Ring/Module Learning with Errors under Linear Leakage – Hardness and Applications

Zhedong Wang[1,2], Qiqi Lai[3,2⋆], Feng-Hao Liu[4]

[1] School of Cyber Science and Engineering, Shanghai Jiao Tong University,
Shanghai, China. wzdstill@sjtu.edu.cn.
[2] State Key Laboratory of Cryptology, P. O. Box 5159, Beijing ,100878,China.
[3] School of Computer Science, Shaanxi Normal University, Xi'an, China.
laiqq@snnu.edu.cn.
[4] Washington State University, Pullman, WA, USA. feng-hao.liu@wsu.edu.

Abstract. This paper studies the hardness of decision Module Learning with Errors (MLWE) under linear leakage, which has been used as a foundation to derive more efficient lattice-based zero-knowledge proofs in a recent paradigm of Lyubashevsky, Nguyen, and Seiler (PKC 21). Unlike in the plain LWE setting, it was unknown whether this problem remains provably hard in the module/ring setting.

This work shows a reduction from the standard search MLWE to decision MLWE with linear leakage. Thus, the main problem remains hard asymptotically as long as the non-leakage version of MLWE is hard. Additionally, we also refine the paradigm of Lyubashevsky, Nguyen, and Seiler (PKC 21) by showing a more fine-grained tradeoff between efficiency and leakage. This can lead to further optimizations of lattice proofs under the paradigm.

1 Introduction

Ring/Module Learning with Errors (RLWE/MLWE) is an important foundation in the category of lattice-based cryptography, which is a plausible direction for post-quantum cryptography. RLWE/MLWE facilities more efficient constructions of public-key encryption, e.g., several candidates in the NIST PQC call, as well as advanced crypto systems including identity-based encryption [1,27,46,47] and fully homomorphic encryption [16], in comparison to those based on the plain LWE [2, 3, 14]. Due to the efficiency advantage, this problem has drawn a lot of attentions since its proposal [29, 37, 38, 45].

Zero-knowledge proof (ZKP) is a key technical tool in many applications with strong privacy requirements. Towards quantum-safe solutions, researchers have put a lot of efforts in the direction of lattice-based ZKP [7, 9, 11, 19, 20, 23, 32, 35, 36, 39]. Despite feasibility results (though not practical) in the standard common reference string (CRS) model [43], many new highly efficient solutions are constructed in the random oracle model in recent years, using the technique of

[⋆] Corresponding author

Fiat-Shamir with aborts [32]. In recent years, tremendous progress has been made to optimize the concrete efficiency, e.g., improving the proof sizes for showing knowledge of an $\boldsymbol{s}$ with small coefficients satisfying $\mathbf{A}\boldsymbol{s} = \boldsymbol{t}$, from 384 KB [11] to 47 KB [23]. Additionally, research in this line has deep impacts on the design of efficient lattice-based signatures [19, 20, 34] and as well other privacy-preserving protocols [24, 25, 30].

Recently, Lyubashevsky, Nguyen, and Seiler [36] identified a new paradigm that can improve the proof size by roughly 30% over the prior best constructions [23, 35], by using *leakage* to trade efficiency. More specifically, they derive a novel modified rejection sampling strategy, called *subset rejection sampling*, that leaks one bit of the randomness of a one-time commitment, which is used in the commit-and-prove paradigm. This key technique to the efficiency improvements, is allowing smaller proofs. For security, as long as the one-time commitment is leakage resilient against this class of leakage, then the overall scheme is secure.

Now, the question turns to whether we can prove that the one-time commitment is leakage resilient to one bit. To do this, the work [36] showed that this task can be reduced to a leakage version of the *decisional* MLWE problem against linear functions,[5] i.e., as long as the decisional MLWE problem is leakage resilient for linear functions (over the coefficients of the secret and the error), then so is the one-time commitment against the same class, implying security against the bit leakage applied to any linear function.

Despite the fact that there are reduction results showing positive results in the plain-LWE settings [4, 36, 41], it was identified as an important open question in [36] whether the same results carry to the ring setting. On the other hand, the work [36] speculated that one-bit leakage will not hurt security, at least under the currently best known attacks. However, it is not clear whether the leakage version of $\mathsf{RLWE}/\mathsf{MLWE}$ is inherently hard or we just have not found an attack yet by exploiting the ring structure with the leakage. This motivates the main research goal of the work:

> (**Main Research Goal**) Determine whether the leakage version of decisional $\mathsf{RLWE}/\mathsf{MLWE}$ against linear functions (as required in [36]) is inherently hard (as $\mathsf{RLWE}/\mathsf{MLWE}$).

The new rejection sampling paradigm [36] provides a promising opportunity for achieving more practical lattice proofs, with numerous identified applications. Therefore, it is crucial to thoroughly investigate the underlying hardness foundations, to ensure that using leakage to trade efficiency does not hurt security in a provable manner. This would enhance the confidence in the practical adoption of this emerging paradigm. Our main goal is the key to achieve this.

1.1 Our Results

This work provides an affirmative answer to the main task. Particularly, our main result is to prove the following informal theorem.

[5] In fact, the work [36] only needs a slightly weaker version known as extended (R)LWE.

Theorem 1.1 (Main Result, Informally Stated) *Under the hardness of search* RLWE *(for appropriate parameters), the decisional* RLWE *under leakage of linear functions (required as [36]) is hard.*

In summary, our result provides stronger theoretical justification, increasing confidence in the foundation of the design paradigm [36]. This result has practical implications, as it can be applied to enhance the efficiency of Zero-Knowledge Proofs (ZKP) and other lattice-based cryptographic systems. Additionally, our reduction can be generalized to the module setting, i.e., MLWE, offereing more flexibility in terms of design choices.

An important aspect of our contribution is that our reduction works in the full-splitting setting, where $q\mathcal{R}$ completely splits into linear factors. Before this paper, as far as we know, there is limited knowledge regarding MLWE/RLWE with leakage in the full-splitting setting, as compared to the low-splitting setting [31]. Given that the full-splitting structure plays a crucial role in various efficient lattice proofs, including some recent works of [7, 23, 33] for establishing more general relations and improving efficiency, our advances in this setting are significant.[6] We present further details in the technical overview (Section 1.2) and Section 4.

Our second contribution is to refine the subset rejection sampling strategy of [36], showing a more fine-grained tradeoff between efficiency and leakage, in the case of full-splitting underlying ring. Particularly, as listed in Table 1, if we allow $\log_2 6$ bits of leakage, the rejection sampling parameter α can be slightly improved from 175.67 to 171.42, which slightly reduces the proof size from 16.56 KB to 16.48 KB. This can be further stretched – using $\log q$ bits of leakage to improve the parameters by a factor of 52% (83.138 versus 175.67), which reduces the proof size by a factor of 10% (14.96 KB versus 16.56 KB). Here, q is the underlying modulus. By Theorem 1.1, the problem remains hard in these leakage settings, at least asymptotically. An interesting open problem is to determine concrete security of ℓ-bit leakage and the efficiency tradeoff, finding the optimal parameters for the best efficiency. We present more details later in the technical overview and Section 4.4.

