$  of degree  $< N$  and every point  $x \in \mathbb{Z}_p$ ,

$$\Pr \left[ \begin{array}{l} b = 1 \wedge \\ \text{PC.Open}(\text{ck}, c, h(X), \delta) = 1 \wedge \\ \text{PC.Verify}(\text{ck}, c, x, h(x), \rho) = 1 \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^{\lambda, N}) \\ (c, \delta) \leftarrow \text{PC.Com}(\text{ck}, h(X)) \\ (y, \rho) \leftarrow \text{PC.Eval}(\delta, x) \\ (\pi, b) \leftarrow \langle \mathcal{P}(\text{ck}, c, \delta), \mathcal{V}(\text{ck}, c) \rangle \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

- **Soundness.** For all PPT adversaries  $\mathcal{P}^*$ , there exists an expected polynomial time extractor  $\mathcal{E}$  such that

$$\Pr \left[ \begin{array}{l} b = 1 \wedge \\ \text{PC.Verify}(\text{ck}, c, x, y, \rho) = 1 \wedge \\ (\text{PC.Open}(\text{ck}, c, h(X), \delta) = 0 \vee y \neq h(x)) \end{array} \middle| \begin{array}{l} \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (c, x, y, \rho) \leftarrow \mathcal{P}^*(\text{ck}) \\ (\pi, b) \leftarrow \langle \mathcal{P}^*(\text{ck}, c), \mathcal{V}(\text{ck}, c) \rangle \\ (h(X), \delta) \leftarrow \mathcal{E}^{\mathcal{P}^*}(\text{ck}, c, x, y, \rho) \end{array} \right] \leq \text{negl}(\lambda)$$

- **Simulatability.** There exists a PPT simulator  $\mathcal{S}$  such that for all PPT adversaries  $\mathcal{A}$ ,

$$\left| \Pr \left[ \begin{array}{l} \mathcal{A}(c, \rho, \pi) = 1 \\ \text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N) \\ (h(X), x) \leftarrow \mathcal{A}(\text{ck}) \\ (c, \pi, \rho) \leftarrow \mathcal{S}(\text{ck}, x, h(x)) \end{array} \right] - \Pr \left[ \begin{array}{l} \mathcal{A}(c, \rho, \pi) = 1 \\ \text{ck} \leftarrow \text{PC.Setup}(1^\lambda) \\ (h(X), x) \leftarrow \mathcal{A}(\text{ck}) \\ (c, \delta) \leftarrow \text{PC.Com}(\text{ck}, h(X)) \\ (y, \rho) \leftarrow \text{PC.Eval}(\delta, x) \\ (\pi, b) \leftarrow \langle \mathcal{P}(\text{ck}, c, \delta), \mathcal{V}(\text{ck}, c) \rangle \end{array} \right] \right| \leq \text{negl}(\lambda).$$

### 3 Revisiting the Ajtai Commitment Scheme

In this section, we revisit the Ajtai commitment scheme [1] and its proof of opening knowledge (POK) protocol, which we utilize as the main building block for our polynomial commitment scheme. Our construction is based on the previous work by Baum et al. [7], which builds a zkSNARK from the Ajtai commitment scheme.

For the Ajtai commitment scheme, we use it to commit coefficients of an input polynomial from  $\mathbb{Z}_p[X]$ , so it needs to provide a compact commitment for messages in  $\mathbb{Z}_p$ . In the applications of polynomial commitments,  $p$  is usually set to a large prime to attain negligible soundness error. However, the previous approach does not effectively handle such cases, so naive adaptation results in parameter blow-ups.<sup>2</sup> To address this, we introduce a novel encoding map for the Ajtai commitment scheme, which transforms integers in  $\mathbb{Z}_p$  into a polynomial with coefficients much smaller than  $p$  while still preserving the homomorphic property. This results in a more compact commitment size, preventing parameter blow-ups.

For our POK protocol, we achieve simulatability without introducing rejection sampling. Previous constructions used the rejection sampling method [31] to achieve simulatability. However, this approach can degrade the prover’s performance since the repetition rate grows exponentially with the number of committed polynomials. This is particularly problematic for use cases of polynomial

<sup>2</sup> Baum et al. [7] presented an effective encoding method for a finite field  $GF(p^k)$  with a small prime  $p$ , but it does not cover large primes.

commitments, such as PIOP, which require committing to multiple polynomials. To address this issue, we eliminate rejection sampling in the POK protocol through a newly developed randomized encoding technique. In the rest of this section, we provide more details on our new approaches to the Ajtai commitment scheme and its POK protocol.

### 3.1 New Encoding Method

Let us first recall the basic Ajtai commitment scheme. For a message  $\vec{m} \in R^\ell$  and a randomness  $\vec{\mu}$  sampled from a distribution over  $R^{\mu+\nu}$ , the commitment  $\vec{m} \in R_q^\mu$  is obtained as

$$\vec{m} = \mathbf{A}_0 \vec{m} + \mathbf{A}_1 \vec{\mu} \pmod{q}$$

where  $\mathbf{A}_0 \in R_q^{\mu \times \ell}$  and  $\mathbf{A}_1 \in R_q^{\mu \times (\mu+\nu)}$  are (fixed) random matrices. In particular, the binding property of the Ajtai commitment scheme heavily depends on the size of  $\vec{m}$ .

In the prior work [7], a simple encoding method is used, which embeds elements of  $\mathbb{Z}_p$  into the coefficients of a polynomial in  $R$ . Therefore, the commitment modulus  $q$  should be sufficiently larger than  $p$  to meet the security requirements. If  $p$  is a large prime, this approach can lead to significant inefficiencies due to the scaling of commitments. To address this issue, we introduce a novel encoding map, which maps messages from  $\mathbb{Z}_p$  for large primes into elements of a polynomial ring with small coefficients. Our encoding map is inspired by [19, 20], which proposes a novel packing method for homomorphic encryption to instantiate high-precision arithmetic.

**Lemma 9.** *Let  $b$  and  $r$  be positive integers such that  $r \mid d$  and  $p = b^r + 1$  is prime. Then,  $R/(X^{d/r} - b)$  is isomorphic to  $\mathbb{Z}_p^{d/r}$  as  $\mathbb{Z}$ -modules.*

*Proof.* Using the fact that  $R/(X^{d/r} - b) = \mathbb{Z}[X]/(X^d + 1, X^{d/r} - b)$ , we obtain the following isomorphism.

$$\begin{aligned} R/(X^{d/r} - b) &= \bigoplus_{i=0}^{d/r-1} X^i \cdot \mathbb{Z}[X^{d/r}]/(X^d + 1, X^{d/r} - b) \\ &= \bigoplus_{i=0}^{r-1} X^i \cdot \mathbb{Z}_p[X^{d/r}]/(X^{d/r} - b) \cong \prod_{i=0}^{d/r-1} \mathbb{Z}_p[X^{d/r}]/(X^{d/r} - b). \end{aligned}$$

We note that  $\mathbb{Z}_p[X^{d/r}]/(X^{d/r} - b) \cong \mathbb{Z}_p$  via an isomorphism  $f(X^{d/r}) \mapsto f(b)$ . Therefore, we have  $R/(X^{d/r} - b) \cong \mathbb{Z}_p^{d/r}$  with respect to the following isomorphism  $\varphi$  from  $R/(X^{d/r} - b)$  to  $\mathbb{Z}_p^{d/r}$ :

$$\varphi : \vec{a} = \sum_{i=0}^{d/r-1} \sum_{j=0}^{r-1} a_{(d/r)j+i} X^{(d/r)j+i} \mapsto \left( \sum_{j=0}^{r-1} a_{(d/r)j} b^j, \dots, \sum_{j=0}^{r-1} a_{(d/r)(j+1)-1} b^j \right)$$

We note that the isomorphism  $\varphi$  is defined over the polynomial representation on  $R$  of an element  $\bar{\mathbf{a}}$  in  $R/(X^{d/r} - b)$ . It is well-defined, because for any two polynomial representations  $\mathbf{a}$  and  $\mathbf{a}'$  in  $R$  of  $\bar{\mathbf{a}}$ , we have  $\mathbf{a} = \mathbf{a}' \pmod{X^{d/r} - b}$  and  $\varphi$  maps  $X^{d/r} - b$  to zero.  $\square$

We recall that the space of encoded message in the Ajtai commitment scheme is  $R^\ell$ , so we need to design an encoding map from  $\mathbb{Z}_p^{d/r}$  to  $R$ . The main idea is to utilize the isomorphism between  $\mathbb{Z}_p^{d/r}$  and  $R/(X^{d/r} - b)$ , with  $R$  serving as the representation set for elements of  $R/(X^{d/r} - b)$ . As stated in the above lemma, the isomorphism  $\varphi : R/(X^{d/r} - b) \rightarrow \mathbb{Z}_p^{d/r}$  can be naturally extended to the map over  $R$ , which can be considered as a decoding procedure, the inverse of encoding. Hence, we aim to construct an encoding map that outputs a polynomial representation of  $\mathbf{a} = \varphi^{-1}(\bar{\mathbf{a}}) \in R/(X^{d/r} - b)$  in  $R$  with small coefficients for  $\bar{\mathbf{a}} \in \mathbb{Z}_p^{d/r}$ . To achieve this, we present an algorithm in Alg. 1 that outputs a polynomial representation of  $\mathbf{a}$  with an upper bound  $\|\mathbf{a}\|_\infty \leq \frac{1}{2}(b+2)$  since  $|a_{i,j}| \leq \frac{1}{2}b$  and  $|c_{i,j}| \leq 1$ .

---

**Algorithm 1** New encoding method
 

---

**Input:**  $\bar{\mathbf{a}} = (a_0, \dots, a_{d/r-1}) \in \mathbb{Z}_p^{d/r}$   
**Output:**  $\mathbf{a} \in R$

- 1: **for**  $0 \leq i < d/r$  **do**
- 2:     **if**  $a_i = p - 1 \pmod{p}$  **then**
- 3:          $(a_{i,0}, \dots, a_{i,r-1}) \leftarrow (0, \dots, 0, b)$
- 4:     **else**
- 5:          $(a_{i,0}, \dots, a_{i,r-1})$  is the base- $b$  representation of  $0 \leq a_i < b^r$
- 6:     **end if**
- 7: **end for**
- 8: **for**  $0 \leq i < d/r, 0 \leq j < r$  **do**
- 9:     **if**  $a_{i,j} > b/2$  **then**
- 10:          $a_{i,j} \leftarrow a_{i,j} - b, \quad c_{i,j} \leftarrow 1$
- 11:     **else**
- 12:          $c_{i,j} \leftarrow 0$
- 13:     **end if**
- 14: **end for**
- 15:  $\mathbf{a} \leftarrow \sum_{i=0}^{d/r-1} \sum_{j=0}^{r-1} a_{i,j} X^{(d/r)j+i} + \sum_{i=0}^{d/r-1} \sum_{j=0}^{r-1} c_{i,j} X^{(d/r)(j+1)+i}$

---

Below, we provide the encoding and decoding algorithms for our commitment scheme.

- $\text{Ecd}(\bar{\mathbf{a}}) \rightarrow \mathbf{a}$ : Given an element  $\bar{\mathbf{a}} = (a_0, \dots, a_{d/r-1}) \in \mathbb{Z}_p^{d/r}$ , it runs Alg. 1 and outputs a ring element  $\mathbf{a}$  where  $\|\mathbf{a}\|_\infty \leq \frac{b+2}{2}$ .
- $\text{Dcd}(\mathbf{a}) \rightarrow \bar{\mathbf{a}}$ : Given a ring element  $\mathbf{a} = \sum_{i=0}^{d/r-1} \sum_{j=0}^{r-1} a_{i,j} X^{(d/r)j+i} \in R$ , it outputs  $\bar{\mathbf{a}} = \varphi(\mathbf{a}) = \left( \sum_{j=0}^{r-1} a_{0,j} b^j, \dots, \sum_{j=0}^{r-1} a_{d/r-1,j} b^j \right) \in \mathbb{Z}_p^{d/r}$ .

We note that our encoding maps a message vector in  $\mathbb{Z}_p^{d/r}$  into a polynomial in  $R$  whose norm is bounded by  $\frac{b+2}{2}$ , which is much smaller than  $p$ , while preserving homomorphic property. This allows for a compact commitment size when committing messages from large prime fields, as the commitment modulus can be chosen to be small while still maintaining the binding properties. As an abuse of notation, we often put an integer  $a \in \mathbb{Z}_p$  as an input for the encoding algorithms. In this case,  $\text{Ecd}(a)$  outputs a ring element  $\mathbf{a} = \text{Ecd}(\vec{a})$ , where  $\vec{a} = (a, 0, \dots, 0)$  such that  $\|\mathbf{a}\|_1 \leq \frac{(b+2)r}{2}$  since there are at most  $r$  non-zero coefficients.

Now we explain how our encoding method affects the commitment modulus  $q$ . As mentioned above, the binding property of the Ajtai commitment  $\vec{\mathbf{m}} = \mathbf{A}_0 \vec{\mathbf{m}} + \mathbf{A}_1 \vec{\mu} \pmod{q}$  is closely related to the size of a message  $\vec{\mathbf{m}}$ , which is reduced to the hardness of  $\text{MSIS}_{R, \mu, q, \beta}$  where  $\beta = 2\|\vec{\mathbf{m}}\|\|\vec{\mu}\|_2$ . Following the analysis from [24], the MSIS problem is believed to be computationally hard when  $\beta$  satisfies the following condition for a constant  $\delta \approx 1.005$ :

$$\beta \leq \min\{q, 2^{2\sqrt{\mu d \log q \log \delta}}\}$$

Assuming the second term is smaller than  $q$ , it holds that  $\log q = \Omega(\log^2 \beta)$  when other parameters are fixed. If we naively embed the elements in  $\mathbb{Z}_p$  into coefficients of a polynomial in  $R$ , it holds that  $\beta = O(\log p) = O(r \log b)$ , but with our method, it holds that  $\beta = O(\log b + \log r)$ . Hence, our encoding method roughly reduces the size  $\log q$  of the commitment modulus by a factor of  $r^2$ .

### 3.2 Randomized Encoding

The rejection sampling is yet another bottleneck in the POK protocol for the Ajtai commitment scheme. To be precise, in the POK protocol of the Ajtai commitment scheme, the response of the form  $\vec{\mathbf{t}} = \vec{\mathbf{g}} + \mathbf{c} \cdot \vec{\mathbf{m}}$  and  $\vec{\mathbf{\tau}} = \vec{\mathbf{\gamma}} + \mathbf{c} \cdot \vec{\mu}$  is transmitted to the verifier, where  $\vec{\mathbf{m}}$  is an encoded message,  $\vec{\mu}$  is a commitment randomness,  $\mathbf{c}$  is a random challenge from a verifier, and  $\vec{\mathbf{g}}, \vec{\mathbf{\gamma}}$  are masks for these values. Since the responses  $\vec{\mathbf{t}}$  and  $\vec{\mathbf{\tau}}$  contain partial information about  $\vec{\mathbf{m}}$  and  $\vec{\mu}$ , which inhibits the simulatability, the previous construction of the POK protocol used rejection sampling method [31] to ensure that they follow an independent distribution from  $\vec{\mathbf{m}}$  and  $\vec{\mu}$ .

Recently, Kim et al. [27] presented a POK protocol for BDLOP [8], which is another lattice-based commitment scheme that does not rely on the rejection sampling technique. This work is based on the observation that the protocol can be still simulatable even if the verifier obtains from the response some partial information on the commitment randomness which is sampled from a discrete Gaussian distribution. Unfortunately, this framework is not simply compatible with the Ajtai scheme where the commitment is represented as a linear combination of both the message  $\vec{\mathbf{t}}$  and randomness  $\vec{\mathbf{\tau}}$ , different from BDLOP whose POK statement depends only on the commitment randomness. If we directly apply the same approach, then a POK would leak nonnegligible amount of information about the committed message.

