Toward Practical Lattice-based Proof of Knowledge from Hint-MLWE

Duhyeong Kim[1], Dongwon Lee[2], Jinyeong Seo[2], and Yongsoo Song[2]

Intel Labs
duhyeong1204@gmail.com
Seoul National University
{dongwonlee95, jinyeong.seo, y.song}@snu.ac.kr

Abstract. In the last decade, zero-knowledge proof of knowledge protocols have been extensively studied to achieve active security of various cryptographic protocols. However, the existing solutions simply seek zero-knowledge for both message and randomness, which is an overkill in many applications since protocols may remain secure even if some information about randomness is leaked to the adversary.
We develop this idea to improve the state-of-the-art proof of knowledge protocols for RLWE-based public-key encryption and BDLOP commitment schemes. In a nutshell, we present new proof of knowledge protocols without using noise flooding or rejection sampling which are provably secure under a computational hardness assumption, called Hint-MLWE. We also show an efficient reduction from Hint-MLWE to the standard MLWE assumption.
Our approach enjoys the best of two worlds because it has no computational overhead from repetition (abort) and achieves a polynomial overhead between the honest and proven languages. We prove this claim by demonstrating concrete parameters and compare with previous results. Finally, we explain how our idea can be further applied to other proof of knowledge providing advanced functionality.

Keywords: Zero-knowledge · Proof of Plaintext Knowledge · BDLOP · Hint-MLWE.

1 Introduction

In the last decade, lattice cryptography has arisen as one of the most promising foundations due to its versatility and robustness against quantum attacks. In particular, it has wide application in advanced cryptographic primitive for privacy-preserving computational protocols such as homomorphic encryption (e.g. [9, 8]), multi-party computation (e.g. [14, 18, 5]) and commitment (e.g. [6, 4, 16]).

When building a protocol secure against an active adversary, we often use a strategy to first design a protocol in the semi-honest model and then compile it into a maliciously secure version. In particular, zero-knowledge proof of knowledge is often used for the compilation step. To the best of our knowledge, there are two major methodology to achieve zero-knowledge in lattice-based proof of knowledge protocols: noise flooding and rejection sampling. In the usual sigma protocols, the prover generates an auxiliary randomness $\mathbf{r}$, receives a challenge γ from the verifier, and then sends a response $\mathbf{z} := \mathbf{y} + \gamma \cdot \mathbf{r}$ where $\mathbf{r}$ is the randomness of the input ciphertext/commitment. First, the noise flooding technique samples $\mathbf{y}$ from an exponentially large distribution to fully hide the information of $\gamma \cdot \mathbf{r}$. On the other hand, the rejection sampling [20] makes the random variable $\mathbf{z}$ independent to $\mathbf{r}$ by manipulating its probability distribution. This technique has an advantage in that the ratio $\|\mathbf{z}\|_2/\|\gamma \cdot \mathbf{r}\|_2$ is very small, but instead can abort the protocol repeatedly until generating an accepting transcript. Both methods commonly aim to prevent any information leakage on the randomness of the input ciphertext/commitment, which results in the semantic security of the protocol including the zero-knowledge of the message.

This work starts from the observation that the previous approach can be an overkill since it provides zero-knowledge for both message and randomness, while the primary goal of the zero-knowledge proof is mostly to ensure that there is no information leakage on the message from the transcripts. In other words, we do not always have to achieve the zero-knowledge for randomness, but it is allowed to reveal some information about it as long as the message privacy is guaranteed.

A natural question is how such partial information leakage of randomness affects the security of proof of knowledge protocols. It is obvious that if an adversary obtains the randomness in clear, then the message can also be recovered directly. Hence, we aim to precisely analyze the conditional distribution of the randomness $\mathbf{r}$ for given $\mathbf{y} + \gamma \cdot \mathbf{r}$, which indicates how much information about the randomness is hidden against the adversary.

We apply this idea to a proof of plaintext knowledge (PPK) protocol for the RLWE-based public-key encryption scheme [8, 17] and a Proof of Opening Knowledge (POK) protocol for the BDLOP commitment scheme [6] and its applications [4, 16]. We show that it is possible to build secure PPK and POK protocols without noise flooding or rejection sampling while achieving a polynomial overhead between the honest and proven languages. Finally, we present concrete parameter sets and show that our solution outperforms the state-of-the-art results.

1.1 Technical Overview

Previous work statistically erased the information of the input randomness by using noise flooding or rejection sampling techniques so that it suffices to consider the standard (algebraic) LWE assumption for the security of public key and ciphertext/commitment. On the other hand, in our work, we allow partial information leakage on the randomness from the transcripts without using such techniques, and we rigorously analyze how such leakage affects the secrecy of the message.

We first explain how the transcripts in each proof of knowledge protocol can be interpreted as an algebraic LWE instance with hints on the LWE secret and error, and then we show how the security reduction can be done under the hardness assumption of a variant of LWE called *Hint-MLWE*.

PPK for RLWE-based Public-Key Encryption. For a public key pk, let $\mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r})$ be a ciphertext which we want to prove the plaintext knowledge through the PPK protocol where m and $\mathbf{r}$ denote the message and encryption randomness, respectively. Then, for zero-knowledge of the message m, we need to prove that the transcript of the protocol does not leak any information on m, which consists of $\mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r})$, random ciphertexts $\mathtt{Enc}_{\mathsf{pk}}(y_i, \mathbf{u}_i)$, challenges γ_i and responses $(v_i, \mathbf{z}_i) := (u_i, \mathbf{y}_i) + \gamma_i \cdot (m, \mathbf{r})$ for $0 \leq i < \ell$. Since $\mathtt{Enc}_{\mathsf{pk}}(v_i, \mathbf{z}_i)$ can be naturally generated from $\mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r})$, $(v_i, \mathbf{z}_i)$ and pk as $\mathtt{Enc}_{\mathsf{pk}}(u_i, \mathbf{y}_i) = \mathtt{Enc}_{\mathsf{pk}}(v_i, \mathbf{z}_i) - \gamma_i \cdot \mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r})$, it is enough to consider the following tuple:

$$\left(\mathsf{pk}, \mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r}), (v_0, \mathbf{z}_0), \ldots, (v_{\ell-1}, \mathbf{z}_{\ell-1})\right)$$

for given challenges $\gamma_0, \ldots, \gamma_{\ell-1}$.

Following the methodology of Chen et al. [11], we adopt the BFV scheme [8, 17] and assume the condition $p \mid q$ for the plaintext modulus p and the ciphertext modulus q, which enables to make each v_i to be uniform random in R_p with $u_i \leftarrow \mathcal{U}(R_p)$ for a polynomial ring $R := \mathbb{Z}[X]/(X^n + 1)$. Since $\mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r}) = \mathsf{pk} \cdot r^{(0)} + ((q/p) \cdot m + r^{(1)}, r^{(2)})$ for $\mathbf{r} := (r^{(0)}, r^{(1)}, r^{(2)}) \in R^3$, the remaining components $(\mathsf{pk}, \mathtt{Enc}_{\mathsf{pk}}(m, \mathbf{r}), \mathbf{z}_1, \ldots, \mathbf{z}_\ell)$ can be re-phrased as the following three tuples:

$$\left(\mathsf{pk}, \mathsf{pk} \cdot r^{(0)} + ((q/p) \cdot m + r^{(1)}, r^{(2)})\right) \qquad \text{(Public Key \& Ciphertext)}$$
$$(\gamma_0 \cdot \mathbf{r} + \mathbf{y}_i, \ldots, \gamma_{\ell-1} \cdot \mathbf{r} + \mathbf{y}_{\ell-1}) \qquad \text{(Hints on the randomness } \mathbf{r}\text{)}$$

POK for BDLOP Commitment. For a commitment key $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$ which consists of two matrices over R_q, let $\mathtt{Com}_{\mathsf{ck}}(\mathbf{m}, \mathbf{r}) = (\mathbf{B}_0\mathbf{r}, \mathbf{B}_1\mathbf{r} + \mathbf{m})$ be the commitment of the message $\mathbf{m}$ with the commitment randomness $\mathbf{r}$, where we want to prove the zero-knowledge of the message through the POK protocol. The transcript of the POK protocol consists of $\mathtt{Com}_{\mathsf{ck}}(\mathbf{m}, \mathbf{r})$, $\mathbf{w} = \mathbf{B}_0\mathbf{y}$, a challenge γ and the response $\mathbf{z} = \gamma \cdot \mathbf{r} + \mathbf{y}$. Similarly to the above PPK, $\mathbf{w}$ can be generated from the other components, and hence it suffices to consider the following tuple $(\mathbf{B}, \mathbf{B}\mathbf{r} + (\mathbf{0}, \mathbf{m})^\top, \gamma \cdot \mathbf{r} + \mathbf{y})$ where $\mathbf{B}$ is the concatenation of $\mathbf{B}_0$ and $\mathbf{B}_1$. In BDLOP, we can express $\mathbf{B} = \mathbf{R} \cdot [\mathbf{I} \mid \mathbf{A}]$ for some invertible matrix $\mathbf{R}$ and a matrix $\mathbf{A}$ where both are public. Therefore, it suffices to show that the secrecy of message $\mathbf{m}$ is guaranteed for arbitrary $\mathbf{m}$ when the tuple $(\mathbf{A}, [\mathbf{I} \mid \mathbf{A}]\mathbf{r} + \mathbf{R}^{-1} \cdot (\mathbf{0}, \mathbf{m})^\top, \gamma \cdot \mathbf{r} + \mathbf{y})$ is given.

Security Reduction from Hint-MLWE. For the security proof, we define a variant of Module-LWE (MLWE), which we call *Hint-MLWE*, and prove that the secrecy of the message m is guaranteed under the hardness assumption of Hint-MLWE. To be precise, the Hint-MLWE problem gives MLWE samples $(\mathbf{A}, [\mathbf{I} \mid \mathbf{A}]\mathbf{r})$ with a bounded number of hints on secret and error as $(\gamma_0 \cdot \mathbf{r} + \mathbf{y}_0, \ldots, \gamma_{\ell-1} \cdot \mathbf{r} + \mathbf{y}_{\ell-1})$ where $\mathbf{A} \leftarrow \mathcal{U}(R_q^{m \times d})$, $\mathbf{r} \leftarrow \chi, \mathbf{y}_i \leftarrow \xi$ for some distributions χ, ξ over R^{d+m}, and $\gamma_0, \ldots, \gamma_{\ell-1}$ are chosen from some distribution $\mathcal{C}$ over

R^ℓ. The Hint-MLWE assumption implies that it is hard to distinguish between the MLWE samples and the uniform samples $(\mathbf{a}, \mathbf{b})$ for $\mathbf{b} \leftarrow \mathcal{U}(R_q^m)$ even if the hints on the secret and error are given.

We can directly apply the Hint-MLWE assumption to the security proofs: For PPK, regarding $(\mathsf{pk}, \mathsf{pk}\cdot r^{(0)}+(r^{(1)}, r^{(2)}))$ as two RLWE samples, the Hint-RLWE[1] assumption under proper parameters implies that one cannot distinguish it from $(\mathsf{pk}, \mathbf{b})$ for $\mathbf{b} \leftarrow \mathcal{U}(R_q^2)$ even when the hints are given. Similarly for POK, the tuple $(\mathbf{A}, [\mathbf{I}\ |\mathbf{A}]\mathbf{r}, \gamma\cdot\mathbf{r}+\mathbf{y})$ is computationally indistinguishable with $(\mathbf{A}, \mathbf{u}, \gamma\cdot\mathbf{r}+\mathbf{y})$ for an uniform random vector $\mathbf{u}$ under the Hint-MLWE assumption. Therefore, there is no information leakage on the input message for both PPK and POK protocols.

Hardness of Hint-MLWE. We also prove that there exists an efficient reduction from standard MLWE to Hint-MLWE under a discrete Gaussian setting, so we can apply analyses on the concrete hardness of MLWE which has been studied extensively. Roughly speaking, when we set χ, ξ as discrete Gaussian with width σ_1, σ_2 respectively, then it corresponds to the MLWE problem with secret key and error distributions as discrete Gaussian with width parameter σ, such that $\frac{1}{\sigma^2} = 2(\frac{1}{\sigma_1^2} + \frac{B}{\sigma_2^2})$ for some $B > 0$ determined by the challenge distribution $\mathcal{C}$. Therefore, the concrete hardness of Hint-MLWE is measured by the width parameters σ of underlying MLWE.

Our reduction result gives several implications on the effect of information leakage from hints to security. For example, setting $\sigma_2 \gg \sigma_1$ corresponds to the noise flooding technique and our reduction gives the same result: We can easily check $\sigma \approx \sigma_1$, which implies the hints does not leak any non-negligible information of the secret. It also provides the relation between the number of hints and the hardness of Hint-MLWE: When there are exponentially many hints given so that $\sigma \to 0$, then the underlying MLWE would easily solvable, which implies the hints leak entire information of secret.

1.2 Related Works

In previous literature, there have been proposed several variants of LWE with different forms of side information. In [2, 27], a variant called extended-LWE was firstly proposed which gives a hint on LWE secret and error vectors in a form of a "noisy" inner product, i.e., $(\mathbf{A}, [\mathbf{I}\ |\mathbf{A}]\mathbf{r}, \langle \mathbf{r}, \mathbf{z}\rangle + f)$ for a small integer f and given small vectors $\mathbf{z}$, with a reduction from standard LWE. Later, extended-LWE has been modified in various forms according to its usage. In [10], for example, the noisy hint was substituted by the "exact" inner product (*i.e.,* $f = 0$), and the problem was generalized into the multi-secret version, which was used to prove the hardness of LWE with a binary secret. In [23], the exact inner product hint was restricted to its "sign" value, and this variant of extend-LWE was applied for the construction of an efficient lattice-based commitment scheme.

[1] Note that we can naturally define the Hint-RLWE problem as a special case ($d = 1$) of Hint-MLWE.

Note that all the above variants of extended-LWE have commonly been proved to be computationally hard under non-algebraic setting[2], while the hardness of Hint-LWE can also be proved in algebraic setting (in both ring and module settings).

The Hint-LWE problem was firstly defined in [19], which publishes a hint on the LWE error with *additive* Gaussian noise, i.e., $(\mathbf{A}, \mathbf{As} + \mathbf{e}, \mathbf{e} + \mathbf{f})$ for a small vector $\mathbf{f}$. The main differences between the Hint-LWE problems in [19] and our paper are as following: (1) We consider *multiple* hints on *both* LWE secret and error, while [19] only considers a single hint on LWE error, (2) We also consider the *multiplication of challenges* to the LWE secret and error in the hints while [19] did not, (3) We prove the hardness of Hint-LWE under *discrete* Gaussian setting while [19] uses continuous Gaussian (Hence, [19] is not able to consider the hint on LWE secret which should be discrete). A multi-secret version was considered in [19], but we note that our Hint-LWE problem can also be naturally generalized to the multi-secret version.

There has been a recent work that analyzes the concrete security of LWE with side information [13]. It considers various forms of hints (e.g., exact, approximate, modulo inner product, short vector) and shows how these hints can be used to improve the performance of the LWE attack algorithms. Note that a hint with a (discrete) Gaussian noise in this paper has not been considered in [13], which might be of independent interest.

2 Preliminaries

2.1 Notation

We use bold lower-case and upper-case letters to denote column vectors, and matrices respectively. 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.

Let n be a power of two and q be an integer. 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 a polynomial $f = \sum_{i=0}^{n-1} f_i X^i \in R$, the ℓ^p $(p \geq 1)$ and ℓ^∞ norms are defined as follows:

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

For a vector of polynomials $\mathbf{f} = (f^{(0)}, \ldots, f^{(m-1)}) \in R^m$, we write

$$\|\mathbf{f}\|_p := \sqrt[p]{\sum_{i=0}^{m-1} \left\|f^{(i)}\right\|_p^p}, \qquad \|\mathbf{f}\|_\infty := \max_{0 \leq i < m} \left\|f^{(i)}\right\|_\infty$$

[2] In [23], the authors applied the module extended-LWE to instantiate the commitment scheme, but they only provided the proof for plain extended-LWE.

For a polynomial $c \in R$, we denote the vector of its coefficients by a bold letter $\mathbf{c}$ and the corresponding negacyclic matrix by $\mathbf{M}(c)$. For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we denote the matrix norm of $\mathbf{A}$ by $\|\mathbf{A}\|_2 := \max_{\mathbf{x} \in \mathbb{R}^n} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$.

We denoted the largest and the smallest singular value of a real-value matrix $\mathbf{A}$ by $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$, respectively.

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 $\rho_{\mathbf{c}} : \mathbb{R}^n \to (0, 1]$ centered at $\mathbf{c} \in \mathbb{R}^n$ as $\rho_{\mathbf{c}}(\mathbf{x}) := \exp(-\pi \cdot (\mathbf{x} - \mathbf{c})^\top (\mathbf{x} - \mathbf{c}))$. In general, for a positive definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$, we define the elliptical Gaussian function $\rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}} : \mathbb{R}^n \to (0, 1]$ as $\rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{x}) := \exp(-\pi \cdot (\mathbf{x} - \mathbf{c})^\top \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{c}))$.

Let $\Lambda \subseteq \mathbb{R}^n$ be a lattice and $\mathbf{v} \in \mathbb{R}^n$. The discrete Gaussian distribution $\mathcal{D}_{\mathbf{v}+\Lambda, \mathbf{c}, \sqrt{\mathbf{\Sigma}}}$ is defined as a distribution over the coset $\mathbf{v} + \Lambda$, whose probability mass function is $\mathcal{D}_{\mathbf{v}+\Lambda, \mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{x}) = \rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{x}) / \rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{v} + \Lambda)$ for $\mathbf{x} \in \mathbf{v} + \Lambda$ where $\rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{v} + \Lambda) := \sum_{\mathbf{y} \in \mathbf{v} + \Lambda} \rho_{\mathbf{c}, \sqrt{\mathbf{\Sigma}}}(\mathbf{y}) < \infty$. Note that $\mathcal{D}_{\mathbf{v}+\Lambda, \mathbf{c}, \sqrt{\mathbf{\Sigma}}}$ is identical to the distribution of $\mathbf{c} + \mathbf{x}$ where $\mathbf{x} \leftarrow \mathcal{D}_{(\mathbf{v}-\mathbf{c})+\Lambda, \mathbf{0}, \sqrt{\mathbf{\Sigma}}}$. When $\mathbf{c} = \mathbf{0}$, then we omit $\mathbf{c}$ in the subscripts of both ρ and $\mathcal{D}$. When $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ for $\sigma > 0$ where $\mathbf{I}_n$ is the $(n \times n)$ identity matrix, then we substitute $\sqrt{\mathbf{\Sigma}}$ by σ in the subscript and refer to σ as the width parameter of $\mathcal{D}_{\Lambda, \mathbf{c}, \sigma}$. We denote by $x \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma}$ for $x \in R$ when we sample its corresponding coefficient vector $\mathbf{x}$ from $\mathcal{D}_{\mathbb{Z}^n, \sigma}$.

2.3 Module SIS/LWE

Definition 1. *Let m, d be positive integers, and $0 < \beta < q$. Then, the goal of the Module-SIS (MSIS) problem is to find, for a given matrix $\mathbf{A} \leftarrow \mathcal{U}(R_q^{m \times d})$, $\mathbf{x} \in R_q^d$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$ and $\|\mathbf{x}\|_2 \leq \beta$. We say that a PPT adversary $\mathcal{A}$ has advantages ε in solving $\mathsf{MSIS}_{R,d,m,q,\beta}$ if*

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

Definition 2. *Let d, m, q be positive integers, and χ be a distribution over R^{d+m}. Then, the goal of the Module-LWE (MLWE) problem is to distinguish $(\mathbf{A}, \mathbf{u})$ from $(\mathbf{A}, [\mathbf{I}_m | \, \mathbf{A}]\mathbf{r})$ for $\mathbf{A} \leftarrow \mathcal{U}(R_q^{m \times d})$, $\mathbf{u} \leftarrow \mathcal{U}(R_q^m)$, and $\mathbf{r} \leftarrow \chi$. We say that a PPT adversary $\mathcal{A}$ has advantages ε in solving $\mathsf{MLWE}_{R,d,m,q,\chi}$ if*

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

The MLWE problem with $d = 1$ is called the Ring-LWE problem and denoted by $\mathsf{RLWE}_{R,m,q,\chi}$.

2.4 RLWE-based Public-Key Encryption

We describe the BFV scheme [8, 17], which is a standard RLWE-based public-key encryption with homomorphic property, to describe our PPK protocol.

• $\underline{\texttt{BFV.Setup}(1^\lambda)}$: Given a security parameter λ, outputs the parameter set $\mathsf{pp} = (R, q, p, \chi)$ where χ is a distribution over R^2, and p, q are odd integers such that $p \mid q$.

The parameters p, q do not need to satisfy $p \mid q$ in general, but Chen et al. [11] introduced this condition to simplify the proof of plaintext knowledge. We make the same assumption to take its advantage in the protocol construction. The scaling factor will be denoted by $\Delta := q/p \in \mathbb{Z}$.

• $\underline{\texttt{BFV.Gen}(\mathsf{pp})}$: Given a public parameter $\mathsf{pp} = (R, q, p, \chi)$, sample $(e, s) \leftarrow \chi$. Outputs a secret key s and a public key $\mathsf{pk} = (-a{\cdot}s{+}e, a) \in R_q^2$ where $a \leftarrow \mathcal{U}(R_q)$.