1.2 Technical Overview

In this section, we present an overview of our techniques. First we describe the computational problem of the main focus – decisional Module Learning with Errors (MLWE) with linear leakage, and then the hardness results and applications.

Problem Statement. Let $\mathcal{R}_q$ denote some (polynomial) residual ring of degree d and modulus q, e.g., $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$, and later on $\mathcal{R}$ refers the underlying ring and q is the modulus. We notice that the MLWE problem can be stated as the following: given a ring matrix/vector $\mathbf{A} \in \mathcal{R}_q^{m\times n}$ and a ring vector $\boldsymbol{b} = \mathbf{A}\cdot\boldsymbol{s} + \boldsymbol{e}$

[6] In the very recent work [33], while the full-splitting structure is not required to prove the ℓ_2 norm, it is still necessary to prove the ℓ_∞ norm or the knowledge of the component-wise product of two vectors.

	rep.	α	Size of $\boldsymbol{z}_i$: $\hat{k}\eta d \log_2(12 \cdot \alpha)$	$\mathfrak{l}$
Rej_0	≈ 6	2241.41	22.042KB	0
Rej_1	≈ 6	175.67	16.56 KB	1
Rej_2	≈ 6	171.42	16.48 KB	$\log_2 6$
Rej_3	≈ 6	83.138	14.96 KB	≈ 32

Table 1. Rough comparison of efficiency under different rejection sampling algorithms for the opening protocol with full-splitting underlying ring in Fig. 2. Here rep. denotes prover's expected repetition times, $\mathfrak{l}$ the number of leakage bits, α the derivation of the discrete Gaussian, which will influences the proof size of $\boldsymbol{z}_i$. The concrete parameters are listed using the following example setting: the dimension η of $\boldsymbol{z}_i$ is 3, the ring dimension d is 1024, the modulus q is roughly 2^{32}, and the boosting parameter $\hat{k}$ is 4.

where $\boldsymbol{s} \in \mathcal{R}_q^n$ is some secret ring vector and $\boldsymbol{e} \in \mathcal{R}_q^m$ is some small error ring vector, the search problem asks to find the secret $\boldsymbol{s}$ and the decision version asks to distinguish $\boldsymbol{b}$ from a uniformly random vector. The module setting captures both the RLWE and plain LWE as special cases – if the module rank is one, i.e., $n = 1$, then the problem is RLWE. On the other hand, if the underlying ring has degree $d = 1$, then this is the plain LWE. All these variants have been extensively studied [29, 37, 42] and we have strong confidence in their hardness.

To study the leakage version of the MLWE, we first define the leakage function of our interests, which is the class required in [36]. Let $L_{\boldsymbol{a},\boldsymbol{a}'}(\boldsymbol{s}, \boldsymbol{e})$ be defined as $\langle \phi(\boldsymbol{a}), \phi(\boldsymbol{s}) \rangle + \langle \phi(\boldsymbol{a}'), \phi(\boldsymbol{e}) \rangle \in \mathbb{Z}_q$, where ϕ is the coefficient embedding function, i.e., it maps a ring element into a vector of $\mathbb{Z}_q^d$ that represents the coefficient vector with respect to the power basis $(1, X, X^2, \ldots, X^{d-1})$, and maps ring vectors $\mathcal{R}_q^n$ to $\mathbb{Z}_q^{nd}$, analogously. In this work, we consider the class that contains all such functions regarding the inner product of the coefficient embeddings over both the secret $\boldsymbol{s}$ and the error $\boldsymbol{e}$.[7] Again, we would emphasize that leakage over *both* the secret and error is a critical requirement in the paradigm of [36].

Given the above context, MLWE with linear leakage can be defined in a simple way – the adversary/solver is given $L_{\boldsymbol{a},\boldsymbol{a}'}(\boldsymbol{s}, \boldsymbol{e})$ in addition to the regular MLWE samples. The task of the problem then becomes to find the secret $\boldsymbol{s}$ or distinguish $\boldsymbol{b}$ from the uniform vector, given the leakage. We notice that this problem is very related to another notion called extended MLWE [12] with the following difference: the extended MLWE chooses $\boldsymbol{a}, \boldsymbol{a}'$ from a small discrete Gaussian distribution, yet our leakage version of MLWE allows the adversary to specify $\boldsymbol{a}, \boldsymbol{a}'$ in the beginning of the experiment. Thus, our leakage version of MLWE is stronger than the extended MLWE.

For the application need, we consider the case where the secret $\boldsymbol{s}$ is sampled according to the discrete Gaussian distribution, the same as the error $\boldsymbol{e}$.

Some Prior Results. We first review previous works and then discuss their limitations, particularly the obstacles they face in analyzing the foundation of [36].

[7] In fact, our leakage class in the main body is slightly more general, i.e., the leakage function can include slight multiplicative shifts. Nevertheless, this simplified version is sufficient to demonstrate our core ideas in the introduction.

- In the context of plain LWE, it was demonstrated that the extended LWE is provably as hard as LWE [4,41]. However, as highlighted in [36], this technique does not extend to the ring/module setting due to either a loss of exponential reduction or a dimension mismatch during the reduction transformation.
- The work [12] (and the later journal version [13]) studied a version of extended MLWE. Their leakage function takes the form $\langle \boldsymbol{z}, \boldsymbol{e} \rangle$, where the inner product is defined according to the ring vectors. It should be noted that there exists a gap between the reduction in [13] and the application of ZKP in [36], as the latter requires the leakage is over both the secret and error, and the inner product is defined according to the vectors over $\mathbb{Z}_q$ (under the coefficient embeddings). Besides, their reduction limits to the MLWE (i.e. module rank $k \geq 2$), and is unable to capture the case of RLWE.
- Two recent and concurrent works [22,28] considered the case of MLWE with leakages. Among them, [22] examined a scenario where the leakage is applied to the error but not the secret, and [28] examined the scenario where the leakage is applied to both the secrete and error. However, these two works both have several limitations. Specifically, the leakage function in [22] takes the form $\boldsymbol{e} \cdot \mathbf{Z} + \boldsymbol{e}'$, where $\mathbf{Z}$ is a low-norm ring matrix specified by the adversary and $\boldsymbol{e}'$ is an independent Gaussian error hidden to the adversary. As their analysis relies on the inclusion of $\boldsymbol{e}'$, their results are not expected to be applicable to our setting and are therefore insufficient for analyzing the framework of [36]. [28] specified the leakage function with the form $c \cdot \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{e} \end{pmatrix} + \boldsymbol{y}$, where $\boldsymbol{y}$ is a gaussian vector hidden to adversary. The analysis of [28] is also unable to be directly applied for the framework of [36], as the latter requires to analyze the leakage function with the form $\left\langle \phi \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{e} \end{pmatrix}, \phi(\boldsymbol{z}) \right\rangle$, which can not be simulated by the function in [28].

