Efficient Lattice-based Sublinear Arguments for R1CS without Aborts

Intak Hwang[1], Jinyeong Seo[1], and Yongsoo Song[1]

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

Abstract. We propose a new lattice-based sublinear argument for R1CS that not only achieves efficiency in concrete proof size but also demonstrates practical performance in both proof generation and verification. To reduce the proof size, we employ a new encoding method for large prime fields, resulting in a compact proof for R1CS over such fields. We also devise a new proof technique that randomizes the input message. This results in fast proof generation performance, eliminating rejection sampling from the proving procedure. Compared to Ligero (CCS 2017), a hash-based post-quantum SNARK, our proof system yields a comparable proof size and proof generation performance, and excels in verification performance by an order of magnitude.

Keywords: Lattices · Zero-knowledge proofs · Hint-MLWE · R1CS

1 Introduction

In recent years, publicly verifiable zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) have undergone extensive research and developments. In particular, proof systems based on the discrete logarithm problem (e.g. [12, 15]) have gained renown for their efficiency in cryptographic applications. When considering (plausibly) post-quantum succinct zero-knowledge proof systems, hash-based constructions such as Aurora [9] and Ligero [3] have emerged as practical candidates.

Meanwhile, in the realm of lattice-based cryptography, significant progress has been made, rapidly replacing its pre-quantum counterparts in signature [18] and encryption schemes [14]. This advancement also extends to lattice-based proof systems, whose proof sizes scale linearly with the input size. Based on the efficient commitment scheme called BDLOP [8], there has been a line of research [5, 19, 25, 26] that proves various arithmetic relations, such as product and linear relations, with small proof sizes. This advancement in linear size proof systems results in efficient constructions in ring and group signatures [24, 27].

Unfortunately, there is still a relatively limited number of studies on practical lattice-based SNARK constructions to date, despite several previous research efforts [4,7,11,13,31]. Compared to the linear-scale proof systems, which are mainly based on the BDLOP scheme, these sublinear-scale proof systems are commonly built on top of the Ajtai [2] commitment scheme due to its intrinsic property

where commitment size is almost independent of the input size, as opposed to the BDLOP scheme where commitment size grows linearly with the input size. However, the main challenge in proof systems with Ajtai commitments is the restriction in its message space, which should be kept small due to security issues, hindering practical construction, whereas BDLOP commitment schemes have no limitations on message size, making it easier to construct proof systems. Additionally, these protocols often contain sequential rejection sampling to achieve zero-knowledgeness. However, sequential rejection samplings greatly degrade concrete proving performance since the repetition rate grows exponentially with the number of sequential rejection samplings.

The recent work [31] and [11] have successfully demonstrated the practicality of lattice-based SNARKs in terms of proof size, overcoming limitations in message space. However, the zero-knowledge version of [11] was not presented in the literature, and the protocol of [31] includes sequential rejection samplings to achieve the zero-knowledge property, which inhibits practical proving performance. In addition, to the best of our knowledge, these prior work on lattice-based SNARKs analyzed only the size of proofs, but the concrete performance of proof generation or verification has never been demonstrated in the literature, even though lattice-based cryptography is renowned for its efficiency in computational performance.

1.1 Our Contributions

In this paper, we design a new lattice-based SNARK system that is not only practical in terms of proof size but also efficient in the concrete performance of proof generation and verification. Similar to [3, 9, 11, 31], our primary goal is to prove the satisfiability of R1CS (Rank-1 Constraint System) over a finite field $\mathbb{F}$, an NP-complete problem that generalizes arithmetic circuit satisfiability. As our proof system is also built on top of the Ajtai commitment scheme, it inherits the aforementioned issues: restrictions in message space and sequential rejection sampling.

First of all, the base field $\mathbb{F}$ in R1CS generally has a large size to obtain a negligible soundness error. We solve the issue by adopting the encoding method of [16, 17], which transforms elements of a large prime field $\mathbb{Z}_p$ into ring elements with small coefficients. As a result, we can take $\mathbb{Z}_p$ as a message space while maintaining the security of the Ajtai commitment, allowing us to compose arguments for R1CS over $\mathbb{Z}_p$.

We also combine the methodology of [22] to avoid sequential rejection samplings from the BDLOP-based proof systems. Briefly speaking, the conventional construction exploits the rejection sampling method to make the transcript independent of the input, as it regards the commitment randomness as a fixed value. In contrast, we consider the distribution of commitment randomness in security analysis and prove that the whole transcript of our protocol can be simulated independent of the secret input. However, this method is not directly applicable to Ajtai-based proof systems since the rejection sampling technique is also used for hiding a message different from the BDLOP-based proof systems. Hence,

we resolve this issue by randomizing messages, leveraging the property that the arithmetic relation still holds unless the inputs are equal modulo p. This new proof technique enables us to completely eliminate rejection sampling from our protocol, providing efficiency in proof generation.

Regarding the concrete performance, our proof system is comparable to the prior work Ligero [3]. For a circuit size N, Ligero [3] achieves a sublinear proof size of $O(\sqrt{N})$ (which is larger than polylogarithmic complexity of Aurora [9]), but it enjoys much faster proof generation. This work asymptotically achieves a sublinear complexity of $O(\sqrt{N \log N})$ and has a similar concrete proof size to [3] when the size of the base field is approximately 2^{128}, as presented in Table 3. However, for a larger field size, such as 2^{256}, our proof system yields a smaller proof about by a factor of two compared to Ligero. For the concrete performance of proof generation and verification for our proof system, we summarize the benchmark results in Fig. 10 and Table 4 for a field size about 2^{128}, together with Ligero, and AuroraSNARK. While the proof generation is slightly slower than Ligero, our proof system is still significantly faster than that of Aurora and our verification procedure outperforms the prior work by an order of magnitude.

1.2 Technical Overview

Efficient Encoding Method. For a polynomial ring $R = \mathbb{Z}[X]/(X^n + 1)$, one can prove that $\mathbb{Z}_p^m \cong R/(X^m - b)$ where $m \mid n$ and $p = b^{n/m} + 1$ is a prime. Then, there exists an isomorphism between them. We refer to the process of converting a vector in $\mathbb{Z}_p^m$ to a ring element $R/(X^m - b)$ as encoding, denoted by $\texttt{Ecd}$, and the reverse process as decoding, denoted by $\texttt{Dcd}$. We note that each ring element in $R/(X^m - b)$ has a representation in R with coefficients bounded by $\frac{b+2}{2}$. Since we typically set b to be much smaller than p, the encoding procedure involves converting a vector of large prime field elements into ring elements with small coefficients.

In our proof system, we utilize this encoding method to convert messages from the large prime field $\mathbb{Z}_p$ to small ring elements and vice versa for the security of the Ajtai commitment scheme. From an information-theoretic perspective, this encoding does not necessarily improve efficiency in terms of proof size for linear-scale proof systems, as it essentially expands m elements of size p into n elements of size b, which are of similar sizes. However, since our proof system yields sublinear proof size, an increase in input dimension leads to higher compression rates. As a result, this encoding method not only provides advantages in terms of achieving security but also results in smaller proof sizes.

Message Randomization. To achieve simulatability (i.e., zero-knowledgeness), our proof system randomizes the input message $\vec{x} \in \mathbb{Z}_p^m$, more precisely, the encoded messages $\texttt{Ecd}(\vec{x})$. During proof generation, the prover frequently sends a response in the form of $\boldsymbol{y} + \boldsymbol{c} \cdot \texttt{Ecd}(\vec{x})$ to the verifier, where $\boldsymbol{y}$ is a random noise for masking $\texttt{Ecd}(\vec{x})$, sampled by the prover, and $\boldsymbol{c}$ is a random challenge sampled by the verifier. Since this response leaks information about $\vec{x}$, which must be

kept secret from the verifier, previous constructions use the rejection sampling method [23] to make it independent of $\vec{x}$.

Since the verifier usually checks the satisfiability of R1CS with the decoded value, the prover can replace the value of $\mathtt{Ecd}(\vec{x})$ with $\boldsymbol{x}$ where $\boldsymbol{x} = \mathtt{Ecd}(\vec{x}) \pmod{X^m - b}$. Using this property, we randomize $\mathtt{Ecd}(\vec{x})$ by generating $\boldsymbol{x}$ from $\mathcal{D}_{\mathtt{Ecd}(m)+P\mathbb{Z}^n,\mathfrak{s}_1}$, a sample from the discrete Gaussian distribution over the coset $\mathtt{Ecd}(\vec{x}) + P\mathbb{Z}^n$ where P is the negacyclic matrix corresponding to $(X^m - b)$, and $\mathtt{Ecd}(\vec{x})$ is regarded as a coefficient vector over $\mathbb{Z}^n$. Then, $\boldsymbol{x} = \mathtt{Ecd}(\vec{x}) \pmod{X^m - b}$ holds, and we can prove that $\boldsymbol{y} + \boldsymbol{c} \cdot \boldsymbol{x}$ is statistically close to the distribution $\boldsymbol{y} + \boldsymbol{c} \cdot \boldsymbol{x}'$ for $\boldsymbol{x}' \leftarrow \mathcal{D}_{P\mathbb{Z}^n,\mathfrak{s}_1}$ if $\boldsymbol{y}$ is sampled from $\mathcal{D}_{\mathbb{Z}^n,\mathfrak{s}_2}$.

Note that the latter distribution does not contain any information about $\vec{x}$, making it independent of $\vec{x}$. Thus, by randomizing messages, we can achieve simulatability in transcripts, which results in the complete elimination of rejection sampling from our proof system.

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 $\|$ for the concatenation of two vectors.

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

For a vector $\vec{v} = (v_0, \ldots, v_{m-1}) \in \mathbb{Z}^m$, the ℓ^p $(p \geq 1)$ and ℓ^∞ norms are defined as follows:

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

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

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 $\rho : \mathbb{R}^n \to (0,1]$ 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 $\rho_{\sqrt{\Sigma}} : \mathbb{R}^n \to (0,1]$ 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 σ in the subscript and refer to σ as the width parameter of $\mathcal{D}_{\vec{c}+\Lambda,\sigma}$. For a polynomial $\boldsymbol{x}$, we denote by $\boldsymbol{x} \leftarrow \mathcal{D}_{\vec{c}+\Lambda,\sqrt{\Sigma}}$ if we sample its coefficient vector from $\mathcal{D}_{\vec{c}+\Lambda,\sqrt{\Sigma}}$.

2.3 Useful Lemmas

Definition 1 ([30, Def. 3.1]). *For an n-dimensional lattice Λ and positive real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^* \backslash \{0\}) \leq \varepsilon$.*

Definition 2 ([32, Def. 2.3]). *Let Σ 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 ([30, Lem. 3.3]). *For any n-dimensional lattice Λ 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 $\dim(\mathrm{span}(\Lambda \cap \mathcal{B}(r))) = n$ and $\mathcal{B}(r)$ is the n-dimensional ball with radius r centered at the origin.

Lemma 2 ([22, Lem. 5]). *For a positive-definite matrix Σ, $\sqrt{\Sigma} \geq \eta_\varepsilon(\Lambda)$ holds if $\left\|\Sigma^{-1}\right\|_2 \leq \eta_\varepsilon(\Lambda)^{-2}$.*

Lemma 3 ([32, Lem. 2.4]). *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 ([6, Lem. 2.4]). *Let $\Lambda \subseteq \mathbb{R}^n$ be any n-dimensional lattice. For any $0 < \varepsilon < 1/3$, $\sigma \geq \eta_\varepsilon(\Lambda)$, and any $\vec{c} \in \mathbb{R}^n$,*

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

Lemma 5 ([30, Lem. 4.4]). *Let $\Lambda \subseteq \mathbb{R}^n$ be any n-dimensional lattice. For any $0 < \varepsilon < 1/3$, $\sigma \geq \eta_\varepsilon(\Lambda)$, and any $\vec{c} \in \mathbb{R}^n$,*

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

Lemma 6 ([32, Fact 2.1]). *Let $\Sigma_1, \Sigma_2 > 0$ be positive-definitive matrices, let $\Sigma_0^{-1} = \Sigma_1^{-1} + \Sigma_2^{-1} > 0$ 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 ([10, Lem. 3.1]). *Let n be a power of two, and let $0 \leq i, j < 2n$ such that $i \neq j$. Then, $2(X^i - X^j)^{-1}$ is an element of R such that*

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

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

Lemma 8. *Let $k > 0$ and $\Lambda \subseteq \mathbb{Z}^n$ be a full-rank lattice. Let $\vec{x}_i \leftarrow \mathcal{D}_{\vec{c}_i + \Lambda, \sqrt{\Sigma}}$ for $0 \leq i < k$ and $\vec{c}_i \in \mathbb{R}^n$. If $\sqrt{\Sigma} \geq \sqrt{2} \cdot \eta_\varepsilon(\Lambda)$ for some $0 < \varepsilon < 1/2$, then the distribution of $\sum_{i=0}^{k-1} \vec{x}_i$ is within statistical distance $4k\varepsilon$ of $\mathcal{D}_{\vec{c} + \Lambda, \sqrt{k\Sigma}}$ where $\vec{c} = \sum_{i=0}^{k-1} \vec{c}_i$.*

Lemma 9 (Schwartz-Zippel Lemma). *Let $\mathbb{F}$ be a finite field, and let $g : \mathbb{F}^m \to \mathbb{F}$ be a non-zero m-variate polynomial of total degree at most d. Then, for $(x_0, \ldots, x_{m-1}) \leftarrow \mathcal{U}(\mathbb{F}^m)$,*

$$\Pr[g(x_0, \ldots, x_{m-1}) = 0] \leq \frac{d}{|\mathbb{F}|}$$

2.4 Module SIS/LWE

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

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

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

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

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

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

2.5 Hint-MLWE

A variant of the MLWE problem known as the *Hint-MLWE* problem has been introduced in recent literature [22, 29]. This variant helps us for proving the simulatability of lattice-based proof systems without relying on the rejection sampling method [23].

Definition 5 (The Hint-MLWE Problem). *Let ν, ℓ, n be positive integers, χ, ψ be distributions over $R^{\nu+\ell}$, and $\boldsymbol{c}_0, \ldots \boldsymbol{c}_{h-1}$ be elements in R. The Hint-MLWE problem, denoted by $\mathsf{HintMLWE}^{\boldsymbol{c}_0,\ldots,\boldsymbol{c}_{h-1}}_{R,\nu,q,\chi,\psi_0,\ldots,\psi_{h-1}}$, asks an adversary $\mathcal{A}$ to distinguish the following two cases:*

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

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

Theorem 1 ([22, Thm 1]). *Let ν, ℓ, n be positive integers and $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{h-1}$ be elements in R. For $\sigma_1, \sigma_{2,0}, \ldots \sigma_{2,h-1} > 0$, let $\sigma > 0$ be a real number defined as $1/\sigma^2 = 2(1/\sigma_1^2 + \sum_{i=0}^{h-1} B_i^2/\sigma_{2,i}^2)$, where B_i's are upper bounds for $\|\boldsymbol{c}_i\|_1$'s. If $\sigma \ge \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ for $0 < \varepsilon \le 1/2$, then there exists an efficient reduction from $\mathsf{MLWE}_{R,\nu,q,\sigma}$ to $\mathsf{HintMLWE}^{\boldsymbol{c}_0,\ldots,\boldsymbol{c}_{h-1}}_{R,\nu,q,\sigma_1,\sigma_{2,0},\ldots,\sigma_{2,h-1}}$ that reduces the advantage by $O(\varepsilon)$.*

2.6 Proof of Knowledge and Simulatability

In this subsection, we review the definition of a secure proof-of-knowledge protocol. The conventional construction regards each input as a fixed value. However, this approach is not sufficient for our protocol to achieve the zero-knowledgeness property. Hence, we follow the definition from recent literature [22], which considers the distributions of inputs together, as described below.