• $\underline{\texttt{BFV.Enc}_{\mathsf{pk}}(m, \mathbf{r})}$: For a public key $\mathsf{pk} = (b, a)$, message $m \in R_p$, and encryption randomness $\mathbf{r} = (r_0, r_1, r_2) \in R^3$, output the ciphertext $\mathbf{c} = r_2 \cdot (b, a) + (r_0 + \Delta \cdot m, r_1) \pmod q$.

• $\underline{\texttt{BFV.Dec}(s, \mathbf{c})}$: For a secret key s and a ciphertext $\mathbf{c} = (c_0, c_1) \in R_q^2$, output $m = \lfloor \Delta^{-1} \cdot (c_0 + c_1 \cdot s) \rceil \pmod p$.

The encryption randomness $\mathbf{r}$ is generally chosen to be small so that the decryption works correctly. Note that the additive homomorphism holds for both message and randomness: For messages $m_1, m_2 \in R_p$, $\gamma \in R$, and $\mathbf{r}_1, \mathbf{r}_2 \in R^3$, it holds that

$$\begin{aligned} \texttt{Enc}_{\mathsf{pk}}(m_1, \mathbf{r}_1) + \texttt{Enc}_{\mathsf{pk}}(m_2, \mathbf{r}_2) &= \texttt{Enc}_{\mathsf{pk}}(m_1 + m_2, \mathbf{r}_1 + \mathbf{r}_2) \pmod q \\ \gamma \cdot \texttt{Enc}_{\mathsf{pk}}(m_1, \mathbf{r}_1) &= \texttt{Enc}_{\mathsf{pk}}(\gamma \cdot m_1, \gamma \cdot \mathbf{r}_1) \pmod q. \end{aligned}$$

2.5 Lattice-based Commitment Scheme

We first recall the definition of commitment scheme.

Definition 3 (Commitment Scheme). *A commitment scheme consists of the following three algorithms:*

- $\texttt{Gen}(1^\lambda)$*: Given a security parameter* λ*, it generates a commitment key* ck.
- $\texttt{Com}_{\mathsf{ck}}(m, r)$*: Given a commitment key* ck*, a message* m*, and randomness* r*, it outputs a commitment* c.
- $\texttt{Open}_{\mathsf{ck}}(c, m, r)$*: Given a commitment* c*, a message* m*, and randomness* r*, it outputs either 0 or 1.*

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

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

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

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

$$\Pr\left[(\mathtt{Open}_{\mathsf{ck}}(c,m,r) = \mathtt{Open}_{\mathsf{ck}}(c,m',r') = 1) \wedge (m \neq m') \;\middle|\; \substack{\mathsf{ck}\leftarrow\mathtt{Gen}(1^\lambda); \\ (c,m,r,m',r')\leftarrow\mathcal{A}(\mathsf{ck})} \right].$$

Below, we present the BDLOP commitment scheme, whose binding and hiding properties rely on the hardness of $\mathsf{MSIS}_{R,\mu,q,\beta_{\mathtt{BDLOP}}}$ and $\mathsf{MLWE}_{R,\nu,q,\chi}$, respectively, where χ is a distribution for commitment randomness. We refer the reader to [6] for more details.

• $\underline{\mathtt{BDLOP.Gen}(1^\lambda)}$: Given a security parameter λ, it outputs a commitment key $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$ which are generated as follows:

– $\mathbf{B}_0 = \left[\mathbf{I}_\mu |\ \mathbf{B}_0'\right] \in R_q^{\mu\times(\mu+\nu+k)}$ where $\mathbf{B}_0' \leftarrow \mathcal{U}(R_q^{\mu\times(\nu+k)})$.
– $\mathbf{B}_1 = \left[\mathbf{0}^{k\times\mu} |\ \mathbf{I}_k |\ \mathbf{B}_1'\right] \in R_q^{k\times(\mu+\nu+k)}$ where $\mathbf{B}_1' \leftarrow \mathcal{U}(R_q^{k\times\nu})$.

• $\underline{\mathtt{BDLOP.Com}_{\mathsf{ck}}(\mathbf{m},\mathbf{r})}$: Given a commitment key ck, a message $\mathbf{m} \in R_q^k$, and randomness $\mathbf{r} \in R^{\mu+\nu+k}$, it outputs $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$ where $\mathbf{c}_0 = \mathbf{B}_0\mathbf{r} \pmod q$ and $\mathbf{c}_1 = \mathbf{B}_1\mathbf{r} + \mathbf{m} \pmod q$.

• $\underline{\mathtt{BDLOP.Open}_{\mathsf{ck}}(\mathbf{c},\mathbf{m},\mathbf{r})}$: Given a commitment $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$, a message $\mathbf{m}$, and randomness $\mathbf{r}$, it outputs 1 if and only if $\mathbf{c} = \mathtt{BDLOP.Com}_{\mathsf{ck}}(\mathbf{m},\mathbf{r})$ and $\|\mathbf{r}\|_2 \leq \beta_{\mathtt{BDLOP}}$.

In [6], there is a weaker version of opening algorithm supporting for efficient proof of opening knowledge, which we will describe in Sec. 5.1. The commitment scheme also satisfy the additive homomorphism for both message and randomness as well as the BFV scheme.

2.6 Proof of Knowledge and Simulatability

In this subsection, we present a new approach to building a secure proof-of-knowledge protocol. The conventional construction involves a zero-knowledge proof for the prover's secret input and randomness used in generating statements to be proved. However, our new definition primarily relies on the idea that the leakage of some information on randomness does not lead to an attack against the prover's secret input, which is formally described below.

Definition 4. *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 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:*

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

In this definition, we reformulate zero-knowledge condition on the prover's secret input by simulatability. The main difference between our simulatability property and the conventional zero-knowledge proof is whether randomness is perfectly hidden together or not. Since the essential purpose of secure proof-of-knowledge protocol is to hide the prover's secret input rather than a randomness, it suffices to satisfy our simulatability property for the desired security requirement. It is worth noting that similar approaches have been considered in [15, 23].

Our definition utilizes two languages $\mathbf{L} \subseteq \mathbf{L}'$, called the honest and proven languages respectively, to address common scenarios in lattice-based construction. There have been studies, such as [7, 25], which reduce the communication cost by weakening extractors' power in the knowledge soundness property. Since our instantiations of proof-of-knowledge in this paper also employ these methods, our definition makes use of two languages to cover these cases. The gap between $\mathbf{L}$ and $\mathbf{L}'$ is often referred as *soundness slack*.

2.7 Useful Lemmas

Lemma 1 ([20, Lemma 4.4]). *For any $k > 0$,* $\Pr\left[\|\mathbf{x}\|_\infty < k\sigma \mid \mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}\right] > 1 - 2n \cdot \exp(-\pi k^2)$.

Lemma 2 ([4, Lemma 2.5]). $\Pr\left[\|\mathbf{x}\|_2 < \sigma\sqrt{n/\pi} \mid \mathbf{x} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}\right] > 1 - 2^{-n/8}$.

Lemma 3 (Simplified Convolution Lemma [28]). *Let $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2$ be positive definite matrices such that $\mathbf{\Sigma}_3^{-1} := \mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}$ satisfies $\sqrt{\mathbf{\Sigma}_3} \geq \eta_\varepsilon(\mathbb{Z}^n)$ for $0 < \varepsilon < 1/2$. Then for an arbitrary $\mathbf{c} \in \mathbb{Z}^n$, the distribution*

$$\left\{\mathbf{x}_1 + \mathbf{x}_2 \mid \mathbf{x}_1 \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sqrt{\mathbf{\Sigma}_1}},\ \mathbf{x}_2 \leftarrow \mathcal{D}_{\mathbb{Z}^n, \mathbf{c}, \sqrt{\mathbf{\Sigma}_2}}\right\}$$

is within statistical distance 2ε of $\mathcal{D}_{\mathbb{Z}^n, \mathbf{c}, \sqrt{\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2}}$.

Definition 5 (Smoothing parameter [26]). *For an n-dimensional lattice Λ and positive real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^* \backslash \{\mathbf{0}\}) \leq \varepsilon$.*

Definition 6 ([28, Definition 2.3]). *Let $\mathbf{\Sigma}$ be a positive-definite matrix. We say that $\sqrt{\mathbf{\Sigma}} \geq \eta_\varepsilon(\Lambda)$ if $\eta_\varepsilon(\sqrt{\mathbf{\Sigma}}^{-1} \cdot \Lambda) \leq 1$, i.e., $\rho\left(\sqrt{\mathbf{\Sigma}}^\top \cdot \Lambda^* \backslash \{\mathbf{0}\}\right) \leq \varepsilon$.*

Lemma 4 ([26, Lemma 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 $\dim(\mathrm{span}(\Lambda \cap r\mathcal{B})) = n$ and $\mathcal{B}$ is the n-dimensional unit ball centered at the origin.

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

Proof. Note that the matrix norm equals to the largest singular value, and hence $\sqrt{\sigma_{\min}(\mathbf{\Sigma})} = 1/\sqrt{\sigma_{\max}(\mathbf{\Sigma}^{-1})} = 1/\sqrt{\|\mathbf{\Sigma}^{-1}\|_2} \geq \eta_\varepsilon(\Lambda)$. Therefore, it holds that $\sum_{\mathbf{x} \in \Lambda^* \backslash \{\mathbf{0}\}} \exp\left(-\pi \sigma_{\min}(\mathbf{\Sigma}) \cdot \mathbf{x}^\top \mathbf{x}\right) \leq \varepsilon$ by Def. 5.

Since $\mathbf{\Sigma}$ is positive-definite, it holds that $\mathbf{x}^\top \mathbf{\Sigma} \mathbf{x} \geq \sigma_{\min}(\mathbf{\Sigma}) \cdot \mathbf{x}^\top \mathbf{x}$ for any $\mathbf{x} \in \Lambda^*$, and we obtain

$$\sum_{\mathbf{x} \in \Lambda^* \backslash \{\mathbf{0}\}} \exp(-\pi \cdot \mathbf{x}^\top \mathbf{\Sigma} \mathbf{x}) \leq \sum_{\mathbf{x} \in \Lambda^* \backslash \{\mathbf{0}\}} \exp\left(-\pi \sigma_{\min}(\mathbf{\Sigma}) \cdot \mathbf{x}^\top \mathbf{x}\right) \leq \varepsilon,$$

which implies $\eta_\varepsilon(\sqrt{\mathbf{\Sigma}}^{-1} \cdot \Lambda) \leq 1$. □

Lemma 6 ([7, Lemma 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 $\mathbb{Q}[X]/(X^n + 1)$.

3 Hint-MLWE

In this section, we introduce a variant of the MLWE problem called *Hint-MLWE*. The Hint-MLWE problem is inspired by the structure of transcripts generated by lattice-based proof of knowledge protocols. They often include partial information about secret values such as the MLWE secret and the errors in MLWE instances, which are obtained by adding random errors to them. Since these 'hints' on the secret values may affect the security of MLWE, noise flooding or rejection sampling have utilized to ensure that no useful information is leaked from a transcript.

Apart from these previous approaches, we aim to precisely measure how much information on the secret values can be leaked from a transcript and its impact on the security of the protocol. In this context, we come up with the Hint-MLWE problem where the adversary is given the MLWE problem with some hints about secrets and errors. As expected, this problem is useful for proving the security of proof-of-knowledge protocols which we will deal with in Sec. 4 and 5.

To return, we will show that our goal can be achieved if both the secret values and the errors for generating hints are drawn from (discrete) Gaussian distributions by precisely analyzing the conditional distribution of the secret values for given hints. We start by giving a formal definition of the Hint-MLWE problem.

Definition 7 (The Hint-MLWE Problem). *Let d, m, ℓ be positive integers, χ, ξ be distributions over R^{d+m}, and $\mathcal{C}$ be a distribution over R^{ℓ}. The Hint-MLWE problem, denoted by $\mathsf{HintMLWE}^{\ell,\xi,\mathcal{C}}_{R,d,m,q,\chi}$, asks an adversary $\mathcal{A}$ to distinguish the following two cases:*

1. $\left(\mathbf{A}, [\mathbf{I}_m|\mathbf{A}]\mathbf{r}, \gamma_0, \ldots, \gamma_{\ell-1}, \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ *for* $\mathbf{A} \leftarrow \mathcal{U}(R_q^{m\times d})$, $\mathbf{r} \leftarrow \chi$, $\mathbf{y}_i \leftarrow \xi$, $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{C}$, *and* $\mathbf{z}_i = \gamma_i \cdot \mathbf{r} + \mathbf{y}_i$ *for* $0 \le i < \ell$.
2. $\left(\mathbf{A}, \mathbf{u}, \gamma_0, \ldots, \gamma_{\ell-1}, \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ *for* $\mathbf{A} \leftarrow \mathcal{U}(R_q^{m\times d})$, $\mathbf{u} \leftarrow \mathcal{U}(R_q^m)$, $\mathbf{r} \leftarrow \chi$, $\mathbf{y}_i \leftarrow \xi$, $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{C}$, *and* $\mathbf{z}_i = \gamma_i \cdot \mathbf{r} + \mathbf{y}_i$ *for* $0 \le i < \ell$.

We call the $d = 1$ case of Hint-MLWE as the Hint-RLWE problem and denote it by $\mathsf{HintRLWE}^{\ell,\xi,\mathcal{C}}_{R,m,q,\chi}$.

We often refer $(\mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1})$ as hints since it contains partial information about the secret $\mathbf{r}$. When χ and ξ are spherical discrete Gaussian distributions, we replace them with their width parameters in the Hint-MLWE notation for simplicity.

Lemma 7. *Let $\ell > 0$ be an integer and $\sigma_1, \sigma_2 > 0$ be reals. For $\gamma_0, ..., \gamma_{\ell-1} \in R$, let $\mathbf{\Gamma}_i$ be the negacyclic matrix corresponding to γ_i and $\mathbf{\Sigma}_0 := (\frac{1}{\sigma_1^2} \cdot \mathbf{I} + \frac{1}{\sigma_2^2} \cdot \sum_{i=0}^{\ell-1} \mathbf{\Gamma}_i^\top \mathbf{\Gamma}_i)^{-1}$. Then, the following two distributions over $R^{\ell+1}$ are statistically identical:*

$$\left\{(r, z_0, \ldots, z_{\ell-1}) \;\middle|\; r \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1},\ y_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2},\ z_i = \gamma_i \cdot r + y_i\right\}$$

$$\left\{(\hat{r}, z_0, \ldots, z_{\ell-1}) \;\middle|\; \begin{array}{c} r \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1},\ y_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2},\ z_i = \gamma_i \cdot r + y_i, \\ \mathbf{c} = \frac{1}{\sigma_2^2}\mathbf{\Sigma}_0 \cdot \sum_{i=0}^{\ell-1} \mathbf{\Gamma}_i^\top \mathbf{z}_i,\ \hat{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\mathbf{c},\sqrt{\mathbf{\Sigma}_0}} \end{array}\right\}$$

Proof. We claim that two random variables have the same probability mass function. The probability that the first random variable outputs $(v, w_0, \ldots, w_{\ell-1}) \in R^{\ell+1}$ can be written as following:

$$\begin{aligned}
&\Pr\left[r = v,\ \gamma_i \cdot r + y_i = w_i \mid r \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}, y_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}\right] \\
=&\mathcal{D}_{\mathbb{Z}^n,\sigma_1}(\mathbf{v}) \cdot \prod_{i=0}^{\ell-1} \mathcal{D}_{\mathbb{Z}^n,\sigma_2}(\mathbf{w}_i - \mathbf{\Gamma}_i \mathbf{v}) \\
\propto& \exp\left[-\pi\left(\frac{1}{\sigma_1^2}\cdot \mathbf{v}^\top \mathbf{v} + \frac{1}{\sigma_2^2}\cdot \sum_{i=0}^{\ell-1} (\mathbf{w}_i - \mathbf{\Gamma}_i \mathbf{v})^\top (\mathbf{w}_i - \mathbf{\Gamma}_i \mathbf{v})\right)\right] \\
=& \exp\left[-\pi\left((\mathbf{v}-\mathbf{c})^\top \Sigma_0^{-1}(\mathbf{v}-\mathbf{c}) - \mathbf{c}^\top \Sigma_0^{-1}\mathbf{c} + \frac{1}{\sigma_2^2}\cdot \sum_{i=0}^{\ell-1} \mathbf{w}_i^\top \mathbf{w}_i\right)\right]
\end{aligned}$$

where $\mathbf{c} = \frac{1}{\sigma_2^2}\mathbf{\Sigma}_0 \cdot \sum_{i=0}^{\ell-1} \mathbf{\Gamma}_i^\top \mathbf{w}_i$.

Hence, the conditional probability $\Pr\left[r = v \mid \gamma_i \cdot r + y_i = w_i\right]$ is proportional to $\exp\left[-\pi(\mathbf{v}-\mathbf{c})^\top \mathbf{\Sigma}_0^{-1}(\mathbf{v}-\mathbf{c})\right]$ for any $w_1, \ldots, w_\ell \in R$, which implies

$$\Pr\left[r = v \mid \gamma_i \cdot r + y_i = w_i\right] \equiv \rho_{\sqrt{\mathbf{\Sigma}_0}}(\mathbf{v}-\mathbf{c}) \equiv \Pr\left[\hat{r} = v \mid \gamma_i \cdot r + y_i = w_i\right].$$

Therefore, the given two distributions are statistically identical. □

At a high level, Lem. 7 implies that the conditional distribution of r given $(\gamma_0 \cdot r + y_0 \ldots, \gamma_{\ell-1} \cdot r + y_{\ell-1})$ follows a (possibly not balanced) discrete Gaussian distribution again. Namely, the distribution of the first component of r given $z_i = \gamma_i \cdot r + y_i$ can be expressed as the Gaussian distribution over $\mathbb{Z}^n$ with parameters $\mathbf{c} = \frac{1}{\sigma_2^2}\mathbf{\Sigma}_0 \cdot \sum_{i=0}^{\ell-1} \mathbf{\Gamma}_i^\top \mathbf{z}_i$ and $\mathbf{\Sigma}_0$. Based on Lem. 7, we prove the hardness of Hint-MLWE under the MLWE assumption when the secret and errors are sampled from discrete Gaussian distributions.

Theorem 1 (Hardness of Hint-MLWE). *Let d, k, m, q, ℓ be positive integers and $\mathcal{C}$ be a distribution over R^ℓ. Let $B > 0$ be a real number which satisfies $\left\|\sum_{j=0}^{\ell-1} \mathbf{M}(\gamma_j)^\top \mathbf{M}(\gamma_j)\right\|_2 \leq B$ for any possible $(\gamma_0, \ldots, \gamma_{\ell-1})$ sampled from $\mathcal{C}$. For $\sigma_1, \sigma_2 > 0$, let $\sigma > 0$ be a real number defined as $\frac{1}{\sigma^2} = 2(\frac{1}{\sigma_1^2} + \frac{B}{\sigma_2^2})$. If $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ for $0 < \varepsilon \leq 1/2$, then there exists an efficient reduction from $\mathsf{MLWE}_{R,d,m,q,\sigma}$ to $\mathsf{HintMLWE}_{R,d,m,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}}$ that reduces the advantage by at most $(d+m) \cdot 2\varepsilon$.*

Proof. Let $(\gamma_1, ..., \gamma_\ell) \leftarrow \mathcal{C}$, and let $\mathbf{\Sigma}_0 = (\sigma_1^{-2} \cdot \mathbf{I}_n + \sigma_2^{-2} \cdot \sum_{j=0}^{\ell-1} \mathbf{\Gamma}_j^\top \mathbf{\Gamma}_j)^{-1}$ where $\mathbf{\Gamma}_j := \mathbf{M}(\gamma_j)$ is the corresponding negacyclic matrix of γ_j for $0 \leq j < \ell$.

Let $(\mathbf{A}, \mathbf{b}) \in R_q^{m \times d} \times R_q^m$ be given $\mathsf{MLWE}_{R,d,m,q,\sigma}$ instances. Our reduction starts by sampling some polynomials in R:

$$r_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1},\ y_{i,j} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2} \text{ for } 0 \leq i < d+m, \text{ and } 0 \leq j < \ell$$

$$t_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\mathbf{c}_i,\sqrt{\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n}} \text{ for } \ \mathbf{c}_i = \frac{1}{\sigma_2^2}\mathbf{\Sigma}_0 \cdot \sum_{j=0}^{\ell-1} \mathbf{\Gamma}_j^\top (\mathbf{\Gamma}_j \mathbf{r}_i + \mathbf{y}_{i,j}) \text{ and } 0 \leq i < d+m$$

We write $(r_0, \ldots, r_{d+m-1})$, $(y_0, \ldots, y_{d+m-1})$, and $(t_0, \ldots, t_{d+m-1})$ as $\mathbf{r}$, $\mathbf{y}$, and $\mathbf{t}$ respectively. Note that $\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n$ is positive-definite, since the smallest singular value of $\mathbf{\Sigma}_0$ is $\left(\sigma_1^{-2} + \sigma_2^{-2} \cdot \left\|\sum_{j=0}^{\ell-1} \mathbf{\Gamma}_j^\top \mathbf{\Gamma}_j\right\|_2\right)^{-1} \geq (\sigma_1^{-2} + \sigma_2^{-2} \cdot B)^{-1} = 2\sigma^2 > \sigma^2$.