To overcome this limitation, we propose a randomized message encoding method where the output follows a discrete Gaussian distribution. We point out that the decoding procedure outputs the same result if inputs are congruent modulo  $X^{d/r} - b$ , i.e.,  $\text{Dcd}(\mathbf{a}) = \text{Dcd}(\mathbf{a}')$  whenever  $\mathbf{a} = \mathbf{a}' \pmod{X^{d/r} - b}$ . Since we aim to construct a commitment scheme for messages in  $\mathbb{Z}_p$ , it is allowed to incorporate a randomization procedure during the message encoding process as long as its decoding remains the same. To be precise, let  $\vec{m} \in \mathbb{Z}_p^{d/r}$  be an input message, and  $\mathbf{m}$  be the output of the randomized encoding. Since our goal is to have  $\mathbf{m} = \text{Ecd}(\vec{m}) \pmod{X^{d/r} - b}$ , we can sample it from a discrete Gaussian over a coset  $\text{Ecd}(\vec{m}) + P\mathbb{Z}^d$ , where  $P \in \mathbb{R}^{d \times d}$  is the negacyclic matrix corresponding to  $X^{d/r} - b$ , then it satisfies all the required conditions. Below, we present our randomized encoding algorithm.

- $\text{R.Ecd}(\vec{a}, \mathfrak{s}) \rightarrow \mathbf{a}$ : Given an element  $\vec{a} \in \mathbb{Z}_p^{d/r}$  and a positive real  $\mathfrak{s} > 0$ , output a ring element  $\mathbf{a} \leftarrow \mathcal{D}_{\text{Ecd}(\vec{a}) + P\mathbb{Z}^d, \mathfrak{s}P}$  where  $P \in \mathbb{Z}^{d \times d}$  is the negacyclic matrix of  $X^{d/r} - b$ .

As an abuse of notation, we often put an  $\ell d/r$ -dimensional vector  $\vec{a} = \vec{a}_0 \parallel \dots \parallel \vec{a}_{\ell-1} \in \mathbb{Z}_p^{\ell d/r}$  as an input for the above algorithms. In this case,  $\text{R.Ecd}(\vec{a})$  outputs a vector of ring element  $\vec{\mathbf{a}} = (\mathbf{a}_0, \dots, \mathbf{a}_{\ell-1}) \in R^\ell$  where  $\mathbf{a}_i = \text{R.Ecd}(\vec{a}_i; \mathfrak{s})$  for  $0 \leq i < \ell$ . We also note that we use the covariant matrix of the form  $\mathfrak{s}P$  for the sake of efficiency. We first sample from  $\mathcal{D}_{P^{-1}\text{Ecd}(\vec{a}) + \mathbb{Z}^d, \mathfrak{s}}$ , and then multiply it by  $P$ , since sampling from a spherical discrete Gaussian over a coset of the integer lattice is much more efficient than sampling from a coset of an arbitrary lattice. Finally, we remark that  $P^{-1}$  corresponds to the negacyclic matrix of the polynomial  $(X^{d/r} - b)^{-1} = -\frac{1}{b^r+1}(X^{d-d/r} + bX^{d-2d/r} + \dots + b^{r-1})$ . Thus, it holds the followings:

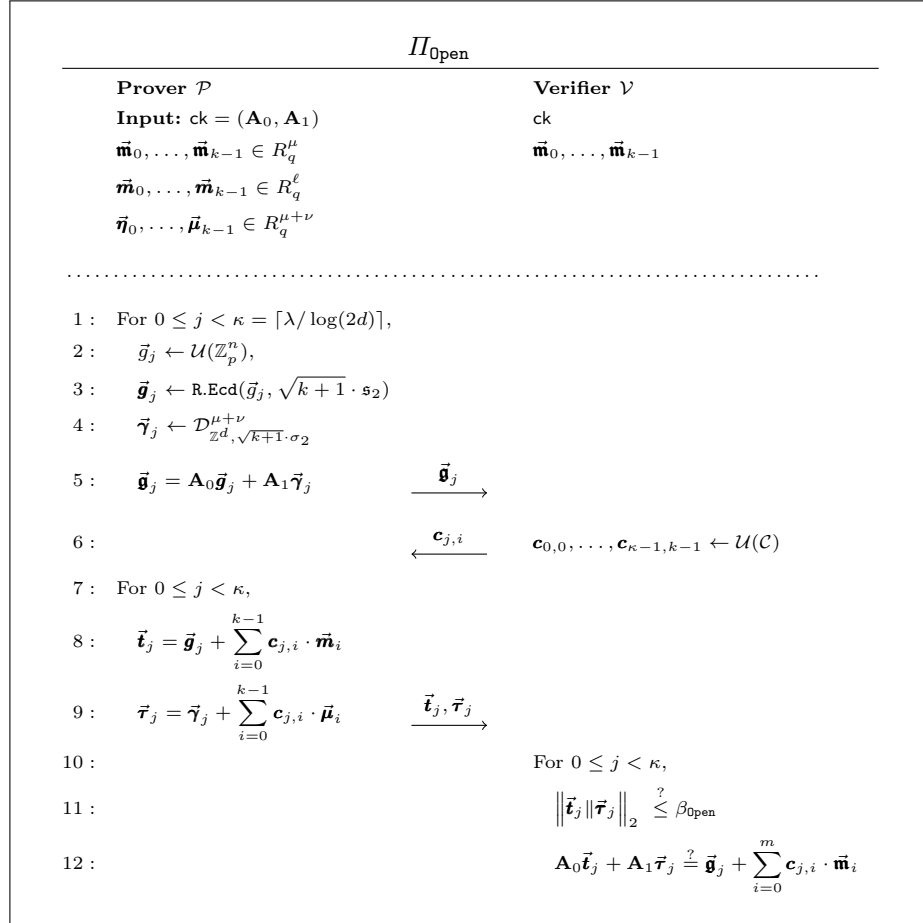
$$\|P\|_2 \leq \|X^{d/r} - b\|_1 = b + 1, \quad \|P^{-1}\|_2 \leq \|(X^{d/r} - b)^{-1}\|_1 = \frac{b^r - 1}{(b-1)(b^r+1)} \leq \frac{1}{b-1}$$

### 3.3 A New Proof of Opening Knowledge Protocol

In this subsection, we present our variant of the Ajtai commitment scheme, which incorporates a new encoding method for compact proof sizes and a randomized encoding technique for simulatability. The algorithms (**Setup**, **Com**, **Open**) are defined as follows for the message space  $\mathbb{Z}_p^n$ .

- **Setup**( $1^\lambda$ )  $\rightarrow$  ck: Given a security parameter  $\lambda$ , it generates a commitment key ck =  $(\mathbf{A}_0, \mathbf{A}_1)$  where  $n = \ell d/r$ ,  $\mathbf{A}_0 \leftarrow \mathcal{U}(R_q^{\mu \times \ell})$ , and  $\mathbf{A}_1 = [\mathbf{A}'_1 | \mathbf{I}_\mu] \in R_q^{\mu \times (\mu + \nu)}$  with  $\mathbf{A}'_1 \leftarrow \mathcal{U}(R_q^{\mu \times \nu})$ .
- **Com**(ck,  $\vec{m}$ )  $\rightarrow$   $(\vec{\mathbf{m}}, \delta)$ : Given a message  $\vec{m} \in \mathbb{Z}_p^n$ , it returns a commitment  $\vec{\mathbf{m}} = \mathbf{A}_0 \vec{m} + \mathbf{A}_1 \vec{\mu} \in R_q^\mu$  and an opening  $\delta = (\vec{\mathbf{m}}, \vec{\mu})$ , where  $\vec{\mathbf{m}} \leftarrow \text{R.Ecd}(\vec{m}; \mathfrak{s}_1)$  and  $\vec{\mu} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu + \nu}$ .
- **Open**(ck,  $\vec{\mathbf{m}}, \vec{m}, \delta$ )  $\rightarrow$  b: Given a commitment  $\vec{\mathbf{m}}$ , a message  $\vec{m}$ , and an opening  $\delta = (\vec{\mathbf{m}}, \vec{\mu})$ , it outputs 1 if it satisfies  $\|2\vec{\mathbf{m}}\|_2 \|2\vec{\mu}\|_2 < 2d\beta_{\text{open}}$ ,  $\vec{\mathbf{m}} = \mathbf{A}_0 \vec{m} + \mathbf{A}_1 \vec{\mu} \pmod{q}$ , and  $\vec{m} = \frac{p+1}{2} \cdot \text{Dcd}(2\vec{\mathbf{m}}) \pmod{p}$ ; otherwise, it outputs 0.

The commitment scheme defined above is computationally hiding if  $\text{MLWE}_{R,\nu,q,\sigma_1}$  is hard, and computationally binding if  $\text{MSIS}_{R,\mu,q,Ad\beta_{\text{open}}}$  is hard. Next, we define the POK protocol  $\Pi_{\text{open}}$  as in Fig. 1. Our POK protocol adapts the batching technique from [7], hence it can prove the knowledge of openings of  $k$  commitments simultaneously. For the challenge set  $\mathcal{C}$ , we use the set of monomials  $\{1, X, \dots, X^{2d-1}\}$ , following the technique from [10]. The completeness and soundness of the protocol are as follows.



**Fig. 1.** Batched proof of opening protocol for the Ajtai commitment scheme

**Theorem 2 (Completeness).** *The proof of opening knowledge protocol  $\Pi_{\text{open}}$  in Fig. 1 has completeness if it satisfies the following conditions for  $\varepsilon = \text{negl}(\lambda)$ :*

- $\mathfrak{s}_1, \sqrt{k+1} \cdot \mathfrak{s}_2 \geq \eta_\varepsilon(\mathbb{Z}^{d\ell})$



$$\begin{aligned}
 & - \sigma_1, \sqrt{k+1} \cdot \sigma_2 \geq \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)}) \\
 & - \beta_{\text{open}} = \sqrt{\left( (k\sigma_1 + \sqrt{k+1} \cdot \sigma_2)^2 \cdot (\mu+\nu) + (b+1)^2 \cdot (k\mathfrak{s}_1 + \sqrt{k+1} \cdot \mathfrak{s}_2)^2 \cdot \ell \right) d}
 \end{aligned}$$

*Proof.* We use Lem. 5 to attain upper bounds for  $\|\vec{\mathbf{t}}_j\|_2$  and  $\|\vec{\mathbf{r}}_j\|_2$ . Let  $P_\ell = P \otimes I_\ell \in \mathbb{R}^{d\ell \times d\ell}$ . Since  $\mathbf{g}_j \sim \mathcal{D}_{\text{Ecd}(\vec{g}_j) + P_\ell \mathbb{Z}^{d\ell}, \sqrt{k+1}\mathfrak{s}_2 P_\ell} \equiv P_\ell \cdot \mathcal{D}_{P_\ell^{-1} \text{Ecd}(\vec{g}_j) + \mathbb{Z}^{d\ell}, \sqrt{k+1}\mathfrak{s}_2}$ , the following holds with overwhelming probability, assuming  $\sqrt{k+1} \cdot \mathfrak{s}_2 \geq \eta_\varepsilon(\mathbb{Z}^{d\ell})$ :

$$\|\mathbf{g}_j\|_2 \leq \mathfrak{s}_2 \cdot \|P_\ell\|_2 \cdot \sqrt{(k+1)d\ell} \leq (b+1)\mathfrak{s}_2 \cdot \sqrt{(k+1)d\ell}$$

Similarly, we have  $\|\mathbf{m}_i\|_2 \leq (b+1)\mathfrak{s}_1 \cdot \sqrt{d\ell}$ , assuming that  $\mathfrak{s}_1 \geq \eta_\varepsilon(\mathbb{Z}^{d\ell})$ . Thus, for  $\vec{\mathbf{t}}_j = \vec{\mathbf{g}}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i$ , the following holds with overwhelming probability, given that  $\mathbf{c}_{j,i}$ 's are monomials:

$$\|\vec{\mathbf{t}}_j\|_2 \leq \|\vec{\mathbf{g}}_j\|_2 + \sum_{i=0}^{k-1} \|\vec{\mathbf{m}}_i\|_2 \leq (b+1)(k\mathfrak{s}_1 + \sqrt{k+1} \cdot \mathfrak{s}_2) \sqrt{d\ell}$$

Similarly, we obtain the following result for  $\vec{\mathbf{r}}_j = \vec{\mathbf{r}}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \vec{\boldsymbol{\mu}}_i$ , assuming that  $\sigma_1, \sqrt{k+1} \cdot \sigma_2 \geq \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$ :

$$\|\vec{\mathbf{r}}_j\|_2 \leq \|\vec{\mathbf{r}}_j\|_2 + \sum_{i=0}^{k-1} \|\vec{\boldsymbol{\mu}}_i\|_2 \leq (b+1)(k\sigma_1 + \sqrt{k+1} \cdot \sigma_2) \sqrt{d(\mu+\nu)}$$

As a result, we have  $\|\vec{\mathbf{t}}_j\|_2 \|\vec{\mathbf{r}}_j\|_2 \leq \beta_{\text{open}}$  with overwhelming probability.  $\square$

**Theorem 3 (Soundness).** *The proof of opening knowledge protocol  $\Pi_{\text{open}}$  in Fig. 1 satisfies soundness.*

*Proof.* Let  $\text{ck} \leftarrow \text{Setup}(1^\lambda)$ , and  $\mathcal{P}^*$  be a PPT adversary which works as follows:

$$c = (\vec{\mathbf{m}}_0, \dots, \vec{\mathbf{m}}_{k-1}) \leftarrow \mathcal{P}^*(\text{ck}), (\pi, b) \leftarrow (\mathcal{P}^*(\text{ck}, c), \mathcal{V}(\text{ck}, c))$$

Let  $\alpha$  be the probability that  $b = 1$ . From Lem. 5 of [7], there exists an extractor  $\mathcal{E}^{\mathcal{P}^*}$  that can extract  $(\vec{\mathbf{m}}'_i, \vec{\boldsymbol{\mu}}'_i)$  such that  $\|\vec{\mathbf{m}}'_i\|_2 \|\vec{\boldsymbol{\mu}}'_i\|_2 < 2d\beta_{\text{open}}$ , and  $\mathbf{A}_0 \vec{\mathbf{m}}'_i + \mathbf{A}_1 \vec{\boldsymbol{\mu}}'_i = 2\vec{\mathbf{m}}_i$  for  $0 \leq i < k$  in expected time  $\text{poly}(\lambda)/\alpha$ . Then, for  $\delta_i = \left(\frac{q+1}{2} \cdot \vec{\mathbf{m}}'_i, \frac{q+1}{2} \cdot \vec{\boldsymbol{\mu}}'_i\right)$  and  $\vec{m}_i = \frac{p+1}{2} \cdot \text{Dcd}(\vec{\mathbf{m}}'_i)$ , it always holds that  $\text{Open}(\text{ck}, \vec{\mathbf{m}}_i, \vec{m}_i, \delta_i) = 1$  for  $0 \leq i < k$ . Thus, in case of non-negligible  $\alpha$ , there exists an expected polynomial time extractor that outputs valid message and opening.  $\square$

We now prove the simulatability of our POK protocol. The main idea of our proof is to utilize the property of discrete Gaussian distribution similar to the technique from [27]. We first provide a convolution lemma, as a generalization of Lem. 7.