Definition 6. *Let $\mathbf{L}, \mathbf{L}'$ be NP-languages satisfying $\mathbf{L} \subseteq \mathbf{L}'$. Let $\mathbf{R}, \mathbf{R}'$ be witness relations for $\mathbf{L}$ and $\mathbf{L}'$ respectively i.e., $(t \in \mathbf{L} \Leftrightarrow \exists w\ (t, w) \in \mathbf{R})$ and $(t \in \mathbf{L}' \Leftrightarrow \exists w'\ (t, w') \in \mathbf{R}')$. Let $(\mathcal{P}, \mathcal{V})$ be an interactive protocol where $\mathcal{P}$ takes a secret input m and a public parameter pp as input, and $\mathcal{V}$ only takes a public parameter pp as input. Then $(\mathcal{P}, \mathcal{V})$ is called a secure proof-of-knowledge protocol for the languages $(\mathbf{L}, \mathbf{L}')$ if and only if it satisfies the followings:*

- ***Two Phases:*** *The protocol consists of the following phases.*
  - ***Generate-phase:*** *In generate-phase, the prover first samples randomness r, and then generates a statement t with x and r. At the end of the phase, it sends the statement t to the verifier $\mathcal{V}$.*

- ***Prove-phase:*** *In prove-phase, the prover and the verifier take* (pp, t, x, r) *and* (pp, t) *as input respectively. Then, they interact each other to prove that* $t \in \mathbf{L}'$*. At the end of the phase, the verifier outputs either* 0 *or* 1*.*

We refer the sequence of messages exchanged between $\mathcal{P}$ *and* $\mathcal{V}$ *during the generate-phase and the prove-phase as the transcript, and denote it by* $\mathtt{Tr}(\mathcal{P}(\mathsf{pp}, x), \mathcal{V}(\mathsf{pp}))$*.*

- ***Completeness:*** *If* $\mathcal{P}$ *generates a statement* $t \in \mathbf{L}$ *in the generate-phase, the prove-phase ends with* 1 *except for negligible probability.*
- ***Knowledge Soundness:*** *If there exists an adversarial prover* $\mathcal{P}^*$ *which makes the verifier outputs* 1 *at the prove-phase with non-negligible probability, then there exists an efficient algorithm* $\mathcal{E}$*, called an extractor, which, given black-box access to* $\mathcal{P}^*$*, outputs* w' *such that* $(t, w') \in \mathbf{R}'$ *with non-negligible probability.*
- ***Simulatability:*** *There exists a PPT algorithm* $\mathcal{S}$*, called a simulator, whose input is* pp *and output is* $\mathtt{tr}$ *which is computationally indistinguishable from the transcript from the honest prover* $\mathcal{P}$ *and verifier* $\mathcal{V}$*, for any secret input* x*. In other words, for all PPT algorithm* $\mathcal{A}$*, the following advantage is negligible:*

$$\left|\Pr\left[b = 1 \middle| \begin{smallmatrix} x \leftarrow \mathcal{A}(\mathsf{pp});\ \mathtt{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{pp},x),\mathcal{V}(\mathsf{pp}));\\ b \leftarrow \mathcal{A}(\mathsf{pp},\mathtt{tr}) \end{smallmatrix}\right] - \Pr\left[b = 1 \middle| \begin{smallmatrix} x \leftarrow \mathcal{A}(\mathsf{pp});\ \mathtt{tr} \leftarrow \mathcal{S}(\mathsf{pp});\\ b \leftarrow \mathcal{A}(\mathsf{pp},\mathtt{tr}); \end{smallmatrix}\right]\right|$$

In this definition, the zero-knowledge condition is reformulated by the simulatability of transcripts without knowledge of the prover's secret input. The main distinction between the simulatability property and the conventional zero-knowledge proof is whether input randomness is considered together or not. Since the main objective of a secure proof-of-knowledge protocol is to conceal the prover's secret input, it suffices to satisfy this simulatability property for the desired security requirement.

The definition utilizes two languages, $\mathbf{L} \subseteq \mathbf{L}'$, referred to as the honest and proven languages, respectively, to address common scenarios in lattice-based constructions. Previous studies, such as [10,28], have reduced the communication cost by weakening the extractors' power in the knowledge soundness property, and our instantiations of proof-of-knowledge in this paper also employ these methods. The difference between $\mathbf{L}$ and $\mathbf{L}'$ is often referred to as the *soundness slack*.

2.7 Lattice-based Commitment Scheme

We recall the definition of commitment scheme.

Definition 7 (Commitment Scheme). *A commitment scheme consists of the following three algorithms:*

- $\mathtt{Gen}(1^\lambda)$*: Given a security parameter* λ*, it generates a commitment key* ck*.*
- $\mathtt{Com}_{\mathsf{ck}}(m; \mu)$*: Given a commitment key* ck*, a message* m*, and randomness* μ*, it outputs a commitment* m^**.*

– $\mathtt{Open}_{\mathsf{ck}}(m^*, m, \mu)$*: Given a commitment* m^**, a message* m*, and randomness* μ*, it outputs either 0 or 1.*

where $\mathtt{Gen}$ *is probabilistic and* $\mathtt{Com}, \mathtt{Open}$ *are deterministic. Let* $\mathcal{R}$ *be a distribution for randomness. Then a commitment scheme* $(\mathtt{Gen}, \mathtt{Com}, \mathtt{Open})$ *is said to be secure if it satisfies the following properties:*

– ***Hiding****: For all PPT adversaries* $\mathcal{A}$*, the following advantage is negligible:*

$$\left| \Pr\left[b = b' \;\middle|\; \begin{array}{c} \mathsf{ck} \leftarrow \mathtt{Gen}(1^\lambda); (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{ck}); \mu \leftarrow \mathcal{R}; \\ b \leftarrow \mathcal{U}(\{0,1\}); m^* = \mathtt{Com}_{\mathsf{ck}}(m_b; \mu); b' \leftarrow \mathcal{A}(\mathsf{ck}, m^*); \end{array} \right] - \frac{1}{2} \right|.$$

– ***Binding****: For all PPT adversaries* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[\begin{array}{c} \mathtt{Open}_{\mathsf{ck}}(m^*, m_0, \mu_0) = \mathtt{Open}_{\mathsf{ck}}(m^*, m_1, \mu_1) = 1 \\ \wedge\ m_0 \neq m_1 \end{array} \;\middle|\; \begin{array}{c} \mathsf{ck} \leftarrow \mathtt{Gen}(1^\lambda); \\ (m^*, m_0, m_1, \mu_0, \mu_1) \leftarrow \mathcal{A}(\mathsf{ck}) \end{array} \right].$$

Below, we present the Ajtai commitment scheme [2], whose binding and hiding properties rely on the hardness of $\mathsf{MSIS}_{R,\mu,q,2\beta}$ and $\mathsf{MLWE}_{R,\nu,q,\chi}$, respectively, where χ is a distribution for commitment randomness.

• $\underline{\mathtt{Gen}(1^\lambda)}$: Given a security parameter λ, it outputs a commitment key $\mathsf{ck} = (\mathbf{A}_0, \mathbf{A}_1)$ where $\mathbf{A}_0 \leftarrow \mathcal{U}(R_q^{\mu \times \ell})$, and $\mathbf{A}_1 \leftarrow \mathcal{U}(R_q^{\mu \times \nu})$.

• $\underline{\mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}; \vec{\boldsymbol{\mu}})}$: Given a commitment key $\mathsf{ck} = (\mathbf{A}_0, \mathbf{A}_1)$, a message $\vec{\boldsymbol{m}} \in R^\ell$, and randomness $\vec{\boldsymbol{\mu}} \in R^{\mu+\nu}$, it outputs $\vec{\boldsymbol{m}}^* := \mathbf{A}_0 \vec{\boldsymbol{m}} + [\mathbf{A}_1 | \mathbf{I}_\mu] \vec{\boldsymbol{\mu}} \in R_q^\mu$.

• $\underline{\mathtt{Open}_{\mathsf{ck}}(\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}})}$: Given a commitment $\vec{\boldsymbol{m}}^*$, a message $\vec{\boldsymbol{m}}$, and randomness $\vec{\boldsymbol{\mu}}$, it outputs 1 if and only if $\vec{\boldsymbol{m}}^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}; \vec{\boldsymbol{\mu}})$ and $\|\vec{\boldsymbol{m}} \| \vec{\boldsymbol{\mu}}\|_2 < \beta$.

We note that the above commitment scheme satisfies the additive homomorphism for both message and randomness *i.e.,* for any $\vec{\boldsymbol{m}}_0, \vec{\boldsymbol{m}}_1 \in R^\ell$, $\vec{\boldsymbol{\mu}}_0, \vec{\boldsymbol{\mu}}_1 \in R^{\mu+\nu}$, and $\boldsymbol{c} \in R$, it holds that

$$\mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}_0; \vec{\boldsymbol{\mu}}_0) + \boldsymbol{c} \cdot \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}_1; \vec{\boldsymbol{\mu}}_1) = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}_0 + \boldsymbol{c} \cdot \vec{\boldsymbol{m}}_1; \vec{\boldsymbol{\mu}}_0 + \boldsymbol{c} \cdot \vec{\boldsymbol{\mu}}_1)$$

2.8 Rank-one Constraint System

A rank-one constraint system (R1CS) over a finite field $\mathbb{F}$ is a problem of finding a solution vector $\vec{s} \in \mathbb{F}^M$ to the equation $A\vec{s} \circ B\vec{s} = C\vec{s}$, where the first entry of $\vec{s}$ is fixed to one, and $A, B, C \in \mathbb{F}^{M \times M}$ are matrices that contains $\Omega(M)$ non-zero entries. Below, we give another formulation of R1CS which will be used throughout this paper. We also denote by $N = \Omega(M)$ the total number of non-zero entries in the three matrices A, B, C.

Definition 8 (R1CS problem). *Given matrices* $A, B, C \in \mathbb{F}^{M \times M}$ *each containing* $\Omega(M)$ *nonzero entries, and vectors* $\vec{a}, \vec{b}, \vec{c} \in \mathbb{F}^M$*, the R1CS problem asks to find any solution vector* $\vec{t} \in \mathbb{F}^M$ *which satisfies:*

$$(A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) = (C\vec{t} + \vec{c})$$

where $\circ$ *denotes the component-wise product.*

3 Revisiting Amortized Proofs of Opening Knowledge

In this section, we revisit the amortized proof of opening knowledge (POK) protocols for the Ajtai commitment scheme. The previous construction by Baum et al. [7] achieved a sublinear proof size by leveraging the core property of the Ajtai commitment scheme, where the dimension of commitment is almost independent of the dimension of the input. However, as we mentioned before, it has two main bottlenecks for practical usage: restrictions in message space and rejection sampling.

In our new construction, we address all these issues, making the amortized POK protocol for the Ajtai scheme more practical. To handle the challenge of a large message modulus, we devise a novel encoding method that converts a vector of elements in $\mathbb{Z}_p$ with a large prime p into a small cyclotomic ring element, inspired from recent literature by Chen et al. [16,17]. Using this method, we can commit messages with a large modulus p without introducing large commitment modulus q.

To eliminate rejection sampling, we also devise a new proof technique that randomizes messages. In most applications of sublinear arguments, it proves knowledge of solutions for arithmetic relations on $\mathbb{Z}_p$. Thus, a prover can commit to a message m' instead of m if $m' = m \pmod p$. We focus on this redundancy and leverage it for randomizing messages. This randomization in the message helps us apply the technique from [22] to the Ajtai commitment scheme, eliminating rejection sampling from the POK protocol. In the rest of this section, we delve into the technical details of our improvements.

3.1 Encoding Methods for Large Prime Fields

We first explain how we adapt the encoding method proposed by Chen et al. [16,17] for our own purposes. Initially, their encoding method was proposed to instantiate high-precision arithmetic for lattice-based homomorphic encryption without increasing the ciphertext modulus. Since we aim for a similar goal, committing large prime field elements without increasing the commitment modulus, we modify their encoding method to support large prime field arithmetic. Below, we present the key algebraic property that enables such an encoding /method.

Lemma 10. *Let b, m, and κ be positive integers such that $\kappa = n/m$ is an integer and $p = b^\kappa + 1$ is a prime. Then, $R/(X^m - b)$ is isomorphic to $\mathbb{Z}_p^m$ as a $\mathbb{Z}$-module.*

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

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

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

$$\varphi : \bar{\boldsymbol{a}} = \sum_{i=0}^{m-1}\sum_{j=0}^{\kappa-1} a_{mj+i}X^{mj+i} \mapsto \left(\sum_{j=0}^{\kappa-1} a_{mj}b^j, \sum_{j=0}^{\kappa-1} a_{mj+1}b^j, \ldots, \sum_{j=0}^{\kappa-1} a_{mj+m-1}b^j\right)$$

□

The isomorphism φ is defined over the polynomial representation on R of an element $\bar{\boldsymbol{a}}$ in $R/(X^m-b)$. It is well-defined, because for any two polynomial representations $\boldsymbol{a}$ and $\boldsymbol{a}'$ in R of $\bar{\boldsymbol{a}}$, we have $\boldsymbol{a} = \boldsymbol{a}' \pmod{X^m-b}$ and φ maps $X^m - b$ to zero. As for the inverse map φ^{-1}, which corresponds to encoding of messages in $\mathbb{Z}_p^m$, we aim for its polynomial representation in R to have a small norm. To achieve this, we present an algorithm in Alg. 1 that outputs a polynomial representation $\boldsymbol{a}$ of $\varphi^{-1}(a)$ with an upper bound $\|\boldsymbol{a}\|_\infty \leq \frac{b+2}{2}$ since $|a_{i,j}| \leq b/2$ and $|c_{i,j}| \leq 1$. Below, we provide encoding and decoding algorithms for the large prime field $\mathbb{Z}_p$.

Algorithm 1 Encoding

Input: $\vec{a} = (a_0, \ldots, a_{m-1}) \in \mathbb{Z}_p^m$
Output: $\boldsymbol{a} \in R$

for $0 \leq i < m$ **do**
  if $a_i = p - 1 \pmod p$ **then**
    $(a_{i,0}, \ldots, a_{i,\kappa-1}) \leftarrow (0, \ldots, 0, b)$
  else
    $(a_{i,0}, \ldots, a_{i,\kappa-1})$ is the base-b representation of $0 \leq a_i < b^\kappa$
  end if
end for
for $0 \leq i < m$, $0 \leq j \leq \kappa$ **do**
  if $a_{i,j} > b/2$ **then**
    $a_{i,j} \leftarrow a_{i,j} - b, \quad c_{i,j} \leftarrow 1$
  else
    $c_{i,j} \leftarrow 0$
  end if
end for
$\boldsymbol{a} \leftarrow \sum_{i=0}^{m-1}\sum_{j=0}^{\kappa-1} a_{i,j}X^{mi+j} + \sum_{i=0}^{m-1}\sum_{j=0}^{\kappa-1} c_{i,j}X^{m(i+1)+j}$

• $\underline{\texttt{Ecd}(\vec{a})}$: Given an element $\vec{a} = (a_0, \ldots, a_{m-1}) \in \mathbb{Z}_p^m$, run Alg. 1 and output a ring element $\boldsymbol{a}$ where $\|\boldsymbol{a}\|_\infty \leq \frac{b+2}{2}$.

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, $\texttt{Ecd}(a)$ outputs a ring element $\boldsymbol{a} = \texttt{Ecd}(\vec{a})$,

where $\vec{a} = (a, 0, \ldots, 0)$ such that $\|\boldsymbol{a}\|_1 \leq \frac{(b+2)\kappa}{2}$ since there are at most κ non-zero coefficients.

• $\underline{\mathtt{Dcd}(\boldsymbol{a})}$: Given a ring element $\boldsymbol{a} = \sum_{i=0}^{m-1}\sum_{j=0}^{\kappa-1} a_{i,j}X^{i+mj} \in R$, output $\vec{a} = \varphi(\boldsymbol{a}) = \big(\sum_{j=0}^{\kappa-1} a_{0,j}b^j, \ldots, \sum_{j=0}^{\kappa-1} a_{m-1,j}b^j\big) \in \mathbb{Z}_p^m$.

3.2 Randomized Encoding

Now, we present a randomization procedure for messages, which helps us achieve the simulatability of the POK protocol without using the rejection sampling method. We focus on the fact that the decoding algorithm outputs the same results unless inputs are congruent modulo $X^m - b$—i.e., it holds that $\mathtt{Dcd}(\boldsymbol{a}) = \mathtt{Dcd}(\boldsymbol{a}')$ if $\boldsymbol{a} = \boldsymbol{a}' \pmod{X^m - b}$. Since our objective is to construct sublinear arguments for arithmetic circuits, the verification for the arguments remains valid if the decoding algorithm outputs identical results. Therefore, we aim to incorporate a randomization procedure during the message encoding process. To be precise, a randomized encoding procedure samples a ring element $\boldsymbol{m}$ such that $\boldsymbol{m} = \mathtt{Ecd}(\vec{m}) \pmod{X^m - b}$, where $\vec{m} \in \mathbb{Z}_p^m$ is an input message. Then, it remains to specify the distribution from which $\boldsymbol{m}$ is sampled.

To proceed, we recall that the purpose of randomized encoding is for the simulatability of the POK protocol. Hence, we first analyze the problematic part of the transcripts of the protocol. During the protocol, a prover sends a response of the form $\boldsymbol{y} + \boldsymbol{c} \cdot \mathtt{Ecd}(\vec{m})$, where $\boldsymbol{y}$ is a mask for $\mathtt{Ecd}(\vec{m})$ that a prover samples, and $\boldsymbol{c}$ is a random challenge sampled from a verifier. Without careful treatment, such as the rejection sampling method, the response $\boldsymbol{y} + \boldsymbol{c} \cdot \mathtt{Ecd}(\vec{m})$ includes partial information on $\vec{m}$, which inhibits simulating the transcript without witness knowledge. Our aim is to replace $\mathtt{Ecd}(\vec{m})$ with $\boldsymbol{m}$ so that $\boldsymbol{y} + \boldsymbol{c} \cdot \boldsymbol{m}$ follows a distribution that is independent of $\mathtt{Ecd}(\vec{m})$. For this purpose, we propose the following lemma, inspired by the convolution lemma for the discrete Gaussian distribution [32].