Then, we use the sampled polynomials to transform the given MLWE sample $(\mathbf{A}, \mathbf{b})$ into $\left(\mathbf{A},\ \mathbf{b} + [\mathbf{I}_m\ |\mathbf{A}]\mathbf{t},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ where $\mathbf{z}_j = \gamma_j \cdot \mathbf{r} + \mathbf{y}_j$ for $0 \leq j < \ell$, which are the output of the reduction.

We first assume that $\mathbf{b} = [\mathbf{I}_m\ |\mathbf{A}]\mathbf{r}'$ for $\mathbf{r}' \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma}^{d+m}$. Then, we have $\mathbf{b} + [\mathbf{I}_m\ |\mathbf{A}]\mathbf{t} = [\mathbf{I}_m\ |\mathbf{A}](\mathbf{r}' + \mathbf{t})$ where $\mathbf{r}' + \mathbf{t}$ follows the distributions $\prod_{i=0}^{d+m-1}(\mathcal{D}_{\mathbb{Z}^n,\sigma} + \mathcal{D}_{\mathbb{Z}^n,\mathbf{c}_i,\sqrt{\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n}})$.

Now we show that $\sqrt{\mathbf{\Sigma}_3} \geq \eta_\varepsilon(\mathbb{Z}^n)$ where $\mathbf{\Sigma}_3^{-1} := \sigma^{-2} \cdot \mathbf{I}_n + (\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n)^{-1}$. By Lem. 5, it is enough to show that $\left\|\mathbf{\Sigma}_3^{-1}\right\|_2 \leq \eta_\varepsilon(\mathbb{Z}^n)^{-2}$. Recall that the smallest singular value of $\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n$ is at least σ^2 as discussed above. Therefore, it holds that

$$\left\|\mathbf{\Sigma}_3^{-1}\right\|_2 = \sigma^{-2} + \left\|(\mathbf{\Sigma}_0 - \sigma^2 \cdot \mathbf{I}_n)^{-1}\right\|_2 \leq \sigma^{-2} + \sigma^{-2} = 2\sigma^{-2} \leq \eta_\varepsilon(\mathbb{Z}^n)^{-2}.$$

By Lem. 3, the distributions $\mathcal{D}_{\mathbb{Z}^n,\sigma} + \mathcal{D}_{\mathbb{Z}^n,\mathbf{c}_i,\sqrt{\mathbf{\Sigma}_0 - \sigma^2 \mathbf{I}_n}}$ are within the statistical distance 2ε of $\mathcal{D}_{\mathbb{Z}^n,\mathbf{c}_i,\sqrt{\mathbf{\Sigma}_0}}$. Therefore, the distribution of

$$\left(\mathbf{A},\ \mathbf{b} + [\mathbf{I}_m\ |\mathbf{A}]\mathbf{t},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$$

is within statistical distance $(d+m) \cdot 2\varepsilon$ of

$$\left(\mathbf{A},\ [\mathbf{I}_m\ |\mathbf{A}]\hat{\mathbf{r}},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right) \text{ for } \hat{\mathbf{r}} \leftarrow \prod_{i=0}^{d+m-1} \mathcal{D}_{\mathbb{Z}^n,\mathbf{c}_i,\sqrt{\mathbf{\Sigma}_0}}.$$

As the last step, we apply Lem. 7 on $(\hat{\mathbf{r}}, \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1})$, then its distribution is identical to that of $(\mathbf{r}, \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1})$. As a result, the distribution of $\left(\mathbf{A},\ [\mathbf{I}_m\ |\mathbf{A}]\hat{\mathbf{r}},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ is identical to that of $\left(\mathbf{A},\ [\mathbf{I}_m\ |\mathbf{A}]\mathbf{r},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$, which exactly follows the distribution of samples from $\mathsf{HintMLWE}_{R,d,m,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}}$.

If $\mathbf{b} \leftarrow \mathcal{U}(R_q^m)$, then $\left(\mathbf{A},\ \mathbf{b} + [\mathbf{I}_m\ |\mathbf{A}]\mathbf{t},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ follows the same distribution with $\left(\mathbf{A},\ \mathbf{u},\ \gamma_0, \ldots, \gamma_{\ell-1},\ \mathbf{z}_0, \ldots, \mathbf{z}_{\ell-1}\right)$ where $\mathbf{u} \leftarrow \mathcal{U}(R_q^m)$.

Therefore, the reduction is correct and reduces the advantage at most $(d + m) \cdot 2\varepsilon$. □

Remark 1. When $\mathcal{C}$ samples monomials with leading coefficient ± 1, then it holds that $\left\|\sum_{j=0}^{\ell-1} \mathbf{\Gamma}_j{}^\top \mathbf{\Gamma}_j\right\|_2 \leq \ell$.

Remark 2. Let $a \in R$ with $\|a\|_\infty \leq 1$ and $\|a\|_1 \leq \kappa$. Then, for any real polynomial $b \in \mathbb{R}[X]/(X^n+1)$, we have $\|a \cdot b\|_2 \leq \kappa \cdot \|b\|_2$. Thus, we obtain $\|\mathbf{M}(a)\|_2 = \max_{\mathbf{b}\in\mathbb{Z}^n} \frac{\|\mathbf{M}(a)\mathbf{b}\|_2}{\|\mathbf{b}\|_2} = \max_{b\in\mathbb{R}[X]/(X^n+1)} \frac{\|a\cdot b\|_2}{\|b\|_2} \leq \kappa$.

Therefore, if a distribution $\mathcal{C}$ over R^ℓ outputs a sample $(\gamma_0, \ldots, \gamma_{\ell-1})$ with $\|\gamma_j\|_\infty \leq 1$ and $\|\gamma_j\|_1 \leq \kappa$, then $\left\|\sum_{j=0}^{\ell-1} \mathbf{\Gamma}_j^\top \mathbf{\Gamma}_j\right\|_2 \leq \ell\kappa^2$ holds.

4 Proof of Plaintext Knowledge for RLWE-based Public-Key Encryption

The Proof of Plaintext Knowledge (PPK) protocol is frequently used to attain active security in the constructions of secure multiparty computation protocols [3, 14]. To be precise, the prover would like to send a ciphertext ct to the verifier and convince the verifier that ct is well-formed while revealing no information about the underlying message m.

One can formalize the functionality of PPK protocol using the framework of the secure proof of knowledge protocol in Sec. 2.6. Let $(\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ be a public-key encryption scheme, and pk be a public key for $\mathtt{Enc}$. Then, the public parameter pp is pk, the secret input is the prover's message m, and the honest language $\mathbf{L}$ and the proven language $\mathbf{L}'$ are the set of honestly generated ciphertexts and the set of accepted ciphertexts respectively. In the generation phase, the prover samples encryption randomness r and generates a ciphertext by $c = \mathtt{Enc}_{\mathsf{pk}}(m, r)$. In the proof phase, the verifier checks whether c is valid or not. If it outputs 1, it is the case that $c \in \mathbf{L}'$.

The completeness ensures that an honestly generated ciphertext $c \in \mathbf{L}$ always passes the proof phase except for negligible probability. The soundness ensures that if the prove-phase ends with 1, then $c \in \mathbf{L}'$ and the prover knows encryption randomness u and message m except for negligible probability. Finally, the simulatability ensures that a verifier cannot know the underlying message m from the transcript between the honest prover and verifier. Thus, the construction of PPK protocol based on the proof-of-knowledge framework fulfills all the required functionality.

4.1 PPK based on Hint-RLWE

Now, we provide a concrete instantiation of PPK protocol for the BFV scheme [8, 17]. The main objective of the PPK protocol is to convince the verifier that a ciphertext is generated with small randomness. For $\mathsf{pp} = \mathtt{BFV.Setup}(1^\lambda); \mathsf{pk} \leftarrow \mathtt{BFV.Gen}(\mathsf{pp})$, we first define the witness relationship $\mathbf{R}_{\mathsf{PPK}}$ and $\mathbf{R}'_{\mathsf{PPK}}$ as follows:

$$\mathbf{R}_{\mathsf{PPK}} = \left\{(m, \mathbf{r}, \mathbf{c}) \mid \mathtt{BFV.Enc}_{\mathsf{pk}}(m, 2\mathbf{r}) = \mathbf{c} \ \wedge \ \|\mathbf{r}\|_\infty \leq \beta\right\},$$
$$\mathbf{R}'_{\mathsf{PPK}} = \left\{(m, \mathbf{r}, \mathbf{c}) \mid \mathtt{BFV.Enc}_{\mathsf{pk}}(m, \mathbf{r}) = \mathbf{c} \ \wedge \ \|\mathbf{r}\|_\infty \leq \beta'\right\},$$

Then, $(m, \mathbf{r})$ can be viewed as a witness for the statement about $\mathbf{c}$. The honest language $\mathbf{L}_{\mathsf{PPK}}$ and the proven language $\mathbf{L}'_{\mathsf{PPK}}$ are defined as follows:

$$\mathbf{L}_{\mathsf{PPK}} = \left\{ \mathbf{c} \in R_q^2 \mid \exists (m, \mathbf{r}) \in R_p \times R^3 \text{ s.t. } (m, \mathbf{r}, \mathbf{c}) \in \mathbf{R}_{\mathsf{PPK}} \right\},$$
$$\mathbf{L}'_{\mathsf{PPK}} = \left\{ \mathbf{c} \in R_q^2 \mid \exists (m, \mathbf{r}) \in R_p \times R^3 \text{ s.t. } (m, \mathbf{r}, \mathbf{c}) \in \mathbf{R}'_{\mathsf{PPK}} \right\}.$$

In Fig. 1, we describe the PPK protocol Π_{PPK} for the BFV scheme whose security relies on the hardness of (Hint)RLWE. We remark that an encryption randomness $\mathbf{r}$ is multiplied by 2 in $\mathbf{R}_{\mathsf{PPK}}$ for the honest language due to the weakened knowledge extractor. In the soundness proof, we show that a knowledge extractor can obtain $(X^i - X^j) \cdot (m, \mathbf{r})$ for some $i \neq j$. Since $(X^i - X^j)^{-1} \notin R$ and $2(X^i - X^j)^{-1} \in R$ by Lem. 6, we can finally get $(2m, 2\mathbf{r})$ rather than $(m, \mathbf{r})$. The prior work [5] had the same issue, but it resolved the problem by changing the proven language of PPK. To be precise, the previous PPK protocol does not guarantee the validity of $\mathbf{c}$, but the validity of $2\mathbf{c}$ instead. However, this approach induces another issue that $2\mathbf{c}$ is an encryption of $2m$, not m. Hence, we tweak the relation $\mathbf{R}_{\mathsf{PPK}}$ of the honest prover so that we can guarantee that the ciphertext $\mathbf{c}$ itself is a valid encryption of m in the proven language.

Since the membership decision for $\mathbf{R}_{\mathsf{PPK}}$ and $\mathbf{R}'_{\mathsf{PPK}}$ can be done in polynomial time, both $\mathbf{L}_{\mathsf{PPK}}$ and $\mathbf{L}'_{\mathsf{PPK}}$ are NP-languages. The bounds β_i and β'_i are parameters that will be determined later after $\mathcal{P}$ and $\mathcal{V}$ are designated.

Theorem 2. *Let ℓ be a positive integer, $\sigma_1, \sigma_2 > 0$ and $\kappa = \sqrt{\ln(2n/\varepsilon)/\pi}$ for a negligible $\varepsilon > 0$. Let $\mathsf{pp} = (R, q, p, \chi) \leftarrow \mathtt{BFV.Setup}(1^\lambda), \mathsf{pk} \leftarrow \mathtt{BFV.Gen}(\mathsf{pp})$, $C = \{X^j : 0 \leq j < 2n\}$, $\beta = \kappa\sigma_1$, and $\beta' = 2n\kappa(\sigma_1 + \sigma_2)$. If $(2n)^{-\ell}$ is negligible, then Π_{PPK} is a secure proof-of-knowledge protocol for the pair of NP-languages $(\mathbf{L}_{\mathsf{PPK}}, \mathbf{L}'_{\mathsf{PPK}})$ under the hardness assumption of $\mathsf{RLWE}_{R,1,q,\chi}$ and $\mathsf{HintRLWE}_{R,2,q,\sigma_1}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}$.*

Proof. We show the completeness, knowledge soundness, and simulatability of Π_{PPK} as below.

Completeness: Suppose that both prover and verifier honestly follow the protocol. Then, the ciphertext $\mathbf{c}$ generated by the prover satisfies the honest language $\mathbf{L}_{\mathsf{PPK}}$ since $\|\mathbf{r}\|_\infty < \beta$ except for a negligible probability ε from Cor. 1. The equality $\mathtt{BFV.Enc}(v_i, \mathbf{z}_i) = \mathbf{w}_i + \gamma_i \cdot \mathbf{c}$ follows from the fact that $v_i = u_i + \gamma_i \cdot m$ and $\mathbf{z}_i = \mathbf{y}_i + \gamma_i \cdot \mathbf{r}$. It remains to show that $\|\mathbf{z}_i\|_\infty < (1 + \sigma_2/\sigma_1) \cdot \beta$ for $0 \leq i < \ell$.

Let $\mathbf{z}_i = (z_i^{(0)}, z_i^{(1)}, z_i^{(2)})$. From the definition, $z_i^{(j)}$ follows the distribution $\mathcal{D}_{\mathbb{Z}^n,\sigma_1} + \gamma_i \cdot \mathcal{D}_{\mathbb{Z}^n,\sigma_2}$ for $0 \leq j < 3$. Note that $\gamma_i \cdot \mathcal{D}_{\mathbb{Z}^n,\sigma_2}$ is statistically identical to $\mathcal{D}_{\mathbb{Z}^n,\sigma_2}$ regardless of γ_i, as γ_i is a monomial with the leading coefficient 1 and $\mathcal{D}_{\mathbb{Z}^n,\sigma_2}$ is spherical with center zero. Then, $z_i^{(j)}$ follows the distribution $\mathcal{D}_{\mathbb{Z}^n,\sigma_1} + \mathcal{D}_{\mathbb{Z}^n,\sigma_2}$ for all $0 \leq i < \ell$, which is bounded by $(1 + \sigma_2/\sigma_1) \cdot \beta = (\sigma_1 + \sigma_2) \cdot \kappa$ with an overwhelming probability. Therefore, the verifier outputs 1 except for a negligible probability.

Soundness: Since the soundness error $(2n)^{-\ell}$ is negligible, it suffices to show the existence of an efficient knowledge extractor which can generate a witness

PPK protocol Π_{PPK}

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
Input: $m \in R_p$		pk
$\mathsf{pk} \in R_q^2$		

Generate-phase:

$\mathbf{r} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_1}$		
$\mathbf{c} = \mathtt{BFV.Enc}_{\mathsf{pk}}(m, 2\mathbf{r})$	$\xrightarrow{\mathbf{c}}$	

Prove-phase:

For $0 \le i < \ell$:		
$u_i \leftarrow \mathcal{U}(R_p)$		
$\mathbf{y}_i \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_2}$		
$\mathbf{w}_i = \mathtt{BFV.Enc}_{\mathsf{pk}}(u_i, 2\mathbf{y}_i)$		
	$\xrightarrow{\mathbf{w}_i}$	
		$(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{U}(C^\ell)$
	$\xleftarrow{\gamma_i}$	
For $0 \le i < \ell$:		
$v_i = u_i + \gamma_i \cdot m \pmod p$		
$\mathbf{z}_i = \mathbf{y}_i + \gamma_i \cdot \mathbf{r}$		
	$\xrightarrow{(v_i, \mathbf{z}_i)}$	
		For $0 \le i < \ell$, check:
		$\mathtt{BFV.Enc}_{\mathsf{pk}}(v_i, 2\mathbf{z}_i) = \mathbf{w}_i + \gamma_i \cdot \mathbf{c} \pmod q$
		$\lVert \mathbf{z}_i \rVert_\infty \le (1 + \sigma_2/\sigma_1) \cdot \beta$

Fig. 1. Our PPK protocol for the BFV scheme.

Simulator $\mathcal{S}_{\mathsf{PPK}}$

Input

$\mathsf{pk} \in R_q^2$

---

1. Sample $\mathbf{c} \leftarrow \mathcal{U}(R_q^2)$ and $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{U}(C^\ell)$.
2. Sample $\mathbf{r} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_1}$.
3. Sample $\mathbf{y}_i \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_2}$, and compute $\mathbf{z}_i = \mathbf{y}_i + \gamma_i \cdot \mathbf{r}$ for $0 \le i < \ell$.
4. Sample $v_i \leftarrow \mathcal{U}(R_p)$, and compute $\mathbf{w}_i = \mathtt{BFV.Enc}_{\mathsf{pk}}(v_i, 2\mathbf{z}_i) - \gamma_i \cdot \mathbf{c} \pmod q$ for $0 \le i < \ell$.
5. Output $\mathsf{tr} = (\mathbf{c}, (\mathbf{w}_i, \gamma_i, (v_i, \mathbf{z}_i))_{0 \le i < \ell})$.

Fig. 2. Simulator for Π_{PPK}.

from two accepting transcripts $(\mathbf{c}, \mathbf{w}_i, \gamma_i, (v_i, \mathbf{z}_i))$ and $(\mathbf{c}, \mathbf{w}_i, \gamma_i', (v_i', \mathbf{z}_i'))$ such that $\gamma_i \neq \gamma_i'$ for some $0 \le i < \ell$. We define an extractor $\mathcal{E}$ as follows:

1. Find an index i such that $\gamma_i \neq \gamma_i'$, and set $\bar{\gamma}_i = \gamma_i - \gamma_i'$. It is shown in Lem. 6 that $2\bar{\gamma}_i^{-1}$ is an element of R with $\left\|2\bar{\gamma}_i^{-1}\right\|_\infty \le 1$.
2. Compute and output $(m, \mathbf{r})$ as follows:

$$m = \frac{p+1}{2} \cdot 2\bar{\gamma}_i^{-1} \cdot (v_i - v_i') \pmod p$$
$$\mathbf{r} = 2\bar{\gamma}_i^{-1} \cdot (\mathbf{z}_i - \mathbf{z}_i') \pmod q$$

From $\mathtt{BFV.Enc}_{\mathsf{pk}}(v_i, 2\mathbf{z}_i) = \mathbf{w}_i + \gamma_i \cdot \mathbf{c}$ and $\mathtt{BFV.Enc}_{\mathsf{pk}}(v_i', 2\mathbf{z}_i') = \mathbf{w}_i + \gamma_i' \cdot \mathbf{c}$, we get $\mathtt{BFV.Enc}_{\mathsf{pk}}(v_i - v_i', 2(\mathbf{z}_i - \mathbf{z}_i')) = \bar{\gamma}_i \cdot \mathbf{c}$. We also note that $\frac{p+1}{2} = \frac{q+1}{2} \pmod p$ if p and q are odd integers such that $p \mid q$. Then, we obtain the following equality:

$$\begin{aligned}
\mathtt{BFV.Enc}_{\mathsf{pk}}(m, \mathbf{r}) &= 2\bar{\gamma}_i^{-1} \cdot \mathtt{BFV.Enc}_{\mathsf{pk}}\left(\frac{p+1}{2}(v_i - v_i'), \mathbf{z}_i - \mathbf{z}_i'\right) \pmod q \\
&= 2\bar{\gamma}_i^{-1} \cdot \frac{q+1}{2} \cdot \mathtt{BFV.Enc}_{\mathsf{pk}}\left(v_i - v_i', 2(\mathbf{z}_i - \mathbf{z}_i')\right) \pmod q \\
&= 2\bar{\gamma}_i^{-1} \cdot \frac{q+1}{2} \cdot \bar{\gamma}_i \cdot \mathbf{c} = \mathbf{c} \pmod q.
\end{aligned}$$

Meanwhile, we get $\|\mathbf{r}\|_\infty \le n \cdot \|\mathbf{z}_i - \mathbf{z}_i'\|_\infty \le \beta'$ since $\mathbf{r} = 2\bar{\gamma}_i^{-1} \cdot (\mathbf{z}_i - \mathbf{z}_i') \in R$ and $\left\|2\bar{\gamma}_i^{-1}\right\|_\infty \le 1$. Therefore, the output $(m, \mathbf{r}, \mathbf{c})$ satisfies the relation $\mathbf{R}'_{\mathsf{PPK}}$, so $\mathcal{E}$ is an knowledge extractor for Π_{PPK}.

Simulatability. We show that $\mathcal{S}_{\mathsf{PPK}}$ in Fig 2 is a simulator for the protocol Π_{PPK}. Let $\mathcal{D}_0(m)$ and $\mathcal{D}_1$ be the distribution of the transcripts generated by the honest prover and verifier of Π_{PPK} for each message $m \in R_p$ and that generated by $\mathcal{S}_{\mathsf{PPK}}$, respectively. We prove these distributions are computationally indistinguishable by the hybrid argument: Let $\mathcal{H}_0(m) = \mathcal{D}_0(m)$, $\mathcal{H}_1(m)$, $\mathcal{H}_2$ and $\mathcal{H}_3 = \mathcal{D}_1$ be the distributions of tr which are defined as follows:

$\mathcal{H}_0(m)$: $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{pk}, m), \mathcal{V}(\mathsf{pk}))$ for $\mathsf{pp} = \mathtt{BFV.Setup}(1^\lambda); \mathsf{pk} \leftarrow \mathtt{BFV.Gen}(\mathsf{pp})$ and given $m \in R_p$.