**Lemma 10.** *Let  $k > 1$  be an integer,  $\Sigma \in \mathbb{R}^{n \times n}$  be positive-definitive matrices, and  $\Lambda \subseteq \mathbb{Z}^n$  be an  $n$ -dimensional lattice. If  $\sqrt{\Sigma/2} \geq \eta_\varepsilon(\Lambda)$  for some  $0 < \varepsilon < 1/2$ , then for any  $\vec{u}_0, \dots, \vec{u}_{k-1} \in \mathbb{R}^n$ , the distribution of  $\sum_{i=0}^{k-1} \vec{x}_i$ , where  $\vec{x}_i \leftarrow \mathcal{D}_{\vec{u}_i + \Lambda, \sqrt{\Sigma}}$ , is within a statistical distance of  $8(k-1)\varepsilon$  from  $\mathcal{D}_{\vec{u} + \Lambda, \sqrt{k\Sigma}}$ , where  $\vec{u} = \sum_{i=0}^{k-1} \vec{u}_i$ .*

*Proof.* We first define  $\vec{y}_j := \sum_{i=0}^{j-1} \vec{x}_i$  and  $\vec{v}_j := \sum_{i=0}^{j-1} \vec{u}_i$ . We note that the sample space of  $\vec{y}_j$  is identical to  $\vec{v}_j + \Lambda$ . By induction, it suffices to show that the distribution of  $\vec{y}_{j+1}$  is within a statistical distance of  $8(j+1)\varepsilon$  from  $\mathcal{D}_{\vec{v}_{j+1} + \Lambda, \sqrt{(j+1)\Sigma}}$  whenever  $\vec{y}_j$  is within a statistical distance of  $8j\varepsilon$  from  $\mathcal{D}_{\vec{v}_j + \Lambda, \sqrt{j\Sigma}}$ .

From Lem. 7, the distribution of  $\vec{y}_j + \vec{x}_{j+1}$  is within statistical distance of  $8(j+1)\varepsilon$  from  $\mathcal{D}_{\vec{v}_{j+1} + \Lambda, \sqrt{(j+1)\Sigma}}$  if  $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$  and  $\sqrt{\frac{j+1}{j+2}\Sigma} \geq \eta_\varepsilon(\Lambda)$ , which are true from the assumption  $\sqrt{\Sigma/2} \geq \eta_\varepsilon(\Lambda)$ .  $\square$

Using the above lemma, one can freely add or decompose discrete Gaussian distributions defined over cosets of the same lattice  $\Lambda$ , under proper conditions. Next, we provide a useful lemma to be used in our simulatability proof.

**Lemma 11.** *Let  $k > 0$  be an integer,  $\mathfrak{S}, \Sigma_0, \dots, \Sigma_{k-1} \in \mathbb{R}^{n \times n}$  be positive-definitive matrices,  $C_0, \dots, C_{k-1} \in \mathbb{Z}^{n \times n}$  be invertible matrices, and  $\Lambda \subseteq \mathbb{Z}^n$  be an  $n$ -dimensional lattice. If  $\sqrt{(\mathfrak{S}^{-1} + \sum_{i=0}^{k-1} C_i^\top \Sigma_i^{-1} C_i)^{-1}} \geq \eta_\varepsilon(\Lambda)$ , then for any  $\vec{u} \in \mathbb{R}^n$ , the following two distributions are within a statistical distance of  $2\varepsilon$ .*

$$\left\{ (\vec{z}_0, \dots, \vec{z}_{k-1}) \mid \vec{x} \leftarrow \mathcal{D}_{\vec{u} + \Lambda, \sqrt{\mathfrak{S}}}, \vec{y}_i \leftarrow \mathcal{D}_{-C_i \vec{u} + \Lambda, \sqrt{\Sigma_i}}, \vec{z}_i = C_i \vec{x} + \vec{y}_i \right\}$$

$$\left\{ (\vec{z}'_0, \dots, \vec{z}'_{k-1}) \mid \vec{x}' \leftarrow \mathcal{D}_{\Lambda, \sqrt{\mathfrak{S}}}, \vec{y}'_i \leftarrow \mathcal{D}_{\Lambda, \sqrt{\Sigma_i}}, \vec{z}'_i = C_i \vec{x}' + \vec{y}'_i \right\}$$

*Proof.* We first note that the sample spaces of the above two distributions are identical to  $\Lambda^k$  since  $C_i$ 's are integer matrices and  $\Lambda \in \mathbb{Z}^n$ . Thus, it is enough to show that

$$\Pr[\vec{z} = \vec{w}] \in [1 - 2\varepsilon, 1 + 2\varepsilon] \cdot \Pr[\vec{z}' = \vec{w}] \quad (1)$$

for all  $\vec{w} = \vec{w}_0 \parallel \dots \parallel \vec{w}_{k-1} \in \Lambda^k$  where  $\vec{z} = \vec{z}_0 \parallel \dots \parallel \vec{z}_{k-1}$ , and  $\vec{z}' = \vec{z}'_0 \parallel \dots \parallel \vec{z}'_{k-1}$ . The first term of Eq. (1) can be computed as follows:

$$\begin{aligned} \Pr[\vec{z} = \vec{w}] &= \sum_{\vec{v} \in \vec{u} + \Lambda} \Pr[\vec{x} = \vec{v}] \cdot \prod_{i=0}^{k-1} \Pr[\vec{y}_i = \vec{w}_i - C_i \cdot \vec{v}] \\ &= \sum_{\vec{v} \in \vec{u} + \Lambda} \rho_{\sqrt{\mathfrak{S}}}(\vec{v}) \cdot \prod_{i=0}^{k-1} \rho_{C_i^{-1} \sqrt{\Sigma_i}}(\vec{v} - C_i^{-1} \cdot \vec{w}_i) \end{aligned} \quad (2)$$

Let  $\mathfrak{S}_i^{-1} = \mathfrak{S}^{-1} + \sum_{j=0}^{i-1} C_j^\top \Sigma_j^{-1} C_j$  for  $0 \leq i < k$ . Then, we have the following from Lem. 6:

$$\rho_{\sqrt{\mathfrak{S}_i}}(\vec{v} - \vec{a}_i) \cdot \rho_{C_i^{-1} \sqrt{\Sigma_i}}(\vec{v} - C_i^{-1} \vec{w}_i) = \rho_{\sqrt{\mathfrak{S}_{i+1}}}(\vec{v} - \vec{a}_{i+1}) \cdot \rho_{\sqrt{\mathfrak{S}_i + C_i^{-1} \Sigma_i C_i^{-\top}}}(\vec{b}_{i+1})$$

where  $\vec{a}_0 = 0$ ,  $\vec{b}_0 = C_0^{-1} \cdot \vec{w}_0$  and,  $\vec{a}_{i+1} = (\mathfrak{S}_i + C_i^{-1} \Sigma_i C_i^{-\top})(\mathfrak{S}_i^{-1} \vec{a}_i + C_i^\top \Sigma_i^{-1} \vec{w}_i)$ , and  $\vec{b}_{i+1} = \vec{a}_i - \vec{b}_i$ . Note that both  $\vec{a}_i$  and  $\vec{b}_i$  are independent of the value of  $\vec{v}$ . Then, Eq. (2) can be rewritten as follows:

$$\Pr[\vec{z} = \vec{w}] = \prod_{i=0}^{k-1} \rho_{\sqrt{\mathfrak{S}_i + C_i^{-1} \Sigma_i C_i^{-\top}}}(\vec{b}_{i+1}) \cdot \left( \sum_{\vec{v} \in \vec{u} + \Lambda} \rho_{\sqrt{\mathfrak{S}_k}}(\vec{v} - \vec{a}_k) \right)$$

Similarly, the second term of Eq. (1) can be computed as follows:

$$\Pr[\vec{z}^\top = \vec{w}] = \prod_{i=0}^{k-1} \rho_{\sqrt{\mathfrak{S}_i + C_i^{-1} \Sigma_i C_i^{-\top}}}(\vec{b}_{i+1}) \cdot \left( \sum_{\vec{v} \in \Lambda} \rho_{\sqrt{\mathfrak{S}_k}}(\vec{v} - \vec{a}_k) \right)$$

By the given condition, we know that  $\sqrt{\mathfrak{S}_k} \geq \eta_\varepsilon(\Lambda)$ . Thus, we obtain the following from Lem. 3.

$$\sum_{\vec{v} \in \vec{u} + \Lambda} \rho_{\sqrt{\mathfrak{S}_k}}(\vec{v} - \vec{a}_k) \in [1 - 2\varepsilon, 1 + 2\varepsilon] \cdot \sum_{\vec{v} \in \Lambda} \rho_{\sqrt{\mathfrak{S}_k}}(\vec{v} - \vec{a}_k)$$

Therefore, we prove that  $\Pr[\vec{z} = \vec{w}] \in [1 - 2\varepsilon, 1 + 2\varepsilon] \cdot \Pr[\vec{z}^\top = \vec{w}]$   $\square$

The above lemma essentially says that if we suitably choose the masking vector  $\vec{y}_i$  in the linear combination  $\vec{z}_i = C_i \cdot \vec{x} + \vec{y}_i$ , we can efficiently simulate the resulting distribution without knowing each  $\vec{u}_i$ . We sketch our proof briefly, incorporating the above lemmas. To achieve simulatability without rejection sampling, we sample mask vectors  $\vec{g}_j$  for the responses  $\vec{t}_j$  as a randomized encoding of  $\vec{g}_j$ , a uniformly sampled vector from  $\mathbb{Z}_p^n$ . Then, it can be decomposed into multiple discrete Gaussian distribution over cosets  $\text{Ecd}(-C_{i,j} \vec{m}_i) + P\mathbb{Z}^{d\ell}$  for  $0 \leq i \leq m$  and  $\text{Ecd}(\vec{g}_j + \sum_{i=0}^{k-1} C_{i,j} \vec{m}_i) + P\mathbb{Z}^{d\ell}$ , using Lem. 10. We note that the distribution of  $\vec{g}_j + \sum_{i=0}^{k-1} C_{i,j} \vec{m}_i$  is identical to  $\vec{g}_j$  since  $\vec{g}_j$  follows the uniform random distribution over  $\mathbb{Z}_p^n$ . Finally, we use Lem. 11 to simulate the distribution of the responses  $\vec{t}_j$ . Below, we provide a proof of the simulatability of our protocol.

**Theorem 4 (Simulatability).** *The proof of opening knowledge protocol  $\Pi_{\text{open}}$  in Fig. 1 satisfies simulatability if it satisfies the followings for  $\varepsilon = \text{negl}(\lambda)$ :*

- $\frac{s_2}{\sqrt{2}}, s_{\text{open}} \geq \frac{1}{b-1} \cdot \eta_\varepsilon((P \otimes I_\ell) \mathbb{Z}^{d\ell})$ , where  $s_{\text{open}} = (s_1^{-2} + \kappa s_2^{-2})^{-1/2}$
- $\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_{\text{open}}}{\sqrt{2}} \geq \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$ , where  $\sigma_{\text{open}} = \frac{1}{\sqrt{2}} \cdot (\sigma_1^{-2} + \kappa \sigma_2^{-2})^{-1/2}$
- The advantage of  $\text{MLWE}_{R,\nu,q,\sigma_{\text{open}}}$  is  $\text{negl}(\lambda)$

$$\mathcal{S}_{\text{open}}(\text{ck})$$


---

- 1 :  $\vec{\mathbf{m}}_i \leftarrow \mathcal{U}(R_q^\mu), \mathbf{c}_{i,0}, \dots, \mathbf{c}_{i,\kappa-1} \leftarrow \mathcal{U}(\mathcal{C})$  for  $0 \leq i < k$
- 2 :  $\vec{\mathbf{m}}_i \leftarrow \text{R.Ecd}(\vec{0}, \mathbf{s}_1), \vec{\boldsymbol{\mu}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}$  for  $0 \leq i < k$
- 3 :  $\vec{g}_j \leftarrow \mathcal{U}(\mathbb{Z}_p^n), \vec{g}_j \leftarrow \text{R.Ecd}(\vec{g}_j, \sqrt{k+1} \cdot \mathbf{s}_2), \vec{\boldsymbol{\gamma}}_j \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{k+1} \cdot \sigma_2}$  for  $0 \leq j < \kappa$
- 4 :  $\vec{\mathbf{t}}_j = \vec{g}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i, \vec{\boldsymbol{\tau}}_j = \vec{\boldsymbol{\gamma}}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\boldsymbol{\mu}}_i$  for  $0 \leq j < \kappa$
- 5 :  $\vec{\mathbf{g}}_j = \mathbf{A}_0 \vec{\mathbf{t}}_j + \mathbf{A}_1 \vec{\boldsymbol{\tau}}_j - \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i \pmod{q}$  for  $0 \leq j < \kappa$
- 6 : Output  $(\vec{\mathbf{m}}, \vec{\mathbf{g}}, \vec{\mathbf{c}}, \vec{\mathbf{t}}, \vec{\boldsymbol{\tau}})$

Fig. 2. Simulator for  $\Pi_{\text{open}}$ 

*Proof.* Let  $\vec{m} = \vec{m}_0 \| \dots \| \vec{m}_{k-1}$  be messages chosen by a PPT adversary  $\mathcal{A}(\text{ck})$ , where  $\text{ck} \leftarrow \text{Setup}(1^\lambda)$ . We prove that the algorithm  $\mathcal{S}_{\text{open}}$  in Fig. 2 is an efficient simulator for the protocol  $\Pi_{\text{open}}$  using hybrid arguments. We first define a distribution  $\mathcal{H}_0(\text{ck}, \vec{m})$  as follows, where  $\vec{\mathbf{m}} = \vec{\mathbf{m}}_0 \| \dots \| \vec{\mathbf{m}}_{k-1}, \vec{\mathbf{g}} = \vec{\mathbf{g}}_0 \| \dots \| \vec{\mathbf{g}}_{\kappa-1}, \vec{\mathbf{c}} = (\mathbf{c}_{0,0}, \dots, \mathbf{c}_{k-1, \kappa-1}), \vec{\mathbf{t}} = \vec{\mathbf{t}}_0 \| \dots \| \vec{\mathbf{t}}_{\kappa-1}$ , and  $\vec{\boldsymbol{\tau}} = \vec{\boldsymbol{\tau}}_0 \| \dots \| \vec{\boldsymbol{\tau}}_{\kappa-1}$ .

$$\left\{ (\vec{\mathbf{m}}, \vec{\mathbf{g}}, \vec{\mathbf{c}}, \vec{\mathbf{t}}, \vec{\boldsymbol{\tau}}) \left| \begin{array}{l} \vec{\mathbf{m}}_i \leftarrow \text{R.Ecd}(\vec{m}_i, \mathbf{s}_1), \vec{\boldsymbol{\mu}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{m}}_i = \mathbf{A}_0 \vec{\mathbf{m}}_i + \mathbf{A}_1 \vec{\boldsymbol{\mu}}_i \text{ for } 0 \leq i < k \\ \mathbf{c}_{j,0}, \dots, \mathbf{c}_{j,k-1} \leftarrow \mathcal{U}(\mathcal{C}), \vec{g}_j \leftarrow \mathcal{U}(\mathbb{Z}_p^n) \\ \vec{g}_j \leftarrow \text{R.Ecd}(\vec{g}_j, \sqrt{2} \mathbf{s}_2), \vec{\boldsymbol{\gamma}}_j \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{2} \sigma_2}^{\mu+\nu} \\ \vec{\mathbf{t}}_j = \vec{g}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i, \vec{\boldsymbol{\tau}}_j = \vec{\boldsymbol{\gamma}}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\boldsymbol{\mu}}_i \\ \vec{\mathbf{g}}_j = \mathbf{A}_0 \vec{\mathbf{t}}_j + \mathbf{A}_1 \vec{\boldsymbol{\tau}}_j - \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i \text{ for } 0 \leq j < \kappa \end{array} \right. \right\}$$

We note that the above distribution corresponds to the distribution of transcripts from  $\Pi_{\text{open}}$ .