Lemma 11. *Let $h > 0$ be an integer, $\mathfrak{S} \in \mathbb{R}^{n\times n}$ be a positive-definitive matrix, $\mathfrak{s}' > 0$ be positive reals, and $\Lambda \subseteq \mathbb{Z}^n$ be an n-dimensional lattice. Given $\boldsymbol{c}_0, \ldots, \boldsymbol{c}_{h-1} \in R$, let $C_i \in \mathbb{R}^{n\times n}$ be the negacyclic matrix corresponding to $\boldsymbol{c}_i$ for $0 \leq i < h$. Let $\mathfrak{S}_h > 0$ be a positive-definitive matrices such that $\mathfrak{S}_h^{-1} = \mathfrak{S}^{-1} + \sum_{i=0}^{h-1} \mathfrak{s}_i'^{-2} \cdot C_i^\top C_i$. If $\sqrt{\mathfrak{S}_h} \geq \eta_\varepsilon(\Lambda)$ for some $0 < \varepsilon < 1/2$, then for any $\vec{u}, \vec{u}' \in \mathbb{Z}^n$, the following two distributions are within a statistical distance of 2ε.*

$$\left\{(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{h-1}) \;\middle|\; \boldsymbol{x} \leftarrow \mathcal{D}_{\vec{u}+\Lambda,\sqrt{\mathfrak{S}}},\ \boldsymbol{y}_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\mathfrak{s}_i'},\ \boldsymbol{z}_i = \boldsymbol{c}_i \cdot \boldsymbol{x} + \boldsymbol{y}_i\right\}$$

$$\left\{(\boldsymbol{z}_0', \ldots, \boldsymbol{z}_{k-1}') \;\middle|\; \boldsymbol{x}' \leftarrow \mathcal{D}_{\vec{u}'+\Lambda,\sqrt{\mathfrak{S}}},\ \boldsymbol{y}_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\mathfrak{s}_i'},\ \boldsymbol{z}_i' = \boldsymbol{c}_i \cdot \boldsymbol{x}' + \boldsymbol{y}_i\right\}$$

Proof. Since $\vec{u}, \vec{u}' \in \mathbb{Z}^n$ and $\Lambda \subseteq \mathbb{Z}^n$, the sample spaces of the above two distributions are identical to R^h. Thus, it is enough to show that

$$\left| \Pr[\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}}] - \Pr[\vec{\boldsymbol{z}}' = \vec{\boldsymbol{w}}] \right| \leq 2\varepsilon \cdot \Pr[\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}}] \tag{1}$$

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

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

where $\vec{v}$ and $\vec{w}_i$ are the coefficient vectors of $\boldsymbol{v}$ and $\boldsymbol{w}_i$ respectively. Let $\mathfrak{S}_i = \mathfrak{S}^{-1} + \sum_{j=0}^{i-1} \mathfrak{s}_j'^{-2} \cdot C_j^\top C_j$ for $0 \leq i < h$. Then, we have the following by Lem. 6

$$\rho_{\sqrt{\mathfrak{S}_i}}(\vec{v} - \vec{a}_i) \cdot \rho_{\mathfrak{s}'_i \cdot C_i^{-1}}(\vec{v} - C_i^{-1}\vec{w}_i) = \rho_{\sqrt{\mathfrak{S}_{i+1}}}(\vec{v} - \vec{a}_{i+1}) \cdot \rho_{\sqrt{\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}}}(\vec{b}_{i+1})$$

where $\vec{a}_0 = 0$, $\vec{b}_0 = C_0^{-1}\vec{w}_0$ and, $\vec{a}_{i+1} = \left(\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}\right)\left(\mathfrak{S}_i^{-1}\vec{a}_i + \mathfrak{s}_i'^{-2} C_i^\top \vec{w}_i\right)$, 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

$$\Pr[\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}}] = \prod_{i=0}^{h-1} \rho_{\sqrt{\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}}}(\vec{b}_{i+1}) \cdot \sum_{\vec{v} \in \vec{u} + \Lambda} \rho_{\sqrt{\mathfrak{S}_h}}(\vec{v} - \vec{a}_h).$$

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

$$\Pr[\vec{\boldsymbol{z}}' = \vec{\boldsymbol{w}}] = \prod_{i=0}^{h-1} \rho_{\sqrt{\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}}}(\vec{b}_{i+1}) \cdot \sum_{\vec{v} \in \vec{u}' + \Lambda} \rho_{\sqrt{\mathfrak{S}_h}}(\vec{v} - \vec{a}_h).$$

Hence, if $\sqrt{\mathfrak{S}_h} \geq \eta_\varepsilon(\Lambda)$ for $0 < \varepsilon < 1/2$, we obtain the followings by Lem. 3

$$\begin{aligned} & \left| \Pr[\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}}] - \Pr[\vec{\boldsymbol{z}}' = \vec{\boldsymbol{w}}] \right| \\ = \; & \prod_{i=0}^{h-1} \rho_{\sqrt{\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}}}(\vec{b}_{i+1}) \cdot \left| \rho_{\sqrt{\mathfrak{S}_h}}(\vec{u} - \vec{a}_h + \Lambda) - \rho_{\sqrt{\mathfrak{S}_h}}(\vec{u}' - \vec{a}_h + \Lambda) \right| \\ \leq \; & 2\varepsilon \cdot \prod_{i=0}^{h-1} \rho_{\sqrt{\mathfrak{S}_i + \mathfrak{s}_i'^2 (C_i^\top C_i)^{-1}}}(\vec{b}_{i+1}) \cdot \rho_{\sqrt{\mathfrak{S}_h}}(\vec{u} - \vec{a}_h + \Lambda) = 2\varepsilon \cdot \Pr[\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}}]. \end{aligned}$$

Therefore, the two given distributions are within a statistical distance of 2ε. □

The above lemma essentially states that if a prover samples a mask $\boldsymbol{y}$ from a discrete Gaussian distribution over an n-dimensional integer lattice $\mathbb{Z}^n$, and a randomized encoding $\boldsymbol{m}$ of a message $\vec{m}$ follows a discrete Gaussian distribution

over a coset $\mathtt{Ecd}(\vec{m})+\Lambda$ of some lattice $\Lambda \subseteq \mathbb{Z}^n$, then $\boldsymbol{y}+\boldsymbol{c}\cdot\boldsymbol{m}$ is statistically close to a distribution that is independent of $\vec{m}$. Since our goal is to have $\boldsymbol{m} = \mathtt{Ecd}(\vec{m})$ (mod $X^m - b$), we can set $\Lambda = P\mathbb{Z}^n$, where $P \in \mathbb{R}^{n\times n}$ is the negacyclic matrix corresponding to $X^m - b$, then it satisfies all the required conditions. Below, we present our randomized encoding algorithm.

• $\underline{\mathtt{R.Ecd}(a;\mathfrak{s})}$: Given an element $\vec{a} \in \mathbb{Z}_p^m$ and a positive real $\mathfrak{s} > 0$, output a ring element $\boldsymbol{a} \leftarrow \mathcal{D}_{\mathtt{Ecd}(\vec{a})+P\mathbb{Z}^n,\mathfrak{s}P}$ where $P \in \mathbb{R}^{n\times n}$ is the negacyclic matrix of $X^m - b$.

As an abuse of notation, we often put an ℓm-dimensional vector $\vec{a} = \vec{a}_0\| \cdots \|\vec{a}_{\ell-1} \in \mathbb{Z}_p^{\ell m}$ as an input for the above algorithms. In this case, $\mathtt{R.Ecd}(\vec{a})$ outputs a vector of ring element $\vec{\boldsymbol{a}} = (\boldsymbol{a}_0, \ldots, \boldsymbol{a}_{\ell-1}) \in R^\ell$ where $\boldsymbol{a}_i = \mathtt{R.Ecd}(\vec{a}_i;\mathfrak{s})$ for $0 \le i < \ell$.

To analyze an upper bound for the randomized encoding, we utilize the fact that $\mathcal{D}_{\mathtt{Ecd}(\vec{a})+P\mathbb{Z}^n,\mathfrak{s}P} = P \cdot \mathcal{D}_{P^{-1}\mathtt{Ecd}(\vec{a})+\mathbb{Z}^n,\mathfrak{s}}$. Then, we can apply the tail bounds from Lem. 4 and 5 to $\mathcal{D}_{P^{-1}\mathtt{Ecd}(\vec{a})+\mathbb{Z}^n,\mathfrak{s}}$. Additionally, we note that P^{-1} corresponds to the negacyclic matrix of the polynomial $(X^m-b)^{-1} = -\frac{1}{b^\kappa+1}\big(X^{n-m} + bX^{n-2m} + \cdots + b^{\kappa-1}\big)$. Then, it holds that $\|P\|_2 \le \|X^m - b\|_1 = b+1$ and $\big\|P^{-1}\big\|_2 \le \big\|(X^m-b)^{-1}\big\|_1 = \frac{b^\kappa-1}{(b-1)(b^\kappa+1)} \le \frac{1}{b-1}$.

3.3 New Proof of Opening Knowledge Protocol

Incorporating the aforementioned randomized encoding method for large prime fields, we present an improved version of the amortized POK protocol for the Ajtai commitment scheme, which achieves the binding property with small parameter overheads and simulatability without the rejection sampling method. We first define the witness relations for POK as follows:

$$\mathbf{R}_{\mathtt{Open}} := \{(\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \mid \vec{\boldsymbol{m}}^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \wedge \|\vec{\boldsymbol{m}}\|\vec{\boldsymbol{\mu}}\|_2 < \beta_{\mathtt{Open}}\}$$
$$\mathbf{R}'_{\mathtt{Open}} := \{(\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \mid 2\vec{\boldsymbol{m}}^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \wedge \|\vec{\boldsymbol{m}}\|\vec{\boldsymbol{\mu}}\|_2 < 2n \cdot \beta_{\mathtt{Open}}\}$$

where $\mathsf{ck} \leftarrow \mathtt{Gen}(1^\lambda)$. Then, $(\vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}})$ can be viewed as a witness for the statement about $\vec{\boldsymbol{m}}^*$. The honest language $\mathbf{L}_{\mathtt{Open}}$ and the proven language $\mathbf{L}'_{\mathtt{Open}}$ are defined as follows:

$$\mathbf{L}_{\mathtt{Open}} = \big\{\vec{\boldsymbol{m}}^* \in R_q^\mu \mid \exists(\vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \in R^\ell \times R^{\mu+\nu} \text{ s.t. } (\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \in \mathbf{R}_{\mathtt{Open}}\big\}$$
$$\mathbf{L}'_{\mathtt{Open}} = \big\{\vec{\boldsymbol{m}}^* \in R_q^\mu \mid \exists(\vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \in R^\ell \times R^{\mu+\nu} \text{ s.t. } (\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \in \mathbf{R}'_{\mathtt{Open}}\big\}$$

We note that $\mathbf{L}_{\mathtt{Open}} \subseteq \mathbf{L}'_{\mathtt{Open}}$ since $(\vec{\boldsymbol{m}}^*, \vec{\boldsymbol{m}}, \vec{\boldsymbol{\mu}}) \in \mathbf{R}_{\mathtt{Open}}$ implies $(\vec{\boldsymbol{m}}^*, 2\vec{\boldsymbol{m}}, 2\vec{\boldsymbol{\mu}}) \in \mathbf{R}'_{\mathtt{Open}}$.

We present the amortized POK protocol $\Pi_{\mathtt{Open}}$ in Fig. 1, the verification procedure $\mathcal{V}_{\mathtt{Open}}$ in Fig. 2, and the simulator $\mathcal{S}_{\mathtt{Open}}$ in Fig. 3.

$\Pi_{\mathtt{Open}}(\vec{\boldsymbol{m}}_0^*, \ldots, \vec{\boldsymbol{m}}_{k'-1}^*)$

	Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
	Input: $\mathsf{ck} \in R_q^{\mu \times (\mu+\nu+\ell)}$		ck
	$\vec{\boldsymbol{m}}_0, \ldots, \vec{\boldsymbol{m}}_{k'-1} \in R^{\ell}$		
	$\vec{\boldsymbol{\mu}}_0, \ldots, \vec{\boldsymbol{\mu}}_{k'-1} \in R^{\mu+\nu}$		
1 :	For $0 \le j < r$,		
2 :	$\vec{\boldsymbol{n}}^{(j)} \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \sqrt{k'}\mathfrak{s}_2}, \vec{\boldsymbol{\nu}}^{(j)} \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{k'}\sigma_2}$		
3 :	$\vec{\boldsymbol{n}}^{*(j)} = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{n}}^{(j)}; \vec{\boldsymbol{\nu}}^{(j)})$	$\xrightarrow{\vec{\boldsymbol{n}}^{*(j)}}$	
4 :			For $0 \le j < r$,
5 :		$\xleftarrow{\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{k'-1}^{(j)}}$	$\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{k'-1}^{(j)} \leftarrow \mathcal{U}(C)$
6 :	For $0 \le j < r$,		
7 :	$\vec{\boldsymbol{m}}^{(j)} = \vec{\boldsymbol{n}}^{(j)} + \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i$		
8 :	$\vec{\boldsymbol{\mu}}^{(j)} = \vec{\boldsymbol{\nu}}^{(j)} + \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\mu}}_i$	$\xrightarrow{\vec{\boldsymbol{m}}^{(j)}, \vec{\boldsymbol{\mu}}^{(j)}}$	
9 :			Run $\mathcal{V}_{\mathtt{Open}}$

Fig. 1. Amortized POK protocol for the Ajtai commitment scheme

$\mathcal{V}_{\mathtt{Open}}$

1 :	For $0 \le j < r$,
2 :	$\left\|\vec{\boldsymbol{m}}^{(j)}\right\|_2 \overset{?}{<} ((b+1)\mathfrak{s}_1 + \mathfrak{s}_2) \cdot \sqrt{k'\ell n}$
3 :	$\left\|\vec{\boldsymbol{m}}^{(j)}\right\|_\infty \overset{?}{<} 5((b+1)\mathfrak{s}_1 + \mathfrak{s}_2) \cdot \sqrt{k'}$
4 :	$\left\|\vec{\boldsymbol{\mu}}^{(j)}\right\|_2 \overset{?}{<} (\sigma_1 + \sigma_2) \cdot \sqrt{k'(\mu+\nu)n}$
5 :	$\left\|\vec{\boldsymbol{\mu}}^{(j)}\right\|_\infty \overset{?}{<} 5(\sigma_1 + \sigma_2) \cdot \sqrt{k'}$
6 :	$\mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}^{(j)}, \vec{\boldsymbol{\mu}}^{(j)}) \overset{?}{=} \vec{\boldsymbol{n}}^{*(j)} + \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i^* \pmod q$

Fig. 2. Verification procedure for $\Pi_{\mathtt{Open}}$

$\mathcal{S}_{\mathtt{Open}}(\mathsf{ck})$

1 : $\vec{\boldsymbol{m}}_i'^{*} \leftarrow \mathcal{U}(R_q^{\mu})$ for $0 \le i < k'$

2 : $\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{k'-1}^{(j)} \leftarrow \mathcal{U}(C)$ for $0 \le j < r$

3 : $\vec{\boldsymbol{n}}^{(j)} \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \sqrt{k'}\mathfrak{s}_2}, \vec{\boldsymbol{\nu}}^{(j)} \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{k'}\sigma_2}$ for $0 \le j < r$

4 : $\vec{\boldsymbol{m}}_i' \leftarrow \mathtt{R.Ecd}(0, \mathfrak{s}_1),\ \vec{\boldsymbol{\mu}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ for $0 \le i < k'$

5 : $\vec{\boldsymbol{m}}'^{(j)} := \vec{\boldsymbol{n}}^{(j)} + \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i',\ \vec{\boldsymbol{\mu}}^{(j)} := \vec{\boldsymbol{\nu}}^{(j)} + \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\mu}}_i$ for $0 \le j < r$

6 : $\vec{\boldsymbol{n}}^{*(j)} := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}'^{(j)}; \vec{\boldsymbol{\mu}}^{(j)}) - \sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i'^{*}$ for $0 \le j < r$

7 : Output $\{\vec{\boldsymbol{m}}_i'^{*}\}_{i=0}^{k'-1}, \{\vec{\boldsymbol{n}}^{*(j)}\}_{j=0}^{r-1}, \{(\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{k'-1}^{(j)})\}_{j=0}^{r-1}, \{\vec{\boldsymbol{m}}'^{(j)}\}_{j=0}^{r-1}, \{\vec{\boldsymbol{\mu}}^{(j)}\}_{j=0}^{r-1}$

Fig. 3. Simulator for $\Pi_{\mathtt{Open}}$

Below, we prove that $\Pi_{\mathtt{Open}}$ is a secure POK protocol for the languages $(\mathbf{L}_{\mathtt{Open}}, \mathbf{L}'_{\mathtt{Open}})$.

Theorem 2. *Let* $\mathsf{ck} \leftarrow \mathtt{Gen}(1^{\lambda})$, $C = \{1, X, \ldots, X^{2n-1}\}$ *and* $r \ge (\lambda+2)/\log(2n)$. *Let* $\mathfrak{s}_1, \mathfrak{s}_2, \sigma_1, \sigma_2, \sigma, \beta_{\mathtt{Open}}$ *be positive reals satisfying the following conditions for some* $0 < \varepsilon < 2^{-\lambda}$.