  We note that it is unknown whether our results can be inferred from those of [22,28] or vice versa, so we consider them as incomparable results.
- Another approach to analyze leakage is by employing the lossy-matrix technique [5,15,31], yet the current developments have several limitations. For instance, the work [5] is only applicable to the plain LWE setting due to the absence of the leftover hash lemma in the ring setting at that time. The work [15] is limited to the search version, and it was unclear how to extend their techniques to the decision version. The work [31] derived a ring-leftover hash lemma (LHL), and generalized the analysis of [5] to the module setting, i.e., MLWE. Nonetheless, there are subtleties where their analytical techniques [31] cannot be applied, as elaborated below.

  Particularly, let n be the module rank, d be the ring dimension, and $q\mathcal{R}$ splits into c factors for $1 \leq c \leq d$. Their result (particularly the ring LHL) requires that $n = \omega(c)$ in order to guarantee the required entropy lower bound. Thus, in the low-splitting setting (e.g., $c = 2$), the techniques [31] can be used to analyze for $n = O(1)$, e.g., $n = 2$. However, for the high-splitting (e.g., $c = d$), then their technique requires $n > d$. To choose more competitive parameters in many practical works, n is set to be $O(1)$ (even

1 for the RLWE), e.g., [23]. Thus, the technique of [31] is not sufficient to analyze these practical parameter choices in the full/high splitting setting.

- Besides the reductions of several versions of MLWEwith leakages, the work [18] also considers the concrete security estimation of (M)LWE with side information. We note that the reduction and concrete security estimation are two different perspectives for studying the hardness of (M)LWE with leakages. Combining the two ways provides us more comprehensive understanding about the hardness of this problem.

To summarize, we observe that the setting involving low module rank and high-splitting is not well understood compared to other settings. As many efficient lattice proofs rely on specific algebraic advantages in this setting, e.g., [7, 23], and [33] for more general relationships, there is a strong motivation to address the challenges and develop new analytical techniques for the foundation and applications.

Our New Analysis. To achieve this, we prove a new reduction from search MLWE (without leakage) to decisional MLWE with linear leakage, meaning that the linear leakage does not decrease the hardness up to a polynomial factor. Our proof structure is similar to that of [29,31,37], consisting of six steps as Figure 1. Below we briefly elaborate on the intermediate problems and the technical advances over the prior work, i.e., why prior analyses do not go through directly and how our new techniques solve the challenges.

Our reduction works in the case where $q\mathcal{R}$ splits completely into d ideals with linear degree, i.e., $q\mathcal{R} = \mathfrak{q}_1 \dots \mathfrak{q}_d$. Next we describe the notations in the diagram – S and D to denote search and decision version. LE denotes leakage of linear error and LS denotes leakage of linear secret (and error), and (A)/(W) denotes average-case/worst-case over the secret distribution. The $\mathfrak{q}_i$-MLWE problem asks the solver to find $\boldsymbol{s} \mod \mathfrak{q}_i$. The decisional MLWELE^i is to distinguish $\boldsymbol{b}+\boldsymbol{h}$ where $\boldsymbol{h}$ is either from A^i or A^{i-1} defined as follow. A^i is uniformly random mod $\mathfrak{q}_j\mathcal{R}$ for all $j \leq i$, and 0 mod all the other ideals, i.e., $\mathfrak{q}_j\mathcal{R}$'s for $j > i$.

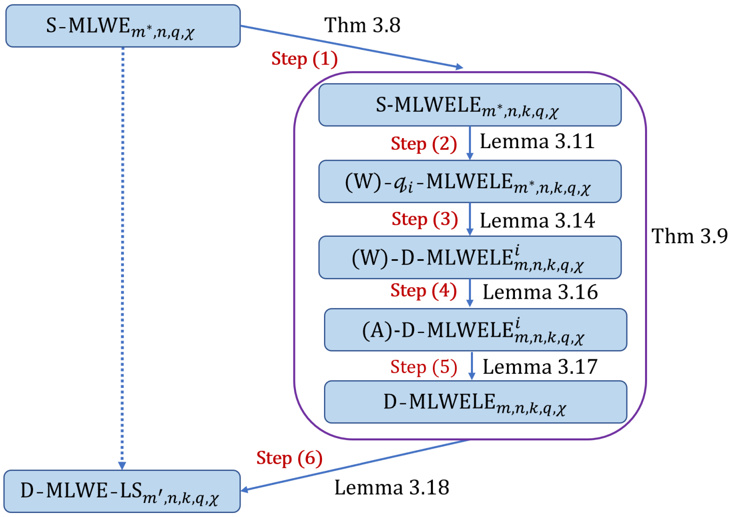

S-MLWE$_{m^*,n,q,\chi}$
Thm 3.8
Step (1)
S-MLWELE$_{m^*,n,k,q,\chi}$
Step (2)
Lemma 3.11
(W)-$\mathfrak{q}_i$-MLWELE$_{m^*,n,k,q,\chi}$
Step (3)
Lemma 3.14
Thm 3.9
(W)-D-MLWELE$^i_{m,n,k,q,\chi}$
Step (4)
Lemma 3.16
(A)-D-MLWELE$^i_{m,n,k,q,\chi}$
Step (5)
Lemma 3.17
D-MLWELE$_{m,n,k,q,\chi}$
Step (6)
D-MLWE-LS$_{m',n,k,q,\chi}$
Lemma 3.18

Fig. 1. Our reduction route

In our reduction route, we introduce an intermediate problem denoted as MLWELE, for which the leakage function is only applied to the error. We first establish the one-way hardness of MLWELE on the hardness of search MLWE. The idea of this step is directly from a random guess of leakage, resulting $1/q$ reduction loss. Then we further show a search-to-decision reduction of MLWELE, which follows the framework from [31,37], but makes several important changes. Finally, we show a reduction from the intermediate problem MLWELE to our target MLWE-LS problem.

Now we briefly discuss each step in the figure. As Steps (1), (3), (4) and (5) follow essentially the same idea from the prior work [31, 37], we do not repeat the ideas. So next we focus on Steps (2) and (6).