**Claim 1:**  $\mathcal{H}_0(\text{ck}, \vec{m})$  and  $\mathcal{H}_1(\text{ck}, \vec{m})$  are statistically indistinguishable, where  $\mathcal{H}_1(\text{ck}, \vec{m})$  is defined as follows:

$$\left\{ (\vec{\mathbf{m}}, \vec{\mathbf{g}}, \vec{\mathbf{c}}, \vec{\mathbf{t}}, \vec{\boldsymbol{\tau}}) \left| \begin{array}{l} \vec{\mathbf{m}}_i \leftarrow \text{R.Ecd}(\vec{m}_i, \mathbf{s}_1), \vec{\boldsymbol{\mu}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{m}}_i = \mathbf{A}_0 \vec{\mathbf{m}}_i + \mathbf{A}_1 \vec{\boldsymbol{\mu}}_i \text{ for } 0 \leq i < k \\ \mathbf{c}_{j,0}, \dots, \mathbf{c}_{j,k-1} \leftarrow \mathcal{U}(\mathcal{C}), \vec{g}_j \leftarrow \mathcal{U}(\mathbb{Z}_p^n) \\ \vec{g}_{j,i} \leftarrow \text{R.Ecd}(-C_{j,i} \vec{m}_i, \mathbf{s}_2) \text{ for } 0 \leq i < k, \vec{g}_{j,k} \leftarrow \text{R.Ecd}(\vec{g}_j + \sum_{i=0}^{k-1} C_{j,i} \vec{m}_i, \mathbf{s}_2) \\ \vec{\boldsymbol{\tau}}_{j,0}, \dots, \vec{\boldsymbol{\tau}}_{j,k} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_2}^{\mu+\nu} \\ \vec{\mathbf{t}}_j = \sum_{i=0}^{k-1} (\vec{g}_{j,i} + \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i) + \vec{g}_{j,k}, \vec{\boldsymbol{\tau}}_j = \sum_{i=0}^{k-1} (\vec{\boldsymbol{\tau}}_{j,i} + \mathbf{c}_{j,i} \cdot \vec{\boldsymbol{\mu}}_i) + \vec{\boldsymbol{\tau}}_{j,k} \\ \vec{\mathbf{g}}_j = \mathbf{A}_0 \vec{\mathbf{t}}_j + \mathbf{A}_1 \vec{\boldsymbol{\tau}}_j - \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mathbf{m}}_i \text{ for } 0 \leq j < \kappa \end{array} \right. \right\}$$

The difference between  $\mathcal{H}_0(\text{ck}, \vec{m})$  and  $\mathcal{H}_1(\text{ck}, \vec{m})$  lies in how  $\vec{g}_j$  and  $\vec{\gamma}_j$  are sampled, which we underlined in the above. Let  $P_\ell = P \otimes I_\ell \in \mathbb{R}^{d\ell \times d\ell}$ . In  $\mathcal{H}_0(\text{ck}, \vec{m})$ ,  $\vec{g}_j \sim \mathcal{D}_{\text{Ecd}(\vec{g}_j) + P_\ell \mathbb{Z}^{d\ell}, \sqrt{k+1} \mathfrak{s}_2 P_\ell}$  and  $\vec{\gamma}_j \sim \mathcal{D}_{\mathbb{Z}^d, \sqrt{k+1} \sigma_2}^{\mu+\nu}$ . Meanwhile, in  $\mathcal{H}_1(\text{ck}, \vec{m})$ ,

$$\begin{aligned} \vec{g}_j &= \sum_{i=0}^k \vec{g}_{j,i} \sim \sum_{i=0}^{k-1} \mathcal{D}_{\text{Ecd}(-C_{j,i} \vec{m}_i) + P_\ell \mathbb{Z}^{d\ell}, \mathfrak{s}_2 P_\ell} + \mathcal{D}_{\text{Ecd}(\vec{g}_j + \sum_{i=0}^{k-1} C_{j,i} \vec{m}_i) + P_\ell \mathbb{Z}^{d\ell}, \mathfrak{s}_2 P_\ell} \\ \vec{\gamma}_j &= \sum_{i=0}^k \vec{\gamma}_{j,i} \sim \sum_{i=0}^k \mathcal{D}_{\mathbb{Z}^d, \sigma_2}^{\mu+\nu} \end{aligned}$$

where  $C_{j,i}$  is the negacyclic matrix of  $\mathbf{c}_{j,i}$ . From Lem. 10, the distribution of  $\vec{g}_j$  in  $\mathcal{H}_1(\text{ck}, \vec{m})$  is within a statistical distance of  $8k\varepsilon$  from  $\mathcal{D}_{\text{Ecd}(\vec{g}_j) + P_\ell \mathbb{Z}^{d\ell}, \sqrt{k+1} \mathfrak{s}_2 P_\ell}$  if  $\frac{\mathfrak{s}_2}{\sqrt{2}} P_\ell \geq \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$ , which is implied by  $\frac{\mathfrak{s}_2}{\sqrt{2}} \geq \frac{1}{b-1} \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$  from Lem. 2 and  $\|P^{-1}\|_2 \leq \frac{1}{b-1}$ . In a similar manner, from Lem. 10,  $\sum_{i=0}^k \mathcal{D}_{\mathbb{Z}^d, \sigma_2}^{\mu+\nu}$  is within a statistical distance of  $8k\varepsilon$  from  $\mathcal{D}_{\mathbb{Z}^d, \sqrt{k+1} \sigma_2}^{\mu+\nu}$  if  $\frac{\sigma_2}{\sqrt{2}} \geq \eta_\varepsilon(\mathbb{Z}^d)$ . Therefore,  $\mathcal{H}_0(\text{ck}, \vec{m})$  and  $\mathcal{H}_1(\text{ck}, \vec{m})$  are within a statistical distance  $16k\varepsilon$ , which is  $\text{negl}(\lambda)$ .

**Claim 2:**  $\mathcal{H}_1(\text{ck}, \vec{m})$  and  $\mathcal{H}_2(\text{ck}, \vec{m})$  are computationally indistinguishable, where  $\mathcal{H}_2(\text{ck}, \vec{m})$  is defined as follows:

$$\left\{ (\vec{\mathbf{m}}, \vec{\mathbf{g}}, \vec{\mathbf{c}}, \vec{\mathbf{t}}, \vec{\mathbf{\tau}}) \left| \begin{array}{l} \vec{\mathbf{m}}_i \leftarrow \text{R.Ecd}(\vec{m}_i, \mathfrak{s}_1), \vec{\mu}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{m}}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i < k \\ \text{The rest is the same as } \mathcal{H}_1(\text{ck}, \vec{m}) \end{array} \right. \right\}$$

The difference between  $\mathcal{H}_1(\text{ck}, \vec{m})$  and  $\mathcal{H}_2(\text{ck}, \vec{m})$  lies in how  $\vec{\mathbf{m}}_i$ 's are sampled. Then, distinguishing  $\mathcal{H}_1(\text{ck}, \vec{m})$  and  $\mathcal{H}_2(\text{ck}, \vec{m})$  is at least hard as

$\text{HintMLWE}_{R, \nu, q, \sigma_1, \sigma_2, \dots, \sigma_2}^{\mathbf{c}_0, \dots, \mathbf{c}_{\kappa-1}}$  since  $(\vec{\mathbf{m}}_i, \vec{\gamma}_{0,i} + \mathbf{c}_{0,i} \cdot \vec{\mu}_i, \dots, \vec{\gamma}_{\kappa-1,i} + \mathbf{c}_{\kappa-1,i} \cdot \vec{\mu}_i)$  can be considered as a  $\text{HintMLWE}$  instance. By Thm. 1, there is an efficient reduction from  $\text{MLWE}_{R, \nu, q, \sigma_{\text{open}}}$  to  $\text{HintMLWE}_{R, \nu, q, \sigma_1, \sigma_2, \dots, \sigma_2}^{\mathbf{c}_0, \dots, \mathbf{c}_{\kappa-1}}$  if  $\sigma_{\text{open}} \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$ . Therefore,  $\mathcal{H}_1(\text{ck}, \vec{m})$  and  $\mathcal{H}_2(\text{ck}, \vec{m})$  are computationally indistinguishable due to the hardness of  $\text{MLWE}_{R, \nu, q, \sigma_{\text{open}}}$ .

**Claim 3:**  $\mathcal{H}_2(\text{ck}, \vec{m})$  and  $\mathcal{H}_3(\text{ck})$  are statistically indistinguishable,  $\mathcal{H}_3(\text{ck})$  is defined as follows:

$$\left\{ (\vec{\mathbf{m}}, \vec{\mathbf{g}}, \vec{\mathbf{c}}, \vec{\mathbf{t}}, \vec{\mathbf{\tau}}) \left| \begin{array}{l} \vec{\mathbf{m}}_i \leftarrow \text{R.Ecd}(\vec{0}, \mathfrak{s}_1), \vec{\mu}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{m}}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i < k \\ \mathbf{c}_{j,0}, \dots, \mathbf{c}_{j,k-1} \leftarrow \mathcal{U}(\mathcal{C}), \vec{g}_j \leftarrow \mathcal{U}(\mathbb{Z}_p^n) \\ \vec{g}_{j,i} \leftarrow \text{R.Ecd}(\vec{0}, \mathfrak{s}_2) \text{ for } 0 \leq i < k, \vec{g}_{j,k} \leftarrow \text{R.Ecd}(\vec{g}_j, \mathfrak{s}_2) \\ \text{The rest is the same as } \mathcal{H}_2(\text{ck}, \vec{m}) \end{array} \right. \right\}$$

The difference between  $\mathcal{H}_2(\text{ck}, \vec{m})$  and  $\mathcal{H}_3(\text{ck})$  lies in how  $\vec{\mathbf{m}}_i$ ,  $\vec{g}_{j,i}$ , and  $\vec{g}_{j,i}$  are sampled. From Lem. 11, the distribution of  $(\vec{g}_{0,i} + \mathbf{c}_{0,i} \cdot \vec{\mathbf{m}}_i, \dots, \vec{g}_{\kappa-1,i} + \mathbf{c}_{\kappa-1,i} \cdot \vec{\mathbf{m}}_i)$  in  $\mathcal{H}_2(\text{ck}, \vec{m})$  and  $\mathcal{H}_3(\text{ck}, \vec{m})$  are within statistical distance  $2\varepsilon$ , if the following condition holds.

$$\sqrt{\left( \mathfrak{s}_1^{-2} (P_\ell P_\ell^\top)^{-1} + \mathfrak{s}_2^{-2} \cdot \sum_{j=0}^{\kappa-1} C_j^\top (P_\ell P_\ell^\top)^{-1} C_j \right)^{-1}} \geq \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$$

From Lem. 2, the above condition is implied by  $s_{\text{open}} \geq \frac{1}{b-1} \cdot \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$  since  $\|C_j\|_2 = 1$  and  $\|P^{-1}\|_2 \leq \frac{1}{b-1}$ . For the distribution of  $\vec{g}_{j,k}$ , it is a randomized encoding of  $\vec{g}_j + \sum_{i=0}^{k-1} C_{j,i} \vec{m}_i$  in  $\mathcal{H}_2(\text{ck}, \vec{m})$ , while it corresponds to a randomized encoding of  $\vec{g}_j$  in  $\mathcal{H}_3(\text{ck})$ . We note that the distributions of  $\vec{g}_j + \sum_{i=0}^{k-1} C_{j,i} \vec{m}_i$  and  $\vec{g}_j$  are identical since  $\vec{g}_j$  is uniformly sampled from  $\mathbb{Z}_p^n$ . Thus, we can regard  $\vec{g}_{j,1}$  in  $\mathcal{H}_2$  and  $\mathcal{H}_3$  as following an identical distribution. Therefore,  $\mathcal{H}_2(\text{ck}, \vec{m})$  and  $\mathcal{H}_3(\text{ck})$  are within a statistical distance  $2\varepsilon$ , which is  $\text{negl}(\lambda)$ .

**Claim 4:**  $\mathcal{H}_3(\text{ck})$  and  $\mathcal{H}_4(\text{ck})$  are statistically indistinguishable, where  $\mathcal{H}_4(\text{ck})$  is defined as follows.

$$\left( \vec{m}, \vec{g}, \vec{c}, \vec{t}, \vec{r} \right) \left\{ \begin{array}{l} \vec{m}_i \leftarrow \text{R.Ecd}(\vec{0}, s_1), \vec{\mu}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{m}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i < k \\ \mathbf{c}_{j,0}, \dots, \mathbf{c}_{j,k-1} \leftarrow \mathcal{U}(\mathcal{C}), \vec{g}_j \leftarrow \mathcal{U}(\mathbb{Z}_p^n) \\ \vec{g}_j \leftarrow \text{R.Ecd}(\vec{g}_j, \sqrt{2}s_2), \vec{r}_j \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{2}\sigma_2}^{\mu+\nu} \\ \vec{t}_j = \vec{g}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{m}_i, \vec{r}_j = \vec{r}_j + \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{\mu}_i \\ \vec{g}_j = \mathbf{A}_0 \vec{t}_j + \mathbf{A}_1 \vec{r}_j - \sum_{i=0}^{k-1} \mathbf{c}_{j,i} \cdot \vec{m}_i \text{ for } 0 \leq j < \kappa \end{array} \right.$$

We note that the above distribution is identical to the distribution of simulated transcript from  $\mathcal{S}_{\text{open}}(\text{ck})$ . The difference between  $\mathcal{H}_3(\text{ck})$  and  $\mathcal{H}_4(\text{ck})$  lies in how  $\vec{g}_j$  and  $\vec{r}_j$  are sampled, similar to Claim 1. Thus, following the proof of Claim 1, they are within a statistical distance of  $16k\varepsilon$ , which is  $\text{negl}(\lambda)$ .

Therefore, we can conclude that  $\mathcal{H}_0(\text{ck}, \vec{m})$  and  $\mathcal{H}_4(\text{ck})$  are computationally indistinguishable under the given conditions. Since  $\mathcal{H}_0(\text{ck}, \vec{m})$  corresponds to the distribution of real transcripts from  $\Pi_{\text{open}}$ , and  $\mathcal{H}_4(\text{ck})$  corresponds to the distribution of simulated transcripts from  $\mathcal{S}_{\text{open}}$ , there is no PPT adversary  $\mathcal{A}$  that can distinguish between the two distributions with non-negligible probability.  $\square$

## 4 Lattice-based Polynomial Commitment Scheme

In this section, we present our polynomial commitment scheme based on the modified Ajtai commitment scheme and its POK protocol described in the previous section. We basically follow the construction in [13, 14], which builds a polynomial commitment scheme based on the Pedersen commitment scheme. Their main strategy in polynomial evaluation is to rearrange the coefficients of a polynomial to reduce communication cost while preserving secrecy. To be precise, let  $h(X) = \sum_{i=0}^{N-1} h_i X^i \in \mathbb{Z}_p[X]$ , and  $N = nm$ . If we define  $\vec{h}_i = (h_{ni}, h_{ni+1}, \dots, h_{n(i+1)-i})$  for  $0 \leq i < m$ , then  $h(x) = \left\langle \sum_{i=0}^{m-1} x^{ni} \cdot \vec{h}_i, \vec{x} \right\rangle$  for  $\vec{x} = (1, x, \dots, x^{n-1})$  for all  $x \in \mathbb{Z}_p$ . Hence, it suffices to return  $\sum_{i=0}^{m-1} x^{ni} \cdot \vec{h}_i$  as an evaluation proof, which is of size  $O(n)$ . However, it still contains information about each coefficient, which inhibits the evaluation hiding property. To resolve this issue, we pick random blinders  $b_1, \dots, b_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p)$ , and define  $h_m =$

$(b_1, \dots, b_{n-1}, 0)$ , and  $h_{m+1} = (0, -b_1, \dots, -b_{n-1})$ . Then,  $\langle x \cdot \vec{h}_m + \vec{h}_{m+1}, \vec{x} \rangle = 0$ , so it follows the distribution of random vectors whose inner product with  $\vec{x}$  is 0. Thus, sending  $\sum_{i=0}^{m-1} x^{ni} \cdot \vec{h}_i + x \cdot \vec{h}_m + \vec{h}_{m+1}$  achieves the desired goal of the evaluation proof, revealing nothing about the coefficients but the evaluation result  $h(x)$ .