- $\mathfrak{s}_1 \ge \sqrt{2} \cdot \frac{b+1}{b-1} \cdot \eta_{\varepsilon}(\mathbb{Z}^n), \quad \mathfrak{s}_2 \ge (b+1)\sqrt{2r} \cdot \eta_{\varepsilon}(\mathbb{Z}^n)$
- $\sigma_1 \ge 2\sqrt{2} \cdot \eta_{\varepsilon}(\mathbb{Z}^n), \quad \sigma_2 \ge 2\sqrt{2r} \cdot \eta_{\varepsilon}(\mathbb{Z}^n),\ \sigma = \frac{1}{\sqrt{2}}(\sigma_1^{-2} + r \cdot \sigma_2^{-2})^{-1/2}$
- $\beta_{\mathtt{Open}} = \big((b+1)\mathfrak{s}_1 + \mathfrak{s}_2\big) \cdot \sqrt{k'\ell n} + (\sigma_1 + \sigma_2) \cdot \sqrt{k'(\mu+\nu)n}$

If $\vec{\boldsymbol{m}}_i \leftarrow \mathtt{R.Ecd}(\vec{m}_i; \mathfrak{s}_1)$ *for some* $\vec{m}_i \in \mathbb{Z}_p^{\ell m}$ *and* $\vec{\boldsymbol{\mu}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ *for* $0 \le i < k'$, *then* $\Pi_{\mathtt{Open}}$ *is a secure proof-of-knowledge protocol for* $(\mathbf{L}_{\mathtt{Open}}, \mathbf{L}'_{\mathtt{Open}})$ *under the hardness assumption of* $\mathsf{MLWE}_{R,\nu,q,\sigma}$ *and* $\mathsf{MSIS}_{R,\mu,q,4n\beta_{\mathtt{Open}}}$.

Proof. We show the completeness, soundness and simulatability of $\Pi_{\mathtt{Open}}$.

Completeness. If the prover is honest, $\vec{\boldsymbol{m}}_i^{*} = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}_i; \vec{\boldsymbol{\mu}}_i)$ holds for $0 \le i < k'$. Consequently, line 6 of $\mathcal{V}_{\mathtt{Open}}$ always holds, so we need to show that the upper-bound check conditions in lines 2 to 5 are valid. Since $\boldsymbol{c}_i^{(j)}$'s are monomials with coefficients equal to 1, $\boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i$ and $\boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\mu}}_i$ follow the distributions $\mathcal{D}_{\boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i + (P\mathbb{Z}^n)^{\ell}, \mathfrak{s}_1 P \otimes I_{\ell}}$ and $\mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$, respectively. Using Lem. 8, the distribution of $\sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \vec{\boldsymbol{m}}_i$ and $\sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \vec{\boldsymbol{\mu}}_i$ are within statistical distance of $O(\varepsilon)$ from $\mathcal{D}_{\sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \vec{\boldsymbol{m}}_i + (P\mathbb{Z}^n)^{\ell}, \sqrt{k'}\mathfrak{s}_1 P \otimes I_{\ell}}$ and $\mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{k'}\sigma_1}$ since $\mathfrak{s}_1, \sigma_1 \ge \sqrt{2} \cdot \eta_{\varepsilon}(\mathbb{Z}^n)$. Applying the tail bounds, presented in Lem. 4, 5, to $\vec{\boldsymbol{n}}^{(j)}$ and $\sum_{i=0}^{k'-1} \boldsymbol{c}_i^{(j)} \vec{\boldsymbol{m}}_i$, we have the following upper bounds, except for negligible probabilities:

$$\left\|\vec{\boldsymbol{m}}^{(j)}\right\|_2 \le \big((b+1)\mathfrak{s}_1 + \mathfrak{s}_2\big) \cdot \sqrt{k'\ell n}, \quad \left\|\vec{\boldsymbol{m}}^{(j)}\right\|_{\infty} \le 5\big((b+1)\mathfrak{s}_1 + \mathfrak{s}_2\big) \cdot \sqrt{k'}$$

Similarly, we obtain the following results for $\vec{\boldsymbol{\mu}}^{(j)}$:

$$\left\|\vec{\boldsymbol{\mu}}^{(j)}\right\|_2 \leq \big(\sigma_1+\sigma_2\big)\cdot\sqrt{k'\ell n}, \quad \left\|\vec{\boldsymbol{\mu}}^{(j)}\right\|_\infty \leq 5\big(\sigma_1+\sigma_2\big)\cdot\sqrt{k'}$$

Thus, the upper-bound check conditions in lines 2 to 5 hold, except for negligible probabilities.

Soundness. The soundness of the protocol is directly derived from Lem. 5 of [7]. If there exists an adversarial prover who succeeds in cheating an honest verifier with a non-negligible probability, it implies the existence of a knowledge extractor $\mathcal{E}$, which can extract a witness $(\vec{\boldsymbol{m}}'_i, \vec{\boldsymbol{\mu}}'_i)$ satisfying $2\vec{\boldsymbol{m}}^*_i = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}'_i, \vec{\boldsymbol{\mu}}'_i)$ and $\left\|\vec{\boldsymbol{m}}'_i \| \vec{\boldsymbol{\mu}}'_i\right\|_2 < 2n\beta_{\mathtt{0pen}}$ for $0 \leq i < k'$ with a non-negligible probability. Therefore, a prover cannot be accepted by an honest verifier without knowing the witness for $2\vec{\boldsymbol{m}}^*_i$ for $0 \leq i < k'$ assuming the hardness of $\mathsf{MSIS}_{R,\mu,q,4n\beta_{\mathtt{0pen}}}$.

Simulatability. We show that $\mathcal{S}_{\mathtt{0pen}}$ in Fig. 3 efficiently simulates $\mathit{\Pi}_{\mathtt{0pen}}$. Let $\mathcal{D}_0(\vec{m})$ and $\mathcal{D}_1$ represent the distributions of the transcript generated by the honest prover and the verifier of $\mathit{\Pi}_{\mathtt{0pen}}$ with input message $\vec{m} = (\vec{m}_0, \ldots, \vec{m}_{k'-1})$, respectively, and those generated by $\mathcal{S}_{\mathtt{0pen}}$. We show that these distributions are computationally indistinguishable using the hybrid argument.

We first define the distributions $\mathcal{H}_0(\vec{m})$, $\mathcal{H}_1(\vec{m})$, $\mathcal{H}_2(\vec{m})$, and $\mathcal{H}_3$ as follows, where $\vec{\boldsymbol{n}}^{(j)}_i \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n,\mathfrak{s}_2}$, and $\vec{\boldsymbol{\nu}}^{(j)}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n,\sigma_2}$ for $0 \leq i < k'$, $0 \leq j < r$.

- $\mathcal{H}_0(\vec{m}) := \Big(\mathbf{A}_0\vec{\boldsymbol{m}}_i + [\mathbf{A}_1|\mathbf{I}_\mu]\vec{\boldsymbol{\mu}}_i,\ \vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}_i,\ \vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i\vec{\boldsymbol{\mu}}_i\Big)$
- $\mathcal{H}_1(\vec{m}) := \Big(\mathbf{A}_0\vec{\boldsymbol{m}}_i + \vec{\boldsymbol{m}}'^{*}_i,\ \vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}_i,\ \vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i\vec{\boldsymbol{\mu}}_i\Big)$
- $\mathcal{H}_2(\vec{m}) := \Big(\vec{\boldsymbol{m}}'^{*}_i,\ \vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}_i,\ \vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i\vec{\boldsymbol{\mu}}_i\Big)$
- $\mathcal{H}_3 := \Big(\vec{\boldsymbol{m}}'^{*}_i,\ \vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}'_i,\ \vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i\vec{\boldsymbol{\mu}}_i\Big)$

If $\mathcal{H}_0(\vec{m})$ and $\mathcal{H}_3$ are computationally indistinguishable, then $\mathcal{D}_0(\vec{m})$ and $\mathcal{D}_1$ are also computationally indistinguishable, as the latter distributions can be derived from the former ones. To be precise, $\sum_{i=0}^{k'-1}\vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}_i$ and $\sum_{i=0}^{k'-1}\vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{\mu}}_i$ in $\mathcal{H}_0(\vec{m})$ are within statistical distance $O(\varepsilon)$ of $\vec{\boldsymbol{m}}^{(j)}$ and $\vec{\boldsymbol{\mu}}^{(j)}$ in $\mathcal{D}_0(\vec{m})$, respectively, by Lem. 8. Similarly, $\sum_{i=0}^{k'-1}\vec{\boldsymbol{n}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{m}}'_i$ and $\sum_{i=0}^{k'-1}\vec{\boldsymbol{\nu}}^{(j)}_i + \boldsymbol{c}^{(j)}_i \cdot \vec{\boldsymbol{\mu}}_i$ in $\mathcal{H}_3$ are within statistical distance $O(\varepsilon)$ of $\vec{\boldsymbol{m}}'^{(j)}$ and $\vec{\boldsymbol{\mu}}^{(j)}$ in $\mathcal{D}_1$, respectively, by Lem. 8. Thus, it is sufficient to prove that $\mathcal{H}_0(\vec{m})$ and $\mathcal{H}_3$ are computationally indistinguishable, which involves proving the following claims.

Claim 1: $\mathcal{H}_0(\vec{m})$ and $\mathcal{H}_1(\vec{m})$ are computationally indistinguishable.

Distinguishing between the two given distributions is equivalent to solving $\mathsf{HintMLWE}^{\boldsymbol{c}^{(0)}_i,\ldots,\boldsymbol{c}^{(r-1)}_i}_{R,\nu,q,\sigma_1,\sigma_2,\ldots,\sigma_2}$. Since $\sigma \geq \sqrt{2}\cdot\eta_\varepsilon(\mathbb{Z}^n)$ holds under the given conditions for σ_1 and σ_2, $\mathsf{HintMLWE}^{\boldsymbol{c}^{(0)}_i,\ldots,\boldsymbol{c}^{(r-1)}_i}_{R,\nu,q,\sigma_1,\sigma_2,\ldots,\sigma_2}$ can be reduced from $\mathsf{MLWE}_{R,\nu,q,\sigma}$ by Thm. 1. Therefore, the two given distributions are computationally indistinguishable under the hardness assumption of $\mathsf{MLWE}_{R,\nu,q,\sigma}$.

Claim 2: $\mathcal{H}_1(\vec{m})$ and $\mathcal{H}_2(\vec{m})$ are statistically identical.

Since $\vec{\boldsymbol{m}}_i'^*$'s are uniform random samples from $\mathcal{U}(R_q^\mu)$, the distribution of $\mathbf{A}_0\vec{\boldsymbol{m}}_i + \vec{\boldsymbol{m}}_i'^*$ and $\vec{\boldsymbol{m}}_i'^*$ are identical.

Claim 3: $\mathcal{H}_2(\vec{m})$ and $\mathcal{H}_3$ are within negligible statistical distance.

We use Lem. 11 to prove the claim Let $\mathfrak{S}$ be a positive definitive matrices satisfying $\mathfrak{S}^{-1} = \mathfrak{s}_1^{-2} \cdot (PP^\top)^{-1} + r\mathfrak{s}_2^{-2} \cdot I_n$. Then, we need to show $\sqrt{\mathfrak{S}} \geq \eta_\varepsilon(P\mathbb{Z}^n) \iff P^{-1}\sqrt{\mathfrak{S}} \geq \eta_\varepsilon(\mathbb{Z}^n)$. By Lem. 2, it suffices to show that $\left\|P^\top\mathfrak{S}^{-1}P\right\|_2 \leq \eta_\varepsilon(\mathbb{Z})^{-2}$. Since $\|P\|_2 \leq b+1$ and $\left\|P^{-1}\right\|_2 \leq (b-1)^{-1}$, we have the followings:

$$\left\|P^\top\mathfrak{S}^{-1}P\right\|_2 \leq (b+1)^2\left((b-1)^{-2}\mathfrak{s}_1^{-2} + r\mathfrak{s}_2^{-2}\right)$$

Then, it holds that $\left\|P^\top\mathfrak{S}^{-1}P\right\|_2 \leq \eta_\varepsilon(\mathbb{Z}^n)$ under the given conditions for $\mathfrak{s}_1$ and $\mathfrak{s}_2$ Therefore, by Lem. 11, each distribution of the components of $\vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i$ and $\vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}_i'$ is within statistical distance $O(\varepsilon)$.

By Claim 1,2, and 3, the distributions $\mathcal{H}_0(\vec{m})$ and $\mathcal{H}_3$ are computationally indistinguishable, which implies $\mathcal{D}_0(\vec{m})$ and $\mathcal{D}_1$ are also computationally indistinguishable.

Therefore, $\Pi_{\texttt{Open}}$ satisfies the completeness, soundness, and simulatability under the given conditions. □

4 Sublinear Argument for R1CS

In this section, we present sublinear zero-knowledge argument for R1CS based on our new amortized POK protocol $\Pi_{\texttt{Open}}$. Our protocol mainly proves the knowledge of the solution vector $\vec{t} \in \mathbb{Z}_p^M$ for the following equations:

$$(A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) = (C\vec{t} + \vec{c}) \pmod{p} \tag{3}$$

where $A, B, C \in \mathbb{Z}_p^{M\times M}$ and $\vec{a}, \vec{b}, \vec{c} \in \mathbb{Z}_p^M$ are public to both prover and verifier. The high-level idea is similar to the previous work [7]. We embed the constraints of R1CS, such as $A, B, C, \vec{a}, \vec{b}, \vec{c}$, into polynomial equations over $\mathbb{Z}_p$. Then, the satisfiability of R1CS is reduced to the satisfiability of these polynomial equations.

Reduction to Inner Product Relations. Before composing the polynomial equations, we adapt optimization techniques from recent literature [19,31], which converts satisfiability of linear relations to satisfiability of inner product relations. To prove that $\vec{t}$ is a solution for Eq. (3), we show the following equation holds for a random challenge $\vec{v} = (1, v, \ldots, v^{M-1})$ where $v \leftarrow \mathcal{U}(\mathbb{Z}_p)$:

$$\left\langle (A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) - (C\vec{t} + \vec{c}), \vec{v} \right\rangle = 0 \pmod{p}$$

By applying the Schwartz-Zippel lemma, this equation holds only with a probability of $O(1/p)$, which will be set to be negligible, if $\vec{t}$ is not a solution for Eq. (3). We can rewrite the above equation as follows, where $V = \mathrm{diag}(\vec{v})$:

$$\left\langle \vec{t}, A^\top V B\vec{t} + B^\top V\vec{a} + A^\top V\vec{b} - C^\top\vec{v} \right\rangle = \langle \vec{c}, \vec{v}\rangle - \left\langle \vec{a}, V\vec{b} \right\rangle \pmod p$$

If we introduce a new indeterminate $\vec{d}$, and set $g_0 := \langle \vec{c}, \vec{v}\rangle - \left\langle \vec{a}, V\vec{b} \right\rangle$, it is equivalent to showing that the following two equations hold:

$$\begin{cases} \left\langle \vec{t}, \vec{d} \right\rangle = g_0 \pmod p \\ \vec{d} = A^\top V B\vec{t} + B^\top V\vec{a} + A^\top V\vec{b} - C^\top\vec{v} \pmod p \end{cases}$$

Instead of the second equation, we show the following equation holds for a random challenge $\vec{w} = (1, w, \ldots, w^{M-1})$ for $w \leftarrow \mathcal{U}(\mathbb{Z}_p)$:

$$\begin{aligned} &\left\langle \vec{d} - A^\top V B\vec{t} - B^\top V\vec{a} - A^\top V\vec{b} + C^\top\vec{v}, \vec{w} \right\rangle = 0 \\ &\iff \langle \vec{t}, B^\top V A\vec{w}\rangle - \left\langle \vec{d}, \vec{w} \right\rangle = \left\langle C^\top\vec{v} - B^\top V\vec{a} - A^\top V\vec{b}, \vec{w} \right\rangle \end{aligned}$$

If we set $\vec{f} = (f_0, \ldots, f_{M-1}) := B^\top V A\vec{w}$ and $g_1 := \left\langle C^\top\vec{v} - B^\top V\vec{a} - A^\top V\vec{b}, \vec{w} \right\rangle$, then proving the satisfiability of the R1CS finally reduces to proving the satisfiability of the following inner product relations:

$$\begin{cases} \left\langle \vec{t}, \vec{d} \right\rangle = g_0 \pmod p \\ \left\langle \vec{t}, \vec{f} \right\rangle - \left\langle \vec{d}, \vec{w} \right\rangle = g_1 \pmod p \end{cases} \tag{4}$$

Reduction to Polynomial Equations. We now reduces the satisfiability of Eq. (4) to the satisfiability of polynomial equations over $\mathbb{Z}_p$. For an indeterminant Y, we define polynomials $t(Y)$ and $d(Y)$ as follows:

$$\begin{aligned} t(Y) &:= Y^{\ell m} \cdot \sum_{i=0}^{M-1} t_i Y^i, \quad d(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} d_i Y^{M-i} \\ f(Y) &:= Y^{\ell m} \cdot \sum_{i=0}^{M-1} f_i Y^{M-i}, \quad w(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} w^i Y^i \end{aligned}$$

Then, the coefficient of $Y^{M+2\ell m}$ of $t(Y) \cdot d(Y)$, $t(Y) \cdot f(Y)$, and $d(Y) \cdot w(Y)$ are equal to $\left\langle \vec{t}, \vec{d} \right\rangle$, $\left\langle \vec{t}, \vec{f} \right\rangle$, and $\left\langle \vec{d}, \vec{w} \right\rangle$, respectively. Hence, proving the satisfiability of Eq. (4) is reduced to the proving the coefficient of $Y^{M+2\ell m}$ of the following polynomial is zero for some random challenge $x \leftarrow \mathcal{U}(\mathbb{Z}_p)$ by applying the Schwartz-Zippel lemma.