For Step (2), we would like to prove the following – if we can find $\boldsymbol{s} \mod \mathfrak{q}_i$ for some i, then we can find $\boldsymbol{s}$ (given leakage of error). To achieve this, we first try to apply the automorphism argument of [31,37] – finding $\sigma(\boldsymbol{s}) \mod \mathfrak{q}_i$ implies finding $\boldsymbol{s} \mod \mathfrak{q}_j$ for another j. By going through all the automorphisms, we would recover $\boldsymbol{s}$ modulo every ideal, and thus by the Chinese Remainder Theorem recover $\boldsymbol{s}$. This idea faces a subtlety in the presence of leakage – the reduction needs to simulate $L_{\sigma(\boldsymbol{a})}(\sigma(\boldsymbol{e}))$ faithfully in order to call the underlying solver that finds $\sigma(\boldsymbol{s}) \mod \mathfrak{q}_i$. For general leakage functions, this task is unclear. Fortunately for the linear leakage in our case, we can prove $\langle\phi(\boldsymbol{a}), \phi(\boldsymbol{e})\rangle = \langle\phi(\sigma(\boldsymbol{a})), \phi(\sigma(\boldsymbol{e}))\rangle$ under the coefficient embedding in the cyclotomic rings of two's powers. This implies that the linear leakage is invariant under automorphism, and thus our reduction can faithfully simulate the leakage and complete the process as in the prior work.

We remark that this invariance of linear leakage under automorphism is nontrivial. Particularly, it requires that the bases corresponding to the coefficient embeddings are invariant (up to some re-ordering and sign) under automorphisms. This requirement however, does not always hold, and even for some rings such a basis does not exist. Currently, we only know that the Normal Integral Basis (NIB) mentioned in [31] and the power basis of cyclotomic rings with 2's powers considered in this paper meet this requirement.

For Step (6), our target is to show a reduction from MLWE with linear leakage of error to MLWE with linear leakage of both secret and error – if we can transform an instance of MLWELE to a valid instance of MLWE-LS or a random sample to another random sample, then we can distinguish the instance of MLWELE from random sample by invoking the distinguisher of MLWE-LS. Our idea is somehow similar to the hardness reduction of HNF-MLWE in prior works [6, 29], but needs very subtle analysis due to the introduced hints and leakage. Briefly, let $(\boldsymbol{a}, b, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1 e, \cdots, c_k e)\rangle$ be the instance of MLWELE, where $\boldsymbol{z}$ and $(c_1, \cdots, c_k)$ are the hints of leakage. The goal is to simulate an instance of MLWE-LS with the form: $(\boldsymbol{a}', b', \boldsymbol{z}', (c'_1, \cdots, c'_k), \langle \phi(\boldsymbol{z}'), \phi(c'_1(s, e), \cdots, c'_k(s, e))\rangle$. We can use the similar approach of the hardness reduction of HNF-MLWE to transform b to $b' = \langle \boldsymbol{a}', \bar{\boldsymbol{e}}\rangle + e$, where $\bar{\boldsymbol{e}}$ is the error vector corresponding to the invoked instances, and e is the error of the initial instance. Thus the leakage in MLWE-LS can be simulated by the linear combination of the leakages of error obtained during calling the MLWELE oracle.

We would like to point out a subtle issue involved in this transformation where the hints $\boldsymbol{z}', (c'_1, \cdots, c'_k)$ should be consistent with the leakage. As described above, our reduction is similar to the approach of the hardness reduction of HNF-MLWE, and thus requires to sample $n + 1$ instances of MLWELE. Therefore, we need to determine the hint vectors $(\boldsymbol{z}', c'_1, \cdots, c'_k)$ of MLWE-LS from $n+1$ tuples of hints of MLWELE. We tackle this barrier by a precise design of hints $\boldsymbol{z}'$ and $(c'_1, \cdots, c'_k)$, which makes use of the linearity of our leakage function and the ability of the adversary of MLWELE. The details can be referred to Lemma 3.18 in Section 3.

Our Second Contribution. Under the hardness of MLWE with linear leakage, our second contribution shows how to further improve the generalized rejection sampling paradigm of [36], deriving a more fine-grained tradeoff between efficiency and leakage. We elaborate on the high level ideas below.

Briefly speaking, the rejection sampling-style lattice proofs have the following structure: $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}$, where c is some small ring element, $\boldsymbol{s}$ is some small secret, $\boldsymbol{y}$ is some Gaussian mask, and $\boldsymbol{z}$ is the proof message sent to the verifier. To achieve zero-knowledge, $\boldsymbol{y}$ must wipe out the information of $\boldsymbol{s}$. If $\boldsymbol{y}$ is super-polynomially larger than $c\boldsymbol{s}$, then this is the well-known smudging noise technique [41]. However, this would require a very large proof $\boldsymbol{z}$. To reduce the size, Lyubashevsky [32] introduced the rejection sampling technique where $\boldsymbol{z}$ might be set to $\perp$ with a certain probability. In this way, the dependency on $\boldsymbol{s}$ can be removed with a much smaller $\boldsymbol{y}$. To further improve the size, [36] identified a new way – by imposing an additional condition on $\langle \phi(\boldsymbol{z}), c\phi(\boldsymbol{s})\rangle \geq 0$ (or rejecting the case when the inner product is negative), one can further reduce the size of $\boldsymbol{y}$. This comes at the price of leaking one bit, i.e., the sign bit. If

MLWE under linear leakage is hard, then leaking this bit would not hurt security of the protocol.

To further improve the size of $\boldsymbol{y}$, we observe that we can use a stronger condition $\langle \phi(\boldsymbol{z}), c\phi(\boldsymbol{s}) \rangle \geq T$ for some parameter $T > 0$. Intuitively, a larger T can result in smaller proof, yet at the cost of more leakage. If we completely leak $\langle \phi(\boldsymbol{z}), c\phi(\boldsymbol{s}) \rangle$, the size of $\boldsymbol{y}$ can be minimized. However, if the whole $\mathbb{Z}_q$ element is leaked, then the concrete hardness might be affected by the attack of [18]. Even though [18] does not solve the MLWE asymptotically, leaking $\log q$-bits (i.e., one element in $\mathbb{Z}_q$) might decrease the concrete security by a noticeable amount, whereas leaking one or two bits might not (as the framework of [18] does not apply). Therefore, stretching the leakage too much might be worse in practice. We leave it as an interesting open problem to determine the best tradeoff between leakage and concrete security.

A recent work [28] developed new ideas to improve the proof of knowledge protocols in [36]. Specifically, their approach can remove the computational overhead from repetition (abort) in the framework of [36]. We clarify that same as [36], our framework also requires more computational overhead compared with [28]. However, our framework can achieve better communication overhead than [28]. Concretely, the output size of our improved algorithm is s-maller than [28]. As a fair comparison, we calculate the parameters under the same benchmark defined in [28]. By accurate calculation, we can achieve output size that is approximately 3.6x smaller than their output size. More details of comparison can be referred to the end part of Section 4.4.