$\mathcal{H}_1(m)$: $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{pk}, m), \mathcal{V}(\mathsf{pk}))$ for $\mathsf{pk} \leftarrow \mathcal{U}(R_q^2)$ and given $m \in R_p$.

$\mathcal{H}_2$: $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{PPK}}(\mathsf{pk})$ for $\mathsf{pk} \leftarrow \mathcal{U}(R_q^2)$.

$\mathcal{H}_3$: $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{PPK}}(\mathsf{pk})$ for $\mathsf{pp} = \mathtt{BFV.Setup}(1^\lambda)$; $\mathsf{pk} \leftarrow \mathtt{BFV.Gen}(\mathsf{pp})$.

Claim 1: $\mathcal{H}_0(m)$ and $\mathcal{H}_1(m)$ are computationally indistinguishable for any message $m \in R_p$ under the hardness assumption of $\mathsf{RLWE}_{R,1,q,\chi}$.

For a given RLWE sample pk, one can pick any message $m \in R_p$ and generate the transcript $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(m, \mathsf{pk}), \mathcal{V}(\mathsf{pk}))$. When pk is sampled from the RLWE distribution (resp. the uniform distribution), then tr follows $\mathcal{H}_0(m)$ (resp. $\mathcal{H}_1$). Therefore, $\mathcal{H}_0(m)$ and $\mathcal{H}_1$ are computationally indistinguishable if $\mathsf{RLWE}_{R,1,q,\chi}$ is hard.

Claim 2: $\mathcal{H}_1(m)$ and $\mathcal{H}_2$ are computationally indistinguishable for any message $m \in R_p$ under the hardness assumption of $\mathsf{HintRLWE}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}_{R,2,q,\sigma_1}$.

Let $\mathcal{A}$ be an algorithm that distinguishes $\mathcal{H}_1(m)$ and $\mathcal{H}_2$ with an advantage ε' for a message $m \in R_p$. Then, we can construct an algorithm $\mathcal{B}$ solving $\mathsf{HintRLWE}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}_{R,2,q,\sigma_1}$ by exploiting $\mathcal{A}$.

The algorithm $\mathcal{B}$ first receives a sample $\left(\mathbf{a}, \mathbf{b}, \gamma_0, \dots, \gamma_{\ell-1}, \mathbf{z}_0, \dots, \mathbf{z}_{\ell-1}\right)$ from the Hint-RLWE challenger. Let $\mathsf{pk} = \mathbf{a}$, $\mathbf{c} = 2 \cdot \mathbf{b} + ((q/p)m, 0) \pmod q$, $v_i := u_i + \gamma_i \cdot m \pmod p$ for $u_i \leftarrow \mathcal{U}(R_p)$, and $\mathbf{w}_i := \mathtt{BFV.Enc}_{\mathsf{pk}}(v_i, 2\mathbf{z}_i) - \gamma_i \cdot \mathbf{c} \pmod q$ for $0 \le i < \ell$. The algorithm $\mathcal{B}$ runs $\mathcal{A}(\mathsf{pk}, \mathsf{tr})$ for the transcript $\mathsf{tr} := (\mathbf{c}, (\mathbf{w}_i, \gamma_i, (v_i, \mathbf{z}_i))_{1 \le i \le \ell})$, and it outputs the response from $\mathcal{A}$.

If $\mathbf{b} = [\mathbf{I}_2 \ |\mathbf{a}]\mathbf{r}$ where $\mathbf{r} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_1}$, $\mathbf{y}_i \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_2}$, $\mathbf{z}_i = \gamma_i \cdot \mathbf{r} + \mathbf{y}_i$ for $0 \le i < \ell$. Then, $\mathbf{c} = \mathtt{BFV.Enc}_{\mathsf{pk}}(m, 2\mathbf{r})$ holds. Moreover, it holds that $\mathbf{w}_i = \mathtt{BFV.Enc}_{\mathsf{pk}}(u_i, 2\mathbf{y}_i)$ since $p \mid q$. Therefore, tr follows the distribution $\mathcal{H}_1(m)$.

Otherwise, if $\mathbf{b}$ is sampled from $\mathcal{U}(R_q^2)$, $\mathbf{c}$ and v_i become uniform over R_q^2 and R_p, respectively. Therefore, tr follows the distribution $\mathcal{H}_2$.

Thus, the algorithm $\mathcal{B}$ solves $\mathsf{HintRLWE}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}_{R,2,q,\sigma_1}$ with the same advantage ε', and ε' should be negligible by the hardness assumption, and therefore $\mathcal{H}_1(m)$ and $\mathcal{H}_2$ are computationally indistinguishable for any message $m \in R_p$.

Claim 3: $\mathcal{H}_2$ and $\mathcal{H}_3$ are computationally indistinguishable under the hardness assumption of $\mathsf{RLWE}_{R,q,\chi_s,\chi_e}$.

For a given RLWE sample pk, one can generate the transcript $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{PPK}}(\mathsf{pk})$. When pk is sampled from the RLWE distribution (resp. the uniform distribution), then tr follows $\mathcal{H}_2$ (resp. $\mathcal{H}_3$). Therefore, if one can distinguish $\mathcal{H}_2$ and $\mathcal{H}_3$ with advantage $\varepsilon' > 0$, then it can also solve $\mathsf{RLWE}_{R,q,\chi_s,\chi_e}$ with advantage ε'.

By Claim 1,2 and 3, the distributions $\mathcal{H}_0(m)$ and $\mathcal{H}_3$ are computationally indistinguishable for any message $m \in R_p$, and hence $\mathit{\Pi}_{\mathtt{PPK}}$ is simulatable assuming that $\mathsf{RLWE}_{R,q,\chi_s,\chi_e}$ and $\mathsf{HintRLWE}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}_{R,2,q,\sigma_1}$ are hard to solve.

Thus, the completeness, knowledge soundness, and simulatability of $\mathit{\Pi}_{\mathsf{PPK}}$ are completely proved.

Remark 3. Let $\sigma > 0$ be a real number such that $\sigma^{-2} = 2(\sigma_1^{-2} + \ell\sigma_2^{-2})$. If $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ for some negligible $\varepsilon > 0$, then the hardness of $\mathsf{HintRLWE}_{R,2,q,\sigma_1}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}$ is reduced from the hardness of $\mathsf{RLWE}_{R,2,q,\sigma}$ by Thm. 1, since samples from $\mathcal{U}(C^\ell)$ are monomials (see Rem. 1).

Soundness Slack. In the previous work [5], the value β'/β is used to describe soundness slack between $\mathbf{L}_{\mathsf{PPK}}$ and $\mathbf{L}'_{\mathsf{PPK}}$. This measurement correctly captures the intuition of soundness slack since it represents an overhead derived from the noise flooding. However, this context does not perfectly fit with our case since the security of our protocol eventually depends on $\kappa\sigma$ (rather than $\beta = \kappa\sigma_1$) if we reduce the hardness of Hint-RLWE from RLWE. Thus we use the quantity $\beta'/\kappa\sigma = \frac{2n(\sigma_1+\sigma_2)}{\sigma}$ as an alternative measurement for soundness slack in our protocol since it precisely describes how much cost is incurred to achieve the security against a malicious adversary.

Parameter Setting. We explain a methodology to choose optimal parameter sets for $\mathit{\Pi}_{\mathsf{PPK}}$ following the conditions of Thms 1 and 2. We denote by λ_{Snd} and λ_{ZK} the security parameters of soundness and simulatability of our protocol, respectively. The soundness security stands for the soundness error of the protocol so it is determined by the size of the challenge space. The zero-knowledge security is originally intended to denote a statistical distance between the simulator and real accepting conversation because simulators in the previous studies [5, 18] are based on statistical indistinguishability. Since our simulator is based on computational indistinguishability, we only account for statistical advantage for λ_{ZK} neglecting computational ones.

We now set the parameters k, ℓ, σ_1, and σ_2 for given λ_{Snd} and λ_{ZK}. We first consider the soundness security. We set $\ell = \lceil \lambda_{\mathsf{Snd}} / \log 2n \rceil$ so that $(2n)^{-\ell} \leq 2^{-\lambda_{\mathsf{Snd}}}$ holds. Then, we set the parameters σ_1, σ_2 which are related to the zero-knowledge security λ_{ZK}. Note that indistinguishability for $\mathcal{S}_{\mathsf{PPK}}$ comes from computational hardness of $\mathsf{RLWE}_{R,q,\chi_s,\chi_e}$ and $\mathsf{HintRLWE}_{R,2,q,\sigma_1}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}$. From Thm. 1, the hardness of $\mathsf{HintRLWE}_{R,2,q,\sigma_1}^{\ell,\sigma_2,\mathcal{U}(C^\ell)}$ is reduced from $\mathsf{RLWE}_{R,2,q,\sigma}$ with statistical advantage 6ε where $\frac{1}{\sigma^2} = 2(\frac{1}{\sigma_1^2} + \frac{\ell}{\sigma_2^2})$, and $\varepsilon > 0$ is some value satisfying $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$. Since we use standard HE parameter sets presented in [1] for $\mathsf{RLWE}_{R,q,\chi_s,\chi_e}$, it is computationally hard. Thus, it suffices to consider the hardness of $\mathsf{RLWE}_{R,q,2,\sigma}$ and the advantage 6ε occurred during reduction for the zero-knowledge security λ_{ZK}. We set $\varepsilon = 2^{-\lambda_{\mathsf{ZK}}}/6$ and $\sigma = \sqrt{2} \cdot \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} \simeq \sqrt{2} \cdot \sqrt{\frac{\lambda_{\mathsf{ZK}} + \ln(12n)}{\pi}}$ so that $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ for given λ_{ZK}. Note that standard HE parameters presented in [1] use $3.2 \cdot \sqrt{2\pi}$ as width parameter for error distribution of RLWE. Since the value of σ is larger than that value for $\lambda_{\mathsf{ZK}} = 128$, it does not affect on the hardness assumption of RLWE with our parameter.

Note that the soundness slack of our protocol is determined by $\sigma_1 + \sigma_2$ when σ is fixed. Hence, we aim to choose σ_1 and σ_2 so that the soundness slack is minimized for given σ. It is easy to show that the best parameters are such that $\sigma_1 = \sqrt{\ell^{\frac{1}{3}} + 1} \cdot \sigma$, $\sigma_2 = \ell^{\frac{1}{3}} \cdot \sigma_1$ and Therefore, the soundness slack of our protocol is calculated as $2n(\sigma_1 + \sigma_2)/\sigma = 2n(1 + \ell^{\frac{1}{3}})^{\frac{3}{2}}$.

Finally, we set the parameter κ which is related to the completeness. If we set $\kappa = \sqrt{\ln(2n/\varepsilon)/\pi}$ for a negligible ε', a honestly generated conversation gets accepted with an overwhelming probability by Thm. 2.

4.2 Extension to Multi-prover PPK

Among versatile applications of PPK protocol, we focus on its usage on the SPDZ multi-party computation (MPC) protocol [14] which utilizes somewhat homomorphic encryption (HE). To achieve active security, SPDZ runs a zero-knowledge PPK protocol for HE ciphertexts so that they are ensured to be honestly generated.

There have been several follow-up studies [5, 18] that improve the efficiency of the PPK protocol in SPDZ. The current state-of-the-art PPK protocol for SPDZ is called k-prover PPK protocol [5], which consists of k parties who play roles of both prover and verifier. In this protocol, all parties verify the validity of a single (accumulated) ciphertext instead of verifying multiple ciphertexts by repeatedly running Π_{PPK} for each party. This reduces the computational cost of verification by a factor of k. However, for this purpose, all parties must be online to jointly generate a shared challenge. Therefore, the noise flooding method is enforced to achieve zero-knowledge since the rejection sampling method would lead to a slowdown due to potentially having to rerun the protocol multiple times[5]. Hence, it achieves a faster verification procedure at the expense of increased communication cost due to the larger ciphertext size resulting from the noise flooding method.

We note that our PPK protocol can be naturally extended to the k-prover case, as described in Appendix A. Compared to the previous work, which uses the noise flooding, our method significantly reduces soundness slack, which incurs a smaller ciphertext size and reduced communication cost. Additionally, we note that the previous work was based on the BGV scheme [9], but we use BFV as a substitute.

Parameter Setting. A parameter setting for the k-party PPK protocol for BFV can be done in a similar manner. The only difference is that the bounds β and β' become k times larger since each party adds k commitments or responses during the prove-phase, but it does not affect the soundness slack as both of them get increased by the same factor. As a result, the soundness slack is still $2n(1 + \ell^{\frac{1}{3}})^{\frac{3}{2}}$. In asymptotic scale, the soundness slack for our PPK protocol is $2n(1 + \ell^{\frac{1}{3}})^{\frac{3}{2}} = O(n \cdot \sqrt{\ell}) = O(n \cdot \sqrt{\lambda_{\mathsf{Snd}}/\log n})$ since $\ell = O(\lambda_{\mathsf{Snd}}/\log n)$. Meanwhile, the soundness slack in the previous PPK protocol [5] accompanies the exponential factor $2^{\lambda_{\mathsf{ZK}}}$ which comes from the noise flooding technique.

5 Proof of Opening Knowledge for BDLOP

The commitment scheme has been used extensively as a core building block of various cryptographic schemes (e.g. [22, 24, 21]). In these applications, the Proof of Opening Knowledge (POK) protocol is usually incorporated together to ensure the security against active adversaries. While the existing constructions of POK rely on zero-knowledge proofs for both input message and commitment randomness, we aim to construct a more efficient POK protocol that allows us to leak partial information of the randomness while still guaranteeing the full message privacy.

Such POK protocol can be implemented using the secure proof-of-knowledge framework in Sec. 2.6. Let $(\mathtt{Gen}, \mathtt{Com}, \mathtt{Open})$ be a commitment scheme, and ck be a commitment key generated by $\mathtt{Gen}$. Then, the public parameter pp is ck, the secret input x is the prover's message m, and the honest language $\mathbf{L}$ and the proven language $\mathbf{L}'$ are the set of honestly generated commitments and the set of accepted commitments, respectively. Then, the completeness guarantees that the prove-phase ends with 1 if the commitment $c \in \mathbf{L}$. The soundness guarantees that if the prove-phase ends with 1, then $c \in \mathbf{L}'$ and the prover knows randomness r and message m used for generating the commitment c. Finally, the simulatability guarantees that the transcript between the prover and the verifier does not leak any information about input message m.

In the rest of this section, we present a concrete instantiation of the POK protocol for the BDLOP commitment scheme [6] based on the hardness assumption of Hint-MLWE, and we provide a concrete parameter set of our POK protocol with a comparison to prior work. It is worth noting that our POK protocol is free from aborting, contrary to previous constructions in [6, 23] using rejection sampling. This work also answers the open questions stated in [23], whether it would be possible to achieve any security proof for POK without rejection.

5.1 POK without Abort based on Hint-MLWE

In this subsection, we propose a POK protocol for the BDLOP commitment scheme [6], which is one of the most widely used building blocks for lattice-based cryptographic primitives [24, 21]. While our protocol leaks some information about commitment randomness, it still satisfies security conditions to be a key ingredient for the construction of the advanced proof techniques such as proofs for product relation [4] and proofs for linear relation [16]. We discuss how our POK protocol can be extended to cover these applications in the next subsection.

We first recall soundness slack that arises in lattice-based proof-of-knowledge construction. The BDLOP scheme follows the proof style presented in [25], so a knowledge extractor can only obtain a witness of the form $\bar{\gamma} \cdot (\mathbf{m}, \mathbf{r})$, where $\bar{\gamma}$ is an element from the difference set $\bar{C} := \{\gamma - \gamma' \mid \gamma, \gamma' \in C\}$ given a challenge set C. Hence, it requires a weakened version of the opening algorithm to accommodate soundness slack. Below, we present the weakened opening algorithm for BDLOP.

- $\underline{\mathtt{BDLOP.WeakOpen}_{\mathsf{ck}}(\mathbf{c}, \mathbf{m}, \mathbf{r}, \bar{\gamma})}$: Given a commitment $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)$, a message m, randomness $\mathbf{r}$, and an element $\bar{\gamma} \in \bar{C}$, it outputs 1 if and only if $\mathbf{c} =$

$\texttt{BDLOP.Com}_{\mathsf{ck}}(\mathbf{m}, \mathbf{r})$ and $\|\bar{\gamma}\mathbf{r}\|_2 < 2\beta'_{\texttt{BDLOP}}$.

Then, the witness relations for POK are defined as follows:

$$\mathbf{R}_{\texttt{Open}} := \{(\mathbf{c}, \mathbf{m}, \mathbf{r}) \mid \texttt{BDLOP.Open}_{\mathsf{ck}}(\mathbf{c}, \mathbf{m}, \mathbf{r}) = 1\}$$
$$\mathbf{R}'_{\texttt{Open}} := \{(\mathbf{c}, \mathbf{m}, \mathbf{r}, \bar{\gamma}) \mid \texttt{BDLOP.WeakOpen}_{\mathsf{ck}}(\mathbf{c}, \mathbf{m}, \mathbf{r}, \bar{\gamma}) = 1\}$$

where $\mathsf{ck} \leftarrow \texttt{BDLOP.Gen}(1^\lambda)$. We note that $(\mathbf{m}, \mathbf{r}, \bar{\gamma})$ serves the role of witness in $\mathbf{R}'_{\texttt{Open}}$. The corresponding honest language and proven language are defined as follows:

$$\mathbf{L}_{\texttt{Open}} := \{\mathbf{c} \in R_q^{\mu+k} \mid \exists(\mathbf{m}, \mathbf{r})\ (\mathbf{c}, \mathbf{m}, \mathbf{r}) \in \mathbf{R}_{\texttt{BDLOP}}\}$$
$$\mathbf{L}'_{\texttt{Open}} := \{\mathbf{c} \in R_q^{\mu+k} \mid \exists(\mathbf{m}, \mathbf{r}, \bar{\gamma})\ (\mathbf{c}, \mathbf{m}, \mathbf{r}, \bar{\gamma}) \in \mathbf{R}'_{\texttt{BDLOP}}\}$$

In Fig. 3, we describe our new POK protocol for the BDLOP commitment scheme. We assume that q is a prime integer satisfying $q = 5 \pmod 8$, and $C := \{\gamma \in R \mid \|\gamma\|_1 = \kappa \wedge \|\gamma\|_\infty \leq 1\}$, the set of polynomials with ternary coefficients in $\{0, \pm 1\}$ and hamming weight $\kappa > 0$. Then, it is known that every element of $\bar{C}$ except 0 is invertible in R_q [25, Cor 1.2].

We formulate the security of our POK protocol $\Pi_{\texttt{Open}}$ for the BDLOP commmitment scheme as the following theorem. Then, the binding property depends on the hardness of $\mathsf{MSIS}_{R,\mu,q,8\kappa\beta'_{\texttt{BDLOP}}}$ under the weakened opening algorithm as in the prior work[6].

Theorem 3. *Let ν, μ, k, q be positive integers, $\sigma_1, \sigma_2 > 0$, $\beta'_{\texttt{BDLOP}} = (\kappa\sigma_1 + \sigma_2)\sqrt{(\mu+\nu+k)n/\pi}$, and $\mathsf{ck} \leftarrow \texttt{BDLOP.Gen}(1^\lambda)$. If $\binom{n}{\kappa}^{-1} \cdot 2^{-\kappa}$ and $2^{-(\mu+\nu+k)n/8}$ are negligible, then $\Pi_{\texttt{Open}}$ is a secure proof-of-knowledge protocol for $(\mathbf{L}_{\texttt{Open}}, \mathbf{L}'_{\texttt{Open}})$ under the hardness assumption of $\mathsf{HintMLWE}^{1,\sigma_2,\mathcal{U}(C)}_{R,\nu,\mu+k,q,\sigma_1}$.*

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

Completeness: Suppose that both the prover and the verifier are honest. Since the relation $\mathbf{B}_0\mathbf{z} = \mathbf{w} + \gamma \cdot \mathbf{c}_0 \pmod q$ always holds, we only need to check the condition $\|\mathbf{z}\|_2 < \beta'_{\texttt{BDLOP}} = (\kappa\sigma_1 + \sigma_2)\sqrt{(\mu+\nu+k)n/\pi}$. By Lem. 2, we have $\|\mathbf{r}\|_2 < \sigma_1\sqrt{(\mu+\nu+k)n/\pi}$ and $\|\mathbf{y}\|_2 < \sigma_2\sqrt{(\mu+\nu+k)/\pi}$ with probability larger than $1 - 2^{-(\mu+\nu+k)n/8}$. Then, we obtain $\|\mathbf{z}\|_2 = \|\mathbf{y} + \gamma \cdot \mathbf{r}\|_2 < (\kappa\sigma_1 + \sigma_2)\sqrt{(\mu+\nu+k)/\pi}$ with probability larger than $(1-2^{-(\mu+\nu+k)n/8})^2$ as $\|\gamma\|_1 = \kappa$.

Therefore, the verifier outputs 1 except for negligible probability since the value $2^{-(\mu+\nu+k)n/8}$ is negligible.

Soundness: Since the soundness error $1/|C| = \frac{1}{\binom{n}{\kappa}\cdot 2^\kappa}$ is negligible, it suffices to show the existence of efficient knowledge extractor for $\mathbf{R}'_{\texttt{Open}}$. We refer the detailed construction of knowledge extractor to [6].