However, directly adapting their technique to the Ajtai commitment scheme is not straightforward, as it requires converting all the proof techniques based on the properties of  $\mathbb{Z}_p$  into the setting of  $R$ . First, there is an issue with the extractability of the polynomial commitment scheme. In [13, 14], extractability is directly attained from evaluation proofs at  $N$  different points by computing the inverse of the Vandermonde matrix over  $\mathbb{Z}_p$ . However, in the Ajtai commitment scheme, such an extraction technique is not allowed since encoded messages and randomness are elements of  $R$ . Hence, we additionally attach the proof of opening knowledge to provide extractability. Second, the aforementioned blinding technique is also not converted straightforwardly due to the difference between  $\mathbb{Z}_p$  and  $R$ . To resolve this, we extend the randomized encoding in the previous method so that blinders sampled from proper discrete Gaussian distributions effectively hide information of coefficients. In the rest of this section, we present more details on how we address these issues arising from the difference between  $\mathbb{Z}_p$  and  $R$ .

#### 4.1 Our Polynomial Commitment

Our polynomial commitment scheme is defined as follows.

- **PC.Setup**( $1^\lambda, N$ )  $\rightarrow$  **ck**: Given a security parameter  $\lambda$  and a polynomial degree upper bound  $N = mn$ , it generates a commitment key  $\text{ck} = (\mathbf{A}_0, \mathbf{A}_1)$  where  $n = d\ell/r$ ,  $\mathbf{A}_0 \leftarrow \mathcal{U}(R_q^{\mu \times \ell})$ , and  $\mathbf{A}_1 = [\mathbf{A}'_1 | \mathbf{I}_\mu] \in R_q^{\mu \times (\mu + \nu)}$  with  $\mathbf{A}'_1 \leftarrow \mathcal{U}(R_q^{\mu \times \nu})$ .
- **PC.Com**( $\text{ck}, h(X)$ )  $\rightarrow$   $(\vec{\mathbf{h}}, \delta)$ : Given a polynomial  $h(X) = \sum_{i=0}^{N-1} h_i X^i \in \mathbb{Z}_p[X]$ , it generates a commitment  $\vec{\mathbf{h}}$  and an opening  $\delta = (\vec{\mathbf{h}}, \vec{\boldsymbol{\eta}})$  as follows, where  $\vec{\mathbf{h}} = \vec{\mathbf{h}}_0 \| \dots \| \vec{\mathbf{h}}_{m+1}$ ,  $\vec{\mathbf{h}} = \vec{\mathbf{h}}_0 \| \dots \| \vec{\mathbf{h}}_{m+1}$ , and  $\vec{\boldsymbol{\eta}} = \vec{\boldsymbol{\eta}}_0 \| \dots \| \vec{\boldsymbol{\eta}}_{m+1}$ :
  1. Define  $\vec{h}_i = (h_{ni}, \dots, h_{(n+1)i-1})$  and  $0 \leq i < m$ .
  2. Sample  $b_1, \dots, b_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p)$ , define  $\vec{h}_m = (b_1, \dots, b_{n-1}, 0)$  and  $\vec{h}_{m+1} = (0, -b_1, \dots, -b_{n-1})$ .
  3.  $\vec{\mathbf{h}}_i \leftarrow \text{R.Ecd}(\vec{h}_i; \mathfrak{s}_1)$ ,  $\vec{\boldsymbol{\eta}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu + \nu}$ , and  $\vec{\mathbf{h}}_i = \mathbf{A}_0 \vec{\mathbf{h}}_i + \mathbf{A}_1 \vec{\boldsymbol{\eta}}_i \pmod{q}$  for  $0 \leq i \leq m$ .
  4.  $\vec{\mathbf{h}}_{m+1} \leftarrow \text{R.Ecd}(\vec{h}_{m+1}, \sqrt{m+2} \cdot \mathfrak{s}_3)$ ,  $\vec{\boldsymbol{\eta}}_{m+1} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}^{\mu + \nu}$ , and  $\vec{\mathbf{h}}_{m+1} = \mathbf{A}_0 \vec{\mathbf{h}}_{m+1} + \mathbf{A}_1 \vec{\boldsymbol{\eta}}_{m+1} \pmod{q}$ .
- **PC.Open**( $\text{ck}, \vec{\mathbf{h}}, h(X), \delta$ )  $\rightarrow$   $b$ : Given a commitment  $\vec{\mathbf{h}}$ , an opening  $\delta = (\vec{\mathbf{h}}, \vec{\boldsymbol{\eta}})$ , and polynomial  $h(X)$  of degree  $< N$ , output 1 if it satisfies the following conditions; otherwise, output 0:

1.  $\left\| 2\vec{h}_i \| 2\vec{\eta}_i \right\|_2 \stackrel{?}{\leq} 2d\beta_{\text{PC.Open}}$  for  $0 \leq i \leq m$
  2.  $\left\| 2\vec{h}_{m+1} \| 2\vec{\eta}_{m+1} \right\|_2 \stackrel{?}{\leq} 2\beta_{\text{PC}}$ , where  $\beta_{\text{PC}} = \beta_{\text{PC.Eval}} + \frac{(b+1)(m+1)dr}{2} \cdot \beta_{\text{PC.Open}}$
  3.  $\vec{h}_i \stackrel{?}{=} \mathbf{A}_0 \vec{h}_i + \mathbf{A}_1 \vec{\eta}_i \pmod{q}$  for  $0 \leq i \leq m+1$
  4.  $h(X) \stackrel{?}{=} \sum_{i=0}^{m-1} X^{ni} \cdot \langle \vec{h}_i, \vec{X} \rangle + X \cdot \langle \vec{h}_m, \vec{x} \rangle + \langle \vec{h}_{m+1}, \vec{X} \rangle \pmod{p}$ , where  $\vec{X} = (1, X, \dots, X^{n-1})$ , and  $\vec{h}_i = \frac{p+1}{2} \cdot \text{Dcd}(2\vec{h}_i) \pmod{p}$  for  $0 \leq i \leq m+1$
- $\text{PC.Eval}(x, \delta) \rightarrow (y, \rho)$ : Given an opening  $\delta = (\vec{h}, \vec{\eta})$  and an evaluation point  $x \in \mathbb{Z}_p$ , it generates an evaluation result  $y$  and an evaluation proof  $\rho$  as follows:
1. Compute  $\vec{e} = \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{h}_i + \text{Ecd}(x) \cdot \vec{h}_m + \vec{h}_{m+1}$  and  $\vec{\epsilon} = \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{\eta}_i + \text{Ecd}(x) \cdot \vec{\eta}_m + \vec{\eta}_{m+1}$ .
  2. Return an evaluation proof  $\rho = (\vec{e}, \vec{\epsilon})$ , and an evaluation result  $y = \langle \text{Dcd}(\vec{e}), (1, x, \dots, x^{n-1}) \rangle \pmod{p}$ .
- $\text{PC.Verify}(\text{ck}, \vec{h}, x, y, \rho) \rightarrow b$ : Given a commitment  $\vec{h}$ , an evaluation point  $x$ , an evaluation result  $y$ , and a proof  $\rho = (\vec{e}, \vec{\epsilon})$ , output 1 if  $\|\vec{e}\| \|\vec{\epsilon}\|_2 \leq \beta_{\text{PC.Eval}}$ ,  $y = \langle \text{Dcd}(\vec{e}), (1, x, \dots, x^{n-1}) \rangle \pmod{p}$ , and  $\mathbf{A}_0 \vec{e} + \mathbf{A}_1 \vec{\epsilon} = \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{h}_i + \text{Ecd}(x) \cdot \vec{h}_m + \vec{h}_{m+1} \pmod{q}$ ; otherwise, output 0.

The above polynomial commitment scheme is computationally hiding if both  $\text{MLWE}_{R, \nu, q, \sigma_1}$  and  $\text{MLWE}_{R, \nu, q, \sigma_3}$  are hard, and binding if  $\text{MSIS}_{R, \mu, q, 4\beta_{\text{PC}}}$  is hard. We also define a POK protocol  $\Pi_{\text{PC.Open}}$  for the polynomial commitment as  $\Pi_{\text{Open}}$  in Fig. 1, with the prover's input  $(\text{ck}, \vec{h}_0 \| \dots \| \vec{h}_m, \vec{h}_0 \| \dots \| \vec{h}_m, \vec{\eta}_0 \| \dots \| \vec{\eta}_m)$ , the verifier's input  $(\text{ck}, \vec{h}_0 \| \dots \| \vec{h}_m)$ , and replacing the value  $\beta_{\text{Open}}$  by  $\beta_{\text{PC.Open}}$ . The evaluation binding property is ensured by the hardness of the MSIS problem as follows.

**Theorem 5 (Evaluation Binding).** *The polynomial commitment scheme (PC.Setup, PC.Com, PC.Open, PC.Eval, PC.Verify) is evaluation binding if  $\text{MSIS}_{R, \mu, q, 2\beta_{\text{PC.Eval}}}$  is hard.*

*Proof.* Let  $\text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N)$ , and  $\mathcal{A}(\text{ck})$  be a PPT adversary that outputs  $(c, x, y, y', \rho, \rho')$ . Let  $\alpha$  be the probability that  $\text{PC.Verify}(\text{ck}, c, x, y, \rho) = 1$ ,  $\text{PC.Verify}(\text{ck}, c, x, y', \rho') = 1$ , and  $y \neq y'$ . Then, for  $\rho = (\vec{e}, \vec{\epsilon})$  and  $\rho' = (\vec{e}', \vec{\epsilon}')$ , it holds that  $\mathbf{A}_0(\vec{e} - \vec{e}') + \mathbf{A}_1(\vec{\epsilon} - \vec{\epsilon}') = \mathbf{0}$ , and  $\|(\vec{e} - \vec{e}')\| \|(\vec{\epsilon} - \vec{\epsilon}')\|_2 \leq 2\beta_{\text{PC.Eval}}$ . Also,  $\vec{e} \neq \vec{e}'$  since  $y = \langle \text{Dcd}(\vec{e}), (1, x, \dots, x^{n-1}) \rangle$ ,  $y' = \langle \text{Dcd}(\vec{e}'), (1, x, \dots, x^{n-1}) \rangle$ , and  $y \neq y'$ . Then,  $\mathcal{A}$  solves  $\text{MSIS}_{R, \mu, q, 2\beta_{\text{PC.Eval}}}$  with probability  $\alpha$ . Therefore,  $\alpha$  should be  $\text{negl}(\lambda)$  due to the hardness of  $\text{MSIS}_{R, \mu, q, 2\beta_{\text{PC.Eval}}}$ .  $\square$

For the evaluation hiding property, we extend the randomized encoding technique to our modified Ajtai commitment scheme. Notably, we sample the blinders  $\vec{h}_m$  and  $\vec{h}_{m+1}$  via randomized encoding of  $\vec{h}_m$  and  $\vec{h}_{m+1}$ . This offers simulatability not only in the encoded state in  $R$  but also in the decoded state in  $\mathbb{Z}_p$ . The proof follows a similar workflow as the simulatability proof for the POK protocol  $\Pi_{\text{Open}}$ .



**Theorem 6 (Evaluation Hiding).** *The polynomial commitment scheme (PC.Setup, PC.Com, PC.Open, PC.Eval, PC.Verify) is evaluation hiding if it satisfies the followings for  $\varepsilon = \text{negl}(\lambda)$ :*

- $\frac{1}{\sqrt{2}}\mathfrak{s}_3, \mathfrak{s}_{\text{PC.Eval}} \geq \frac{1}{b-1} \cdot \eta_\varepsilon ((P \otimes I_\ell)\mathbb{Z}^{d\ell})$  for  $\mathfrak{s}_{\text{PC.Eval}} = \left(\mathfrak{s}_1^{-2} + \left(\frac{(b+1)r}{2}\right)^2 \mathfrak{s}_3^{-2}\right)^{-1/2}$
- $\frac{1}{\sqrt{2}}\sigma_3, \frac{1}{\sqrt{2}}\sigma_{\text{PC.Eval}} \geq \eta_\varepsilon (\mathbb{Z}^{d(\mu+\nu)})$  for  $\sigma_{\text{PC.Eval}} = \frac{1}{\sqrt{2}} \cdot \left(\sigma_1^{-2} + \left(\frac{(b+1)r}{2}\right)^2 \sigma_3^{-2}\right)^{-1/2}$
- The advantage of  $\text{MLWE}_{R,\nu,q,\sigma_{\text{PC.Eval}}}$  is  $\text{negl}(\lambda)$

*Proof.* We refer to Appendix A.

The security of the POK protocol  $\Pi_{\text{PC.Open}}$  for the polynomial commitment scheme is as follows. The basic idea of the proof is similar to the case of  $\Pi_{\text{Open}}$ .

**Theorem 7 (Completeness).** *The proof of opening knowledge protocol  $\Pi_{\text{PC.Open}}$  has completeness if it satisfies the following conditions for  $\varepsilon = \text{negl}(\lambda)$ :*

- $\mathfrak{s}_1, \sqrt{m+2} \cdot \mathfrak{s}_2, \sigma_1, \sqrt{m+2} \cdot \sigma_2 \geq \eta_\varepsilon(\mathbb{Z}^d)$
- $\beta_{\text{PC.Open}} = \sqrt{\frac{\left(\left(\frac{(b+1)r}{2}\right)^2 \cdot \sigma_1 + \sqrt{m+2} \cdot \sigma_2\right)^2 \cdot (\mu+\nu) + (b+1)^2 \cdot \left(\frac{(b+1)r}{2}\right)^2 \cdot \ell}{\left(\frac{(b+1)r}{2}\right)^2 \cdot \ell}}$
- $\beta_{\text{PC.Eval}} = \sqrt{\frac{\left(\frac{(b+1)r}{2} \cdot \sigma_1 + \sqrt{m+2} \cdot \sigma_2\right)^2 \cdot (\mu+\nu) + (b+1)^2 \cdot \left(\frac{(b+1)r}{2} \cdot \sigma_1 + \sqrt{m+2} \cdot \sigma_2\right)^2 \cdot \ell}{\left(\frac{(b+1)r}{2}\right)^2 \cdot \ell}}$

*Proof.* Following the proof of Thm. 2, we obtain the following upper bounds. For  $\beta_{\text{PC.Open}}$ , setting  $k = m + 1$  in Thm. 2 yields the given upper bound. For  $\beta_{\text{PC.Eval}}$ , we can estimate upper bounds for  $\vec{e}$  and  $\vec{\varepsilon}$  as follows

$$\|\vec{e}\|_2 \leq \sum_{i=0}^m \frac{(b+1)r}{2} \|\vec{h}_i\|_2 + \|\vec{h}_{m+1}\|_2, \quad \|\vec{\varepsilon}\|_2 \leq \sum_{i=0}^m \frac{(b+1)r}{2} \|\vec{\eta}_i\|_2 + \|\vec{\eta}_{m+1}\|_2$$

since  $\|\text{Ecd}(x^k)\| \leq \frac{(b+1)r}{2}$  for any  $k$ . Then, applying upper bounds for  $\vec{h}_i$  and  $\vec{\eta}_i$  estimated from Lem. 5 yields the given upper bound for  $\beta_{\text{PC.Eval}}$   $\square$

Since we cannot compute the inverse Vandermonde matrix in  $R$  in general, we bypass the problem in extractability by additionally offering the proof of opening knowledge for the commitments. Then, our extractor extracts openings by rewinding the POK protocol, unlike [13, 14].