$$t(Y) \cdot d(Y) - g_0 \cdot Y^{M+2\ell m} + x \cdot \left(t(Y) \cdot f(Y) - d(Y) \cdot w(Y) - g_1 \cdot Y^{M+2\ell m} \right) \tag{5}$$

Committing and Verifying. To prove that the coefficient of $Y^{M+2\ell m}$ in Eq. (5) is zero, a prover initially commits $\vec{t} = (t_0, \ldots, t_{M-1})$ and $\vec{d} = (d_0, \ldots, d_{M-1})$ using the Ajtai scheme. We recall that for each commitment, it can contain at most ℓm elements of $\mathbb{Z}_p$. Therefore, we set $M = k \cdot \ell m$, so that k commitments are generated for each vector. For $0 \le i < k$, commitments $\vec{\boldsymbol{t}}_i^*$ and $\vec{\boldsymbol{d}}_i^*$ for $\vec{t}$ and $\vec{d}$ are generated as follows, where $\vec{t}_i = (t_{i\ell m}, \ldots, t_{(i+1)\ell m-1})$, $\vec{d}_i = (d_{i\ell m}, \ldots, d_{(i+1)\ell m-1})$, and $\mathfrak{s}_1, \sigma_1 > 0$.

$$\vec{\boldsymbol{t}}_i \leftarrow \mathtt{R.Ecd}(\vec{t}_i; \mathfrak{s}_1), \quad \vec{\boldsymbol{\tau}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n,\sigma_1}, \quad \vec{\boldsymbol{t}}_i^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{t}}_i; \vec{\boldsymbol{\tau}}_i)$$
$$\vec{\boldsymbol{d}}_i \leftarrow \mathtt{R.Ecd}(\vec{d}_i; \mathfrak{s}_1), \quad \vec{\boldsymbol{\delta}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n,\sigma_1}, \quad \vec{\boldsymbol{d}}_i^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{d}}_i; \vec{\boldsymbol{\delta}}_i)$$

Then, for $y \leftarrow \mathcal{U}(\mathbb{Z}_p)$, $\vec{y} := (1, \ldots, y^{\ell m-1})$ and $\vec{y}' := (y^{\ell m-1}, \ldots, 1)$, it holds the followings.

$$\left\langle \vec{y}, \mathtt{Dcd}\left(\sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i\right)\right\rangle = t(y)$$
$$\left\langle \vec{y}', \mathtt{Dcd}\left(\sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i\right)\right\rangle = d(y)$$

Thus, a prover can prove the evaluations of $t(Y)$ and $d(Y)$ at the point y by sending $\sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i$ and $\sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i$ instead of $\vec{t}$ and $\vec{d}$, effectively reducing communication costs. However, directly sending them reveals information about $\vec{t}$ and $\vec{d}$. To resolve this, we mask it with some random noise $\boldsymbol{e}_0, \boldsymbol{e}_1$ and $\boldsymbol{e}_2$ so that $\vec{\boldsymbol{t}} := \boldsymbol{e}_0 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i$ and $\vec{\boldsymbol{d}} := \boldsymbol{e}_1 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i$ appears independent of $\vec{t}$ and $\vec{d}$ to a verifier. Then, we have $\left\langle \vec{y}, \mathtt{Dcd}(\vec{t})\right\rangle = e_0(y) + t(y)$ and $\left\langle \vec{y}', \mathtt{Dcd}(\vec{d})\right\rangle = e_1(y) + d(y)$ where

$$(e_0, \ldots, e_{3\ell-1}) := \mathtt{Dcd}(\boldsymbol{e}_0 \| \boldsymbol{e}_1 \| \boldsymbol{e}_2)$$
$$e_0(Y) := \sum_{i=0}^{\ell m-1} e_i Y^i, \quad e_1(Y) := \sum_{i=0}^{\ell m-1} e_{\ell m+i} Y^{\ell m-1-i}, \quad e_2(Y) := \sum_{i=0}^{\ell m-1} e_{2\ell m+i} Y^i$$

Hence, a prover can prove evaluations of $e_0(Y) + t(Y)$ and $e_1(Y) + d(Y)$ at the point y, with reduced communication cost while keeping $\vec{t}$ and $\vec{d}$ secret. To recap, our objective is to prove the coefficient of $Y^{M+2\ell m}$ is zero in Eq. (5), which is equal to the coefficient of $Y^{M+2\ell m}$ of the following polynomial:

$$\begin{aligned} h(Y) := {} & (e_0(Y) + t(Y)) \cdot (e_1(Y) + d(Y)) - g_0 \cdot Y^{M+2\ell m} - e_2(Y) \\ & + x \cdot \big((e_0(Y) + t(Y)) \cdot f(Y) - (e_1(Y) + d(Y)) \cdot w(Y) - g_1 \cdot Y^{M+2\ell m}\big) \end{aligned}$$

since the degree of $e_0(Y)$, $e_1(Y)$ and $e_2(Y)$ are smaller than ℓm. To prove the coefficient of $Y^{M+2\ell m}$ in $h(Y) = \sum_{i=0}^{2M+2\ell m-1} h_i Y^i$ is zero, a prover commits

its coefficients h_i except for the $(M + 2\ell m)$-th coefficient. Then, using a similar method as for $e_0(Y) + t(Y)$ and $e_1(Y) + d(Y)$, a prover can efficiently prove evaluations of $e_2(Y) + h(Y) - h_{M+2\ell m} \cdot Y^{M+2\ell m}$ at the point y. The verifier then checks that it has the same value as:

$$\begin{aligned}&(e_0(y) + t(y)) \cdot (e_1(y) + d(y)) - g_0 \cdot y^{M+2\ell m} \\ &+ x \cdot \big((e_0(y) + t(y)) \cdot f(y) - (e_1(y) + d(y)) \cdot w(y) - g_1 \cdot y^{M+2\ell m}\big)\end{aligned}$$

This results in proving $h_{M+2\ell m} = 0$, which is our goal.

4.1 Proof-of-Knowledge Protocol for R1CS

We present a proof-of-opening knowledge protocol for R1CS, which achieves sublinear communication complexity of $O(\sqrt{M})$ in the input size M. Our protocol follows the commit-and-prove framework. A prover starts by committing input messages $\vec{t} = (t_0, t_1, \ldots, t_{M-1})$, along with other side information $\vec{d} = (d_0, \ldots, d_{M-1})$, and $\vec{h} = (h_0, \ldots, h_{2M+2\ell m-1})$ using the Ajtai commitment scheme. Then, it shows that the committed values satisfy polynomial equations, which we reduce from the satisfiability of R1CS. Finally, it proves its opening knowledge of the committed values using the amortized POK protocol for the Ajtai scheme.

The witness relations for the protocol are defined as follows:

$$\mathbf{R}_{\mathtt{R1CS}} := \left\{ \left(\{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1}, \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \right) \;\middle|\; \begin{array}{c} (\vec{\boldsymbol{t}}_i^*, \vec{\boldsymbol{t}}_i, \vec{\boldsymbol{\tau}}_i) \in \mathbf{R}_{\mathtt{Open}} \\ \wedge (A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) = (C\vec{t} + \vec{c}) \\ \text{s.t. } \vec{t} = \mathtt{Dcd}(\vec{\boldsymbol{t}}_0 \| \cdots \| \vec{\boldsymbol{t}}_{k-1}) \in \mathbb{Z}_p^M \end{array} \right\}$$

$$\mathbf{R}'_{\mathtt{R1CS}} := \left\{ \left(\{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1}, \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \right) \;\middle|\; \begin{array}{c} (\vec{\boldsymbol{t}}_i^*, \vec{\boldsymbol{t}}_i, \vec{\boldsymbol{\tau}}_i) \in \mathbf{R}'_{\mathtt{Open}} \\ \wedge (A\vec{t} + 2\vec{a}) \circ (B\vec{t} + 2\vec{b}) = 2(C\vec{t} + 2\vec{c}) \\ \text{s.t. } \vec{t} = \mathtt{Dcd}(\vec{\boldsymbol{t}}_0 \| \cdots \| \vec{\boldsymbol{t}}_{k-1}) \in \mathbb{Z}_p^M \end{array} \right\}$$

where $\mathsf{ck} \leftarrow \mathtt{Gen}(1^\lambda)$. Then, $(\{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1})$ is a witness for R1CS, which is bound by the commitments. The honest language $\mathbf{L}_{\mathtt{R1CS}}$ and the proven language $\mathbf{L}'_{\mathtt{R1CS}}$ are defined as follows:

$$\mathbf{L}_{\mathtt{R1CS}} = \left\{ \{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1} \;\middle|\; \exists \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \text{ s.t. } \left(\{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1}, \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \right) \in \mathbf{R}_{\mathtt{R1CS}} \right\}$$

$$\mathbf{L}'_{\mathtt{R1CS}} = \left\{ \{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1} \;\middle|\; \exists \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \text{ s.t. } \left(\{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1}, \{\vec{\boldsymbol{t}}_i\}_{i=0}^{k-1}, \{\vec{\boldsymbol{\tau}}_i\}_{i=0}^{k-1} \right) \in \mathbf{R}'_{\mathtt{R1CS}} \right\}$$

We note that what one can prove from the protocol is the knowledge of the solution $\vec{t}$ to the following equations:

$$(A\vec{t} + 2\vec{a}) \circ (B\vec{t} + 2\vec{b}) = 2(C\vec{t} + 2\vec{c}) \pmod{p}$$

However, if we set $\vec{t'} = 2^{-1} \cdot \vec{t}$, then

$$\begin{aligned} & (2A\vec{t'} + 2\vec{a}) \circ (2B\vec{t'} + 2\vec{b}) = 4(C\vec{t'} + \vec{c}) \pmod{p} \\ \iff & (A\vec{t'} + \vec{a}) \circ (B\vec{t'} + \vec{b}) = (C\vec{t'} + \vec{c}) \pmod{p} \end{aligned}$$

which is equivalent to solving the original R1CS. Hence, our protocol is sufficient for proving knowledge of a solution to the given R1CS even if there is a gap between the honest language and the proven language.

We present the proof-of-knowledge protocol for R1CS $\Pi_{\mathtt{R1CS}}$ in Fig. 4, 5, 6, and the verification procedure $\mathcal{V}_{\mathtt{R1CS}}$ in Fig. 7, and the simulator $\mathcal{S}_{\mathtt{R1CS}}$ in Fig. 8, 9.

$\Pi_{\mathtt{R1CS}}(A, B, C, \vec{a}, \vec{b}, \vec{c}, \{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1})$

	Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
	Input: $\mathsf{ck} \in R_q^{\mu \times (\mu+\nu+\ell)}$		ck
	$\vec{\boldsymbol{t}}_0, \ldots, \vec{\boldsymbol{t}}_{k-1} \in R^\ell$		
	$\vec{\boldsymbol{\tau}}_0, \ldots, \vec{\boldsymbol{\tau}}_{k-1} \in R^{\mu+\nu}$		
1 :	**Step 1:**		
2 :	$\vec{t} := \mathtt{Dcd}(\vec{\boldsymbol{t}}_0 \| \cdots \| \vec{\boldsymbol{t}}_{k-1}) \in \mathbb{Z}_p^M$		
3 :		$\xleftarrow{v}$	$v \leftarrow \mathcal{U}(\mathbb{Z}_p)$
4 :	$\vec{v} := (1, v, \ldots, v^{M-1}),\ V := \mathrm{diag}(\vec{v})$,		
5 :	$\vec{d} = \vec{d}_0 \| \cdots \| \vec{d}_{k-1} := A^\top V B \vec{t} + B^\top V \vec{a} + A^\top V \vec{b} - C^\top \vec{v}$		
6 :	For $0 \le i < k$,		
7 :	$\vec{\boldsymbol{d}}_i \leftarrow \mathtt{R.Ecd}(\vec{d}_i; \mathfrak{s}_1), \quad \vec{\boldsymbol{\delta}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$		
8 :	$\vec{\boldsymbol{d}}_i^* := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{d}}_i; \vec{\boldsymbol{\delta}}_i)$	$\xrightarrow{\vec{\boldsymbol{d}}_0^*, \ldots, \vec{\boldsymbol{d}}_{k-1}^*}$	
9 :		$\xleftarrow{w}$	$w \leftarrow \mathcal{U}(\mathbb{Z}_p)$
10 :	$\vec{w} := (1, w, \ldots, w^{M-1})$		

Fig. 4. Proof of R1CS relation(1).

$\Pi_{\mathtt{R1CS}}(A, B, C, \vec{a}, \vec{b}, \vec{c}, \{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1})$

Prover $\mathcal{P}$ **Verifier $\mathcal{V}$**

Step 2:

$\vec{\boldsymbol{e}}_0, \vec{\boldsymbol{e}}_1 \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\cdot\sqrt{k}\cdot \mathfrak{s}_2}$, $\vec{\boldsymbol{e}}_2 \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\cdot\sqrt{2k+2}\cdot \mathfrak{s}_2}$

$\vec{\boldsymbol{\varepsilon}}_0, \vec{\boldsymbol{\varepsilon}}_1 \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\cdot\sqrt{k}\cdot \sigma_2}$, $\vec{\boldsymbol{\varepsilon}}_2 \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\cdot\sqrt{2k+2}\cdot \sigma_2}$

$\vec{\boldsymbol{e}}_0^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{e}}_0; \vec{\boldsymbol{\varepsilon}}_0)$, $\vec{\boldsymbol{e}}_1^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{e}}_1; \vec{\boldsymbol{\varepsilon}}_1)$

$\vec{\boldsymbol{e}}_2^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{e}}_2; \vec{\boldsymbol{\varepsilon}}_2)$ $\xrightarrow{\vec{\boldsymbol{e}}_0^*, \vec{\boldsymbol{e}}_1^*, \vec{\boldsymbol{e}}_2^*}$

. . . . . .

Step 3:

$(t_0, \ldots, t_{M-1}) := \vec{t}, \quad (d_0, \ldots, d_{M-1}) := \vec{d}$

$(e_0, \ldots, e_{3\ell m-1}) := \mathtt{Dcd}(\vec{\boldsymbol{e}}_0 \| \vec{\boldsymbol{e}}_1 \| \vec{\boldsymbol{e}}_2)$

$\vec{f} = (f_0, \ldots, f_{M-1}) := B^\top V A \vec{w}$

$g_0 := \langle \vec{c}, \vec{v} \rangle - \left\langle \vec{a}, V\vec{b} \right\rangle, \quad g_1 := \left\langle C^\top \vec{v} - B^\top V \vec{a} - A^\top V \vec{b}, \vec{w} \right\rangle$

$t(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} t_i Y^i, \quad d(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} d_i Y^{M-i}$

$e_0(Y) := \sum_{i=0}^{\ell m-1} e_i Y^i, \quad e_1(Y) := \sum_{i=0}^{\ell m-1} e_{\ell m+i} Y^{\ell m-1-i}$

$e_2(Y) := \sum_{i=0}^{\ell m-1} e_{2\ell m+i} Y^i$

$f(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} f_i Y^{M-i}, \quad w(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} w^i Y^i$

$\xleftarrow{x}$ $x \leftarrow \mathcal{U}(\mathbb{Z}_p)$

$$\sum_{i=0}^{2M+2\ell m-1} h_i Y^i := \big(e_0(Y) + t(Y)\big)\big(e_1(Y) + d(Y)\big) - e_2(Y) - g_0 Y^{M+2\ell m}$$

$$+ x \cdot \Big((e_0(Y) + t(Y)) f(Y) - (e_1(Y) + d(Y)) w(Y) - g_1 Y^{M+2\ell m}\Big)$$

For $0 \le i < 2k+2$,

$\vec{\boldsymbol{h}}_i \leftarrow \mathtt{R.Ecd}(h_{i\ell m}, \ldots, h_{(i+1)\ell m-1}; \mathfrak{s}_1)$ if $i < k+2$

$\vec{\boldsymbol{h}}_i \leftarrow \mathtt{R.Ecd}(h_{i\ell m+1}, \ldots, h_{(i+1)\ell m}; \mathfrak{s}_1)$ if $i \ge k+2$

$\vec{\boldsymbol{\eta}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}, \quad \vec{\boldsymbol{h}}_i^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{h}}_i; \vec{\boldsymbol{\eta}}_i)$ $\xrightarrow{\vec{\boldsymbol{h}}_0^*, \ldots, \vec{\boldsymbol{h}}_{2k+1}^*}$

Fig. 5. Proof of R1CS relation(2).