Simulatability: In Fig. 4, we describe a simulator $\mathcal{S}_{\texttt{Open}}$ for $\Pi_{\texttt{Open}}$. Let $\mathcal{D}_0(m)$ and $\mathcal{D}_1$ be the distributions of the transcript tr generated by an honest prover and

POK Protocol $\Pi_{\mathtt{Open}}$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Input: $\mathbf{B}_0 \in R_q^{\mu\times(\mu+\nu+k)}$		$\mathbf{B}_0, \mathbf{B}_1$
$\mathbf{B}_1 \in R_q^{k\times(\mu+\nu+k)},\ \mathbf{m} \in R_q^k$		
Generate-phase:		
$\mathbf{r} \leftarrow \mathcal{D}^{\mu+\nu+k}_{\mathbb{Z}^n,\sigma_1}$		
$\mathbf{c}_0 = \mathbf{B}_0\mathbf{r} \pmod q$		
$\mathbf{c}_1 = \mathbf{B}_1\mathbf{r} + \mathbf{m} \pmod q$	$\xrightarrow{\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1)}$	
Prove-phase:		
$\mathbf{y} \leftarrow \mathcal{D}^{\mu+\nu+k}_{\mathbb{Z}^n,\sigma_2}$		
$\mathbf{w} = \mathbf{B}_0\mathbf{y} \pmod q$	$\xrightarrow{\mathbf{w}}$	
	$\xleftarrow{\gamma}$	$\gamma \leftarrow \mathcal{U}(C)$
$\mathbf{z} = \mathbf{y} + \gamma \cdot \mathbf{r}$		
	$\xrightarrow{\mathbf{z}}$	
		Check $\|\mathbf{z}\|_2 < \beta'_{\mathtt{BDLOP}}$
		Check $\mathbf{B}_0\mathbf{z} = \mathbf{w} + \gamma \cdot \mathbf{c}_0 \pmod q$

Fig. 3. The POK protocol for BDLOP.

verifier for a message $\mathbf{m} \in R_q^k$ and that generated by the simulator, respectively, which are defined as follows:

$\mathcal{D}_0(\mathbf{m})$: $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{ck}, \mathbf{m}), \mathcal{V}(\mathsf{ck}))$ for $\mathsf{ck} \leftarrow \mathtt{BDLOP.Gen}(1^\lambda)$ and given $\mathbf{m} \in R_q^k$

$\mathcal{D}_1$: $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{Open}}(\mathsf{ck})$ for $\mathsf{ck} \leftarrow \mathtt{BDLOP.Gen}(1^\lambda)$

Assume that there exists an algorithm $\mathcal{A}$ that distinguishes the distributions $\mathcal{D}_0(\mathbf{m})$ and $\mathcal{D}_1$ with advantage $\varepsilon > 0$ for a message $\mathbf{m} \in R_q^k$. Then, we can construct an efficient algorithm $\mathcal{B}$ for $\mathsf{HintMLWE}^{1,\sigma_2,\mathcal{U}(C)}_{R,\nu,\mu+k,q,\sigma_1}$ using $\mathcal{A}$ which works as follows:

Simulator $\mathcal{S}_{\texttt{Open}}$

Input

$\mathbf{B}_0 \in R_q^{\mu\times(\mu+\nu+k)}, \mathbf{B}_1 \in R_q^{k\times(\mu+\nu+k)}$

---

1. Sample $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+k})$, $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times k})$ and $\gamma \leftarrow \mathcal{U}(C)$.
2. Sample $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k}$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k}$.
3. Compute $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{k\times\mu} & \mathbf{I}_k \end{bmatrix} \mathbf{u} \pmod q$ and parse $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_0 \in R_q^\mu$, $\mathbf{c}_1 \in R_q^k$.
4. Compute $\mathbf{z} = \mathbf{y} + \gamma \cdot \mathbf{r}$, and $\mathbf{w} = \mathbf{B}_0\mathbf{z} - \gamma \cdot \mathbf{c}_0 \pmod q$.
5. Output $(\mathbf{c}, \mathbf{w}, \gamma, \mathbf{z})$.

Fig. 4. Simulator for $\Pi_{\texttt{Open}}$.

1. Receive a Hint-MLWE instance $(\mathbf{A}, \mathbf{u}, \gamma, \mathbf{z})$ from a Hint-MLWE challenger. Write $\mathbf{z} = \begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \end{bmatrix} \in R^{\mu+\nu+k}$ and parse $\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{bmatrix}$ for $\mathbf{A}_0 \in R_q^{\mu\times\nu}$ and $\mathbf{A}_1 \in R_q^{k\times\nu}$.
2. Sample $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times k})$. Set $\mathbf{B}_0 = \begin{bmatrix} \mathbf{I}_\mu \mid \mathbf{V} \mid \mathbf{A}_0 + \mathbf{V}\mathbf{A}_1 \end{bmatrix} \in R_q^{\mu\times(\mu+\nu+k)}$, $\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}^{k\times\mu} \mid \mathbf{I}_k \mid \mathbf{A}_1 \end{bmatrix} \in R_q^{k\times(\mu+\nu+k)}$, and compute $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{k\times\mu} & \mathbf{I}_k \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \pmod q$. Parse $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_0 \in R_q^\mu$, $\mathbf{c}_1 \in R_q^k$.
3. Compute $\mathbf{w} = \mathbf{B}_0\mathbf{z} - \gamma \cdot \mathbf{c}_0 \pmod q$, and set $\mathsf{tr} = (\mathbf{c}, \mathbf{w}, \gamma, \mathbf{z})$, $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$.
4. Send tr to $\mathcal{A}$, receive a response $b = \mathcal{A}(\mathsf{tr})$, and output b.

We first note that ck always follows the identical distribution with a sample from $\texttt{BDLOP.Gen}(1^\lambda)$. If $\mathbf{u} = \begin{bmatrix} \mathbf{I}_{\mu+k} & \mathbf{A} \end{bmatrix} \mathbf{r}$ for $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k}$, then it holds that

$$\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{k\times\mu} & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{I}_{\mu+k} & \mathbf{A} \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \pmod q.$$

By the definition of Hint-MLWE, we can rewrite $\mathbf{z}$ as $\mathbf{z} = \mathbf{y} + \gamma \cdot \mathbf{r}$ for some $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k}$ and $\mathbf{y} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k}$. Then, we can also check that

$$\mathbf{w} = \mathbf{B}_0(\mathbf{y} + \gamma \cdot \mathbf{r}) - \gamma \cdot \mathbf{B}_0\mathbf{r} = \mathbf{B}_0\mathbf{y} \pmod q$$

Therefore, the distribution of tr is identical to $\mathcal{D}_0(\mathbf{m})$.

On the other hand, if $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+k})$, all the variables are defined just as same with $\mathcal{S}_{\texttt{Open}}$ except $\mathbf{c}$ due to the addition of $\begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$. Since $\begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{k\times\mu} & \mathbf{I}_k \end{bmatrix}$ is invertible over $R_q^{(\mu+k)\times(\mu+k)}$, $\begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{k\times\mu} & \mathbf{I}_k \end{bmatrix} \mathbf{u}$ is also uniform over $R_q^{\mu+k}$, and hence the distribution of $\mathbf{c}$ is identical to that sampled from $\mathcal{S}_{\texttt{Open}}$. Therefore, the distribution of tr is identical to $\mathcal{D}_1$.

Thus, the adversary $\mathcal{B}$ has the same advantage ε as $\mathcal{A}$ in distinguishing the Hint-MLWE instance. As a result, distributions $\mathcal{D}_0(\mathbf{m})$ and $\mathcal{D}_1$ are computationally indistinguishable for any message $\mathbf{m} \in R_q^k$ if $\mathsf{HintMLWE}_{R,\nu,\mu+k,q,\sigma_1}^{1,\sigma_2,\mathcal{U}(C)}$ is hard, which implies the simulatability of our $\Pi_{\texttt{Open}}$. □

Parameter Setting. We now present the method for setting parameters in our POK protocol. The binding property of the commitment scheme is based on the hardness of $\mathsf{MSIS}_{R,\mu,q,8\kappa\beta'_{\texttt{BDLOP}}}$, which is identical to the previous construction in [6]. Meanwhile, the simulatability of our POK protocol is based on the $\mathsf{HintMLWE}_{R,\nu,\mu+k,q,\sigma_1}^{1,\sigma_2,\mathcal{U}(C)}$ assumption. Thus, the parameters must be chosen in such a way that all three problems remain computationally hard.

To set the concrete parameters for the Hint-MLWE problem, we use the reduction in Thm. 1. Since a sample $\gamma \leftarrow \mathcal{U}(C)$ satisfies $\|\gamma\|_1 = \kappa$ and $\|\gamma\|_\infty = 1$, we can reduce the hardness of $\mathsf{HintMLWE}_{R,\nu,\mu+k,q,\sigma_1}^{1,\sigma_2,\mathcal{U}(C)}$ from $\mathsf{MLWE}_{R,\nu,\mu+k,q,\sigma}$ where $1/\sigma^2 = 2(1/\sigma_1^2 + \kappa^2/\sigma_2^2)$ by Remark 2. To this end, $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ should hold for some negligible $\varepsilon > 0$. Then, we only need to consider the hardness of $\mathsf{MLWE}_{R,\nu,\mu+k,q,\sigma}$ when setting the parameters for simulatability. Recall that the upper bound of $\|\mathbf{z}\|_2$ is $\beta'_{\texttt{BDLOP}} = (\kappa\sigma_1 + \sigma_2)\sqrt{(\mu+\nu+k)n/\pi}$. Thus, we choose σ_1 and σ_2 which minimizes $\kappa\sigma_1 + \sigma_2$ under the constraints $1/\sigma^2 = 2(1/\sigma_1^2 + \kappa^2/\sigma_2^2)$, $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$.

In Table 1, we present concrete parameters which are calculated according to the aforementioned method. We measure the hardness of MSIS and MLWE in terms of the root Hermite factor δ, targeting for $\delta \approx 1.0043$ which gives 128-bit security. We first set $q \approx 2^{32}$ and $n = 2^7$ as presented in [23] and then adjust the MSIS rank μ and the MLWE rank ν. We also set $\kappa = 32$ to achieve a negligible soundness error $1/|C| = \binom{n}{\kappa}^{-1} \cdot 2^{-\kappa} < 2^{-128}$. We set $\sigma = \sqrt{2} \cdot \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}$ so that the condition $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$ holds by Lem. 4.

Comparison to Rejection Sampling. In the previous work [6, 23], the rejection sampling method is used to attain zero-knowledge or simulatability. Although it reduces the soundness slack significantly, it introduces additional computational overheads due to repetition. To provide comparison with our work, we also calculate concrete parameters in Table 1 which are obtained by using the rejection sampling method in [20] and [23]. We follow the notation from [23] where $\texttt{Rej}_0$ and $\texttt{Rej}_1$ refer to the rejection sampling methods presented in [20] and its improved version, respectively. In [23], they set randomness distribution to be $\mathcal{U}(\{-1,0,1\}^n)$ and the number of rejections $M = 6$. Then, $\texttt{Rej}_0$ and $\texttt{Rej}_1$ output $\mathbf{z}$ whose distribution is statistically close to $\mathcal{D}_{\mathbb{Z}^n,\beta_0}^{\mu+\nu+k}$ and $\mathcal{D}_{\mathbb{Z}^n,\beta_1}^{\mu+\nu+k}$, respectively, where $\beta_0 = 16.89 \cdot \kappa\sqrt{(\mu+\nu+k)n}$ and $\beta_1 = 1.69 \cdot \kappa\sqrt{(\mu+\nu+k)n}$. [3]

Simulatability of $\texttt{Rej}_0$ can be obtained by constructing the simulator that has a negligible statistical distance to the distribution of real transcripts, but the

[3] Since the Gaussian function in [23] is defined as $\rho(\mathbf{x}) = \exp(-1/2 \cdot \mathbf{x}^\top \mathbf{x})$, we multiplied a factor of $\sqrt{2\pi}$ to those presented in [23].

simulator for $\texttt{Rej}_1$ requires additional assumption called Extended-MLWE [23] to achieve indistinguishability since it leaks some information on commitment randomness. We remark that the hardness of the Extended-MLWE problem has been proven only for the non-algebraic setting. In contrast, simulatability for our method depends on the Hint-MLWE problem, and its hardness can be reduced from the MLWE problem by Thm. 1.

We now compare the parameters with ours (Table 1). Note that ν is determined by the hardness of $\mathsf{MLWE}_{R,\nu,\mu+k,q,\chi_{ter}}$ where $\chi_{ter} = \mathcal{U}(\{-1,0,1\}^n)$. As a result, ν needs to be at least 10 for both $\texttt{Rej}_0$ and $\texttt{Rej}_1$ to attain root Hermite factor $\delta \approx 1.0043$, assuming the Extended-MLWE problem is as hard as the MLWE problem. However, our method enables us to set $\nu = 9$ due to the larger upper bound on the commitment randomness $\mathbf{r}$. It is worth noting that both $\texttt{Rej}_0$ and $\texttt{Rej}_1$ have an upper bound on the ratio $\|\mathbf{y}\|_2/\|\gamma\mathbf{r}\|_2$ in terms of the rejection rate, and therefore they try to set $\|\mathbf{r}\|_2$ as small as possible. However, our method is free from this restriction.

Note that μ is determined by the hardness of $\mathsf{MSIS}_{R,\mu+\nu+k,\mu,q,8\kappa\beta_i}$ for $\texttt{Rej}_i$. As a result, μ should be at least 7 for $\texttt{Rej}_0$ to attain root Hermite factor $\delta \approx 1.0043$. In case of $\texttt{Rej}_1$, it reduces μ to 6 due to having a smaller width parameter. Meanwhile, it suffices to set $\mu = 5$ in our case. Therefore, our method gives smaller μ, ν values compared to the prior work under the same security level. Additionally, our method reduces computational overheads since it does not require any repetitions (rejections) to achieve simulatability.

	$\texttt{Rej}_0$	$\texttt{Rej}_1$	**Ours**
μ (MSIS rank)	7	6	**5**
ν (MLWE rank)	10	10	**9**
Repetition	6	6	–
Simulatability	–	Ext-MLWE	**MLWE**

Table 1. Parameters of each POK for BDLOP ($q \approx 2^{32}, n = 2^7$, $\kappa = 32$, $k = 1$)

5.2 Optimizations and Extensions

In the realm of lattice-based cryptography, there are several applications of the BDLOP commitment scheme such as proofs for integral relation [22], group signature [21] and ring signature [24]. In these applications, advanced proof techniques from [4, 16] are employed to verify additional conditions for the input message. These conditions vary depending on applications, but they all stem from the core property of the BDLOP scheme: computational binding.

In this subsection, we briefly describe how our POK protocol can be further extended to advanced proof systems for product relation [4] and for linear relation over $\mathbb{Z}_q$ [16].

Modification in Challenge Set. In recent applications of BDLOP, the modulus q is often set to be $q = 2n+1 \pmod{4n}$ to obtain the isomorphism $R_q \simeq \mathbb{Z}_q^n$. However, this approach has a disadvantage in that some elements of $\bar{C}$ are not invertible in R_q. To cope with this issue, a new challenge distribution $\mathcal{C}$ over $\{\gamma \in R \mid \|\gamma\|_\infty \leq 1\}$ was proposed in [4] where each coefficient is sampled independently from $-1, 0, 1$ with probability $1/2$ for 0 and $1/4$ for each -1 and 1. It has been shown in [4] that the POK protocol using the new challenge distribution $\mathcal{C}$ attains a soundness error of approximately q^{-1}.

The simulatability still holds for this case by simply substituting $\mathcal{U}(C)$ with $\mathcal{C}$ in Theorem 3. Since a sample $\gamma \leftarrow \mathcal{C}$ satisfies $\|\gamma\|_\infty \leq 1$ and $\|\gamma\| \leq n$, the parameter setting procedure for this case is equal to that in Sec. 5.1 except $\kappa = n$ by Remark 2.

Boosting Soundness. As mentioned earlier, the new challenge distribution provides a soundness error of q^{-1}, which is non-negligible in most applications where $q \approx 2^{32}$. To reduce the soundness error further (i.e., $q^{-\ell}$), an optimization technique [4] that amplifies a single challenge into multiple challenges via automorphisms is often used. In this case, the prover sends multiple responses $\mathbf{z}_i = \mathbf{y}_i + \varphi^i(\gamma) \cdot \mathbf{r}$ for $0 \leq i < \ell$ where $\varphi(X) = X^{2n/\ell+1}$, and the verifier checks if $\|\mathbf{z}_i\|_2 < \beta'_{\texttt{BDLOP}}$ and $\mathbf{B}_0\mathbf{z}_i = \mathbf{w}_i + \varphi^i(\gamma) \cdot \mathbf{c}_0 \pmod{q}$ for $0 \leq i < \ell$.

A further improvement [22, Appendix A.6] was proposed to reduce the size of transcripts by expressing $(\varphi^i(\gamma))_{0 \leq i < \ell}$ as a linear combination of the parsed polynomials $\gamma_i = \sum_{j=0}^{n/\ell-1} \gamma^{(j\ell+i)} X^{j\ell}$ for $0 \leq i < \ell$ of $\gamma = \sum_{j=0}^{n-1} \gamma^{(j)} X^j$. In this case, the prover sends $\mathbf{w}'_i = \mathbf{B}_0\mathbf{y}'_i$ and $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$ for $\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k}$ and verifier checks if $\mathbf{B}_0\mathbf{z}'_i = \mathbf{w}'_i + \gamma_i \cdot \mathbf{c}_0$ for $0 \leq i < \ell$. Then, by computing $\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{y}'_j$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$, and $\mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{w}'_j$ for $0 \leq i < \ell$, one can reconstruct the relations $\mathbf{z}_i = \mathbf{y}_i + \varphi^i(\gamma) \cdot \mathbf{r}$ and $\mathbf{B}_0\mathbf{z}_i = \mathbf{w}_i + \varphi^i(\gamma) \cdot \mathbf{c}_0$. Thus, the soundness property is still maintained. Since $\|\gamma_i\|_1 \leq n/\ell$ while $\left\|\varphi^{(i)}(\gamma)\right\|_1 \leq n$, it results in smaller size of responses.

Adopting these optimizations, the transcript now contains multiple responses $\mathbf{z}'_i$ for $0 \leq i < \ell$, which increases the number of hints from 1 to ℓ in terms of Hint-MLWE. Let $\mathcal{C}'$ be the distribution of $(\gamma_0, \ldots, \gamma_{\ell-1})$ where $\gamma_i = \sum_{j=0}^{n/\ell-1} \gamma^{(j\ell+i)} X^{j\ell}$ for $\gamma = \sum_{j=0}^{n-1} \gamma^{(j)} X^j \leftarrow \mathcal{C}$. Then, the simulatability holds under the hardness assumption of $\mathsf{HintMLWE}_{R,\nu,\mu+k,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}'}$. Meanwhile, the upper bound of $\|\mathbf{z}_i\|_2$ becomes $\beta'_{\texttt{BDLOP}} = (n\sigma_1 + \sqrt{\ell}\sigma_2)\sqrt{(\mu+\nu+k)n/\pi}$ since $\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{y}'_j$ follows $\sum_{j=0}^{\ell-1} \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k}$, which is statistically close to $\mathcal{D}_{\mathbb{Z}^n,\sqrt{\ell}\sigma_2}^{\mu+\nu+k}$ assuming the convolution lemma (Lem. 3).

Applications. We first discuss the simulatability for advanced BDLOP-based proof systems: proof of multiplicative relation [4, Fig. 4] and proof of knowledge for a (ternary) solution to a linear equation [16, Fig. 1 and Fig. 3]. We present new simulatability proofs of these protocols without abortion under the Hint-MLWE assumption in Appendix B.

To summarize briefly, in those protocols the elements of the transcripts are fully simulatable except for $\mathbf{c}$ and $\mathbf{z}_i$ since they are sampled independently from the commitment randomness $\mathbf{r}$. Therefore, it suffices to consider the simulatability of $\mathbf{c}$ and $\mathbf{z}_i$, and it can be shown using the same methodology to Thm. 3, together with the aforementioned modifications. As a result, one can construct simulators for both protocols in a similar way to $\mathcal{S}_{\texttt{Open}}$. Note that our new simulatability proofs for the advanced BDLOP-based proof systems are valid only for non-aborting transcripts, which is the same restriction for zero-knowledge proofs in the previous work [4, 16].

As a benchmark, we present parameters for the protocol in [16, Fig. 3] in Table. 2, which proves knowledge for a ternary solution of a linear equation over $\mathbb{Z}_q$. In [16], a rejection sampling method whose output follows uniform distribution is used. Meanwhile, [23] uses the improved version of the rejection sampling method, $\texttt{Rej}_1$, so that it managed to reduce the parameter μ by 1.

For the parameters in our method, the binding property depends on the hardness of $\mathsf{MSIS}_{R,\mu,q,8n\beta'_{\texttt{BDLOP}}}$. For the simulatability, it depends on the hardness of $\mathsf{HintMLWE}^{\ell,\sigma_2,\mathcal{C}'}_{R,\nu,\mu+k,q,\sigma_1}$. We choose σ_1, σ_2 which minimizes $\beta'_{\texttt{BDLOP}} = (n\sigma_1 + \sqrt{\ell}\sigma_2)\sqrt{(\mu+\nu+k)n/\pi}$ under the constraints $1/\sigma^2 = 2(1/\sigma_1^2 + \ell \cdot (n/\ell)^2/\sigma_2^2)$, and $\sigma \geq \sqrt{2} \cdot \eta_\varepsilon(\mathbb{Z}^n)$. As a result, our method reduces both parameter μ and ν to 7 and 9, respectively. We also note that our method does not require any repetition, so it indeed reduces computation overheads.