2 Preliminaries

Notations. In this paper, $\mathbb{Z}$ and $\mathbb{R}$ denote the sets of integers and real numbers. We use λ to denote the security parameter, which is the implicit input for all algorithms presented in this paper. A function $f(\lambda) > 0$ is negligible and denoted by $\mathsf{negl}(\lambda)$ if for any $c > 0$ and sufficiently large λ, $f(\lambda) < 1/\lambda^c$. A probability is called to be overwhelming if it is $1-\mathsf{negl}(\lambda)$. A column vector is denoted by a bold lower case letter (e.g., $\boldsymbol{x}$). A matrix is denoted by a bold upper case letter (e.g., $\mathbf{A}$). For a vector $\boldsymbol{x}$, its Euclidean norm (also known as the ℓ_2 norm) is defined to be $\|\boldsymbol{x}\| = (\sum_i x_i^2)^{1/2}$. For a matrix $\mathbf{A}$, its ith column vector is denoted by $\boldsymbol{a}_i$ and its transposition is denoted by $\mathbf{A}^\top$. And the norm of an element in $\mathcal{R}_q$ will be the norm of its unique representative with coefficients in $[-(q-1)/2, (q-1)/2]$. For positive $\beta \in \mathbb{R}$, we use S_β to denote the set of all polynomials of infinity norm less than β, i.e., $S_\beta = \{a \in \mathcal{R} \mid \|a\|_\infty \leq \beta\}$.

For positive integers n, q, let $[n]$ denote the set $\{1, ..., n\}$ and $\mathbb{Z}_q$ denote the ring of integers modulo q. For a distribution or a set X, we write $x \overset{\$}{\leftarrow} X$ to denote the operation of sampling an uniformly random x according to X. We denote as $\mathsf{Supp}(X)$ the support of a distribution X. For two distributions X, Y, we let $\mathsf{SD}(X, Y)$ denote their statistical distance. We write $X \overset{s}{\approx} Y$ to mean that they are statistically close, and $X \overset{c}{\approx} Y$ to say that they are computationally indistinguishable.

2.1 Cyclotomic Rings

Throughout this paper, we use $\mathcal{R}$ to denote a polynomial ring of the form $\mathbb{Z}[X]/(\varPhi_m(X))$, where $\varPhi_m(X)$ is the m^{th} cyclotomic polynomial. For an integer $q \in \mathbb{Z}$, we also consider the quotient ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. We recall that for d being a power of 2, the $2d$-th cyclotomic polynomial is given as $\varPhi_{2d}(x) = x^d + 1$. Then the ring of integers of the $2d$-th cyclotomic field $\mathcal{R} = \mathbb{Z}[x]/(x^d + 1)$. Thus, we can use the coefficients of an integer polynomial modulo $(x^n + 1)$ to represent a ring element.

Embedding and Rotation. In this work, we view elements of $\mathcal{R}$ as $\mathbb{Z}^d$ through certain embeddings. For example, for $\mathcal{R} = \mathbb{Z}[x]/(x^d + 1)$ with d a power of 2, we view any $a = a_0 + a_1 x + \cdots + a_{d-1}x^{d-1} \in \mathcal{R}$ for $a_i \in \mathbb{Z}$ as the coefficient vector $(a_0, \cdots, a_{d-1})$, and denote $\phi(a) = (a_0, \cdots, a_{d-1})$; and for $\mathcal{R} = \mathbb{Z}[x]/\varPhi_m(X)$ with m a prime, we view any $b = b_0 + b_1\zeta + \cdots + b_{m-1}\zeta^{m-1} \in \mathcal{R}$ for $b_i \in \mathbb{Z}$ as $(b_0, \cdots, b_{d-1})$, where ζ is the m-th root of unity. Similarly, we denote $\mathsf{Rot}(a)$ as rotation matrix of a, i.e.,

$$\mathsf{Rot}(a) = \begin{bmatrix} \phi(a \bmod q\mathcal{R})^\top \\ \phi(a \cdot x \bmod q\mathcal{R})^\top \\ \vdots \\ \phi(a \cdot x^{d-1} \bmod q\mathcal{R})^\top \end{bmatrix}.$$

It's easy to verify $\phi(sr) = \phi(s) \cdot \mathsf{Rot}(r) = \phi(rs) = \phi(r) \cdot \mathsf{Rot}(s)$ for any $s, r \in \mathcal{R}_q$.

Ideal Factorization. An ideal $I \subset \mathcal{R}$ is an additive subgroup that is closed under multiplication by $\mathcal{R}$. For an integer prime $q \in \mathbb{Z}$, $q\mathcal{R}$ is an ideal of $\mathcal{R}$, and the factorization of $q\mathcal{R}$ is as $q\mathcal{R} = \varPi_i \mathfrak{q}_i^e$, where $\mathfrak{q}_i$ are distinct prime ideals, each of norm $q^{\frac{d}{te}}$ with t the number of distinct ideals.

The number field $\mathbb{Q}[X]/(\varPhi_m(X))$ has $\varphi(m)$ automorphisms σ_k, which are defined by $\sigma_k(\zeta) = \zeta^k$ for $k \in \mathbb{Z}_m^*$. Particularly, for $\mathbb{Q}[X]/(X^d + 1)$, σ_k are defined by $\sigma_k(X) = X^k$. The following lemma says that the automorphisms σ_k "act transitively" on the prime ideals $\mathfrak{q}_i$, i.e., each $\mathfrak{q}_i$ is sent to each $\mathfrak{q}_j$ by some automorphism σ_k.

Lemma 2.1 ([37], Lemma 2.16) *For any $i, j \in \mathbb{Z}_m^*$, we have $\sigma_j(\mathfrak{q}_i) = \mathfrak{q}_{i/j}$.*

Next we recall the Chinese Remainder Theorem (CRT) for $\mathcal{R}$.

Lemma 2.2 (Chinese Remainder Theorem) *Let $\mathfrak{q}_i$ be pairwise coprime ($\mathfrak{q}_i + \mathfrak{q}_j = \mathcal{R}$ for any $i \neq j$) ideals in $\mathcal{R} = \mathbb{Z}[X]/(\varPhi_m(X))$, then natural ring homomorphism is an isomorphism: $\mathcal{R}/\Big(\prod_i \mathfrak{q}_i\Big)\mathcal{R} \to \bigoplus_i (\mathcal{R}/\mathfrak{q}_i\mathcal{R})$.*

2.2 Discrete Gaussian Distribution

For a ring $\mathcal{R}$ of degree d, we can define the discrete Gaussian distribution over it in the following way.

Definition 2.3 *For any positive integer ℓ, the discrete Gaussian distribution over $\mathcal{R}^\ell$ centered around $\boldsymbol{v} \in \mathcal{R}^\ell$ with standard deviation $\sigma > 0$ is given by*

$$D^{\ell \cdot d}_{\boldsymbol{v},\sigma}(\boldsymbol{z}) = \frac{e^{-\|\boldsymbol{z}-\boldsymbol{v}\|^2/2\sigma^2}}{\sum_{\boldsymbol{z}' \in \mathcal{R}^\ell} e^{-\|\boldsymbol{z}'\|^2/2\sigma^2}}.$$

When $\boldsymbol{v} = 0$, we just write $D^{\ell \cdot d}_{\sigma}$ for simplicity.