$\Pi_{\texttt{R1CS}}(A, B, C, \vec{a}, \vec{b}, \vec{c}, \{\vec{\boldsymbol{t}}_i^*\}_{i=0}^{k-1})$

	Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
32 :	**Step 4:**		
33 :	$\vec{\boldsymbol{m}}_0 \| \cdots \| \vec{\boldsymbol{m}}_{4k+1} := (\vec{\boldsymbol{t}}_0 \| \cdots \| \vec{\boldsymbol{t}}_{k-1}) \| (\vec{\boldsymbol{d}}_0 \| \cdots \| \vec{\boldsymbol{d}}_{k-1}) \| (\vec{\boldsymbol{h}}_0 \| \cdots \| \vec{\boldsymbol{h}}_{2k+1})$		
34 :	$\vec{\boldsymbol{\mu}}_0 \| \cdots \| \vec{\boldsymbol{\mu}}_{4k+1} := (\vec{\boldsymbol{\tau}}_0 \| \cdots \| \vec{\boldsymbol{\tau}}_{k-1}) \| (\vec{\boldsymbol{\delta}}_0 \| \cdots \| \vec{\boldsymbol{\delta}}_{k-1}) \| (\vec{\boldsymbol{\eta}}_0 \| \cdots \| \vec{\boldsymbol{\eta}}_{2k+1})$		
35 :	$\vec{\boldsymbol{m}}_0^* \| \cdots \| \vec{\boldsymbol{m}}_{4k+1}^* := (\vec{\boldsymbol{t}}_0^* \| \cdots \| \vec{\boldsymbol{t}}_{k-1}^*) \| (\vec{\boldsymbol{d}}_0^* \| \cdots \| \vec{\boldsymbol{d}}_{k-1}^*) \| (\vec{\boldsymbol{h}}_0^* \| \cdots \| \vec{\boldsymbol{h}}_{2k+1}^*)$		
36 :	Run $\Pi_{\texttt{Open}}(\vec{\boldsymbol{m}}_0^*, \ldots, \vec{\boldsymbol{m}}_{4k+1}^*)$		
	.....		
37 :	**Step 5:**		
38 :		$\xleftarrow{y}$	$y \leftarrow \mathcal{U}(\mathbb{Z}_p)$
39 :	$\vec{\boldsymbol{t}} := \vec{\boldsymbol{e}}_0 + \sum_{i=0}^{k-1} \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i$		
40 :	$\vec{\boldsymbol{\tau}} := \vec{\boldsymbol{\varepsilon}}_0 + \sum_{i=0}^{k-1} \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i$		
41 :	$\vec{\boldsymbol{d}} := \vec{\boldsymbol{e}}_1 + \sum_{i=0}^{k-1} \texttt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i$		
42 :	$\vec{\boldsymbol{\delta}} := \vec{\boldsymbol{\varepsilon}}_1 + \sum_{i=0}^{k-1} \texttt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{\delta}}_i$		
43 :	$\vec{\boldsymbol{h}} := \vec{\boldsymbol{e}}_2 + \sum_{i=0}^{k+1} \texttt{Ecd}(y^{i\ell m}) \cdot \vec{\boldsymbol{h}}_i + \sum_{i=k+2}^{2k+1} \texttt{Ecd}(y^{i\ell m+1}) \cdot \vec{\boldsymbol{h}}_i$		
44 :	$\vec{\boldsymbol{\eta}} := \vec{\boldsymbol{\varepsilon}}_2 + \sum_{i=0}^{k+1} \texttt{Ecd}(y^{i\ell m}) \cdot \vec{\boldsymbol{\eta}}_i + \sum_{i=k+2}^{2k+1} \texttt{Ecd}(y^{i\ell m+1}) \cdot \vec{\boldsymbol{\eta}}_i$	$\vec{\boldsymbol{t}}, \vec{\boldsymbol{d}}, \vec{\boldsymbol{h}}$ $\vec{\boldsymbol{\tau}}, \vec{\boldsymbol{\delta}}, \vec{\boldsymbol{\eta}}$ $\xrightarrow{\quad}$	
45 :			Run $\mathcal{V}_{\texttt{R1CS}}$

Fig. 6. Proof of R1CS relation(3).

$\mathcal{V}_{\mathtt{R1CS}}$

1 : $\left\|\vec{t}\right\|_2, \left\|\vec{d}\right\|_2 \overset{?}{<} \frac{(b+2)\kappa}{2} \cdot \left((b+1)\sqrt{k}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{k\ell n}$

2 : $\left\|\vec{t}\right\|_\infty, \left\|\vec{d}\right\|_\infty \overset{?}{<} \frac{5(b+2)\kappa}{2} \cdot \left((b+1)\sqrt{k}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{k}$

3 : $\|\vec{\tau}\|_2, \left\|\vec{\delta}\right\|_2 \overset{?}{<} \frac{(b+2)\kappa}{2} \cdot \left(\sqrt{k}\sigma_1 + \sigma_2\right) \cdot \sqrt{k(\mu+\nu)n}$

4 : $\|\vec{\tau}\|_\infty, \left\|\vec{\delta}\right\|_\infty \overset{?}{<} \frac{5(b+2)\kappa}{2} \cdot \left(\sqrt{k}\sigma_1 + \sigma_2\right) \cdot \sqrt{k}$

5 : $\left\|\vec{h}\right\|_2 \overset{?}{<} \mathfrak{b}, \quad \|\vec{\eta}\|_2 \overset{?}{<} \beta$

6 : $\left\|\vec{h}\right\|_\infty \overset{?}{<} \frac{5(b+2)\kappa}{2} \cdot \left((b+1)\sqrt{2k+2}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{2k+2}$

7 : $\|\vec{\eta}\|_\infty \overset{?}{<} \frac{5(b+2)\kappa}{2} \cdot \left(\sqrt{2k+2}\sigma_1 + \sigma_2\right) \cdot \sqrt{2k+2}$

8 : $\mathtt{Com}_{\mathsf{ck}}(\vec{t}; \vec{r}) \overset{?}{=} \vec{e}_0^* + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{t}_i^* \pmod q$

9 : $\mathtt{Com}_{\mathsf{ck}}(\vec{d}; \vec{\delta}) \overset{?}{=} \vec{e}_1^* + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{d}_i^* \pmod q$

10 : $\mathtt{Com}_{\mathsf{ck}}(\vec{h}; \vec{\eta}) \overset{?}{=} \vec{e}_2^* + \sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{h}_i^* + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{h}_i^* \pmod q$

11 : $\vec{v} := (1, v, \ldots, v^{M-1}), \quad V := \mathrm{diag}(\vec{v})$

12 : $\vec{w} := (1, w, \ldots, w^{M-1}), \quad w(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} w^i Y^i$

13 : $\vec{f} := (f_0, \ldots, f_{M-1}) := B^\top V A \vec{w}, \quad f(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} f_i Y^{M-i}$

14 : $g_0 := \langle \vec{c}, \vec{v} \rangle - \left\langle \vec{a}, V\vec{b} \right\rangle, \quad g_1 := \left\langle C^\top \vec{v} - B^\top V \vec{a} - A^\top V \vec{b}, \vec{w} \right\rangle$

15 : $\vec{y} := (1, \ldots, y^{\ell m-1}), \quad \vec{y}' := (y^{\ell m-1}, \ldots, 1)$

16 : $\left\langle \vec{y}, \mathtt{Dcd}(\vec{h}) \right\rangle \overset{?}{=} \left\langle \vec{y}, \mathtt{Dcd}(\vec{t}) \right\rangle \cdot \left\langle \vec{y}', \mathtt{Dcd}(\vec{d}) \right\rangle - g_0 \cdot y^{M+2\ell m}$

17 : $\quad + x \cdot \left(\left\langle \vec{y}, \mathtt{Dcd}(\vec{t}) \right\rangle \cdot f(y) - \left\langle \vec{y}', \mathtt{Dcd}(\vec{d}) \right\rangle \cdot w(y) - g_1 \cdot y^{M+2\ell m} \right) \pmod p$

Fig. 7. Verification procedure for $\Pi_{\mathtt{R1CS}}$

$\mathcal{S}_{\mathtt{R1CS}}(\mathsf{ck}, A, B, C, \vec{a}, \vec{b}, \vec{c})$

Step 1:

$\vec{\boldsymbol{t}}_i'^*, \vec{\boldsymbol{d}}_i'^* \leftarrow \mathcal{U}(R_q^\mu)$ for $0 \le i < k$

$\vec{\boldsymbol{h}}_i'^* \leftarrow \mathcal{U}(R_q^\mu)$ for $0 \le i < 2k+2$, $v, w, x, y \leftarrow \mathcal{U}(\mathbb{Z}_p)$

$\vec{v} := (1, v, \ldots, v^{M-1})$, $V := \mathrm{diag}(\vec{v})$, $\vec{w} := (1, w, \ldots, w^{M-1})$

$\vec{\boldsymbol{t}}_i' \leftarrow \mathtt{R.Ecd}(0; \mathfrak{s}_1)$, $\vec{\boldsymbol{\tau}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ for $0 \le i < k$

$\vec{\boldsymbol{d}}_i' \leftarrow \mathtt{R.Ecd}(0; \mathfrak{s}_1)$, $\vec{\boldsymbol{\delta}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ for $0 \le i < k$

$\vec{\boldsymbol{e}}_0, \vec{\boldsymbol{e}}_1 \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \sqrt{k}\kappa b \mathfrak{s}_2}$, $\vec{\boldsymbol{e}}_2 \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \sqrt{2k+2}\kappa b \mathfrak{s}_2}$

$\vec{\boldsymbol{\varepsilon}}_0, \vec{\boldsymbol{\varepsilon}}_1 \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{k}\kappa b \sigma_2}$, $\vec{\boldsymbol{\varepsilon}}_2 \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{2k+2}\kappa b \sigma_2}$

$(e_0, \ldots, e_{3\ell m-1}) := \mathtt{Dcd}(\vec{\boldsymbol{e}}_0 \| \vec{\boldsymbol{e}}_1 \| \vec{\boldsymbol{e}}_2)$, $\vec{f} = (f_0, \ldots, f_{M-1}) := B^\top V A \vec{w}$

$g_0 := \langle \vec{c}, \vec{v} \rangle - \left\langle \vec{a}, V\vec{b} \right\rangle$, $\quad g_1 := \left\langle C^\top \vec{v} - B^\top V \vec{a} - A^\top V \vec{b}, \vec{w} \right\rangle$

$e_0(Y) := \sum_{i=0}^{\ell m-1} e_i Y^i$, $e_1(Y) := \sum_{i=0}^{\ell m-1} e_{\ell m+i} Y^{\ell m-1-i}$, $e_2(Y) := \sum_{i=0}^{\ell m-1} e_{2\ell m+i} Y^i$

$f(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} f_i Y^{M-i}$, $\quad w(Y) := Y^{\ell m} \cdot \sum_{i=0}^{M-1} w^i Y^i$

$h_0' := e_0(y)e_1(y) - e_2(y) - g_0 y^{M+2\ell m} + x \cdot \left(e_0(y)f(y) - e_1(y)w(y) - g_1 y^{M+2\ell m} \right)$

$\vec{\boldsymbol{h}}_0' \leftarrow \mathtt{R.Ecd}(h_0', 0, \ldots, 0; \mathfrak{s}_1)$, $\vec{\boldsymbol{\eta}}_0 \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$

$\vec{\boldsymbol{h}}_i' \leftarrow \mathtt{R.Ecd}(0; \mathfrak{s}_1)$, $\vec{\boldsymbol{\eta}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ for $1 \le i < 2k+2$

$\vec{\boldsymbol{t}}' := \vec{\boldsymbol{e}}_0 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i'$, $\vec{\boldsymbol{\tau}} := \vec{\boldsymbol{\varepsilon}}_0 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i$

$\vec{\boldsymbol{d}}' := \vec{\boldsymbol{e}}_1 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i'$, $\vec{\boldsymbol{\delta}} := \vec{\boldsymbol{\varepsilon}}_0 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{\delta}}_i$

$\vec{\boldsymbol{h}}' := \vec{\boldsymbol{e}}_2 + \sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{\boldsymbol{h}}_i' + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{\boldsymbol{h}}_i'$

$\vec{\boldsymbol{\eta}} := \vec{\boldsymbol{\varepsilon}}_2 + \sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{\boldsymbol{\eta}}_i + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{\boldsymbol{\eta}}_i$

$\vec{\boldsymbol{e}}_0^* := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{t}}; \vec{\boldsymbol{\tau}}) - \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i'^* \pmod q$

$\vec{\boldsymbol{e}}_1^* := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{d}}; \vec{\boldsymbol{\delta}}) - \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{\boldsymbol{d}}_i'^* \pmod q$

$\vec{\boldsymbol{e}}_2^* := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{h}}; \vec{\boldsymbol{\eta}}) - \sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{\boldsymbol{h}}_i'^* - \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{\boldsymbol{h}}_i^* \pmod q$

Output $\{\vec{\boldsymbol{t}}_i'^*\}_{i=0}^{k-1}$, $\{\vec{\boldsymbol{d}}_i'^*\}_{i=0}^{k-1}$, $\{\vec{\boldsymbol{e}}_i^*\}_{i=0}^{2}$, $\{\vec{\boldsymbol{h}}_i'^*\}_{i=0}^{2k+1}, v, w, x, y$, $\vec{\boldsymbol{t}}', \vec{\boldsymbol{d}}', \vec{\boldsymbol{h}}'$, $\vec{\boldsymbol{\tau}}, \vec{\boldsymbol{\delta}}, \vec{\boldsymbol{\eta}}$

Fig. 8. Simulator for $\Pi_{\mathtt{R1CS}}(1)$

$\mathcal{S}_{\mathtt{R1CS}}(\mathsf{ck}, A, B, C, \vec{a}, \vec{b}, \vec{c})$

24 : **Step 2:**

25 : $\vec{\boldsymbol{m}}'_0 \| \cdots \| \vec{\boldsymbol{m}}'_{4k+1} := (\vec{\boldsymbol{t}}'_0 \| \cdots \| \vec{\boldsymbol{t}}'_{k-1}) \| (\vec{\boldsymbol{d}}'_0 \| \cdots \| \vec{\boldsymbol{d}}'_{k-1}) \| (\vec{\boldsymbol{h}}'_0 \| \cdots \| \vec{\boldsymbol{h}}'_{2k+1})$

26 : $\vec{\boldsymbol{\mu}}_0 \| \cdots \| \vec{\boldsymbol{\mu}}_{4k+1} := (\vec{\boldsymbol{\tau}}_0 \| \cdots \| \vec{\boldsymbol{\tau}}_{k-1}) \| (\vec{\boldsymbol{\delta}}_0 \| \cdots \| \vec{\boldsymbol{\delta}}_{k-1}) \| (\vec{\boldsymbol{\eta}}_0 \| \cdots \| \vec{\boldsymbol{\eta}}_{2k+1})$

27 : $\vec{\boldsymbol{m}}'^*_0 \| \cdots \| \vec{\boldsymbol{m}}'^*_{4k+1} := (\vec{\boldsymbol{t}}'^*_0 \| \cdots \| \vec{\boldsymbol{t}}'^*_{k-1}) \| (\vec{\boldsymbol{d}}'^*_0 \| \cdots \| \vec{\boldsymbol{d}}'^*_{k-1}) \| (\vec{\boldsymbol{h}}'^*_0 \| \cdots \| \vec{\boldsymbol{h}}'^*_{2k+1})$

28 : $\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{4k+1}^{(j)} \leftarrow \mathcal{U}(C)$ for $0 \le j < r$

29 : $\vec{\boldsymbol{n}}^{(j)} \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \sqrt{4k+2}\mathfrak{s}_2}, \vec{\boldsymbol{\nu}}^{(j)} \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sqrt{4k+2}\sigma_2}$ for $0 \le j < r$

30 : $\vec{\boldsymbol{m}}'^{(j)} := \vec{\boldsymbol{n}}^{(j)} + \sum_{i=0}^{4k+1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}'_i,\ \vec{\boldsymbol{\mu}}^{(j)} := \vec{\boldsymbol{\nu}}^{(j)} + \sum_{i=0}^{4k+1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\mu}}_i$ for $0 \le j < r$

31 : $\vec{\boldsymbol{n}}^{*(j)} := \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{m}}'^{(j)}; \vec{\boldsymbol{\mu}}^{(j)}) - \sum_{i=0}^{4k+1} \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{m}}'^*_i \pmod q$ for $0 \le j < r$

32 : Output $\{\vec{\boldsymbol{n}}^{*(j)}\}_{j=0}^{r-1}, \{(\boldsymbol{c}_0^{(j)}, \ldots, \boldsymbol{c}_{k'-1}^{(j)})\}_{j=0}^{r-1}, \{\vec{\boldsymbol{m}}'^{(j)}\}_{j=0}^{r-1}, \{\vec{\boldsymbol{\mu}}^{(j)}\}_{j=0}^{r-1}$

Fig. 9. Simulator for $\Pi_{\mathtt{R1CS}}(2)$

Below, we prove that $\Pi_{\mathtt{Open}}$ is a secure POK protocol for the languages $(\mathbf{L}_{\mathtt{Open}}, \mathbf{L}'_{\mathtt{Open}})$.

Theorem 3. *Let* $\mathsf{ck} \leftarrow \mathtt{Gen}(1^\lambda)$ *and* $\log p \ge \lambda + \log(2M + 2\ell m)$. *Let* $\mathfrak{s}_1, \mathfrak{s}_2, \sigma_1, \sigma_2$, $\mathfrak{b}$, β, $\beta_{\mathtt{R1CS}}$ *be positive reals satisfying the following conditions for some* $0 < \varepsilon < 2^{-\lambda}$.