**Theorem 8 (Soundness).** *The proof of opening knowledge protocol  $\Pi_{\text{PC.Open}}$  satisfies soundness.*

*Proof.* Let  $\text{ck} \leftarrow \text{PC.Setup}(1^\lambda, N)$ , and  $\mathcal{P}^*$  be a PPT adversary which works as follows:

$$(\vec{\mathbf{h}}, x, y, \vec{e}, \vec{\varepsilon}) \leftarrow \mathcal{P}^*(\text{ck}), \quad (\pi, b) \leftarrow \left\langle \mathcal{P}^*(\text{ck}, \vec{\mathbf{h}}), \mathcal{V}(\text{ck}, \vec{\mathbf{h}}) \right\rangle$$

Let  $\alpha$  be the probability that  $b = 1$  and  $\text{PC.Verify}(\text{ck}, \vec{\mathbf{h}}, x, y, \vec{e}, \vec{\varepsilon}) = 1$ . Using Lem. 5 in [7], there exists an extractor  $\mathcal{E}^{\mathcal{P}^*}$  that can extract  $(\vec{h}'_i, \vec{\eta}'_i)$  such that  $\|\vec{h}'_i\|_2 \|\vec{\eta}'_i\|_2 \leq 2d\beta_{\text{PC.Open}}$  and  $\mathbf{A}_0 \vec{h}'_i + \mathbf{A}_1 \vec{\eta}'_i = 2\vec{h}_i$  for  $0 \leq i \leq m$  in expected time

$\text{poly}(\lambda)/\alpha$ . Let  $\vec{h}'_{m+1} = 2\vec{e} - \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{h}'_i - \text{Ecd}(x) \cdot \vec{h}'_m$ , and  $\vec{\eta}'_{m+1} = 2\vec{e} - \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{\eta}'_i - \text{Ecd}(x) \cdot \vec{\eta}'_m$ . Then, it holds that

$$\left\| \vec{h}'_{m+1} \parallel \vec{\eta}'_{m+1} \right\|_2 \leq 2\beta_{\text{PC.Eval}} + (b+1)(m+1)dr \cdot \beta_{\text{PC.Open}} = 2\beta_{\text{PC}}$$

since  $\|\text{Ecd}(x^k)\| \leq \frac{(b+1)r}{2}$  for any  $k$ . Also, it holds that

$$\mathbf{A}_0 \vec{h}'_{m+1} + \mathbf{A}_1 \vec{\eta}'_{m+1} = 2\vec{h}_{m+1} \pmod{q}$$

since  $\mathbf{A}_0 \vec{e} + \mathbf{A}_1 \vec{e} = \sum_{i=0}^{m-1} \text{Ecd}(x^{ni}) \cdot \vec{h}_i + \text{Ecd}(x) \cdot \vec{h}_m + \vec{h}_{m+1} \pmod{q}$  from  $\text{PC.Verify}(\text{ck}, \vec{h}, x, y, \vec{e}, \vec{e}) = 1$ . Then, for  $\delta = (\frac{q+1}{2} \cdot \vec{h}', \frac{q+1}{2} \cdot \vec{\eta}')$  and  $h(X) = \sum_{i=0}^{m-1} X^{ni} \cdot \langle \vec{h}_i, \vec{X} \rangle + X \cdot \langle \vec{h}_m, \vec{X} \rangle + \langle \vec{h}_{m+1}, \vec{X} \rangle$ , it holds that  $y = h(x)$  and  $\text{PC.Open}(\text{ck}, \vec{h}, h(X), \delta) = 1$ , where  $\vec{X} = (1, X, \dots, X^{n-1})$ , and  $\vec{h}_i = \frac{p+1}{2} \cdot \text{Dcd}(\vec{h}'_i) \pmod{p}$  for  $0 \leq i \leq m+1$ . Thus, in case of non-negligible  $\alpha$ , there exists an expected polynomial time extractor that outputs  $(h(X), \delta)$  such that  $\text{PC.Open}(\text{ck}, \vec{h}, h(X), \delta) = 1$  and  $y = h(x)$ .  $\square$

**Theorem 9 (Simulatability).** *The proof of opening knowledge protocol  $\Pi_{\text{PC.Open}}$  satisfies simulatability if it satisfies the followings for  $\varepsilon = \text{negl}(\lambda)$ :*

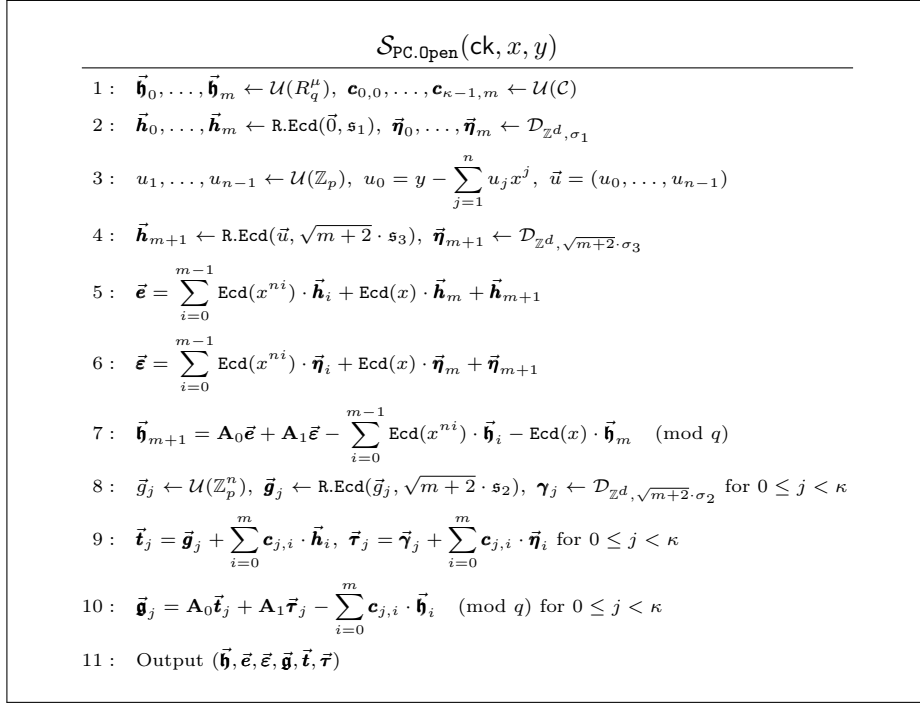
- $\frac{s_2}{\sqrt{2}}, \frac{s_3}{\sqrt{2}}, s_{\text{PC.Open}} \geq \frac{1}{b-1} \cdot \eta_\varepsilon((P \otimes I_\ell) \mathbb{Z}^{d\ell})$  for  $s_{\text{PC.Open}} = (\sigma_1^{-2} + \kappa \sigma_2^{-2} + (\frac{(b+1)r}{2})^2 \sigma_3^{-2})^{-1/2}$
- $\frac{\sigma_2}{\sqrt{2}}, \frac{\sigma_3}{\sqrt{2}}, \frac{\sigma_{\text{PC.Open}}}{\sqrt{2}} \geq \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$  for  $\sigma_{\text{PC.Open}} = \frac{1}{\sqrt{2}} \cdot (\sigma_1^{-2} + \kappa \sigma_2^{-2} + (\frac{(b+1)r}{2})^2 \sigma_3^{-2})^{-1/2}$
- The advantage of  $\text{MLWE}_{R,\nu,q,\sigma_{\text{PC.Open}}}$  is  $\text{negl}(\lambda)$

*Proof.* Let  $\text{ck} \leftarrow \text{PC.Setup}(1^\lambda)$ , and  $(h(X), x)$  be a polynomial and a point chosen by a PPT adversary  $\mathcal{A}$ . We prove that the algorithm  $\mathcal{S}_{\text{PC.Open}}$  in Fig. 3 is an efficient simulator for the distribution of evaluation proofs generated by  $\text{PC.Eval}$ , and real transcripts generated from  $\Pi_{\text{PC.Open}}$ . The proof can be directly derived from Thm. 4 and Thm. 6.  $\square$

## 4.2 Discussions

We first discuss the homomorphic property of our polynomial commitment scheme.

**Homomorphic Property.** In recent years, polynomial commitment schemes with homomorphic properties have become essential for efficient proof batching and recursive proof compositions [12, 17, 28]. Our polynomial commitment scheme naturally possesses homomorphic properties thanks to our new encoding method. Specifically, let  $\alpha \in \mathbb{Z}_p$  and  $\vec{h}, \vec{g}$  be commitments for polynomials  $h(X), g(X) \in \mathbb{Z}_p[X]$ . Then,  $\vec{f} = \vec{h} + \text{Ecd}(\alpha) \cdot \vec{g}$  becomes a commitment for the polynomial  $f(X) = h(X) + \alpha \cdot g(X)$  because  $\text{Ecd}(\alpha)$  is a small-norm element in  $R$ , so it does not significantly degrade the binding property. Thus, our polynomial commitment scheme supports a linear homomorphic property, which we expect


 Fig. 3. Simulator for  $\Pi_{\text{PC.Open}}$ 

to be beneficial for lattice-based proof batching and recursive proof composition systems.

We now discuss the complexity of our polynomial commitment scheme when committing a polynomial of degree less than  $N = nm$ .

**Prover Complexity.** We analyze the prover's complexity in generating commitments, evaluation proofs, and proofs of opening knowledge. For generating each commitment  $\vec{\mathbf{h}}_i = \mathbf{A}_0 \vec{\mathbf{h}}_i + \mathbf{A}_1 \vec{\boldsymbol{\eta}}_i$  for  $0 \leq i \leq m+1$ , it takes  $O(m \cdot \mu(\mu + \nu + \ell))$  ring arithmetic operations over  $R_q$ . For generating an evaluation proof, it takes  $O(m \cdot (\mu + \nu + \ell))$  ring arithmetic operations over  $R_q$  and  $O(n)$  arithmetic operations over  $\mathbb{Z}_p$ . Finally, for the proof of opening knowledge, it takes  $O(m \cdot (\mu + \nu + \ell))$  ring arithmetic operations over  $R_q$ . Assuming polynomial multiplications in  $R_q$  are done in  $O(d \log d)$  complexity, it takes  $O(md \log d \cdot \mu(\mu + \nu + \ell))$  computational complexity in total. We note that the quantity  $\mu d$  affects the security of MSIS and  $\nu d$  affects the security of MLWE and they follow  $O(\text{polylog}(N))$ . Then, the overall prover's complexity follows  $O(\frac{1}{d} \log d \cdot N \cdot \text{polylog}(N))$ , which is essentially  $\tilde{O}(N)$  since  $d$  can be considered independent of  $N$ . However, taking larger values of  $d$  improves the prover's concrete performance.

**Verifier Complexity.** We analyze the verifier's complexity in verifying evaluation proofs and proofs of opening knowledge. For verifying evaluation proofs,

it takes  $O(\mu(\mu + \nu + \ell + m))$  arithmetic operations over  $R_q$  and  $O(n)$  arithmetic operations over  $\mathbb{Z}_p$ . For verifying proofs of opening knowledge, it takes  $O(\mu(\mu + \nu + \ell + m))$  arithmetic operations over  $R_q$ . Then, the overall verifier's complexity follows  $O((n/d + m) \cdot \text{polylog}(N))$  using the same assumption as the prover's complexity.

**Communication Complexity.** We analyze the communication cost in sending commitments, evaluation proofs, and proofs of opening knowledge. For commitments, it sends  $O(m\mu)$  elements in  $R_q$ . For evaluation proofs, it sends  $O(\mu + \nu + \ell)$  elements in  $R_q$ . Finally, for proofs of opening knowledge, it sends  $O(\mu + \nu + \ell)$  elements in  $R_q$ . Hence, the communication cost is  $O(d(m\mu + \nu + \ell)) = O(m \cdot \text{polylog}(N) + n)$  in total.

Therefore, setting  $O(n) = O(m) = O(\sqrt{N})$  results in verifier and communication complexity in  $\tilde{O}(\sqrt{N})$ , which is sublinear in  $N$ .

## 5 Experimental Results

In this section, we present a proof-of-concept implementation of our sublinear polynomial commitment scheme and demonstrate its concrete performance. We implement our protocol using the Rust programming language and convert the interactive protocol into a non-interactive one using the Fiat-Shamir transform. For the functionality of the random oracle, we use the SHA-3 hash function. All experiments were performed with a single thread on a machine with an Intel(R) Xeon(R) Platinum 8268 CPU running at 2.90GHz with 384GB of RAM. Our source code is available at <https://github.com/SNUCP/celpc>.

### 5.1 Parameter Setting

We summarize all parameters that appear in our protocol in Table. 1. Firstly, we set the message modulus  $p = b^r + 1$  to be 255-bit sized primes using  $b = 63388$  and  $r = 16$ . Then, we set the parameters that determine the security of the binding and hiding properties of our polynomial commitment scheme, namely  $q, n, \mu, \nu, \mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3, \sigma_1, \sigma_2, \sigma_3$ . We recall that, to ensure the binding and hiding properties in our protocol,  $\text{MSIS}_{R, \mu, q, 4\beta_{\text{PC}}}$  and  $\text{MLWE}_{R, \nu, q, \sigma_{\text{PC, open}}}$  should be hard. We set the width parameters as follows for  $P_\ell := P \otimes I_\ell$ , so that they satisfy the conditions in Thm. 9.

$$\begin{aligned} \mathfrak{s}_1 &\geq \frac{\sqrt{3}}{b-1} \cdot \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell}), & \mathfrak{s}_2 &\geq \frac{\sqrt{3\kappa}}{b-1} \cdot \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell}), & \mathfrak{s}_3 &\geq \frac{\sqrt{3}(b+1)r}{2b-2} \cdot \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell}) \\ \sigma_1 &\geq 2\sqrt{3} \cdot \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)}), & \sigma_2 &\geq 2\sqrt{3\kappa} \cdot \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)}), & \sigma_3 &\geq \sqrt{3}(b+1)r \cdot \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)}) \end{aligned}$$

We estimate the upper bounds for the smoothing parameters using Lem. 1 for  $\varepsilon = 2^{-\lambda}$  and  $\lambda = 128$ . Once the width parameters are determined, the values of  $q, d, \mu, \nu$  determine the hardness of the MSIS and MLWE problem. To estimate their hardness, we compute the root Hermite factor  $\delta$  and set it to be approximately 1.005. As a result, we set  $\log q \approx 112$ ,  $d = 2048$ ,  $\mu = 1$ , and  $\nu = 2$ .

Type	Parameter	Description
Binding & Hiding	$q$	commitment modulus
	$d$	ring dimension
	$\mu$	MSIS rank
	$\nu$	MLWE rank
	$\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3, \sigma_1, \sigma_2, \sigma_3$	width parameters
Soundness	$b$	base
	$r$	digit lengths
	$p$	message modulus ( $= b^r + 1$ )
	$\kappa$	# of repetition ( $= \lceil \lambda / \log(2d) \rceil$ )
Proof Size	$n$	# of messages per commitments
	$\ell$	ring element per commitment ( $= rn/d$ )
	$m$	# of commitment
	$N$	input polynomial degree ( $= nm$ )

**Table 1.** Parameters

We note that  $d$  is set to be the largest possible value for the concrete efficiency of the prover and verifier. Finally, we adjust the parameters  $n, \ell, m$  to yield the smallest proof size depending on the input polynomial degree  $N$ , as summarized in Table 2.

$N$	$n$	$m$	$\ell$
$2^{19}$	$2^{12}$	$2^7$	$2^5$
$2^{21}$	$2^{13}$	$2^8$	$2^6$
$2^{23}$	$2^{14}$	$2^9$	$2^7$
$2^{25}$	$2^{15}$	$2^{10}$	$2^8$

**Table 2.** Parameters for optimal proof size

## 5.2 Implementation Technique

In this subsection, we present optimization techniques in our implementation.

**Ring Arithmetic.** For efficient ring arithmetic over  $R_q$ , we use Residue Number System (RNS) representation and Number Theoretic Transform (NTT). We recall that we use a commitment modulus  $q$  of size 112 bits, which cannot be covered by 64-bit native integer operations. Hence, we set  $q = q_1 \cdot q_2$ , where  $q_1$  and  $q_2$  are coprime 56-bit integers so that  $R_q \cong R_{q_1} \times R_{q_2}$  by the Chinese

Remainder Theorem. Then, elements in  $R_q$  can be represented as a pair of elements in  $R_{q_1}$  and  $R_{q_2}$ , which is often referred to as a RNS representation. Also, arithmetic over  $R_q$  can be instantiated with arithmetic over  $R_{q_1}$  and  $R_{q_2}$  due to the isomorphism, which can be implemented with 64-bit integer operations. For efficient multiplication in  $R_{q_1}$  and  $R_{q_2}$ , we use NTT operations, which reduce the complexity of polynomial multiplication from  $O(d^2)$  to  $O(d \log d)$ , by setting  $q_1$  and  $q_2$  to be primes such that  $q_1, q_2 = 1 \pmod{2d}$ .