We also need to use the following facts about the discrete Gaussian distribution.

Lemma 2.4 (Generalize of [8]) *For any positive integer ℓ and any real $\sigma > 0$, a sample sampled from $D^{\ell \cdot d}_{\sigma}$ defined as above has norm at most $\sigma\sqrt{\ell d}$ except with probability at most $2^{-2\ell d}$.*

Lemma 2.5 (Lemma 4.3 in [32]) *For any vector $\boldsymbol{v} \in \mathbb{R}^m$ and any $\sigma, r > 0$,*

$$\Pr[|\langle \boldsymbol{z}, \boldsymbol{v} \rangle| > r : \boldsymbol{z} \overset{\$}{\leftarrow} D^m_\sigma] \leq 2e^{-\frac{r^2}{2\|\boldsymbol{v}\|^2\sigma^2}}.$$

2.3 MLWE

Now we introduce the hard problems discussed in this paper, which are denoted as S-MLWE and D-MLWE, and we consider the "*non-dual*" version problems.

Definition 2.6 (S-MLWE [29]) *The search* MLWE *problem with parameters n, m, q, and an error distribution χ such that* $\mathsf{Supp}(\chi) \in \mathcal{R}$ *denoted as S-*$\mathsf{MLWE}_{n,m,q,\chi}$ *is defined as follows. For $\boldsymbol{s} \overset{\$}{\leftarrow} \mathcal{R}^n$, use $A_{q,\boldsymbol{s}}$ to denote the distribution of $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathcal{R}^n_q \times \mathcal{R}_q$, where $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}^n_q$ and $e \overset{\$}{\leftarrow} \chi$. The goal is to find secret $\boldsymbol{s}$ from m samples.*

Definition 2.7 (S-MLWE in HNF [29]) *The search* MLWE *problem with parameters n, m, q, and an error distribution χ such that* $\mathsf{Supp}(\chi) \in \mathcal{R}$ *denoted as S-*$\mathsf{MLWE}_{n,m,q,\chi}$ *is defined as follows. For $\boldsymbol{s} \overset{\$}{\leftarrow} \chi^n$, use $A_{q,\boldsymbol{s}}$ to denote the distribution of $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathcal{R}^n_q \times \mathcal{R}_q$, where $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}^n_q$ and $e \overset{\$}{\leftarrow} \chi$. The goal is to find secret $\boldsymbol{s}$ from m samples.*

Definition 2.8 (D-MLWE in HNF [29]) *The decision* MLWE *problem with parameters n, m, q, and an error distribution χ such that* $\mathsf{Supp}(\chi) \in \mathcal{R}$ *denoted as D-*$\mathsf{MLWE}_{n,m,q,\chi}$ *is defined as follows. For $\boldsymbol{s} \overset{\$}{\leftarrow} \chi^n$, use $A_{q,\boldsymbol{s}}$ to denote the distribution of $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e) \in \mathcal{R}^n_q \times \mathcal{R}_q$, where $\boldsymbol{a} \overset{\$}{\leftarrow} \mathcal{R}^n_q$ and $e \overset{\$}{\leftarrow} \chi$. The goal is to distinguish m samples from either $A_{q,\boldsymbol{s}}$ or $\mathcal{U}(\mathcal{R}^n_q, \mathcal{R}_q)$.*

We notice that the latter two types MLWE problems defined above are the so-called "Hermite Normal Form" version, which can be easily reduced to the standard MLWE via the approach in [6]. For standard MLWE, it is known to be at least as hard as certain standard lattice problems over ideal lattice in the worst case [29]. It should be pointed out that RLWE is the special case of $n = 1$.

3 Hardness: MLWE with Linear Leakage

In this section, we present our main result for the MLWE under linear leakage. First we describe a table of parameters used in this section. Then we define the class of *linear leakage* in the ring/module setting, and Module Learning with Errors, i.e., MLWE in the leakage setting of this class. Finally we present the reduction result.

Parameters	Description
n	MLWE rank
m	number of MLWE samples
q	modulus of MLWE
d	ring dimension
ℓ	number of prime ideal factors of $q\mathcal{R}$
k	number of computing inner product times

Table 2. Notation of parameters in this section

Definition 3.1 *Let $l, q, d, k > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$. For $\boldsymbol{z} = (\boldsymbol{z}_i)_{i \in [k]} \in \mathcal{R}_q^{kl}, \boldsymbol{c} = (c_1, \ldots, c_\nu)^\top \in \mathcal{R}_q^k$, we define the function $L_{\boldsymbol{z},\boldsymbol{c}} : \mathcal{R}_q^l \to \mathbb{Z}_q$ as $L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{x}) = \sum_{i=1}^{k} \langle \phi(\boldsymbol{z}_i), \phi(c_i \boldsymbol{x}) \rangle$, where ϕ is a "coefficient embedding" map from $\mathcal{R}_q^l$ to $\mathbb{Z}_q^{dl}$, i.e., embeds each ring element in $\mathcal{R}_q$ as a vector in $\mathbb{Z}_q^d$.*

Here we can think of $\boldsymbol{x}$ as the secret, and the linear leakage is regarding the inner product of the coefficients as specified above. Additionally, the parameter l is the dimension of the secret key that can be set as m or n, or $m + n$, the parameter k is a dynamic parameter that is related to the latter applications (boosting soundness of ZKP protocol in Section 4), the leakage can also multiplicatively shift the secret to $\phi(c_i \boldsymbol{x})$ specified by the parameters c_i's.

Next we define the search and decision versions of MLWE, with linear leakage. We note that, the hard problems we focus on in this work are with the "Hermite Normal Form". Particularly, the leakage function is defined over both secret and noise. Besides, in our hard problems, the leakage hints ($\boldsymbol{z}$ and c) can be specified by the solver (adversary). The adversary in our definition is less restricted than prior "*Extended LWE*" assumptions for which the hints need to be designated by the challenger, and thus makes our hardness result stronger.

Definition 3.2 (MLWE with Linear Secret Leakage, HNF, Search) *Let $m, n, q, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, χ be error distribution over $\mathcal{R}$. We define the search problem $S\text{-}\mathsf{MLWE}\text{-}\mathsf{LS}_{m,n,k,q,\chi}$ by the experiment between the adversary $\mathcal{A}$ and the challenger $\mathcal{C}$ as:*

- *$\mathcal{A}$ specifies k pairs $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1,\cdots,k\}}$, where $\boldsymbol{z}_i \in \mathcal{R}_q^{m+n}, c_i \in \mathcal{R}_q$, and sends $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1,\cdots,k\}}$ to $\mathcal{C}$.*

- $\mathcal{C}$ *first samples* $\boldsymbol{x} \leftarrow \chi^{n+m}$, $\mathbf{A} \overset{\$}{\leftarrow} \mathcal{R}_q^{m \times n}$, *and computes* $\boldsymbol{b} = [\mathbf{A}|\mathbf{I}_m] \cdot \boldsymbol{x} \in \mathcal{R}_q^m$. *Then, for* $\boldsymbol{z} = (\boldsymbol{z}_i)_{i \in [k]}, \boldsymbol{c} = (c_1, \cdots, c_k)^\top$, $\mathcal{C}$ *computes* $y = L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{x})$. *Finally,* $\mathcal{C}$ *returns* $(\mathbf{A}, \boldsymbol{b}, y)$ *to* $\mathcal{A}$.
- $\mathcal{A}$ *finally attempts to find* $\boldsymbol{s}$.