- $\mathfrak{s}_1 \ge \sqrt{2} \cdot \frac{b+1}{b-1} \cdot \eta_\varepsilon(\mathbb{Z}^n), \quad \mathfrak{s}_2 \ge (b+1)\sqrt{2r+2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$
- $\sigma_1 \ge 2\sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n), \quad \sigma_2 \ge 2\sqrt{2r+2} \cdot \eta_\varepsilon(\mathbb{Z}^n), \ \sigma = \frac{1}{\sqrt{2}}(\sigma_1^{-2} + (r+1) \cdot \sigma_2^{-2})^{-1/2}$
- $\mathfrak{b} = \frac{(b+2)\kappa}{2} \cdot \left((b+1)\sqrt{2k+2}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{(2k+2)\ell n}$
- $\beta = \frac{(b+2)\kappa}{2} \cdot \left(\sqrt{2k+2}\sigma_1 + \sigma_2\right) \cdot \sqrt{(2k+2)(\mu+\nu)n}$
- $\beta_{\mathtt{R1CS}} = 2\mathfrak{b} + 2\beta + 2(b+2)(k+1)\kappa \cdot n\beta_{\mathtt{Open}}$

If $\vec{\boldsymbol{t}}_i \leftarrow \mathtt{R.Ecd}(\vec{t}_i; \mathfrak{s}_1)$ *for some* $\vec{t} = (\vec{t}_0, \ldots, \vec{t}_{k-1}) \in \mathbb{Z}_p^M$ *satisfying* $(A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) = (C\vec{t} + \vec{c})$, $\vec{\boldsymbol{\tau}}_i \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_1}$ *for* $0 \le i < k$, *and* $\Pi_{\mathtt{Open}}$ *is a secure proof-of-knowledge protocol for* $(\mathbf{L}_{\mathtt{Open}}, \mathbf{L}'_{\mathtt{Open}})$, *then* $\Pi_{\mathtt{R1CS}}$ *is a secure proof-of-knowledge protocol for* $(\mathbf{L}_{\mathtt{R1CS}}, \mathbf{L}'_{\mathtt{R1CS}})$ *under the hardness assumption of* $\mathsf{MLWE}_{R,\nu,q,\sigma}$ *and* $\mathsf{MSIS}_{R,\mu,q,2\beta_{\mathtt{R1CS}}}$.

Proof. We show the completeness, soundness, simulatability of $\Pi_{\mathtt{R1CS}}$.

Completeness. If the prover is honest, $\vec{\boldsymbol{t}}_i^* = \mathtt{Com}_{\mathsf{ck}}(\vec{\boldsymbol{t}}_i; \vec{\boldsymbol{\tau}}_i)$ holds for $0 \le i < k'$, and $(A\vec{t} + \vec{a}) \circ (B\vec{t} + \vec{b}) = (C\vec{t} + \vec{c})$ holds for $\vec{t} = \mathtt{Dcd}(\vec{\boldsymbol{t}}_0 \| \ldots \| \vec{\boldsymbol{t}}_{k-1})$. Consequently, lines 8 to 17 of $\mathcal{V}_{\mathtt{R1CS}}$ always hold, so it suffices to show that the upper-bound check conditions in lines 1 to 7 are valid. As an example, we analyze the case of $\vec{\boldsymbol{h}}$; the others can be shown in a similar manner. In this time, we directly apply

the tail bounds from Lem. 4, 5 to $\vec{h}_i$'s and $\vec{e}_2$. Then, the following holds except for negligible probabilities:

$$\left\|\vec{h}_i\right\|_2 \leq (b+1)\mathfrak{s}_1 \cdot \sqrt{\ell n}, \quad \|\vec{e}_2\|_2 \leq \frac{(b+2)\kappa}{2} \cdot \mathfrak{s}_2 \cdot \sqrt{(2k+2)\ell n}$$
$$\left\|\vec{h}_i\right\|_\infty \leq 5(b+1)\mathfrak{s}_1, \quad \|\vec{e}_2\|_\infty \leq \frac{5(b+2)\kappa}{2} \cdot \sqrt{2k+2} \cdot \mathfrak{s}_2$$

Since $\|\mathtt{Ecd}(a)\|_1 \leq \frac{(b+2)\kappa}{2}$ holds for all $a \in \mathbb{Z}_p$, the following holds:

$$\left\|\sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{h}_i + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{h}_i\right\|_2 \leq (2k+2) \cdot \frac{(b+2)\kappa}{2} \cdot (b+1)\mathfrak{s}_1 \cdot \sqrt{\ell n}$$
$$\left\|\sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{h}_i + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{h}_i\right\|_\infty \leq (2k+2) \cdot \frac{(b+2)\kappa}{2} \cdot 5(b+1)\mathfrak{s}_1$$

Therefore, the following holds except for negligible probabilities:

$$\left\|\vec{h}\right\|_2 \leq \mathfrak{b} = \frac{(b+2)\kappa}{2} \cdot \left((b+1)\sqrt{2k+2}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{(2k+2)\ell n}$$
$$\left\|\vec{h}\right\|_\infty \leq \frac{5(b+2)\kappa}{2} \cdot \left(\sqrt{2k+2}\mathfrak{s}_1 + \mathfrak{s}_2\right) \cdot \sqrt{2k+2}$$

Thus, the upper-bound check conditions in lines 1 to 7 hold, except for negligible probabilities.

Soundness. Suppose there is a prover that is accepted by an honest prover of $\Pi_{\mathtt{R1CS}}$ with non-negligible probabilities. By the soundness property of $\Pi_{\mathtt{Open}}$, a prover possess opening knowledges $(\vec{t}'_i \| \vec{\tau}'_i)$, $(\vec{d}'_i \| \vec{\delta}'_i)$, and $(\vec{h}'_i \| \vec{\eta}'_i)$ for $2\vec{t}^*_i$, $2\vec{d}^*_i$, and $2\vec{h}^*_i$, respectively, with sizes bounded by $2n\beta_{\mathtt{Open}}$. Then, a prover can derive opening knowledges $(\vec{e}'_0 \| \vec{\varepsilon}'_0)$, $(\vec{e}'_1 \| \vec{\varepsilon}'_1)$, and $(\vec{e}'_2 \| \vec{\varepsilon}'_2)$ for $2\vec{e}^*_0$, $2\vec{e}^*_1$, and $2\vec{e}^*_2$, respectively, with sizes bounded by $\beta_{\mathtt{R1CS}} = 2\mathfrak{b} + 2\beta + 2(b+2)(k+1)\kappa \cdot n\beta_{\mathtt{Open}}$, based on the equations in lines 8 to 10 in $\mathcal{V}_{\mathtt{R1CS}}$. Since we assume that $\mathsf{MSIS}_{R,\mu,q,2\beta_{\mathtt{R1CS}}}$ is computationally hard, these opening knowledges are bound to the corresponding commitments. Therefore, for the responses $\vec{t}, \vec{d}, \vec{h}$ from the prover, their values are represented as follows:

$$2\vec{t} := \vec{e}'_0 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{t}'_i$$
$$2\vec{d} := \vec{e}'_1 + \sum_{i=0}^{k-1} \mathtt{Ecd}(y^{M-i\ell m+1}) \cdot \vec{d}'_i$$
$$2\vec{h} := \vec{e}'_2 + \sum_{i=0}^{k+1} \mathtt{Ecd}(y^{i\ell m}) \cdot \vec{h}'_i + \sum_{i=k+2}^{2k+1} \mathtt{Ecd}(y^{i\ell m+1}) \cdot \vec{h}'_i$$

From equations in lines 16 to 17 in $\mathcal{V}_{\mathtt{R1CS}}$, it holds that:

$$\begin{aligned}\left\langle \vec{y}, \mathtt{Dcd}(4\vec{\boldsymbol{h}}) \right\rangle = \left\langle \vec{y}, \mathtt{Dcd}(2\vec{\boldsymbol{t}}) \right\rangle \cdot \left\langle \vec{y}', \mathtt{Dcd}(2\vec{\boldsymbol{d}}) \right\rangle - 4g_0 \cdot y^{M+2\ell m} \\ + x \cdot \left(\left\langle \vec{y}, \mathtt{Dcd}(4\vec{\boldsymbol{t}}) \right\rangle \cdot f(y) - \left\langle \vec{y}', \mathtt{Dcd}(4\vec{\boldsymbol{d}}) \right\rangle \cdot w(y) - 4g_1 \cdot y^{M+2\ell m} \right)\end{aligned}$$

By the Schwartz-Zippel lemma on variable y, the coefficient of each y^i for $0 \leq i < 2M + 2\ell m$ on both sides should be equal, except for negligible probabilities. Define $t'_i, d'_i, h'_i \in \mathbb{Z}_p$ as follows:

$$\begin{aligned}\vec{t}' = (t'_0, \ldots, t'_{M-1}) &:= \mathtt{Dcd}(\vec{\boldsymbol{t}}'_0 \| \cdots \| \vec{\boldsymbol{t}}'_{k-1}) \\ \vec{d}' = (d'_0, \ldots, d'_{M-1}) &:= \mathtt{Dcd}(\vec{\boldsymbol{d}}'_0 \| \cdots \| \vec{\boldsymbol{d}}'_{k-1}) \\ (h'_0, \ldots, h'_{M+2\ell m-1}) &:= \mathtt{Dcd}(\vec{\boldsymbol{h}}'_0 \| \cdots \| \vec{\boldsymbol{h}}'_{k+1}) \\ (h'_{M+2\ell m+1}, \ldots, h'_{2M+2\ell m-1}) &:= \mathtt{Dcd}(\vec{\boldsymbol{h}}'_{k+2} \| \cdots \| \vec{\boldsymbol{h}}'_{2k+1})\end{aligned}$$

Note that the coefficient of $y^{M+2\ell m}$ is zero in the left hand side, so the coefficient of $y^{M+2\ell m}$ should also be zero on the right-hand side, as calculated below:

$$\begin{aligned}&\sum_{i=0}^{M-1} t'_i d'_i - 4g_0 + x \cdot \left(\sum_{i=0}^{M-1} (2t'_i f_i - 2d'_i w^i) - 4g_1 \right) \\ &= \left\langle \vec{t}', \vec{d}' \right\rangle - 4g_0 + 2x \cdot \left(\left\langle \vec{t}', \vec{f} \right\rangle - \left\langle \vec{d}', \vec{w} \right\rangle - 2g_1 \right) = 0\end{aligned}$$

By the Schwartz-Zippel lemma on variable x, both $\left\langle \vec{t}', \vec{d}' \right\rangle - 4g_0$ and $\left\langle \vec{t}', \vec{f} \right\rangle - \left\langle \vec{d}', \vec{w} \right\rangle - 2g_1$ should be zero except for negligible probabilities. Then, we have the followings:

$$\begin{aligned}\left\langle \vec{t}', \vec{d}' \right\rangle &= \langle 4\vec{c}, \vec{v} \rangle - \left\langle 2\vec{a}, 2V\vec{b} \right\rangle \\ \left\langle \vec{d}', \vec{w} \right\rangle &= \langle \vec{t}', B^\top V A \vec{w} \rangle + \left\langle 2B^\top V\vec{a} + 2A^\top V\vec{b} - 2C^\top \vec{v}, \vec{w} \right\rangle \\ &= \left\langle A^\top V B\vec{t}' + 2B^\top V\vec{a} + 2A^\top V\vec{b} - 2C^\top \vec{v}, \vec{w} \right\rangle\end{aligned}$$

By the Schwartz-Zippel lemma on variable w, it holds that $\vec{d}' = A^\top V B\vec{t}' + 2B^\top V\vec{a} + 2A^\top V\vec{b} - 2C^\top \vec{v}$ except for negligible probabilities. Then, we have the followings:

$$\begin{aligned}&\left\langle \vec{t}', A^\top V B\vec{t}' + 2B^\top V\vec{a} + 2A^\top V\vec{b} - 2C^\top \vec{v} \right\rangle = \langle 4\vec{c}, \vec{v} \rangle - \left\langle 2\vec{a}, 2V\vec{b} \right\rangle \\ &\iff \left\langle (A\vec{t}' + 2\vec{a}) \circ (B\vec{t}' + 2\vec{b}) - 2(C\vec{t}' + 2\vec{c}), \vec{v} \right\rangle = 0\end{aligned}$$

Finally, by the Schwartz-Zippel lemma on variable v, $(A\vec{t}' + 2\vec{a}) \circ (B\vec{t}' + 2\vec{b}) - 2(C\vec{t}' + 2\vec{c}) = 0$ except for negligible probabilities. Therefore, the prover knows witness knowledge for $\mathbf{L}'_{\mathtt{R1CS}}$.

Simulatability. We show that $\mathcal{S}_{\texttt{R1CS}}$ in Fig. 8 and 9 efficiently simulates accepting transcripts of $\Pi_{\texttt{R1CS}}$. Let $\mathcal{D}_0(\vec{t})$ and $\mathcal{D}_1$ represent the distributions of transcripts generated by the honest prover and the verifier of $\Pi_{\texttt{R1CS}}$ with input message $\vec{t} = (\vec{t}_0, \ldots, \vec{t}_{k-1}) \in \mathbb{Z}_p^M$, respectively, as well as those generated by $\mathcal{S}_{\texttt{R1CS}}$. Similar to the simulatability proof for $\Pi_{\texttt{0pen}}$, we prove their indistinguishability using a hybrid argument. The only difference is that we need to additionally account for the effect of $\vec{\boldsymbol{t}}, \vec{\boldsymbol{d}}, \vec{\boldsymbol{h}}$ and $\vec{\boldsymbol{\tau}}, \vec{\boldsymbol{\delta}}, \vec{\boldsymbol{\eta}}$ in proving simulatability. For simplicity, we focus on proving the simulatability of the transcripts related to $(\vec{\boldsymbol{t}}_i, \vec{\boldsymbol{\tau}}_i, \vec{\boldsymbol{t}}_i^*)$, because the proof can be naturally extend to cover the cases of $(\vec{\boldsymbol{d}}_i, \vec{\boldsymbol{\delta}}_i, \vec{\boldsymbol{d}}_i^*)$ and $(\vec{\boldsymbol{h}}_i, \vec{\boldsymbol{\eta}}_i, \vec{\boldsymbol{h}}_i^*)$.

We first define the distributions $\mathcal{H}_0(\vec{t})$, $\mathcal{H}_1(\vec{t})$, $\mathcal{H}_2(\vec{t})$, and $\mathcal{H}_3$ as follows, where $\vec{\boldsymbol{e}}_{0,i} \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\mathfrak{s}_2}$, $\vec{\boldsymbol{\varepsilon}}_{0,i} \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \frac{(b+2)\kappa}{2}\sigma_2}$, $\vec{\boldsymbol{n}}_i^{(j)} \leftarrow \mathcal{D}^{\ell}_{\mathbb{Z}^n, \mathfrak{s}_2}$, and $\vec{\boldsymbol{\nu}}_i^{(j)} \leftarrow \mathcal{D}^{\mu+\nu}_{\mathbb{Z}^n, \sigma_2}$ for $0 \le i < k$, $0 \le j < r$.

- $\mathcal{H}_0(\vec{t}) := \begin{pmatrix} \mathbf{A}_0\vec{\boldsymbol{t}}_i + [\mathbf{A}_1|\mathbf{I}_\mu]\vec{\boldsymbol{\tau}}_i, & \vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\nu}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\tau}}_i, \\ \vec{\boldsymbol{e}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\varepsilon}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i & \end{pmatrix}$
- $\mathcal{H}_1(\vec{t}) := \begin{pmatrix} \mathbf{A}_0\vec{\boldsymbol{t}}_i + \vec{\boldsymbol{t}}_i'^*, & \vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\nu}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\tau}}_i, \\ \vec{\boldsymbol{e}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\varepsilon}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i & \end{pmatrix}$
- $\mathcal{H}_2(\vec{t}) := \begin{pmatrix} \vec{\boldsymbol{t}}_i'^*, & \vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\nu}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\tau}}_i, \\ \vec{\boldsymbol{e}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i, & \vec{\boldsymbol{\varepsilon}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i & \end{pmatrix}$
- $\mathcal{H}_3 := \begin{pmatrix} \vec{\boldsymbol{t}}_i'^*, & \vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i', & \vec{\boldsymbol{\nu}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{\tau}}_i, \\ \vec{\boldsymbol{e}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i', & \vec{\boldsymbol{\varepsilon}}_{0,i} + \texttt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{\tau}}_i & \end{pmatrix}$

If $\mathcal{H}_0(\vec{t})$ and $\mathcal{H}_3$ are computationally indistinguishable, then $\mathcal{D}_0(\vec{t})$ and $\mathcal{D}_1$ are also computationally indistinguishable, as the latter distributions can be derived from the former ones, as shown in the proof of simulatability for $\Pi_{\texttt{0pen}}$ Thus, it is sufficient to prove that $\mathcal{H}_0(\vec{t})$ and $\mathcal{H}_3$ are computationally indistinguishable, which involves proving the following claims.

Claim 1: $\mathcal{H}_0(\vec{t})$ and $\mathcal{H}_1(\vec{t})$ are computationally indistinguishable.