**Discrete Gaussian Distribution.** Our commitment scheme requires samples from several Gaussian distributions. For distributions with small width parameters, we use the inversion sampling with precomputed cumulative distribution table. For larger width parameters with varying cosets, inversion sampling is infeasible, so we use the convolution sampler from [35] instead. We set the parameters for the convolution sampler to achieve max-log distance up to  $2^{-59}$ , which guarantees  $\geq 100$  bits of security.

**Commitment Compression.** We use the compression technique in [5] to reduce the size of commitments. Instead of sending the full size of commitments  $\vec{h}_0, \dots, \vec{h}_m$ , we can omit the lower  $D$  bits of them while maintaining the binding property. To be precise, suppose we omit the lower  $D$  bits of commitments. Then, it increases the bound of  $\beta_{\text{PC.open}}$  in the POK protocol  $\Pi_{\text{PC.open}}$  by  $(m+1)2^D \cdot \sqrt{\mu d}$ , but reduces the commitment size by a factor of  $(\log q - D) / \log q$ . Hence, in estimating proof size, we omit the lower  $D$  bits of commitments, where  $D$  is set to 24, the largest value that does not alter the MSIS parameter  $\mu$ .

### 5.3 Benchmark Results

We present the benchmark results for our polynomial commitment scheme in Table 3 together with other post-quantum secure polynomial commitment schemes from Brakedown [25] and FRI [9] for the 255-bit base field. Brakedown aims to provide faster prover’s performance, but it provides square root proof size like ours. FRI aims to provide a compact proof size, which is polylogarithmic in size with respect to  $N$ , but it has a slower prover’s performance. We note that both of their constructions are based on cryptographic hash functions. To provide comparison with them, we measure the elapsed time for the prover and the verifier, as well as the communication cost, which includes the size of commitments, evaluation proofs, and the proof of knowledge. The benchmark results for other polynomial commitment schemes are referenced from Fig. 8 in [25].

For the prover’s performance, we measure the elapsed time both with and without randomized encoding, since the benchmark results in [25] only measured in the non-zero-knowledge setting. Considering these differences, our scheme provides comparable performance with Brakedown and faster performance compared to FRI in the non-zero knowledge setting.<sup>3</sup> For the verifier’s performance

<sup>3</sup> In our benchmark for the non-zero knowledge version, we exclude all random sampling procedures.

$N$		$2^{19}$	$2^{21}$	$2^{23}$	$2^{25}$
Prover(seconds)	Ours	3.55	12.8	48.7	188
	Ours w/o ZK	0.97	3.47	13.0	50.9
	Brakedown	0.60	2.41	9.85	39.2
	FRI	6.42	31.9	60.8	353
Verifier(seconds)	Ours	0.14	0.27	0.53	1.07
	Brakedown	0.15	0.30	0.61	0.70
	FRI	0.03	0.03	0.04	0.04
Communication(MB)	Ours	6.07	11.9	23.6	47.5
	Brakedown	10.0	15.8	27.1	49.2
	FRI	0.39	0.49	0.61	0.74

**Table 3.** Benchmark results

and communication costs, our scheme provides comparable results with Brakedown due to similar asymptotic complexity, while FRI-based constructions offer better performance due to its polylogarithmic complexity. We note that another advantage of our polynomial commitment scheme lies in its homomorphic property, which may benefit in the recent proof composition techniques from [17, 28], while Brakedown and FRI don't offer homomorphic properties.

To compare with other lattice-based constructions, we reference the proof size from SLAP [3]. We compare the proof size for the same polynomial degree and similar base field size in Table 4. Although SLAP has an asymptotically better proof size, which is polylogarithmic in  $N$ , ours has a much smaller concrete proof size. Specifically, for a polynomial degree  $N = 2^{20}$ , our proof size is about 4.1 times smaller.

	$N$	$\log p$	Proof size
Ours	$2^{20}$	255	8.93 MB
SLAP	$2^{20}$	276	36.5 MB

**Table 4.** Performance comparison with SLAP [3]

## 6 Conclusion

In this paper, we have introduced a novel polynomial commitment scheme based on lattice-based cryptography. Prior to this work, there have been few studies addressing the construction of concretely efficient lattice-based polynomial commitments, which primarily focused on addressing asymptotic performance. Our proof system not only successfully achieves practical proof sizes through an efficient encoding method for large prime fields, but also provides practical proof generation and verification performance. Compared to other post-quantum secure polynomial commitments, such as Brakedown [25], our proof system yields comparable proof sizes, proof generation, and verification performance.

There are still several ways to improve our polynomial commitment. For the concrete performance, improving the discrete Gaussian sampling algorithm can lead to faster proof generation, as generating random samples currently consumes a significant portion of the total elapsed time. For reducing the proof size, we can enhance its asymptotic scale by adopting the leveled Ajtai commitment [15], which achieves  $\tilde{O}(N^{1/c})$  complexity by generalizing the POK protocols.

## Acknowledgement

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-TB2103-01.

## References

1. Ajtai, M.: Generating hard instances of lattice problems. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 99–108 (1996)
2. Albrecht, M.R., Cini, V., Lai, R.W., Malavolta, G., Thyagarajan, S.A.: Lattice-based snarks: Publicly verifiable, preprocessing, and recursively composable. In: Annual International Cryptology Conference. pp. 102–132. Springer (2022)
3. Albrecht, M.R., Fenzi, G., Lapiha, O., Nguyen, N.K.: Slap: succinct lattice-based polynomial commitments from standard assumptions. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 90–119. Springer (2024)
4. Attema, T., Lyubashevsky, V., Seiler, G.: Practical product proofs for lattice commitments. In: Annual International Cryptology Conference. pp. 470–499. Springer (2020)
5. Bai, S., Galbraith, S.D.: An improved compression technique for signatures based on learning with errors. In: Cryptographers’ Track at the RSA Conference. pp. 28–47. Springer (2014)
6. Banaszczyk, W.: Inequalities for convex bodies and polar reciprocal lattices in  $\mathbb{R}^n$ . *Discrete & Computational Geometry* **13**, 217–231 (1995)
7. Baum, C., Bootle, J., Cerulli, A., Del Pino, R., Groth, J., Lyubashevsky, V.: Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In: Annual International Cryptology Conference. pp. 669–699. Springer (2018)



8. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: International Conference on Security and Cryptography for Networks. pp. 368–385. Springer (2018)
9. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast reed-solomon interactive oracle proofs of proximity. In: 45th international colloquium on automata, languages, and programming. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2018)
10. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 551–572. Springer (2014)
11. Beullens, W., Seiler, G.: Labrador: Compact proofs for r1cs from module-sis. In: Annual International Cryptology Conference. pp. 518–548. Springer (2023)
12. Boneh, D., Drake, J., Fisch, B., Gabizon, A.: Halo infinite: Proof-carrying data from additive polynomial commitments. In: Annual International Cryptology Conference. pp. 649–680 (2021)
13. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 327–357. Springer (2016)
14. Bootle, J., Groth, J.: Efficient batch zero-knowledge arguments for low degree polynomials. In: IACR International Workshop on Public Key Cryptography. pp. 561–588. Springer (2018)
15. Bootle, J., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: A non-pcp approach to succinct quantum-safe zero-knowledge. In: Annual International Cryptology Conference. pp. 441–469. Springer (2020)
16. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. In: 2018 IEEE symposium on security and privacy (SP). pp. 315–334. IEEE (2018)
17. Bünz, B., Chiesa, A., Lin, W., Mishra, P., Spooner, N.: Proof-carrying data without succinct arguments. In: Annual International Cryptology Conference. pp. 681–710 (2021)
18. Bünz, B., Fisch, B., Szepieniec, A.: Transparent snarks from dark compilers. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 677–706 (2020)
19. Chen, H., Iliashenko, I., Laine, K.: When heaan meets fv: a new somewhat homomorphic encryption with reduced memory overhead. In: IMA International Conference on Cryptography and Coding. pp. 265–285. Springer (2021)
20. Chen, H., Laine, K., Player, R., Xia, Y.: High-precision arithmetic in homomorphic encryption. In: Cryptographers’ Track at the RSA Conference. pp. 116–136. Springer (2018)
21. Cini, V., Lai, R.W., Malavolta, G.: Lattice-based succinct arguments from vanishing polynomials. In: Annual International Cryptology Conference. pp. 72–105. Springer (2023)
22. Esgin, M.F., Nguyen, N.K., Seiler, G.: Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 259–288. Springer (2020)
23. Fisch, B., Liu, Z., Vesely, P.: Orbweaver: succinct linear functional commitments from lattices. In: Annual International Cryptology Conference. pp. 106–131. Springer (2023)

24. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 31–51. Springer (2008)
25. Golovnev, A., Lee, J., Setty, S., Thaler, J., Wahby, R.S.: Brakedown: Linear-time and field-agnostic snarks for r1cs. In: Annual International Cryptology Conference. pp. 193–226. Springer (2023)
26. Kate, A., Zaverucha, G.M., Goldberg, I.: Constant-size commitments to polynomials and their applications. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 177–194. Springer (2010)
27. Kim, D., Lee, D., Seo, J., Song, Y.: Toward practical lattice-based proof of knowledge from hint-mlwe. In: Annual International Cryptology Conference. pp. 549–580. Springer (2023)
28. Kothapalli, A., Setty, S., Tzialla, I.: Nova: Recursive zero-knowledge arguments from folding schemes. In: Annual International Cryptology Conference. pp. 359–388. Springer (2022)
29. Kuchta, V., Sakzad, A., Steinfeld, R., Liu, J.K.: Efficient lattice-based polynomial evaluation and batch zk arguments. In: International Conference on Selected Areas in Cryptography. pp. 3–33. Springer (2021)
30. Lee, J.: Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments. In: Theory of Cryptography Conference. pp. 1–34. Springer (2021)
31. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
32. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Practical lattice-based zero-knowledge proofs for integer relations. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. pp. 1051–1070 (2020)
33. Mera, J.M.B., Karmakar, A., Marc, T., Soleimani, A.: Efficient lattice-based inner-product functional encryption. In: IACR International Conference on Public-Key Cryptography. pp. 163–193. Springer (2022)
34. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing* **37**(1), 267–302 (2007)
35. Micciancio, D., Walter, M.: Gaussian sampling over the integers: Efficient, generic, constant-time. In: Annual International Cryptology Conference. pp. 455–485. Springer (2017)
36. Nguyen, N.K., Seiler, G.: Practical sublinear proofs for r1cs from lattices. In: Annual International Cryptology Conference. pp. 133–162. Springer (2022)
37. Peikert, C.: An efficient and parallel gaussian sampler for lattices. In: Annual Cryptology Conference. pp. 80–97. Springer (2010)
38. Tomescu, A., Chen, R., Zheng, Y., Abraham, I., Pinkas, B., Gueta, G.G., Devadas, S.: Towards scalable threshold cryptosystems. In: 2020 IEEE Symposium on Security and Privacy (SP). pp. 877–893. IEEE (2020)
39. Wee, H., Wu, D.J.: Lattice-based functional commitments: Fast verification and cryptanalysis. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 201–235. Springer (2023)
40. Wee, H., Wu, D.J.: Succinct vector, polynomial, and functional commitments from lattices. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 385–416. Springer (2023)
41. Zhang, J., Xie, T., Hoang, T., Shi, E., Zhang, Y.: Polynomial commitment with a One-to-Many prover and applications. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 2965–2982 (2022)

## A Proof of Thm. 6

$$\mathcal{S}_{\text{PC.Eval}}(\text{ck}, x, y)$$


---

- 1:  $\vec{\mathbf{h}}_0, \dots, \vec{\mathbf{h}}_m \leftarrow \mathcal{U}(R_q^\mu)$
- 2:  $\vec{\mathbf{h}}_0, \dots, \vec{\mathbf{h}}_m \leftarrow \text{R.Ecd}(\vec{0}, \mathfrak{s}_1), \vec{\mathbf{\eta}}_0, \dots, \vec{\mathbf{\eta}}_m \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}$
- 3:  $u_1, \dots, u_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), u_0 = y - \sum_{j=1}^n u_j x^j, \vec{u} = (u_0, \dots, u_{n-1})$
- 4:  $\vec{\mathbf{h}}_{m+1} \leftarrow \text{R.Ecd}(\vec{u}, \sqrt{m+2} \cdot \mathfrak{s}_3), \vec{\mathbf{\eta}}_{m+1} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}$
- 5:  $\vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m + \vec{\mathbf{h}}_{m+1}$
- 6:  $\vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{\eta}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{\eta}}_m + \vec{\mathbf{\eta}}_{m+1}$
- 7:  $\vec{\mathbf{h}}_{m+1} = \mathbf{A}_0 \vec{\mathbf{e}} + \mathbf{A}_1 \vec{\mathbf{e}} - \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i - \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m \pmod{q}$
- 8: Output  $(\vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{e}})$

Fig. 4. Simulator for PC.Eval

*Proof.* Let  $\text{ck} \leftarrow \text{PC.Setup}(1^\lambda)$ , and  $(h(X), x)$  be a polynomial and a point chosen by a PPT adversary  $\mathcal{A}$ . We prove that the algorithm  $\mathcal{S}_{\text{PC.Eval}}$  in Fig. 4 is an efficient simulator for the distribution of evaluation proofs generated by PC.Eval using hybrid arguments. We first define a distribution  $\mathcal{H}_0(\text{ck}, h(X), x)$  as follows.

$$\left\{ \begin{array}{l} \vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{e}} \\ \left. \begin{array}{l} b_1, \dots, b_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), \vec{\mathbf{h}}_m = (b_1, \dots, b_{n-1}, 0), \vec{\mathbf{h}}_{m+1} = (0, -b_1, \dots, -b_{n-1}) \\ \vec{\mathbf{h}}_i \leftarrow \text{R.Ecd}(\vec{h}_i, \mathfrak{s}_1), \vec{\mathbf{\eta}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{h}}_i = \mathbf{A}_0 \vec{\mathbf{h}}_i + \mathbf{A}_1 \vec{\mathbf{\eta}}_i \text{ for } 0 \leq i \leq m \\ \vec{\mathbf{h}}_{m+1} \leftarrow \text{R.Ecd}(\vec{h}_{m+1}, \sqrt{m+2} \cdot \mathfrak{s}_3), \vec{\mathbf{\eta}}_{m+1} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}^{\mu+\nu} \\ \vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m + \vec{\mathbf{h}}_{m+1} \\ \vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{\eta}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{\eta}}_m + \vec{\mathbf{\eta}}_{m+1} \\ \vec{\mathbf{h}}_{m+1} = \mathbf{A}_0 \vec{\mathbf{e}} + \mathbf{A}_1 \vec{\mathbf{e}} - \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i - \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m \end{array} \right\} \end{array} \right.$$

We note that the above distribution is identical to the distribution of evaluation proofs generated from PC.Eval

**Claim 1:**  $\mathcal{H}_0(\text{ck}, h(X), x)$  and  $\mathcal{H}_1(\text{ck}, h(X), x)$  are statistically indistinguishable, where  $\mathcal{H}_1(\text{ck}, h(X), x)$  is defined as follows.