	[16]	[23]	**Ours**
μ (MSIS rank)	9	8	**7**
ν (MLWE rank)	10	10	**9**
Repetition	18	6	–
Simulatability	–	Ext-MLWE	**MLWE**

Table 2. Parameters for proof of knowledge of a ternary solution of linear equation over $\mathbb{Z}_q$ $(q \approx 2^{32}, n = 2^7, \ell = 4, k = 19)$

References

1. Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., et al.: Homomorphic encryption standard. In: Protecting Privacy through Homomorphic Encryption, pp. 31–62. Springer (2021)
2. Alperin-Sheriff, J., Peikert, C.: Circular and kdm security for identity-based encryption. In: International Workshop on Public Key Cryptography. pp. 334–352. Springer (2012)
3. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via

threshold fhe. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 483–501. Springer (2012)
4. Attema, T., Lyubashevsky, V., Seiler, G.: Practical product proofs for lattice commitments. In: Annual International Cryptology Conference. pp. 470–499. Springer (2020)
5. Baum, C., Cozzo, D., Smart, N.P.: Using TopGear in overdrive: a more efficient ZKPoK for SPDZ. In: International Conference on Selected Areas in Cryptography. pp. 274–302. Springer (2019)
6. 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)
7. 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)
8. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Annual Cryptology Conference. pp. 868–886. Springer (2012)
9. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)
10. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. pp. 575–584 (2013)
11. Chen, H., Kim, M., Razenshteyn, I., Rotaru, D., Song, Y., Wagh, S.: Maliciously secure matrix multiplication with applications to private deep learning. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 31–59. Springer (2020)
12. Cheon, J.H., Kim, D., Kim, D., Lee, J., Shin, J., Song, Y.: Lattice-Based Secure Biometric Authentication for Hamming Distance. In: Australasian Conference on Information Security and Privacy. pp. 653–672. Springer (2021)
13. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with side information: attacks and concrete security estimation. In: Annual International Cryptology Conference. pp. 329–358. Springer (2020)
14. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Annual Cryptology Conference. pp. 643–662. Springer (2012)
15. Escala, A., Groth, J.: Fine-tuning groth-sahai proofs. In: Public-Key Cryptography–PKC 2014: 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings 17. pp. 630–649. Springer (2014)
16. 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)
17. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012)
18. Keller, M., Pastro, V., Rotaru, D.: Overdrive: making SPDZ great again. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 158–189. Springer (2018)

19. Lee, J., Kim, D., Kim, D., Song, Y., Shin, J., Cheon, J.H.: Instant privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive (2018), Preprint version of [12]
20. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 738–755. Springer (2012)
21. Lyubashevsky, V., Nguyen, N.K., Plancon, M., Seiler, G.: Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 218–248. Springer (2021)
22. 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)
23. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Shorter lattice-based zero-knowledge proofs via one-time commitments. In: Public-Key Cryptography–PKC 2021: 24th IACR International Conference on Practice and Theory of Public Key Cryptography, Virtual Event, May 10–13, 2021, Proceedings, Part I. pp. 215–241. Springer (2021)
24. 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)
25. 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)
26. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM Journal on Computing **37**(1), 267–302 (2007)
27. O'Neill, A., Peikert, C., Waters, B.: Bi-deniable public-key encryption. In: Annual Cryptology Conference. pp. 525–542. Springer (2011)
28. Peikert, C.: An efficient and parallel gaussian sampler for lattices. In: Annual Cryptology Conference. pp. 80–97. Springer (2010)

A k-Prover PPK

We present the protocol (Fig. 5) and the simulator (Fig. 6) for the k-prover PPK. The completeness, soundness, and simulatability of k-Prover PPK can be proved in similar way to Thm. 2. For more details, refer to [5].

k-PPK Protocol $\Pi_{k\text{-PPK}}$

Party $\mathcal{P}_i (0 \leq i < k)$ — **All parties**

Input: $m_i \in R_p, \mathsf{pk} \in R_q^2$ — $\mathsf{pk} \in R_q^2$

. . . .

Generate-phase (for $\mathcal{P}_i$):

$\mathbf{r}_i \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_1}$

$\mathbf{c}_i = \mathtt{BFV.Enc}_{\mathsf{pk}}(m_i, 2\mathbf{r}_i)$ — Broadcast $\mathbf{c}_i$ $\longrightarrow$

. . . .

Prove-phase (for $\mathcal{P}_i$):

For $0 \leq j < \ell$:

$u_{i,j} \leftarrow \mathcal{U}(R_p), \mathbf{y}_{i,j} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_2}$

$\mathbf{w}_{i,j} = \mathtt{BFV.Enc}_{\mathsf{pk}}(u_{i,j}, 2\mathbf{y}_{i,j})$

Broadcast $\mathbf{w}_{i,j}$ $\longrightarrow$

Broadcast γ_j $\longleftarrow$ — $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{U}(C^\ell)$

For $0 \leq j < \ell$:

$v_{i,j} = u_{i,j} + \gamma_j \cdot m_i \pmod p$

$\mathbf{z}_{i,j} = \mathbf{y}_{i,j} + \gamma_j \cdot \mathbf{r}_i$

Broadcast $(v_{i,j}, \mathbf{z}_{i,j})$ $\longrightarrow$

$$\mathbf{c} = \sum_{i=0}^{k-1} \mathbf{c}_i, \ \mathbf{w}_j = \sum_{i=0}^{k-1} \mathbf{w}_{i,j},$$

$$v_j = \sum_{i=0}^{k-1} v_{i,j}, \ \mathbf{z}_j = \sum_{i=0}^{k-1} \mathbf{z}_{i,j}$$

For $0 \leq j < \ell$, check:

$\mathtt{BFV.Enc}_{\mathsf{pk}}(v_j, 2\mathbf{z}_j) = \mathbf{w}_j + \gamma_j \cdot \mathbf{c}$

$\|\mathbf{z}_j\|_\infty < k(1 + \sigma_2/\sigma_1) \cdot \beta$

Fig. 5. PPK protocol for k-prover.

Simulator $\mathcal{S}_{k\text{-PPK}}$

Input

$\mathsf{pk} \in R_q^2$: public key

We denote $I \subset [k]$ as the set of corrupted parties.

1. Sample $\mathbf{c}_{i'} \leftarrow \mathcal{U}(R_q^2)$ for $i' \notin I$ and $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{U}(C^\ell)$.
2. Broadcast $\mathbf{c}_{i'}$ for $i' \notin I$ and receive $\mathbf{c}_i$ for $i \in I$ in the generate-phase.
3. Sample $\mathbf{r}_{i'} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_1}$, $\mathbf{y}_{i',j} \leftarrow \mathcal{D}^3_{\mathbb{Z}^n,\sigma_2}$ and $v_{i',j} \leftarrow \mathcal{U}(R_p)$ for $i' \notin I$ and $0 \leq j < \ell$.
4. Compute $\mathbf{z}_{i',j} = \gamma_j \cdot \mathbf{r}_{i'} + \mathbf{y}_{i',j}$ and $\mathbf{w}_{i',j} = \mathtt{BFV.Enc}_{\mathsf{pk}}(v_{i',j}, 2\mathbf{z}_{i',j}) - \gamma_j \cdot \mathbf{c}_{i'}$ for $i' \notin I$ and $0 \leq j < \ell$.
5. Broadcast $\mathbf{w}_{i',j}$ and receive $\mathbf{w}_{i,j}$ for $i' \notin I$, $i \in I$, and $0 \leq j < \ell$.
6. Broadcast $(\gamma_0, \ldots, \gamma_{\ell-1})$ as a shared challenge.
7. Broadcast $(v_{i',j}, \mathbf{z}_{i',j})$ and receive $(v_{i,j}, \mathbf{z}_{i,j})$ for $i' \notin I$, $i \in I$, and $0 \leq j < \ell$.
8. Output $\mathsf{tr} = (\mathbf{c}_i, \mathbf{w}_{i,j}, \gamma_j, v_{i,j}, \mathbf{z}_{i,j})_{0 \leq i < k, 0 \leq j < \ell}$.

Fig. 6. Simulator for $\Pi_{k\text{-PPK}}$.

B Application of BDLOP

$\mathtt{ALS.Gen}(1^\lambda)$, $\mathtt{ENS.Gen}(1^\lambda)$, and $\mathtt{ENS}'\mathtt{.Gen}(1^\lambda)$ correspond to $\mathtt{BDLOP.Gen}(1^\lambda)$ where $k = 4, 2$ and 19, respectively. Simulatability of these protocols are provided in the following subsections.

B.1 Simulatability of $\Pi_{\mathtt{Prod}}$

In Fig. 9, we describe a simulator $\mathcal{S}_{\mathtt{Prod}}$ for a *non-aborting* transcript of $\Pi_{\mathtt{Prod}}$. Let $\mathcal{D}_0(\mathbf{m})$ and $\mathcal{D}_1$ be the distributions of the transcript tr which is generated by an honest prover and verifier for a message $\mathbf{m} \in R_q^3$ and that generated by the simulator, respectively, which are defined as follows:

$\mathcal{D}_0(\mathbf{m})$: $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{ck}, \mathbf{m}), \mathcal{V}(\mathsf{ck}))$ for $\mathsf{ck} \leftarrow \mathtt{ALS.Gen}(1^\lambda)$ and given $\mathbf{m} \in R_q^3$

$\mathcal{D}_1$: $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{Prod}}(\mathsf{ck})$ for $\mathsf{ck} \leftarrow \mathtt{ALS.Gen}(1^\lambda)$

Let $\mathcal{A}$ be an algorithm that distinguishes the distributions $\mathcal{D}_0(\mathbf{m})$ and $\mathcal{D}_1$ of tr under the condition $\mathtt{Ver}_{\mathtt{Prod}}(\mathsf{tr}) = 1$ with advantage $\varepsilon > 0$ for some message $\mathbf{m} = (m_1, m_2, m_3) \in R_q^3$. Then, given the algorithm $\mathcal{A}$, we can construct an efficient algorithm $\mathcal{B}$ for $\mathsf{HintMLWE}^{\ell,\sigma_2,\mathcal{C}'}_{R,\nu,\mu+4,q,\sigma_1}$ which works as follows:

1. Receive a Hint-MLWE instance $\left(\mathbf{A}, \mathbf{u}, \gamma_0, \ldots, \gamma_{\ell-1}, \mathbf{z}'_0, \ldots, \mathbf{z}'_{\ell-1}\right)$ from the Hint-MLWE challenger. Compute $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$, $\gamma = \sum_{i=0}^{\ell-1} \gamma_i X^i$.

Write $\mathbf{z}_i = \begin{bmatrix}\mathbf{z}_{0,i}\\ \mathbf{z}_{1,i}\end{bmatrix} \in R^{\mu+\nu+4}$, parse $\mathbf{A} = \begin{bmatrix}\mathbf{A}_0\\ \mathbf{A}_1\end{bmatrix}$ for $\mathbf{A}_0 \in R_q^{\mu\times\nu}$ and $\mathbf{A}_1 \in R_q^{4\times\nu}$, and parse $\mathbf{u} = \begin{bmatrix}\mathbf{u}_0\\ \mathbf{u}_1\end{bmatrix}$ for $\mathbf{u}_0 \in R_q^{\mu}$ and $\mathbf{u}_1 = (u_1, u_2, u_3, u_4)^\top \in R_q^4$.

2. Sample $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times 4})$, $\delta_0, \ldots, \delta_{\ell-1} \leftarrow \mathcal{U}(R_q)$. Compute and set $\mathbf{B}_0 = \begin{bmatrix}\mathbf{I}_\mu \mid \mathbf{V} \mid \mathbf{A}_0 + \mathbf{V}\mathbf{A}_1\end{bmatrix} \in R_q^{\mu\times(\mu+\nu+4)}$, $\mathbf{B}_1 = \begin{bmatrix}\mathbf{0}^{4\times\mu} \mid \mathbf{I}_4 \mid \mathbf{A}_1\end{bmatrix} \in R_q^{4\times(\mu+\nu+4)}$, and $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$.
3. Compute $\mathbf{c}_0 = \begin{bmatrix}\mathbf{I}_\mu \mid \mathbf{V}\end{bmatrix}\cdot \mathbf{u}$, $c_j = u_j + m_j$ for $1 \le j \le 3$,

$$\begin{aligned} c_4 =& u_4 + \sum_{i=0}^{\ell-1} \delta_i \varphi^{-i}\Big((\langle \mathbf{b}_3, \mathbf{z}_i\rangle - \varphi^i(\gamma)\cdot u_3) \\ & - m_1(\langle \mathbf{b}_2, \mathbf{z}_i\rangle - \varphi^i(\gamma)\cdot u_2) - m_2(\langle \mathbf{b}_1, \mathbf{z}_i\rangle - \varphi^i(\gamma)\cdot u_1)\Big)\end{aligned}$$

and set $\mathbf{c} = \begin{bmatrix}\mathbf{c}_0\\ \mathbf{c}_1\end{bmatrix} \in R_q^{\mu+4}$ for $\mathbf{c}_1 = (t_1, \ldots, t_4)^\top \in R_q^4$.

4. Compute $\mathbf{w}_i' = \mathbf{B}_0\mathbf{z}_i' - \gamma_i\cdot\mathbf{c}_0 \pmod q$, $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i\rangle - \varphi^i(\gamma)\cdot c_j$ for $0 \le i < \ell$, $1 \le j \le 3$, and $f_4 = \langle \mathbf{b}_4, \mathbf{z}_0\rangle - \gamma\cdot c_4$.
5. Compute $v = \sum_{i=0}^{\ell-1} \delta_i\varphi^{-i}\left(f_{i,1}f_{i,2} + \varphi^i(\gamma) f_{i,3}\right) + f_4$.
6. Set $\mathsf{tr} = (\mathbf{c}, v, \gamma, (\mathbf{w}_i', \delta_i, \mathbf{z}_i')_{0\le i<\ell})$, Send it to $\mathcal{A}$, receive a response $b = \mathcal{A}(\mathsf{ck}, \mathsf{tr})$, and output b.

The overall flow is identical to the proof for $\mathcal{S}_{\texttt{BDLOP}}$ except for the c_4 part. Assume that $\mathbf{u} = \begin{bmatrix}\mathbf{I}_{\mu+4}\ \mathbf{A}\end{bmatrix}\mathbf{r}$ for $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+4}$. First, it is easy to check that $\mathbf{c}_0 = \mathbf{B}_0\mathbf{r}$ and $c_j = \langle \mathbf{b}_j, \mathbf{r}\rangle + m_j$ for $1 \le j \le 3$. By the definition of Hint-MLWE, we can express $\mathbf{z}_i' = \mathbf{y}_i' + \gamma_i\cdot\mathbf{r}$ for $0 \le i < \ell$ where $\mathbf{y}_i' \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+4}$. In other words, $\mathbf{z}_i = \mathbf{y}_i + \varphi^i(\gamma)\cdot\mathbf{r}$ for $\mathbf{z}_i = \sum_{j=0}^{\ell-1}\varphi^j(X^i)\mathbf{z}_i'$ and $\mathbf{y}_i = \sum_{j=0}^{\ell-1}\varphi^j(X^i)\mathbf{y}_i'$. Then it holds that $\langle \mathbf{b}_j, \mathbf{y}_i\rangle = \langle \mathbf{b}_j, \mathbf{z}_i\rangle - \varphi^i(\gamma)\langle \mathbf{b}_j, \mathbf{r}\rangle = \langle \mathbf{b}_j, \mathbf{z}_i\rangle - \varphi^i(\gamma)\cdot u_j$ for $0 \le i < \ell$, $1 \le j \le 4$, and hence we get

$$\begin{aligned} \mathbf{w}_i &= \mathbf{B}_0\mathbf{z}_i - \varphi^i(\gamma)\cdot\mathbf{c}_0 = \mathbf{B}_0\mathbf{y}_i, \\ c_4 &= \langle \mathbf{b}_4, \mathbf{r}\rangle + \sum_{i=0}^{\ell-1}\delta_i\varphi^{-i}\big(\langle \mathbf{b}_3, \mathbf{y}_i\rangle - m_1\langle \mathbf{b}_2, \mathbf{y}_i\rangle - m_2\langle \mathbf{b}_1, \mathbf{y}_i\rangle\big).\end{aligned}$$

Note that $v = \langle \mathbf{b}_4, \mathbf{y}_0\rangle + \sum_{i=0}^{\ell-1}\delta_i\varphi^{-i}\left(\langle \mathbf{b}_1, \mathbf{y}_i\rangle\langle \mathbf{b}_2, \mathbf{y}_i\rangle\right)$ holds for non-abort transcript. Therefore, the distribution of tr is identical to $\mathcal{D}_0(\mathbf{m})$ under the condition $\texttt{Ver}_{\texttt{Prod}}(\mathsf{tr}) = 1$.

Now let us assume that $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+4})$. We can easily check that if $\mathbf{c}$ is determined then there exists a unique solution $\mathbf{u}$ that satisfies the relation between $\mathbf{c}$ and $\mathbf{u}$. Therefore, $\mathbf{c}$ also follows the uniform distribution over $R_q^{\mu+4}$. In simulator $\mathcal{S}_{\texttt{Prod}}$, we can also check that $\mathbf{c}$ follows the uniform distribution over $R_q^{\mu+4}$. By the definition of Hint-MLWE, we can express $\mathbf{z}_i' = \mathbf{y}_i' + \gamma_i\cdot\mathbf{r}$ for $0 \le i < \ell$ where $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+4}$ and $\mathbf{y}_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+4}$. All the other variables are defined just as same with $\mathcal{S}_{\texttt{Prod}}$. Therefore, the distribution of tr is identical to $\mathcal{D}_1$.

$\Pi_{\texttt{Prod}}$

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

Inputs: $\mathbf{B}_0 \in R_q^{\mu \times (\mu+\nu+4)}$, $\mathbf{b}_1, \ldots, \mathbf{b}_4 \in R_q^{\mu+\nu+4}$, $m_1, m_2, m_3 \in R_q$ | Verifier: $\mathbf{B}_0, \mathbf{b}_1, \ldots, \mathbf{b}_4$

Generate-phase:

$\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma_1}^{(\mu+\nu+4)}$

$\mathbf{c}_0 = \mathbf{B}_0 \mathbf{r}$

$c_j = \langle \mathbf{b}_j, \mathbf{r} \rangle + m_j \quad (1 \le j \le 3)$ $\xrightarrow{\mathbf{c}_0, c_1, c_2, c_3}$

Prove-phase:

$\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma_2}^{(\mu+\nu+4)} \quad (0 \le i < \ell)$

$\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j) \mathbf{y}'_j \quad (0 \le i < \ell)$

$\mathbf{w}'_i = \mathbf{B}_0 \mathbf{y}'_i \quad (0 \le i < \ell)$ $\xrightarrow{\mathbf{w}'_i}$

Verifier: $\delta_0, \ldots, \delta_{\ell-1} \leftarrow \mathcal{U}(R_q)$ $\xleftarrow{\delta_0, \ldots, \delta_{\ell-1}}$

$c_4 = \langle \mathbf{b}_4, \mathbf{r} \rangle + \sum_{i=0}^{\ell-1} \delta_i \varphi^{-i} \left(\langle \mathbf{b}_3, \mathbf{y}_i \rangle - m_1 \langle \mathbf{b}_2, \mathbf{y}_i \rangle - m_2 \langle \mathbf{b}_1, \mathbf{y}_i \rangle \right)$

$v = \langle \mathbf{b}_4, \mathbf{y}_0 \rangle + \sum_{i=0}^{\ell-1} \delta_i \varphi^{-i} \left(\langle \mathbf{b}_1, \mathbf{y}_i \rangle \langle \mathbf{b}_2, \mathbf{y}_i \rangle \right)$ $\xrightarrow{c_4, v}$

Verifier: $\gamma \leftarrow \mathcal{C}, \ \gamma = \sum_{i=0}^{\ell-1} \gamma_i X^i$ $\xleftarrow{\gamma_0, \ldots, \gamma_{\ell-1}}$

$\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r} \quad (0 \le i < \ell)$ $\xrightarrow{\mathbf{z}'_i}$

Verifier: $\texttt{Ver}_{\texttt{Prod}}(\mathbf{c}_0, c_1, \ldots, c_4, \mathbf{w}'_i, \delta_i, v, \gamma, \mathbf{z}'_i)$

Fig. 7. Proof of product relation [4].