Distinguishing between the two given distributions is equivalent to solving $\mathsf{HintMLWE}^{\boldsymbol{c}_i^{(0)}, \ldots, \boldsymbol{c}_i^{(r-1)}, \texttt{Ecd}(y^{(i+1)\ell m})}_{R,\nu,q,\sigma_1,\sigma_2,\ldots,\sigma_2,\frac{(b+2)\kappa}{2}\sigma_2}$. Since $\sigma \ge \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ holds under the given conditions for σ_1 and σ_2, $\mathsf{HintMLWE}^{\boldsymbol{c}_i^{(0)}, \ldots, \boldsymbol{c}_i^{(r-1)}, \texttt{Ecd}(y^{(i+1)\ell m})}_{R,\nu,q,\sigma_1,\sigma_2,\ldots,\sigma_2,\frac{(b+2)\kappa}{2}\sigma_2}$ can be reduced from $\mathsf{MLWE}_{R,\nu,q,\sigma}$ by Thm. 1. Therefore, the two given distributions are computationally indistinguishable under the hardness assumption of $\mathsf{MLWE}_{R,\nu,q,\sigma}$.

Claim 2: $\mathcal{H}_1(\vec{t})$ and $\mathcal{H}_2(\vec{t})$ are statistically identical.

Since $\vec{\boldsymbol{t}}_i'^*$'s are uniform random samples from $\mathcal{U}(R_q^\mu)$, the distribution of $\mathbf{A}_0\vec{\boldsymbol{t}}_i + \vec{\boldsymbol{t}}_i'^*$ and $\vec{\boldsymbol{t}}_i'^*$ are identical.

Claim 3: $\mathcal{H}_2(\vec{t})$ and $\mathcal{H}_3$ are within negligible statistical distance.

We use Lem. 11 to prove the claim Let $\mathfrak{S}$ be a positive definitive matrices satisfying $\mathfrak{S}^{-1} = \mathfrak{s}_1^{-2} \cdot (PP^\top)^{-1} + r\mathfrak{s}_2^{-2} \cdot I_n + (\frac{(b+2)\kappa}{2}\mathfrak{s}_2)^{-2} \cdot Y_i^\top Y_i$, where Y_i is the negacyclic matrix corresponding to $\mathtt{Ecd}(y^{(i+1)\ell m})$. Then, we need to show $\sqrt{\mathfrak{S}} \geq \eta_\varepsilon(P\mathbb{Z}^n) \iff P^{-1}\sqrt{\mathfrak{S}} \geq \eta_\varepsilon(\mathbb{Z}^n)$. By Lem. 2, it suffices to show that $\left\|P^\top \mathfrak{S}^{-1} P\right\|_2 \leq \eta_\varepsilon(\mathbb{Z})^{-2}$. Since $\|P\|_2 \leq b+1$, $\left\|P^{-1}\right\|_2 \leq (b-1)^{-1}$, and $\|Y_i\|_2 \leq \frac{(b+2)\kappa}{2}$, we have the followings:

$$\left\|P^\top \mathfrak{S}^{-1} P\right\|_2 \leq (b+1)^2\big((b-1)^{-2}\mathfrak{s}_1^{-2} + (r+1)\mathfrak{s}_2^{-2}\big)$$

Then, it holds that $\left\|P^\top \mathfrak{S}^{-1} P\right\|_2 \leq \eta_\varepsilon(\mathbb{Z}^n)$ under the given conditions for $\mathfrak{s}_1$ and $\mathfrak{s}_2$ Therefore, by Lem. 11, the distributions of $\vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i$ and $\vec{\boldsymbol{e}}_{0,i}^{(j)} + \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i$ are within statistical distance $O(\varepsilon)$ of $\vec{\boldsymbol{n}}_i^{(j)} + \boldsymbol{c}_i^{(j)} \cdot \vec{\boldsymbol{t}}_i'$ and $\vec{\boldsymbol{e}}_{0,i}^{(j)} + \mathtt{Ecd}(y^{(i+1)\ell m}) \cdot \vec{\boldsymbol{t}}_i'$.

By Claim 1,2, and 3, the distributions $\mathcal{H}_0(\vec{t})$ and $\mathcal{H}_3$ are computationally indistinguishable, which implies $\mathcal{D}_0(\vec{t})$ and $\mathcal{D}_1$ are also computationally indistinguishable.

Therefore, $\Pi_{\mathtt{R1CS}}$ satisfies the completeness, soundness, and simulatability under the given conditions. □

Complexity Analysis. We analyze the communication complexity of the protocol $\Pi_{\mathtt{R1CS}}$ and the computational complexity of the prover and the verifier. The communication cost is primarily determined by the size of commitments, which consist of μ ring elements, and responses, which consist of $\mu+\nu+\ell$ ring elements, sent by the prover at each step of the protocol. Therefore, we count the number of them at each step below.

- **Input:** k commitments
- **Step 1:** k commitments
- **Step 2:** 3 commitments
- **Step 3:** $2k+2$ commitments
- **Step 4:** r commitments and r responses for $\Pi_{\mathtt{Open}}$.
- **Step 5:** 3 responses
- **Total:** $(4k+r+5)$ commitments and $(r+3)$ responses

We note that parameters such as q, n, μ, ν, r are determined by the security parameter λ, making them almost independent of the size of the input M. Thus, we can consider each commitment to have a size of $O(1)$. In the case of responses, we can regard them as having a size of $O(\ell \log k)$ since the upper bound of their coefficients has a size of $O(k)$. Therefore, the total communication complexity depends on $O(k + \ell \log k)$. We recall that $M = k\ell m$ holds, so if we set $k = O(\sqrt{M \log M})$, then the total communication complexity is determined as $O(\sqrt{M \log M})$.

For the computational complexity of the prover, it is primarily dominated by two factors: generating commitments and performing matrix and polynomial operations required for constructing arguments for R1CS. For generating

commitments, it takes $O(\mu(\mu+\nu+\ell)) = O(\ell)$ operations in R_q per each commitment. Since a prover generates $O(k)$ commitments, it amounts to a complexity of $O(M)$. In the case of matrix operations, each matrix and vector multiplication, such as in line 5 in Fig. 4, takes $O(N)$ complexity, where $N = \Omega(M)$ represents the number of nonzero entries in the matrices A, B, C. Regarding polynomial operations, the dominating factor is the multiplications in lines 26 to 27 in Fig. 5, which requires $O(M \log M)$ complexity. In summary, the computational complexity for the prover is $O(N \log N)$.

For the verifier, its complexity is dominated by matrix operations in lines 13 to 14 in Fig. 7, which result in a total complexity of $O(N)$.

5 Experimental Results

In this section, we present a proof-of-concept implementation of our sublinear arguments for R1CS 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.[1] For the functionality of the random oracle, we use the SHA-3 hash function. To measure the performance and proof size of Ligero and Aurora, we utilize the libiop library [1], which implements hash-based SNARKs. All experiments were performed with a single thread on a machine with an Intel(R) Xeon(R) Platinum 8268 CPU running at 2.90GHz and 384GB of RAM.

5.1 Parameter Setting

We summarize all parameters that appear in our protocol below.

Type	Parameter	Description
Binding & Hiding	q	commitment modulus
	n	ring dimension
	μ	MSIS rank
	ν	MLWE rank
	$\mathfrak{s}_1, \mathfrak{s}_2, \sigma_1, \sigma_2$	width parameters
Soundness	b	base
	κ	digit lengths
	p	message modulus ($= b^\kappa + 1$)
	r	# repetition
Proof Size	m	messages per ring element ($= n/\kappa$)
	ℓ	ring element per commitment
	k	# commitment
	M	input size ($= k\ell m$)

Table 1. Parameters for R1CS argument

[1] The source code is available at `https://github.com/SNUCP/elsa`.

Firstly, there are parameters that determine the security of the binding and hiding properties of the Ajtai commitment scheme, namely $q, n, \mu, \nu, \mathfrak{s}_1, \mathfrak{s}_2, \sigma_1, \sigma_2$. We recall that, to ensure the binding and hiding properties in our protocol, $\mathsf{MSIS}_{R,\mu,q,2\beta_{\mathtt{R1CS}}}$ and $\mathsf{MLWE}_{R,\mu,q,\nu}$ should be hard, as stated in Thm.3, where $\beta_{\mathtt{R1CS}}$ and σ are constants described in the same theorem. We first set the width parameters as follows $\mathfrak{s}_1 = \sqrt{2} \cdot \frac{b+1}{b-1} \cdot \eta_\varepsilon(\mathbb{Z}^n)$, $\mathfrak{s}_2 = (b+1)\sqrt{2r+2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$, $\sigma_1 = 2\sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$, $\sigma_2 = 2\sqrt{2r+2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$, following the conditions in the theorem. For the smoothing parameter $\eta_\varepsilon(\mathbb{Z}^n)$, we approximate it using the upper bound in Lem.1 by setting $\varepsilon = 2^{-\lambda}$. Once the width parameters are determined, the values of q, n, μ, ν determine the hardness of the MSIS and MLWE problem. To estimate their hardness, we compute the root Hermite factor δ and set it to be approximately 1.0045, similar to [5, 19]. As a result, we use $q \approx 2^{58} \sim 2^{63}$, $n = 2^{11}$, $\mu = \nu = 1$ in our protocol.

The soundness of the protocol is mainly determined by the parameters r and p, where $p = b^\kappa + 1$. In Table 2, we present a list of parameters used in our protocol for security levels approximately equivalent to 2^{128}, 2^{192}, and 2^{256}. For the parameters related to proof size, such as k and ℓ, they are set to be optimal values that result in the smallest proof size for each fixed M.

b	κ	$\lceil \log p \rceil$	r
248	16	128	9
54	32	185	14
156	32	234	18

Table 2. Parameters for Soundness

5.2 Concrete Proof Size

We provide concrete proof sizes based on the parameter settings described in the previous subsection. To calculate the upper bound of each element of commitments, we use the value of q, and for each element of responses, we consider the coefficient upper bounds presented in $\Pi_{\mathtt{Open}}$ and $\Pi_{\mathtt{R1CS}}$. To demonstrate the sublinear growth of proof size, we measure the proof size for input dimensions $M = 2^{18}, 2^{20}$, and 2^{22} in Table 3, alongside Ligero, for field sizes of 128, 192, and 256 bits. We can directly observe that the proof size doubles as the input dimension quadruples, demonstrating the sublinear complexity of our proof system. Regarding the concrete proof size, our system and Ligero have similar proof sizes for a 128-bit field size. However, our system exhibits smaller proof sizes for larger field sizes of 192 and 256 bits.

Input Dimension	128		192		256	
	Ligero	**Ours**	Ligero	**Ours**	Ligero	**Ours**
2^{18}	6.5	**6.3**	14.0	**9.2**	23.1	**11.5**
2^{20}	12.9	**12.8**	28.2	**18.4**	46.8	**22.9**
2^{22}	25.7	**25.7**	56.5	**38.4**	94.4	**47.6**

Table 3. Proof size for R1CS. All units are MB(Megabytes).

5.3 Benchmark Results

We now present benchmark results for the proof generation and verification procedures of our proof system. To optimize ring operations on R_q, we utilize NTT operations, which enable asymptotically faster polynomial multiplications. For obtaining discrete Gaussian samples, we use Karney's algorithm [21].

We measure elapsed time for proof generation and verification for input dimension $M = 2^{18}$, 2^{20}, and 2^{22} when for 128-bits field size. To provide a comparison with other post-quantum SNARKs, we also measured the performance of Ligero and Aurora. We note that all proof systems have the same asymptotic complexity in both proof generation and verification. The results are presented in Table 4 and Fig. 10.

Input Dimension	Prove			Verify		
	Aurora	Ligero	**Ours**	Aurora	Ligero	**Ours**
2^{18}	165	4.8	**8.6**	6.0	3.9	**0.3**
2^{20}	759	20.4	**33.3**	23.8	15.3	**1.1**
2^{22}	3298	80.2	**131.3**	97.4	61.1	**4.1**

Table 4. Benchmark results for our proof system, Ligero, and Aurora. All units are seconds per operation.

In terms of proof generation performance, our proof system is slightly slower than Ligero. This is primarily due to the time taken for generating discrete Gaussian samples in randomized encoding, which accounts for about half of the total time. However, if we were to use rejection sampling instead of our randomized encoding method to achieve simulatability, it could result in much slower performance. In such a scenario, the other half of the elapsed time would be multiplied due to rejection sampling. We also note that our proof system is still significantly faster than Aurora.

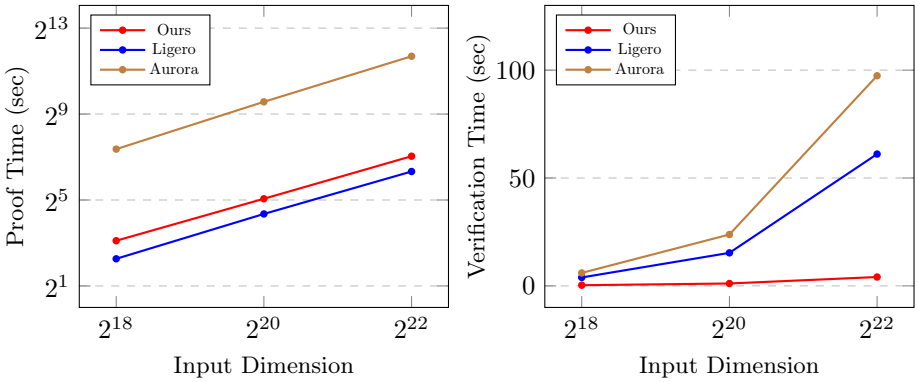

Fig. 10. Benchmark results for our proof system, Ligero, and Aurora. All units are seconds per operation. Proof Times are labeled in logarithmic scale.

For the verification procedure, our proof system outperforms others by an order of magnitude, despite having identical asymptotic complexities. We attribute this result to the efficiency of lattice-based cryptography.

6 Conclusion and Future Work

In this paper, we have introduced a novel post-quantum zero-knowledge proof system for R1CS based on lattice-based cryptography. Prior to this work, there have been few studies [11,31] addressing the construction of efficient lattice-based SNARKs. Furthermore, these studies primarily focused on addressing practical proof size rather than discussing concrete performance in proof generation and verification. Our proof system not only successfully achieves practical proof sizes through an efficient encoding method for large prime fields but also significantly improves practical proof generation and verification performance by introducing a new proof technique based on message randomization. Compared to other post-quantum SNARKs, our proof system yields comparable proof sizes and proof generation performance, while excelling in verification performance, which is an order of magnitude faster than others.

There are still several ways to improve our proof system further. In terms of concrete performance, adopting the discrete Gaussian sampling algorithm from [20] can lead to faster proof generation since generating samples currently consumes a significant portion of the total elapsed time. To improve the proof size, we can improve asymptotic scale by adopting the leveled Ajtai commitment [13]. This approach achieves $O(N^{1/d})$ complexity by generalizing amortized POK protocols. Regarding the verification procedure, the complexity is currently dominated by processing public information. Devising a method for delegating this computation to a prover can lead to sublinear verification complexity, which would be another avenue for improvement.

References

1. libiop. Online: `https://github.com/scipr-lab/libiop`
2. 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)
3. Ames, S., Hazay, C., Ishai, Y., Venkitasubramaniam, M.: Ligero: Lightweight sublinear arguments without a trusted setup. In: Proceedings of the 2017 acm sigsac conference on computer and communications security. pp. 2087–2104 (2017)
4. Attema, T., Cramer, R., Kohl, L.: A compressed σ-protocol theory for lattices. In: Annual International Cryptology Conference. pp. 549–579. Springer (2021)
5. Attema, T., Lyubashevsky, V., Seiler, G.: Practical product proofs for lattice commitments. In: Annual International Cryptology Conference. pp. 470–499. Springer (2020)
6. Banaszczyk, W.: New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen **296**, 625–635 (1993)
7. Baum, C., Bootle, J., Cerulli, A., Del Pino, R., Groth, J., Lyubashevsky, V.: Sublinear 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., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P.: Aurora: Transparent succinct arguments for r1cs. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 103–128. Springer (2019)
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. Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C.: Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In: Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35. pp. 327–357. Springer (2016)
13. 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)
14. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-kyber: a cca-secure module-lattice-based kem. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367. IEEE (2018)
15. 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)
16. 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)

17. 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)
18. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
19. 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)
20. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., et al.: Falcon: Fast-fourier lattice-based compact signatures over ntru. Submission to the NIST's post-quantum cryptography standardization process **36**(5), 1–75 (2018)
21. Karney, C.F.: Sampling exactly from the normal distribution. ACM Transactions on Mathematical Software (TOMS) **42**(1), 1–14 (2016)
22. 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)
23. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
24. Lyubashevsky, V., Nguyen, N.K.: Bloom: Bimodal lattice one-out-of-many proofs and applications. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV. pp. 95–125. Springer (2023)
25. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: shorter, simpler, and more general. In: Annual International Cryptology Conference. pp. 71–101. Springer (2022)
26. 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)
27. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Smile: set membership from ideal lattices with applications to ring signatures and confidential transactions. In: Annual International Cryptology Conference. pp. 611–640. Springer (2021)
28. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I. pp. 204–224. Springer (2018)
29. Mera, J.M.B., Karmakar, A., Marc, T., Soleimanian, A.: Efficient lattice-based inner-product functional encryption. In: IACR International Conference on Public-Key Cryptography. pp. 163–193. Springer (2022)
30. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing **37**(1), 267–302 (2007)
31. Nguyen, N.K., Seiler, G.: Practical sublinear proofs for r1cs from lattices. In: Annual International Cryptology Conference. pp. 133–162. Springer (2022)
32. Peikert, C.: An efficient and parallel gaussian sampler for lattices. In: Annual Cryptology Conference. pp. 80–97. Springer (2010)