$$\left( \begin{array}{l} \vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{\epsilon}} \\ \left. \begin{array}{l} b_1, \dots, b_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), \vec{h}_m = (b_1, \dots, b_{n-1}, 0), \vec{h}_{m+1} = (0, -b_1, \dots, -b_{n-1}) \\ \vec{h}_i \leftarrow \text{R.Ecd}(\vec{h}_i, \mathfrak{s}_1), \vec{\eta}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathfrak{h}}_i = \mathbf{A}_0 \vec{h}_i + \mathbf{A}_1 \vec{\eta}_i \text{ for } 0 \leq i \leq m \\ \vec{h}_{m+1, i} \leftarrow \text{R.Ecd}(-x^{n_i} \cdot \vec{h}_i, \mathfrak{s}_3) \text{ for } 0 \leq i < m, \vec{h}_{m+1, m} \leftarrow \text{R.Ecd}(-x \cdot \vec{h}_m, \mathfrak{s}_3) \\ \vec{h}_{m+1, m+1} \leftarrow \text{R.Ecd}\left(\sum_{i=0}^{m-1} x^{n_i} \cdot \vec{h}_i + x \cdot \vec{h}_m + \vec{h}_{m+1}, \mathfrak{s}_3\right) \\ \vec{\eta}_{m+1, 0}, \dots, \vec{\eta}_{m+1, m+1} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_3}^{\mu+\nu} \\ \vec{\mathbf{e}} = \sum_{i=0}^{m-1} (\vec{h}_{m+1, i} + \text{Ecd}(x^{n_i}) \cdot \vec{h}_i) + (\vec{h}_{m+1, m} + \text{Ecd}(x) \cdot \vec{h}_m) + \vec{h}_{m+1, m+1} \\ \vec{\mathbf{\epsilon}} = \sum_{i=0}^{m-1} (\vec{\eta}_{m+1, i} + \text{Ecd}(x^{n_i}) \cdot \vec{\eta}_i) + (\vec{\eta}_{m+1, m} + \text{Ecd}(x) \cdot \vec{\eta}_m) + \vec{\eta}_{m+1, m+1} \\ \vec{\mathfrak{h}}_{m+1} = \mathbf{A}_0 \vec{\mathbf{e}} + \mathbf{A}_1 \vec{\mathbf{\epsilon}} - \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathfrak{h}}_i - \text{Ecd}(x) \cdot \vec{\mathfrak{h}}_m \end{array} \right\} \end{array} \right.$$

The difference between  $\mathcal{H}_0(\text{ck}, h(X), x)$  and  $\mathcal{H}_1(\text{ck}, h(X), x)$  lies in how  $\vec{h}_{m+1}$  and  $\vec{\eta}_{m+1}$  are sampled. Let  $P_\ell = P \otimes I_\ell \in \mathbb{R}^{d\ell \times d\ell}$ . In  $\mathcal{H}_0(\text{ck}, h(X), x)$ ,  $\vec{h}_{m+1} \sim \mathcal{D}_{\text{Ecd}(\vec{h}_{m+1}) + P_\ell \mathbb{Z}^{d\ell}, \sqrt{m+2} \cdot \mathfrak{s}_3 P_\ell}$  and  $\vec{\eta}_{m+1} \sim \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}^{\mu+\nu}$ . Meanwhile, in  $\mathcal{H}_1(\text{ck}, h(X), x)$ ,

$$\begin{aligned} \vec{h}_{m+1} &= \sum_{i=0}^{m+1} \vec{h}_{m+1, i} \sim \frac{\mathcal{D}_{\text{Ecd}(-x^{n_i} \cdot \vec{h}_i) + P_\ell \mathbb{Z}^{d\ell}, \mathfrak{s}_3 P_\ell} + \mathcal{D}_{\text{Ecd}(-x \cdot \vec{h}_m) + P_\ell \mathbb{Z}^{d\ell}, \mathfrak{s}_3 P_\ell} + \mathcal{D}_{\sum_{i=0}^{m-1} x^{n_i} \cdot \vec{h}_i + x \cdot \vec{h}_m + \vec{h}_{m+1} + P_\ell \mathbb{Z}^{d\ell}, \mathfrak{s}_3 P_\ell}}{3} \\ \vec{\eta}_{m+1} &= \sum_{i=0}^{m+1} \vec{\eta}_{m+1, i} \sim \sum_{i=0}^{m+1} \mathcal{D}_{\mathbb{Z}^d, \sigma_3}^{\mu+\nu} \end{aligned}$$

By Lem. 10, the distribution of  $\vec{h}_{m+1}$  in  $\mathcal{H}_1(\text{ck}, h(X), x)$  is within a statistical distance of  $8(m+1)\varepsilon$  from  $\mathcal{D}_{\text{Ecd}(\vec{h}_{m+1}) + P_\ell \mathbb{Z}^{d\ell}, \sqrt{m+2} \cdot \mathfrak{s}_3 P_\ell}$  if  $\frac{\mathfrak{s}_3}{\sqrt{2}} P_\ell \geq \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$ , which is implied by  $\frac{\mathfrak{s}_3}{\sqrt{2}} \geq \frac{1}{b-1} \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$ . In a similar manner, by Lem. 10,  $\sum_{i=0}^{m+1} \mathcal{D}_{\mathbb{Z}^d, \sigma_3}^{\mu+\nu}$  is within a statistical distance of  $8(m+1)\varepsilon$  from  $\mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}^{\mu+\nu}$  if  $\frac{\sigma_3}{\sqrt{2}} \geq \eta_\varepsilon(\mathbb{Z}^{d(\mu+\nu)})$ . Therefore,  $\mathcal{H}_0(\text{ck}, \vec{m})$  and  $\mathcal{H}_1(\text{ck}, \vec{m})$  are within a statistical distance  $16(m+1)\varepsilon$ , which is  $\text{negl}(\lambda)$ .

**Claim 2:**  $\mathcal{H}_1(\text{ck}, h(X), x)$  and  $\mathcal{H}_2(\text{ck}, h(X), x)$  are computationally indistinguishable, where  $\mathcal{H}_2(\text{ck}, h(X), x)$  is defined as follows.

$$\left( \begin{array}{l} \vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{\epsilon}} \\ \left. \begin{array}{l} b_1, \dots, b_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), \vec{h}_m = (b_1, \dots, b_{n-1}, 0), \vec{h}_{m+1} = (0, -b_1, \dots, -b_{n-1}) \\ \vec{h}_i \leftarrow \text{R.Ecd}(\vec{h}_i, \mathfrak{s}_1), \vec{\eta}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathfrak{h}}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i \leq m \\ \text{The rest is same as } \mathcal{H}_1(\text{ck}, h(X), x). \end{array} \right\} \end{array} \right.$$

The difference between  $\mathcal{H}_1(\text{ck}, h(X), x)$  and  $\mathcal{H}_2(\text{ck}, h(X), x)$  lies in how  $\vec{\mathfrak{h}}$  is sampled. Then, distinguishing  $\mathcal{H}_1(\text{ck}, h(X), x)$  and  $\mathcal{H}_2(\text{ck}, h(X), x)$  is at least hard

as  $\text{HintMLWE}_{R,\nu,q,\sigma_1,\sigma_3}^{\text{Ecd}(x^{n^i})}$  from some  $k$  since  $(\vec{\mathbf{h}}_i, \vec{\mathbf{\eta}}_{m+1,i} + \text{Ecd}(x^{n^i}) \cdot \vec{\mathbf{\eta}}_i)$  can be considered as a  $\text{HintMLWE}$  instance for  $0 \leq i < m$ . Similarly,  $(\vec{\mathbf{h}}_m, \vec{\mathbf{\eta}}_{m+1,m} + \text{Ecd}(x) \cdot \vec{\mathbf{\eta}}_m)$  can be considered as an instance from  $\text{HintMLWE}_{R,\nu,q,\sigma_1,\sigma_3}^{\text{Ecd}(x)}$ . By Thm. 1, there is an efficient reduction from  $\text{MLWE}_{R,\nu,q,\sigma_{\text{PC.Eval}}}$  to  $\text{HintMLWE}_{R,\nu,q,\sigma_1,\sigma_3}^{\text{Ecd}(x^k)}$  for any  $k$  if  $\sigma_{\text{PC.Eval}} \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^d)$ , given that  $\|\text{Ecd}(x^k)\|_1 \leq \frac{(b+1)r}{2}$ . Therefore,  $\mathcal{H}_1(\text{ck}, h(X), x)$  and  $\mathcal{H}_2(\text{ck}, h(X), x)$  are computationally indistinguishable due to the hardness of  $\text{MLWE}_{R,\nu,q,\sigma_{\text{PC.Eval}}}$ .

**Claim 3:**  $\mathcal{H}_2(\text{ck}, h(X), x)$  and  $\mathcal{H}_3(\text{ck}, x, h(x))$  are statistically indistinguishable, where  $\mathcal{H}_3(\text{ck}, x, h(x))$  is defined as follows.

$$\left( \vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{\varepsilon}} \right) \left| \begin{array}{l} u_1, \dots, u_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), u_0 = h(x) - \sum_{j=1}^n b_j x^j, \vec{\mathbf{u}} = (u_0, \dots, u_{n-1}) \\ \vec{\mathbf{h}}_i \leftarrow \text{R.Ecd}(\vec{\mathbf{0}}, \mathfrak{s}_1), \vec{\mathbf{\eta}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{h}}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i \leq m \\ \vec{\mathbf{h}}_{m+1,i} \leftarrow \text{R.Ecd}(\vec{\mathbf{0}}, \mathfrak{s}_3) \text{ for } 0 \leq i \leq m, \vec{\mathbf{h}}_{m+1,m+1} \leftarrow \text{R.Ecd}(\vec{\mathbf{u}}, \mathfrak{s}_3) \\ \text{The rest is same as } \mathcal{H}_2(\text{ck}, h(X), x). \end{array} \right.$$

The difference between  $\mathcal{H}_2(\text{ck}, h(X), x)$  and  $\mathcal{H}_3(\text{ck}, x, h(x))$  lies in how  $\vec{\mathbf{h}}_0, \dots, \vec{\mathbf{h}}_m$ , and  $\vec{\mathbf{h}}_{m+1,0}, \dots, \vec{\mathbf{h}}_{m+1,m+1}$  are sampled. We note that  $\vec{\mathbf{e}} = \sum_{i=0}^{m-1} (\vec{\mathbf{h}}_{m+1,i} + \text{Ecd}(x^{n^i}) \cdot \vec{\mathbf{h}}_i) + (\vec{\mathbf{h}}_{m+1,m} + \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m) + \vec{\mathbf{h}}_{m+1,m+1}$ , so it suffices to consider only the distribution of  $(\vec{\mathbf{h}}_{m+1,i} + \text{Ecd}(x^{n^i}) \cdot \vec{\mathbf{h}}_i)$  for  $0 \leq i < m$ ,  $(\vec{\mathbf{h}}_{m+1,m} + \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m)$ , and  $\vec{\mathbf{h}}_{m+1,m+1}$ . By Lem. 11, the first two distributions in  $\mathcal{H}_2(\text{ck}, h(X), x)$  and  $\mathcal{H}_3(\text{ck}, x, h(x))$  are within a statistical distance  $2\varepsilon$ , if the following holds

$$\sqrt{(\mathfrak{s}_1^{-2} (P_\ell P_\ell^\top)^{-1} + \mathfrak{s}_3^{-2} \cdot X_k^\top (P_\ell P_\ell^\top)^{-1} X_k)^{-1}} \geq \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$$

where  $X_k$  is the negacyclic matrix of  $\text{Ecd}(x^k)$ . By Lem. 2, this condition is implied by  $\mathfrak{s}_{\text{PC.Eval}} \geq \frac{1}{b-1} \cdot \eta_\varepsilon(P_\ell \mathbb{Z}^{d\ell})$  since  $\|X_k\|_2 \leq \frac{(b+1)r}{2}$  and  $\|P^{-1}\|_2 \leq \frac{1}{b-1}$ . For  $\vec{\mathbf{h}}_{m+1,m+1}$ , it is a randomized encoding of  $\sum_{i=0}^{m-1} x^{n^i} \cdot \vec{\mathbf{h}}_i + x \cdot \vec{\mathbf{h}}_m + \vec{\mathbf{h}}_{m+1}$  in  $\mathcal{H}_2(\text{ck}, h(X), x)$ , whereas that of  $\mathcal{H}_3(\text{ck}, x, h(x))$  is a randomized encoding of  $\vec{\mathbf{u}}$ . We note that these distributions are identical in the sense that a uniformly sampled vector, whose inner product with  $(1, x, \dots, x^{n-1})$ , results in  $h(x)$ . Thus, we can regard  $\vec{\mathbf{b}}_{0,m+1}$  in  $\mathcal{H}_2$  and  $\mathcal{H}_3$  as following an identical distribution, given the evaluation point  $x$  and the evaluation result  $h(x)$ . Therefore,  $\mathcal{H}_2(\text{ck}, h(X), x)$  and  $\mathcal{H}_3(\text{ck}, x, h(x))$  are within a statistical distance  $2(m+1)\varepsilon$ , which is  $\text{negl}(\lambda)$ .

**Claim 4:**  $\mathcal{H}_3(\text{ck}, x, h(x))$  and  $\mathcal{H}_4(\text{ck}, x, h(x))$  are statistically indistinguishable, where  $\mathcal{H}_4(\text{ck}, x, h(x))$  is defined as follows.

$$\left( \vec{\mathbf{h}}, \vec{\mathbf{e}}, \vec{\mathbf{e}} \right) \left\{ \begin{array}{l} u_1, \dots, u_{n-1} \leftarrow \mathcal{U}(\mathbb{Z}_p), u_0 = h(x) - \sum_{j=1}^n b_j x^j, \vec{u} = (u_0, \dots, u_{n-1}) \\ \vec{\mathbf{h}}_i \leftarrow \text{R.Ecd}(\vec{0}, \mathbf{s}_1), \vec{\mathbf{\eta}}_i \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sigma_1}^{\mu+\nu}, \vec{\mathbf{h}}_i \leftarrow \mathcal{U}(R_q^\mu) \text{ for } 0 \leq i \leq m \\ \vec{\mathbf{h}}_{m+1} \leftarrow \text{R.Ecd}(\vec{u}, \sqrt{m+2} \cdot \mathbf{s}_3), \vec{\mathbf{\eta}}_{m+1} \leftarrow \mathcal{D}_{\mathbb{Z}^d, \sqrt{m+2} \cdot \sigma_3}^{\mu+\nu} \\ \vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m + \vec{\mathbf{h}}_{m+1} \\ \vec{\mathbf{e}} = \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{\eta}}_i + \text{Ecd}(x) \cdot \vec{\mathbf{\eta}}_m + \vec{\mathbf{\eta}}_{m+1} \\ \vec{\mathbf{h}}_{m+1} = \mathbf{A}_0 \vec{\mathbf{e}} + \mathbf{A}_1 \vec{\mathbf{e}} - \sum_{i=0}^{m-1} \text{Ecd}(x^{n_i}) \cdot \vec{\mathbf{h}}_i - \text{Ecd}(x) \cdot \vec{\mathbf{h}}_m \end{array} \right.$$

The difference between  $\mathcal{H}_3$  and  $\mathcal{H}_4$  lies in how  $\vec{\mathbf{h}}_{m+1}$  and  $\vec{\mathbf{\eta}}_{m+1}$  are sampled, similar to Claim 1. Thus, following the proof of Claim 1, they are within a statistical distance of  $16(m+1)\varepsilon$ , which is  $\text{negl}(\lambda)$ .

Therefore, we can conclude that  $\mathcal{H}_0(\text{ck}, h(X), x)$  and  $\mathcal{H}_4(\text{ck}, x, h(x))$  are computationally indistinguishable under the given conditions. Since  $\mathcal{H}_0(\text{ck}, h(X), x)$  corresponds to the distribution of evaluation transcripts from  $\mathcal{S}_{\text{PC.Eval}}$ , and  $\mathcal{H}_4(\text{ck}, x, h(x))$  corresponds to the distribution of simulated proofs from  $\mathcal{S}_{\text{PC.Eval}}$ , there is no PPT adversary  $\mathcal{A}$  that can distinguish between the two distributions with non-negligible probability.  $\square$