Thus, the adversary $\mathcal{B}$ has the same advantage ε as $\mathcal{A}$ in distinguishing the Hint-MLWE instance. As a result, the distributions $\mathcal{D}_0(\mathbf{m})$ and $\mathcal{D}_1$ of tr under the condition $\mathtt{Ver}_{\mathtt{Prod}}(\mathsf{tr}) = 1$ are computationally indistinguishable for any message $\mathbf{m} \in R_q^3$ if $\mathsf{HintMLWE}^{\ell,\sigma_2,\mathcal{C}'}_{R,\nu,\mu+4,q,\sigma_1}$ is hard, which implies the simulatability of $\Pi_{\mathtt{Prod}}$. □

$\mathtt{Ver}_{\mathtt{Prod}}(\mathbf{c}_0, c_1, c_2, c_3, c_4, \mathbf{w}'_i, \delta_i, v, \gamma, \mathbf{z}'_i)$

1 : Compute $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j,\ \mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{w}'_j \quad (0 \le i < \ell)$

2 : Check $\|\mathbf{z}_i\|_2 < (n\sigma_1 + \sqrt{\ell}\sigma_2)\sqrt{n(\mu+\nu+4)/\pi} \quad (0 \le i < \ell)$

3 : Check $\mathbf{B}_0\mathbf{z}_i = \mathbf{w}_i + \varphi^i(\gamma)\mathbf{c}_0 \quad (0 \le i < \ell)$

4 : Compute $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)c_j \quad (0 \le i < \ell,\ 1 \le j \le 3)$

5 : Compute $f_4 = \langle \mathbf{b}_4, \mathbf{z}_0 \rangle - \gamma \cdot c_4$

6 : Check $\sum_{i=0}^{\ell-1} \delta_i \varphi^{-i}\left(f_{i,1}f_{i,2} + \varphi^i(\gamma)f_{i,3}\right) + f_4 = v$

Fig. 8. Verification procedure for $\Pi_{\mathtt{Prod}}$

Simulator $\mathcal{S}_{\mathtt{Prod}}$

Input

$\mathbf{B}_0 \in R_q^{\mu\times(\mu+\nu+4)}, \mathbf{B}_1 \in R_q^{4\times(\mu+\nu+4)}$

1. Sample $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+4})$, $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times 4})$, $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{C}'$, and $\delta_i \leftarrow \mathcal{U}(R_q)$ for $0 \le i < \ell$.
2. Sample $\mathbf{r} \leftarrow \mathcal{D}^{\mu+\nu+4}_{\mathbb{Z}^n,\sigma_1}$ and $\mathbf{y}'_i \leftarrow \mathcal{D}^{\mu+\nu+4}_{\mathbb{Z}^n,\sigma_2}$ for $0 \le i < \ell$.
3. Compute $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{4\times\mu} & \mathbf{I}_4 \end{bmatrix} \mathbf{u} \pmod q$ and parse $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_0 \in R_q^\mu$ and $\mathbf{c}_1 = (c_1, c_2, c_3, c_4)^\top \in R_q^4$.
4. Compute $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$ and $\mathbf{w}'_i = \mathbf{B}_0\mathbf{z}'_i - \gamma_i \cdot \mathbf{c}_0 \pmod q$ for $0 \le i < \ell$.
5. Let $\mathbf{B}_i$ be the i-th row of $\mathbf{B}_1$ for $1 \le i \le 4$. Compute $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)c_j$ for $0 \le i < \ell,\ 1 \le j \le 3$ and $f_4 = \langle \mathbf{b}_4, \mathbf{z}_0 \rangle - \gamma \cdot c_4$.
6. Compute $v = \sum_{i=0}^{\ell-1} \gamma_i \varphi^{-i}\left(f_{i,1}f_{i,2} + \varphi^i(\gamma)f_{i,3}\right) + f_4$
7. Output $(\mathbf{c}, v, \gamma, (\mathbf{w}'_i, \gamma_i, \mathbf{z}'_i)_{0\le i<\ell})$.

Fig. 9. Simulator for $\Pi_{\mathtt{Prod}}$.

B.2 Simulatability of $\Pi_{\texttt{Lin}}$

In Fig. 12, we describe a simulator $\mathcal{S}_{\texttt{Lin}}$ for a *non-aborting* transcript of $\Pi_{\texttt{Lin}}$. Let $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, m)$ and $\mathcal{D}_1$ be the distributions of the transcript $\texttt{tr}$ which is generated by an honest prover and verifier for a message $m \in R_q$, a matrix $\mathbf{M}$ and a vector $\mathbf{k}$, and that generated by the simulator, respectively, which are defined as follows:

$\mathcal{D}_0(\mathbf{M}, \mathbf{k}, m)$: $\texttt{tr} \leftarrow \texttt{Tr}(\mathcal{P}(\mathsf{ck}, \mathbf{M}, \mathbf{k}, m), \mathcal{V}(\mathsf{ck}, \mathbf{M}, \mathbf{k}))$ for $\mathsf{ck} \leftarrow \texttt{ENS.Gen}(1^\lambda)$ and given
$m \in R_q$, $\mathbf{M} \in \mathbb{Z}_q^{m \times kn}$, $\mathbf{k} \in \mathbb{Z}_q^m$.

$\mathcal{D}_1$: $\texttt{tr} \leftarrow \mathcal{S}_{\texttt{Ter}}(\mathsf{ck}, \mathbf{M}, \mathbf{k})$ for $\mathsf{ck} \leftarrow \texttt{ENS.Gen}(1^\lambda)$

Let $\mathcal{A}$ be an algorithm that distinguishes the distributions $\mathcal{D}_{0,\mathbf{M},\mathbf{k}}(m)$ and $\mathcal{D}_1$ of $\texttt{tr}$ under the condition $\texttt{Ver}_{\texttt{Lin}}(\texttt{tr}) = 1$ with advantage $\varepsilon > 0$ for some message $m \in R_q$, matrix $\mathbf{M}$ and vector $\mathbf{k}$. Then, given the algorithm $\mathcal{A}$, we can construct an efficient algorithm $\mathcal{B}$ for $\mathsf{HintMLWE}^{\ell,\sigma_2,\mathcal{C}'}_{R,\nu,\mu+2,q,\sigma_1}$ which works as follows:

1. Receive a Hint-MLWE instance $(\mathbf{A}, \mathbf{u}, \gamma_0, \ldots, \gamma_{\ell-1}, \mathbf{z}'_0, \ldots, \mathbf{z}'_{\ell-1})$ from the Hint-MLWE challenger. Compute $\gamma = \sum_{j=0}^{\ell-1} \gamma_j X^j$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$ for $0 \le j < \ell$. Write $\mathbf{z}_i = \begin{bmatrix} \mathbf{z}_{0,i} \\ \mathbf{z}_{1,i} \end{bmatrix} \in R^{\mu+\nu+2}$, parse $\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{bmatrix}$ for $\mathbf{A}_0 \in R_q^{\mu \times \nu}$ and $\mathbf{A}_1 \in R_q^{2 \times \nu}$, and parse $\mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \end{bmatrix}$ for $\mathbf{u}_0 \in R_q^\mu$ and $\mathbf{u}_1 = (u_1, u_2)^\top \in R_q^2$.
2. Sample $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu \times 2})$, $\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$, and $g \leftarrow \mathcal{U}(\{a \in R_q | a_0 = \cdots = a_{\ell-1} = 0\})$. Compute and set $\mathbf{B}_0 = \begin{bmatrix} \mathbf{I}_\mu \mid \mathbf{V} \mid \mathbf{A}_0 + \mathbf{V}\mathbf{A}_1 \end{bmatrix} \in R_q^{\mu \times (\mu+\nu+2)}$, $\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}^{2 \times \mu} \mid \mathbf{I}_2 \mid \mathbf{A}_1 \end{bmatrix} \in R_q^{2 \times (\mu+\nu+2)}$, and $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$.
3. Let $\mathbf{b}_i$ be the i-th row of $\mathbf{B}_1$ for $1 \le j \le 2$. Compute $\mathbf{c}_0 = \begin{bmatrix} \mathbf{I}_\mu \mid \mathbf{V} \end{bmatrix} \cdot \mathbf{u}$, $c_1 = u_1 + m$, and $c_2 = u_2 + g$, and set $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_1 = (c_1, c_2)^\top \in R_q^2$.
4. Compute $\mathbf{w}'_i = \mathbf{B}_0 \mathbf{z}'_i - \gamma_i \cdot \mathbf{c}_0 \pmod q$ for $0 \le i < \ell$.
5. Compute $f = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\texttt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j) c_1 - \langle \mathbf{k}, \mathbf{x}_j \rangle \right)$, $h = g + f$, and $v_i = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \texttt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j) \mathbf{b}_1, \mathbf{z}_{i-k} \right\rangle \right) + \langle \mathbf{b}_2, \mathbf{z}_i \rangle - \varphi^i(\gamma)(f + c_2 - h)$ for $0 \le i < \ell$.
6. Set $\texttt{tr} = (\mathbf{c}, h, \gamma, (\mathbf{w}'_i, \mathbf{x}_i, v_i, \mathbf{z}'_i)_{0 \le i < \ell})$, send it to $\mathcal{A}$, receive a response $b = \mathcal{A}(\mathsf{ck}, \texttt{tr})$, and output b.

Assume that $\mathbf{u} = \begin{bmatrix} \mathbf{I}_{\mu+2} \ \mathbf{A} \end{bmatrix} \mathbf{r}$ for $\mathbf{r} \leftarrow \mathcal{D}^{\mu+\nu+2}_{\mathbb{Z}^n,\sigma_1}$. First, it is easy to check that $\mathbf{c}_0 = \mathbf{B}_0 \mathbf{r}$, $c_1 = \langle \mathbf{b}_1, \mathbf{r} \rangle + m$ and $c_2 = \langle \mathbf{b}_2, \mathbf{r} \rangle + g$. By the definition of Hint-MLWE, we can express $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$ for $1 \le i < \ell$ where $\mathbf{y}'_i \leftarrow \mathcal{D}^{\mu+\nu+2}_{\mathbb{Z}^n,\sigma_2}$. Then we get $\mathbf{w}_i = \mathbf{B}_0 \mathbf{z}_i - \varphi^i(\gamma) \cdot \mathbf{c}_0 = \mathbf{B}_0 \mathbf{y}_i \pmod q$ for $0 \le i < \ell$ where $\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j) \mathbf{y}'_j$ $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j) \mathbf{z}'_j$, and $\mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j) \mathbf{w}'_j$. Since $\langle \mathbf{b}_2, \mathbf{z}_i \rangle = \langle \mathbf{b}_2, \mathbf{y}_i \rangle + \varphi^i(\gamma) \langle \mathbf{b}_2, \mathbf{r} \rangle = \langle \mathbf{b}_2, \mathbf{y}_i \rangle + \varphi^i(\gamma) \cdot u_2$ and $f + t_2 - h = u_2$, we get

$$\begin{aligned} v_i &= \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \mathtt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{z}_{i-k}\right\rangle\right) + \langle \mathbf{b}_2, \mathbf{z}_i \rangle - \varphi^i(\gamma)(f + c_2 - h) \\ &= \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \mathtt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{z}_{i-k}\right\rangle\right) + \langle \mathbf{b}_2, \mathbf{y}_i \rangle + \varphi^i(\gamma) \cdot u_2 - \varphi^i(\gamma) \cdot u_2 \\ &= \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \mathtt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{z}_{i-k}\right\rangle\right) + \langle \mathbf{b}_2, \mathbf{y}_i \rangle . \end{aligned}$$

Therefore, the distribution of tr is identical to $\mathcal{D}_{0,\mathbf{M},\mathbf{k},m}$.

Now let us assume that $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+2})$. Since $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{2\times\mu} & \mathbf{I}_2 \end{bmatrix} \cdot \mathbf{u} + \begin{bmatrix} \mathbf{0}^\mu \\ \mathbf{m} \end{bmatrix}$ for $\mathbf{m} = (m, g)^\top \in R_q^2$ and $\begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{2\times\mu} & \mathbf{I}_2 \end{bmatrix}$ is invertible over $R_q^{(\mu+2)\times(\mu+2)}$, $\mathbf{c}$ is also uniform over $R_q^{\mu+2}$ independent to both m and g. In non-abort transcript, h is an element of $\{a \in R_q | a_0 = \cdots = a_{\ell-1} = 0\}$. Since g is uniform over this set, $h = f + g$ is also uniform and independent to f. By the definition of Hint-MLWE, we can express $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$ for $0 \leq i < \ell$ where $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+2}$ and $\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+2}$. Therefore, the distribution of tr is identical to $\mathcal{D}_1$ under the condition $\mathtt{Ver}_{\mathtt{Lin}}(\mathsf{tr}) = 1$.

Thus, the adversary $\mathcal{B}$ has the same advantage ε as $\mathcal{A}$ in distinguishing the Hint-MLWE instance. As a result, distributions $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, m)$ and $\mathcal{D}_1$ of tr under the condition $\mathtt{Ver}_{\mathtt{Lin}}(\mathsf{tr}) = 1$ are computationally indistinguishable for any message $m \in R_q$, matrix $\mathbf{M}$ and vector $\mathbf{k}$ if $\mathsf{HintMLWE}_{R,\nu,\mu+2,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}'}$ is hard, which implies the simulatability of our $\Pi_{\mathtt{Lin}}$. □

$\mathit{\Pi}_{\texttt{Lin}}$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
Inputs: $\mathbf{B}_0 \in R_q^{\mu\times(\mu+\nu+2)}$		$\mathbf{B}_0, \mathbf{b}_1, \mathbf{b}_2$
$\mathbf{b}_1, \mathbf{b}_2 \in R_q^{\mu+\nu+2}$		$\mathbf{M}, \mathbf{k}$
$m \in R_q$		
$\mathbf{M} \in \mathbb{Z}_q^{m\times n}, \mathbf{k} = \mathbf{M}\cdot\texttt{NTT}(m) \in \mathbb{Z}_q^m$		
Generate-phase :		
$\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+2}$		
$\mathbf{c}_0 = \mathbf{B}_0\mathbf{r}$		
$c_1 = \langle \mathbf{b}_1, \mathbf{r}\rangle + m$	$\xrightarrow{\mathbf{c}_0, c_1}$	
Prove-phase :		
$g \leftarrow \mathcal{U}(\{g \in R_q \| g_0 = \cdots = g_{\ell-1} = 0\})$		
$c_2 = \langle \mathbf{b}_2, \mathbf{r}\rangle + g$		
$\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+2} \quad (0 \le i < \ell)$		
$\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{y}'_j,\ \mathbf{w}'_i = \mathbf{B}_0\mathbf{y}'_i \quad (0 \le i < \ell)$	$\xrightarrow{c_2, \mathbf{w}'_i}$	
		$\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$
	$\xleftarrow{\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1}}$	
$h = g + \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\texttt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j)m - \langle \mathbf{k}, \mathbf{x}_j\rangle\right)$		
$v_i = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \texttt{iNTT}(n\mathbf{M}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{y}_{i-k}\right\rangle\right)$		
$\quad + \langle \mathbf{b}_2, \mathbf{y}_i\rangle \quad (0 \le i < \ell)$	$\xrightarrow{h, v_i}$	
		$\gamma \leftarrow \mathcal{C},\ \gamma = \sum_{j=0}^{\ell-1} \gamma_j X^j$
	$\xleftarrow{\gamma_0, \ldots, \gamma_{\ell-1}}$	
$\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r} \quad (0 \le i < \ell)$	$\xrightarrow{\mathbf{z}'_i}$	
		$\texttt{Ver}_{\texttt{Lin}}(\mathbf{c}_0, c_1, c_2, \mathbf{w}'_i, \mathbf{x}_i, h, v_i, \gamma, \mathbf{z}'_i)$

Fig. 10. Proof of linear relation [16]

$\texttt{Ver}_{\texttt{Lin}}(\mathbf{c}_0, c_1, c_2, \mathbf{w}'_i, \mathbf{x}_i, h, v_i, \gamma, \mathbf{z}'_i)$

1 : Compute $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j,\ \mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{w}'_j \quad (0 \le i < \ell)$

2 : Check $\|\mathbf{z}_i\|_2 < (n\sigma_1 + \sqrt{\ell}\sigma_2)\sqrt{n(\mu+\nu+2)/\pi} \quad (0 \le i < \ell)$

3 : Check $\mathbf{B}_0\mathbf{z}_i = \mathbf{w}_i + \varphi^i(\gamma)\mathbf{c}_0 \quad (0 \le i < \ell)$

4 : Check $h_0 = \cdots = h_{k-1} = 0$

5 : Compute $f = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\texttt{iNTT}(d\mathbf{A}^\top \mathbf{x}_j)c_1 - \langle \mathbf{k}, \mathbf{x}_j \rangle\right)$

6 : Check $\sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \texttt{iNTT}(d\mathbf{A}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{z}_{i-k} \right\rangle\right) + \langle \mathbf{b}_2, \mathbf{z}_i \rangle$

7 : $= v_i + \varphi^i(\gamma)(f + c_2 - h) \quad (0 \le i < \ell)$

Fig. 11. Verification procedure for $\Pi_{\texttt{Lin}}$

Simulator $\mathcal{S}_{\texttt{Lin}}$

Input

$\mathbf{B}_0 \in R_q^{\mu \times (\mu+\nu+2)}, \mathbf{B}_1 \in R_q^{2 \times (\mu+\nu+2)}$, $\mathbf{M} \in \mathbb{Z}_q^{m \times n}, \mathbf{k} \in \mathbb{Z}_q^m$

1. Sample $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+2})$, $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times 2})$, and $(\gamma_0, \ldots, \gamma_{\ell-1}) \leftarrow \mathcal{C}'$.
2. Sample $\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$ and $h \leftarrow \mathcal{U}(\{h \in R_q | h_0 = \cdots = h_{\ell-1} = 0\})$.
3. Sample $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+2}$, $\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+2}$, and $\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{y}'_j$ for $0 \le i < \ell$.
4. Compute $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{2\times\mu} & \mathbf{I}_2 \end{bmatrix} \mathbf{u} \pmod q$ and parse $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_0 \in R_q^\mu$, $\mathbf{c}_1 \in R_q^2$.
5. Compute $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$, $\mathbf{w}'_i = \mathbf{B}_0\mathbf{z}'_i - \gamma_i \cdot \mathbf{c}_0 \pmod q$, and $\mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{w}'_j \pmod q$ for $0 \le i < \ell$.
6. Compute $f = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\texttt{iNTT}(d\mathbf{A}^\top \mathbf{x}_j)c_1 - \langle \mathbf{k}, \mathbf{x}_j \rangle\right)$.
7. Compute $v_i = \sum_{j=0}^{\ell-1} \frac{1}{\ell} X^j \sum_{k=0}^{\ell-1} \varphi^k \left(\left\langle \texttt{iNTT}(d\mathbf{A}^\top \mathbf{x}_j)\mathbf{b}_1, \mathbf{z}_{i-k} \right\rangle\right) + \langle \mathbf{b}_2, \mathbf{z}_i \rangle - \varphi^i(\gamma)(f + c_2 - h)$ for $0 \le i < \ell$.
8. Output $(\mathbf{c}, h, \gamma, (\mathbf{w}'_i, \mathbf{x}_i, v_i, \mathbf{z}'_i)_{0 \le i < \ell})$.

Fig. 12. Simulator for $\Pi_{\texttt{Lin}}$.

B.3 Simulatability of Π_{Ter}

In Fig. 15, we describe a simulator $\mathcal{S}_{\texttt{Ter}}$ for non-aborting transcripts of $\Pi_{\texttt{Ter}}$. Let $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$ and $\mathcal{D}_1$ be the distributions of the transcript $\texttt{tr}$ which is generated by an honest prover and verifier for a message $\mathbf{m} \in R_q^k$, a matrix $\mathbf{M}$ and a vector $\mathbf{k}$, and that generated by the simulator, respectively, which are defined as follows:

$\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$: $\mathsf{tr} \leftarrow \mathtt{Tr}(\mathcal{P}(\mathsf{ck}, \mathbf{M}, \mathbf{k}, m), \mathcal{V}(\mathsf{ck}, \mathbf{M}, \mathbf{k}))$ for $\mathsf{ck} \leftarrow \mathtt{ENS}'.\mathtt{Gen}(1^\lambda)$ and given $\mathbf{m} \in R_q^k$, $\mathbf{M} \in \mathbb{Z}_q^{m \times kn}$, $\mathbf{k} \in \mathbb{Z}_q^m$.