The search problem S-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$ *is hard, if it holds: for any* $\boldsymbol{z} = (\boldsymbol{z}_1^\top, \ldots, \boldsymbol{z}_k^\top)^\top \in \mathcal{R}_q^{k(n+m)}, (c_1, \cdots, c_k)^\top \in \mathcal{R}_q^k$ *and every PPT adversary* $\mathcal{A}$ *that*

$$\mathsf{Pr}\big[\mathcal{A}(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), y) = \boldsymbol{s}\big] \leq \mathsf{negl}(\lambda).$$

Definition 3.3 (MLWE with Linear Secret Leakage, HNF, Decision) *Let* $m, n, q, k, d > 0$ *be integers,* $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, χ *be error distribution over* $\mathcal{R}$. *We define the decision problem* D-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$ *by the experiment between adversary* $\mathcal{A}$ *and challenger* $\mathcal{C}$ *as:*

- $\mathcal{A}$ *specifies* k *pairs* $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1, \cdots, k\}}$, *where* $\boldsymbol{z}_i \in \mathcal{R}_q^{m+n}, c_i \in \mathcal{R}_q$, *and sends* $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1, \cdots, k\}}$ *to* $\mathcal{C}$.
- $\mathcal{C}$ *first samples* $\boldsymbol{x} \leftarrow \chi^{n+m}$, $\mathbf{A} \overset{\$}{\leftarrow} \mathcal{R}_q^{m \times n}$, *and computes* $\boldsymbol{b} = [\mathbf{A}|\mathbf{I}_m] \cdot \boldsymbol{x} \in \mathcal{R}_q^m$, *and also samples* $\boldsymbol{u} \overset{\$}{\leftarrow} \mathcal{R}_q^m$. *Then, for* $\boldsymbol{z} = (\boldsymbol{z}_i)_{i \in [k]}, \boldsymbol{c} = (c_1, \cdots, c_k)^\top$, $\mathcal{C}$ *computes* $y = L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{x})$. *Finally,* $\mathcal{C}$ *samples a random bit* $b \in \{0, 1\}$, *and sends* $(\mathbf{A}, \boldsymbol{b}, y)$ *to* $\mathcal{A}$ *if* $b = 1$, *or sends* $(\mathbf{A}, \boldsymbol{u}, y)$ *to* $\mathcal{A}$ *if* $b = 0$.
- $\mathcal{A}$ *finally outputs a bit* b' *as the guess of* b.

The advantage of $\mathcal{A}$ *in the game is defined as* $\mathsf{Adv}_{\mathcal{A},m,n,q,k,d,\chi}^{D\text{-}\mathsf{MLWE}\text{-}\mathsf{LS}} = |\mathsf{Pr}\big[b' = b\big] - \frac{1}{2}|$.

The decision problem D-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$ *is hard, if it holds: for any* $\boldsymbol{z} \in \mathcal{R}_q^{k(n+m)}, (c_1, \cdots, c_k)^\top \in \mathcal{R}_q^k$ *and every PPT adversary* $\mathcal{A}$ *that*

$$\mathsf{Adv}_{\mathcal{A},m,n,q,k,d,\chi}^{D\text{-}\mathsf{MLWE}\text{-}\mathsf{LS}} \leq \mathsf{negl}(\lambda).$$

In addition, we present two intermediate hard problems, denoted as S-MLWELE and D-MLWELE, which will be used in the hardness reduction of D-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$.

Definition 3.4 (MLWE with Linear Error Leakage, Search) *Let* $m, n, q, k, d > 0$ *be integers,* $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, χ *be error distribution over* $\mathcal{R}$. *We define the search problem* S-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$ *by the experiment between adversary* $\mathcal{A}$ *and challenger* $\mathcal{C}$ *as:*

- $\mathcal{A}$ *specifies* k *pairs* $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1, \cdots, k\}}$, *where* $\boldsymbol{z}_i \in \mathcal{R}_q^m, c_i \in \mathcal{R}_q$, *and sends* $\{(\boldsymbol{z}_i, c_i)\}_{i \in \{1, \cdots, k\}}$ *to* $\mathcal{C}$.
- $\mathcal{C}$ *first samples* $\boldsymbol{s} \overset{\$}{\leftarrow} \mathcal{R}^n$, $\boldsymbol{e} \leftarrow \chi^m$, $\mathbf{A} \overset{\$}{\leftarrow} \mathcal{R}_q^{m \times n}$, *and computes* $\boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \in \mathcal{R}_q^m$. *Then, for* $\boldsymbol{z} = (\boldsymbol{z}_i)_{i \in [k]}, \boldsymbol{c} = (c_1, \cdots, c_k)^\top$, $\mathcal{C}$ *computes* $y = L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{e})$. *Finally,* $\mathcal{C}$ *returns* $(\mathbf{A}, \boldsymbol{b}, y)$ *to* $\mathcal{A}$.
- $\mathcal{A}$ *finally attempts to find* $\boldsymbol{s}$.

The search problem S-$\mathsf{MLWELE}_{m,n,k,q,\chi}$ *is hard, if it holds: for any* $\boldsymbol{z} = (\boldsymbol{z}_1^\top, \ldots, \boldsymbol{z}_k^\top)^\top \in \mathcal{R}_q^{km}, (c_1, \cdots, c_k)^\top \in \mathcal{R}_q^k$ *and every PPT adversary* $\mathcal{A}$ *that*

$$\mathsf{Pr}\big[\mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e}, \boldsymbol{z}, (c_1, \cdots, c_k), y) = \boldsymbol{s}\big] \leq \mathsf{negl}(\lambda).$$

Definition 3.5 (MLWE with Linear Error Leakage, Decision) *Let $m, n, q, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, χ be error distribution over $\mathcal{R}$. We define the decision problem D-$\mathsf{MLWELE}_{m,n,k,q,\chi}$ by the experiment between adversary $\mathcal{A}$ and challenger $\mathcal{C}$ as:*