$\mathcal{D}_1$: $\mathsf{tr} \leftarrow \mathcal{S}_{\mathtt{Ter}}(\mathsf{ck}, \mathbf{M}, \mathbf{k})$ for $\mathsf{ck} \leftarrow \mathtt{ENS}'.\mathtt{Gen}(1^\lambda)$

Let $\mathcal{A}$ be an algorithm that distinguishes the distributions $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$ and $\mathcal{D}_1$ of tr under the condition $\mathtt{Ver}_{\mathtt{Ter}}(\mathsf{tr}) = 1$ with advantage $\varepsilon > 0$ for some message $\mathbf{m} = (m_1, ..., m_k) \in R_q^k$, matrix $\mathbf{M}$ and vector $\mathbf{k}$. Then, given the algorithm $\mathcal{A}$, we can construct an efficient algorithm $\mathcal{B}$ for $\mathsf{HintMLWE}_{R,\nu,\mu+k+3,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}'}$ which works as follows:

1. Receive a Hint-MLWE instance $(\mathbf{A}, \mathbf{u}, \gamma_0, \ldots, \gamma_{\ell-1}, \mathbf{z}'_0, \ldots, \mathbf{z}'_{\ell-1})$ from the Hint-MLWE challenger. Compute $\gamma = \sum_{j=0}^{\ell-1} \gamma_j X^j$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j$ for $0 \le i < \ell$. Parse $\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \end{bmatrix}$ for $\mathbf{A}_0 \in R_q^{\mu \times \nu}$ and $\mathbf{A}_1 \in R_q^{(k+3)\times\nu}$, and parse $\mathbf{u} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \end{bmatrix}$ for $\mathbf{u}_0 \in R_q^\mu$ and $\mathbf{u}_1 = (u_1, u_2, ..., u_{k+3})^\top \in R_q^{k+3}$.
2. Sample $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times(k+3)})$, $\delta_0, ..., \delta_{\ell k-1} \leftarrow \mathcal{U}(R_q)$, $\mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$, and $g \leftarrow \mathcal{U}(\{a \in R_q | a_0 = \cdots = a_{\ell-1} = 0\})$, and set $\mathbf{B}_0 = \begin{bmatrix} \mathbf{I}_\mu \mid \mathbf{V} \mid \mathbf{A}_0 + \mathbf{V}\mathbf{A}_1 \end{bmatrix} \in R_q^{\mu\times(\mu+\nu+k+3)}$, $\mathbf{B}_1 = \begin{bmatrix} \mathbf{0}^{(k+3)\times\mu} \mid \mathbf{I}_{k+3} \mid \mathbf{A}_1 \end{bmatrix} \in R_q^{(k+3)\times(\mu+\nu+k+3)}$, and $\mathsf{ck} = (\mathbf{B}_0, \mathbf{B}_1)$.
3. Let $\mathbf{b}_i$ be the i-th row of $\mathbf{B}_1$ for $1 \le j \le k+3$. Compute $\mathbf{c}_0 = \begin{bmatrix} \mathbf{I}_\mu \mid \mathbf{V} \end{bmatrix} \cdot \mathbf{u}$, $c_i = u_i + m_i$ for $1 \le i \le k$, $c_{k+1} = u_{k+1} + g$,
$$c_{k+2} = u_{k+2} + \langle \mathbf{b}_{k+3}, \mathbf{z}_0 \rangle - \gamma \cdot u_{k+3} + \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1} \delta_{ik+j}\varphi^{-i}\big(3m_j \cdot (\langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)\cdot u_j)^2\big),$$
$$c_{k+3} = u_{k+3} + \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1} \delta_{ik+j}\varphi^{-i}\big((3m_j^2 - 1)\cdot(\langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)\cdot u_j)\big).$$
4. Compute $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)\mathbf{c}_j$ for $0 \le i < \ell$, $0 \le j < k$, $f_{k+2} = \langle \mathbf{b}_{k+2}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+2}$, $f_{k+3} = \langle \mathbf{b}_{k+3}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+3}$, and
$$v = \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1} \delta_{ik+j}\varphi^{-i}\Bigg(f_{i,j}\big(f_{i,j} + \varphi^i(\gamma)\big)\big(f_{i,j} - \varphi^i(\gamma)\big)\Bigg) + f_{k+2} + \gamma f_{k+3}$$
5. Parse $\mathbf{M}^\top \mathbf{x}_i = \mathtt{NTT}(\xi_{i,0})||\ldots||\mathtt{NTT}(\xi_{i,k-1})$ for $0 \le i < \ell$, and compute $\tau = \sum_{i=0}^{\ell-1} \frac{1}{\ell} X^i \sum_{s=0}^{\ell-1} \varphi^s\big(\sum_{j=0}^{k-1} n\xi_{i,j}\mathbf{c}_j - \langle \mathbf{k}, \mathbf{x}_i \rangle\big)$, $h = g + \tau$, and
$$v'_i = \sum_{p=0}^{\ell-1} \frac{1}{\ell} X^p \sum_{s=0}^{\ell-1}\sum_{j=1}^{k-1} \varphi^s\big(n\xi_{p,j}\langle \mathbf{b}_j, \mathbf{z}_{i-s} \rangle\big) + \langle \mathbf{b}_{k+1}, \mathbf{z}_i \rangle - \varphi^i(\gamma)(\tau + \mathbf{c}_{k+1} - h).$$

6. Set $\mathbf{c} = \begin{bmatrix}\mathbf{c}_0 \\ \mathbf{c}_1\end{bmatrix}$ for $\mathbf{c}_1 = (c_1, \ldots, c_{k+3})^\top \in R_q^{k+3}$.
7. Compute $\mathbf{w}'_i = \mathbf{B}_0\mathbf{z}'_i - \gamma_i \cdot \mathbf{c}_0 \pmod q$ for $0 \le i < \ell$.
8. Set $\mathsf{tr} = (\mathbf{c}, h, \gamma, v, (\delta_i)_{0\le i<\ell k}, (\mathbf{w}'_i, \mathbf{x}_i, v'_i, \mathbf{z}'_i)_{0\le i<\ell})$, send it to $\mathcal{A}$, receive a response $b = \mathcal{A}(\mathsf{ck}, \mathsf{tr})$, and output b.

Assume that $\mathbf{u} = \begin{bmatrix}\mathbf{I}_{\mu+k+3} \; \mathbf{A}\end{bmatrix}\mathbf{r}$ for $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k+3}$. First, it is easy to check that $\mathbf{c}_0 = \mathbf{B}_0\mathbf{r}$, $c_j = \langle \mathbf{b}_j, \mathbf{r}\rangle + m_j$ for $1 \le j \le k$ and $c_{k+1} = \langle \mathbf{b}_{k+1}, \mathbf{r}\rangle + g$. By the definition of Hint-MLWE, we can express $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}$ for $1 \le i < \ell$ where $\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k+3}$. Then we get $\mathbf{w}_i = \mathbf{B}_0\mathbf{z}_i - \varphi^i(\gamma)\cdot\mathbf{c}_0 = \mathbf{B}_0\mathbf{y}_i \pmod q$ for $0 \le i < \ell$ where $\mathbf{y}_i = \sum_{j=0}^{\ell-1}\varphi^i(X^j)\mathbf{y}'_j$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1}\varphi^i(X^j)\mathbf{z}'_j$, and $\mathbf{w}_i = \sum_{j=0}^{\ell-1}\varphi^i(X^j)\mathbf{w}'_j$ Since $\langle \mathbf{b}_j, \mathbf{z}_i\rangle = \langle \mathbf{b}_j, \mathbf{y}_i\rangle + \varphi^i(\gamma)\langle \mathbf{b}_j, \mathbf{r}\rangle = \langle \mathbf{b}_j, \mathbf{y}_i\rangle + \varphi^i(\gamma)\cdot u_j$, we get

$$c_{k+2} = \langle \mathbf{b}_{k+2}, \mathbf{r}\rangle + \langle \mathbf{b}_{k+3}, \mathbf{y}_0\rangle - \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1}\delta_{ik+j}\varphi^{-i}\left(3m_j\langle \mathbf{b}_j, \mathbf{y}_i\rangle^2\right)$$

$$c_{k+3} = \langle \mathbf{b}_{k+3}, \mathbf{r}\rangle + \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1}\delta_{ik+j}\varphi^{-i}\left((3m_j^2 - 1)\langle \mathbf{b}_j, \mathbf{y}_i\rangle\right).$$

In non-abort transcript, v and v'_i for $0 \le i < \ell$ are identical to those sampled from $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$. Therefore, the distribution of tr is identical to $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$ under the condition $\mathtt{Ver}_{\mathtt{Ter}}(\mathsf{tr}) = 1$.

Now let us assume that $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+k+3})$. Since there exists 1-to-1 correspondence between $\mathbf{u}$ and $\mathbf{c}$, the distribution of $\mathbf{c}$ is also uniform over $R_q^{\mu+k+3}$ independent to both m and g. In the simulator $\mathcal{S}_{\mathtt{Ter}}$, the distribution of $\mathbf{c} = \begin{bmatrix}\mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{(k+3)\times\mu} & \mathbf{I}_{k+3}\end{bmatrix}\mathbf{u}$ is also uniform over $R_q^{\mu+k+3}$ since $\begin{bmatrix}\mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{(k+3)\times\mu} & \mathbf{I}_{k+3}\end{bmatrix}$ is invertible over $R_q^{(\mu+k+3)\times(\mu+k+3)}$. In non-abort transcript, h is an element of $\{a \in R_q | a_0 = \cdots = a_{\ell-1} = 0\}$. Since g is uniform over this set, $h = g + \tau$ is also uniform and independent to τ. By the definition of Hint-MLWE, we can express $\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i\cdot\mathbf{r}$ for $1 \le i < \ell$ where $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k+3}$ and $\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k+3}$. Therefore, the distribution of tr is identical to $\mathcal{D}_1$ under the condition $\mathtt{Ver}_{\mathtt{Lin}}(\mathsf{tr}) = 1$.

Thus, the adversary $\mathcal{B}$ has the same advantage ε as $\mathcal{A}$ in distinguishing the Hint-MLWE instance. As a result, distributions $\mathcal{D}_0(\mathbf{M}, \mathbf{k}, \mathbf{m})$ and $\mathcal{D}_1$ of tr under the condition $\mathtt{Ver}_{\mathtt{Ter}}(\mathsf{tr}) = 1$ are computationally indistinguishable for any message $\mathbf{m} \in R_q^k$, matrix $\mathbf{M}$ and vector $\mathbf{k}$ if $\mathsf{HintMLWE}_{R,\nu,\mu+k+3,q,\sigma_1}^{\ell,\sigma_2,\mathcal{C}'}$ is hard, which implies the simulatability of our $\Pi_{\mathtt{Ter}}$. □

$\Pi_{\texttt{Ter}}$

Prover $\mathcal{P}$

Inputs: $\mathbf{B}_0 \in R_q^{\mu \times (\mu+\nu+k+3)}$

$\mathbf{b}_1, \ldots, \mathbf{b}_{k+3} \in R_q^{\mu+\nu+k+3}$

$m_1, \ldots, m_k \in \texttt{iNTT}(\{-1,0,1\}^n)$

$\mathbf{M} \in \mathbb{Z}_q^{m \times kn}, \mathbf{k} = \mathbf{M} \cdot (\texttt{NTT}(m_1) || \ldots || \texttt{NTT}(m_k)) \in \mathbb{Z}_q^m$

Verifier $\mathcal{V}$

$\mathbf{B}_0, \mathbf{b}_1, \ldots, \mathbf{b}_{k+3}$

$\mathbf{M}, \mathbf{k}$

Generate-phase:

$\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma_1}^{\mu+\nu+k+3}$

$\mathbf{c}_0 = \mathbf{B}_0 \mathbf{r},\ c_j = \langle \mathbf{b}_j, \mathbf{r} \rangle + m_j\ (1 \le j \le k)$ $\xrightarrow{\mathbf{c}_0, c_1, \ldots, c_k}$

Prove-phase:

$g \leftarrow \mathcal{U}(\{g \in R_q | g_0 = \cdots = g_{\ell-1} = 0\})$

$c_{k+1} = \langle \mathbf{b}_{k+1}, \mathbf{r} \rangle + g$

$\mathbf{y}'_i \leftarrow \mathcal{D}_{\mathbb{Z}^n, \sigma_2}^{\mu+\nu+k+3},\ \mathbf{w}'_i = \mathbf{B}_0 \mathbf{y}'_i\ (0 \le i < \ell)$ $\xrightarrow{c_{k+1}, \mathbf{w}'_i}$

$\mathbf{y}_i = \sum_{j=0}^{\ell-1} \varphi^i (X^j) \mathbf{y}'_i\ (0 \le i < \ell)$

Verifier: $\delta_0, \ldots, \delta_{\ell k-1} \leftarrow \mathcal{U}(R_q)$

Verifier: $\mathbf{x}_0, \ldots \mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$ $\xleftarrow{\delta_0, \ldots, \delta_{\ell k-1},\ \mathbf{x}_0, \ldots, \mathbf{x}_{\ell-1}}$

$$c_{k+2} = \langle \mathbf{b}_{k+2}, \mathbf{r} \rangle + \langle \mathbf{b}_{k+3}, \mathbf{y}_0 \rangle - \sum_{i=0}^{\ell-1} \sum_{j=0}^{k-1} \delta_{ik+j} \varphi^{-i} \left(3 m_j \langle \mathbf{b}_j, \mathbf{y}_i \rangle^2 \right)$$

$$c_{k+3} = \langle \mathbf{b}_{k+3}, \mathbf{r} \rangle + \sum_{i=0}^{\ell-1} \sum_{j=0}^{k-1} \delta_{ik+j} \varphi^{-i} \left((3m_j^2 - 1) \langle \mathbf{b}_j, \mathbf{y}_i \rangle \right)$$

$$v = \langle \mathbf{b}_{k+2}, \mathbf{y}_0 \rangle + \sum_{i=0}^{\ell-1} \sum_{j=0}^{k-1} \delta_{ik+j} \varphi^{-i} \left(\langle \mathbf{b}_j, \mathbf{y}_i \rangle^3 \right)$$

$\mathbf{M}^\top \mathbf{x}_i = \texttt{NTT}(\xi_{i,0}) || \ldots || \texttt{NTT}(\xi_{i,k-1})\ (0 \le i < \ell)$

$$h = g + \sum_{i=0}^{\ell-1} \frac{1}{\ell} X^i \sum_{s=0}^{\ell-1} \varphi^s \left(\sum_{j=0}^{k-1} n \xi_{i,j} m_j - \langle \mathbf{k}, \mathbf{x}_i \rangle \right)$$

$$v'_i = \sum_{p=0}^{\ell-1} \frac{1}{\ell} X^p \sum_{s=0}^{\ell-1} \sum_{j=0}^{k-1} \varphi^s \left(\langle n \xi_{p,j} \mathbf{b}_j, \mathbf{y}_{i-s} \rangle \right) + \langle \mathbf{b}_{k+1}, \mathbf{y}_i \rangle\ \ (0 \le i < \ell)$$

$\xrightarrow{c_{k+2}, c_{k+3},\ h, v, v'_i}$

Verifier: $\gamma \leftarrow \mathcal{C},\ \gamma = \sum_{j=0}^{\ell-1} \gamma_j X^j$ $\xleftarrow{\gamma_0, \ldots, \gamma_{\ell-1}}$

$\mathbf{z}'_i = \mathbf{y}'_i + \gamma_i \cdot \mathbf{r}\ (0 \le i < \ell)$ $\xrightarrow{\mathbf{z}'_i}$

Verifier: $\texttt{Ver}_{\texttt{Ter}}(\mathbf{c}_0, c_1, \ldots, c_{k+3}, \mathbf{w}'_i, \delta_i, \mathbf{x}_i, h, v, v'_i, \gamma, \mathbf{z}'_i)$

Fig. 13. Proof of knowledge of a ternary solution for linear system over $\mathbb{Z}_q^{m \times kn}$ [16].

$\texttt{Ver}_{\texttt{Ter}}(\mathbf{c}, \mathbf{w}'_i, \delta_i, \mathbf{x}_i, h, v, v'_i, \gamma, \mathbf{z}'_i)$

Compute $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}'_j,\ \mathbf{w}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{w}'_j \quad (0 \le i < \ell)$

Check $\|\mathbf{z}_i\|_2 < (n\sigma_1 + \sqrt{\ell}\sigma_2)\sqrt{n(\mu + \nu + k + 3)/\pi} \quad (0 \le i < \ell)$

Check $\mathbf{B}_0\mathbf{z}_i = \mathbf{w}_i + \varphi^i(\gamma)\mathbf{c}_0 \quad (0 \le i < \ell)$

Compute $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)\mathbf{c}_j \quad (0 \le i < \ell,\ 0 \le j < k)$

Compute $f_{k+2} = \langle \mathbf{b}_{k+2}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+2}$

Compute $f_{k+3} = \langle \mathbf{b}_{k+3}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+3}$

Check $\sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1} \delta_{ik+j}\varphi^{-i}\Big(f_{i,j}\big(f_{i,j} + \varphi^i(\gamma)\big)\big(f_{i,j} - \varphi^i(\gamma)\big)\Big)$

$+ f_{k+2} + \gamma \cdot f_{k+3} = v$

Check $h_0 = \cdots = h_{\ell-1} = 0$

Parse $\mathbf{M}^\top \mathbf{x}_i = \texttt{NTT}(\xi_{i,0})|| \dots ||\texttt{NTT}(\xi_{i,k-1})\ (0 \le i < \ell)$

Compute $\tau = \sum_{i=0}^{\ell-1} \frac{1}{\ell} X^i \sum_{s=0}^{\ell-1} \varphi^s \Big(\sum_{j=0}^{k-1} n\xi_{i,j}\mathbf{c}_j - \langle \mathbf{k}, \mathbf{x}_i \rangle \Big)$

Check $\sum_{p=0}^{\ell-1} \frac{1}{\ell} X^p \sum_{s=0}^{\ell-1}\sum_{j=1}^{k-1} \varphi^s\big(n\xi_{p,j}\left\langle \mathbf{b}_j, \mathbf{z}_{i-s} \right\rangle\big) + \langle \mathbf{b}_{k+1}, \mathbf{z}_i \rangle$

$= v'_i + \varphi^i(\gamma)(\tau + \mathbf{c}_{k+1} - h) \quad (0 \le i < \ell)$

Fig. 14. Verification procedure for $\Pi_{\texttt{Ter}}$

Simulator $\mathcal{S}_{\mathtt{Ter}}$

Input

$\mathbf{B}_0 \in R_q^{\mu\times(\mu+\nu+k+3)}, \mathbf{B}_1 \in R_q^{(k+3)\times(\mu+\nu+k+3)}, \mathbf{M} \in \mathbb{Z}_q^{m\times kn}, \mathbf{k} \in \mathbb{Z}_q^m$

---

1. Sample $\mathbf{u} \leftarrow \mathcal{U}(R_q^{\mu+k+3})$, $\mathbf{V} \leftarrow \mathcal{U}(R_q^{\mu\times(k+3)})$, and $(\gamma_0,\dots,\gamma_{\ell-1}) \leftarrow \mathcal{C}'$.
2. Sample $\mathbf{x}_0,\dots,\mathbf{x}_{\ell-1} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$ and $h \leftarrow \mathcal{U}(\{h \in R_q | h_0 = \cdots = h_{\ell-1} = 0\})$.
3. Sample $\delta_0,\dots,\delta_{k\ell-1} \leftarrow \mathcal{U}(R_q)$.
4. Sample $\mathbf{r} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_1}^{\mu+\nu+k+3}$ and $\mathbf{y}_i' \leftarrow \mathcal{D}_{\mathbb{Z}^n,\sigma_2}^{\mu+\nu+k+3}$ for $0 \le i < \ell$.
5. Compute $\mathbf{c} = \begin{bmatrix} \mathbf{I}_\mu & \mathbf{V} \\ \mathbf{0}^{(k+3)\times\mu} & \mathbf{I}_{k+3} \end{bmatrix} \mathbf{u} \pmod q$ and parse $\mathbf{c} = \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix}$ for $\mathbf{c}_0 \in R_q^\mu$, $\mathbf{c}_1 \in R_q^{k+3}$.
6. Compute $\mathbf{z}_i' = \mathbf{y}_i' + \gamma_i \cdot \mathbf{r}$, $\mathbf{z}_i = \sum_{j=0}^{\ell-1} \varphi^i(X^j)\mathbf{z}_j'$, and $\mathbf{w}_i' = \mathbf{B}_0\mathbf{z}_i' - \gamma_i \cdot \mathbf{c}_0 \pmod q$ for $0 \le i < \ell$.
7. Compute $f_{i,j} = \langle \mathbf{b}_j, \mathbf{z}_i \rangle - \varphi^i(\gamma)\mathbf{c}_j$ for $0 \le i < \ell,\ 0 \le j < k$.
8. Compute $f_{k+2} = \langle \mathbf{b}_{k+2}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+2}$.
9. Compute $f_{k+3} = \langle \mathbf{b}_{k+3}, \mathbf{z}_0 \rangle - \gamma \cdot \mathbf{c}_{k+3}$.
10. Compute $v = f_{k+2} + \gamma \cdot f_{k+3} + \sum_{i=0}^{\ell-1}\sum_{j=0}^{k-1} \delta_{ik+j}\varphi^{-i}\Big(f_{i,j}(f_{i,j} + \varphi^i(\gamma))(f_{i,j} - \varphi^i(\gamma)) \Big)$.
11. Parse $\mathbf{M}^\top \mathbf{x}_i = \mathtt{NTT}(\xi_{i,0}) || \dots || \mathtt{NTT}(\xi_{i,k-1})$ for $0 \le i < \ell$.
12. Compute $\tau = \sum_{i=0}^{\ell-1} \frac{1}{\ell} X^i \sum_{s=0}^{\ell-1} \varphi^s \Big(\sum_{j=0}^{k-1} n\xi_{i,j}\mathbf{c}_j - \langle \mathbf{k}, \mathbf{x}_i \rangle \Big)$
13. Compute $v_i' = \sum_{p=0}^{\ell-1} \frac{1}{\ell} X^p \sum_{s=0}^{\ell-1}\sum_{j=1}^{k-1} \varphi^s \Big(n\xi_{p,j} \langle \mathbf{b}_j, \mathbf{z}_{i-s} \rangle \Big) + \langle \mathbf{b}_{k+1}, \mathbf{z}_i \rangle - \varphi^i(\gamma)(\tau + \mathbf{c}_{k+1} - h)$ for $0 \le i < \ell$.
14. Output $(\mathbf{c}, \mathbf{w}_i', \delta_i, \mathbf{x}_i, h, v, v_i', \gamma, \mathbf{z}_i')_{0\le i<\ell}$.

Fig. 15. Simulator for $\mathit{\Pi}_{\mathtt{Ter}}$.