- *$\mathcal{A}$ specifies k pairs $\{(\boldsymbol{z}_i, c_i)\}_{i\in\{1,\cdots,k\}}$, where $\boldsymbol{z}_i \in \mathcal{R}_q^m, c_i \in \mathcal{R}_q$, and sends $\{(\boldsymbol{z}_i, c_i)\}_{i\in\{1,\cdots,k\}}$ to $\mathcal{C}$.*
- *$\mathcal{C}$ first samples $\boldsymbol{s} \xleftarrow{\$} \mathcal{R}^n$, $\boldsymbol{e} \leftarrow \chi^m$, $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{m\times n}$, and computes $\boldsymbol{b} = \mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e} \in \mathcal{R}_q^m$, and also samples $\boldsymbol{u} \xleftarrow{\$} \mathcal{R}_q^m$. Then, for $\boldsymbol{z} = (\boldsymbol{z}_i)_{i\in[k]}, \boldsymbol{c} = (c_1, \cdots, c_k)^\top$, $\mathcal{C}$ computes $y = L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{x})$. Finally, $\mathcal{C}$ samples a random bit $b \in \{0, 1\}$, and sends $(\mathbf{A}, \boldsymbol{b}, y)$ to $\mathcal{A}$ if $b = 1$, or sends $(\mathbf{A}, \boldsymbol{u}, y)$ to $\mathcal{A}$ if $b = 0$.*
- *$\mathcal{A}$ finally outputs a bit b' as the guess of b.*

The advantage of $\mathcal{A}$ in the game is defined as $\mathsf{Adv}^{D\text{-}\mathsf{MLWELE}}_{\mathcal{A},m,n,q,k,d,\chi} = |\mathsf{Pr}\big[b' = b\big] - \frac{1}{2}|$.

The decision problem D-$\mathsf{MLWELE}_{m,n,k,q,\chi}$ is hard, if it holds: for any $\boldsymbol{z} \in \mathcal{R}_q^{km}, (c_1, \cdots, c_k)^\top \in \mathcal{R}_q^k$ and every PPT adversary $\mathcal{A}$ that

$$\mathsf{Adv}^{D\text{-}\mathsf{MLWELE}}_{\mathcal{A},m,n,q,k,d,\chi} \leq \mathsf{negl}(\lambda).$$

Remark 3.6 *Notice that in Definition 3.3, it allows that $\mathcal{A}$ specifies leakage query tuples, i.e., $(\boldsymbol{z}_i, c_i)$, before obtaining the MLWE samples. However, for our applications on efficient non-interactive ZKP in Section 4, the leakage hints $(\boldsymbol{z}_i, c_i)$ are sampled by the challenger. Clearly the security in the former case is much stronger, as the adversary in the case is less restricted. Thus, in this section, we just focus on the former one. For completeness, we also present the formal definition on the latter case denoted as "extended MLWE" in Defintion 3.7.*

Definition 3.7 (Extended MLWE, HNF, Decision) *Let $m, n, q, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, χ be error distribution over $\mathcal{R}$. We define the decision problem Extended D-$\mathsf{MLWE}_{m,n,k,q,\chi}$ by the experiment between adversary $\mathcal{A}$ and challenger $\mathcal{C}$ as:*

- *$\mathcal{C}$ first samples $\boldsymbol{x} \leftarrow \chi^{n+m}$, $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{m\times n}$, and computes $\boldsymbol{b} = [\mathbf{A}|\mathbf{I}_m]\cdot\boldsymbol{x} \in \mathcal{R}_q^m$, and also samples $\boldsymbol{u} \xleftarrow{\$} \mathcal{R}_q^m$. Then, $\mathcal{C}$ specifies k pairs $\{(\boldsymbol{z}_i, c_i)\}_{i\in\{1,\cdots,k\}}$, where $\boldsymbol{z}_i \in \mathcal{R}_q^{m+n}, c_i \in \mathcal{R}_q$ and computes $y = L_{\boldsymbol{z},\boldsymbol{c}}(\boldsymbol{x})$. Finally, $\mathcal{C}$ samples a random bit $b \in \{0, 1\}$, and sends $(\mathbf{A}, \boldsymbol{b}, \{(\boldsymbol{z}_i, c_i)\}_{i\in\{1,\cdots,k\}}, y)$ to $\mathcal{A}$ if $b = 1$, or sends $(\mathbf{A}, \boldsymbol{u}, \{(\boldsymbol{z}_i, c_i)\}_{i\in\{1,\cdots,k\}}, y)$ to $\mathcal{A}$ if $b = 0$.*
- *$\mathcal{A}$ finally outputs a bit b' as the guess of b.*

The advantage of $\mathcal{A}$ in the game is defined as $\mathsf{Adv}^{Ext\text{-}D\text{-}\mathsf{MLWE}}_{\mathcal{A},m,n,q,k,d,\chi} = |\mathsf{Pr}\big[b' = b\big] - \frac{1}{2}|$.

The decision problem Extended D-$\mathsf{MLWE}_{m,n,k,q,\chi}$ is hard, if it holds: for any $\boldsymbol{z} \in \mathcal{R}_q^{k(n+m)}, (c_1, \cdots, c_k)^\top \in \mathcal{R}_q^k$ and every PPT adversary $\mathcal{A}$ that

$$\mathsf{Adv}^{Ext\text{-}D\text{-}\mathsf{MLWE}}_{\mathcal{A},m,n,q,k,d,\chi} \leq \mathsf{negl}(\lambda).$$

Now we will give our concrete reductions. To start, we first show a reduction from S-$\mathsf{MLWE}_{m,n,q,\chi}$ to S-$\mathsf{MLWELE}_{m,n,k,q,\chi}$. Generally, a search problem with $\log q$ bits of leakage can only decrease security by a factor of q. Therefore, if $q = \mathsf{poly}(\lambda)$, then the leakage version can be reduced from the non-leakage version of the problem.

Theorem 3.8 *Let $m, n, k, d, q > 0$ be integers, and q is a polynomial of the security parameter λ, $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$, χ be error distribution over $\mathcal{R}$. There exists a* PPT *reduction from S-$\mathsf{MLWE}_{m,n,q,\chi}$ to S-$\mathsf{MLWELE}_{m,n,k,q,\chi}$, such that if ε is the advantage of S-$\mathsf{MLWELE}_{m,n,k,q,\chi}$ solver, then $\varepsilon' = \frac{1}{q}\varepsilon$ is the advantage of S-$\mathsf{MLWE}_{m,n,q,\chi}$ solver.*

Proof. The reduction works as follow. Given $(\mathbf{A}, \boldsymbol{b}) \in \mathcal{R}_q^{m\times n} \times \mathcal{R}_q^m$, where $\boldsymbol{b} = \mathbf{A}\boldsymbol{s} + \boldsymbol{e}, \boldsymbol{s} \overset{\$}{\leftarrow} \mathcal{R}^n, \boldsymbol{e} \leftarrow \chi^m$, it receives $\boldsymbol{z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_k), (c_1, \cdots, c_k)$ from the solver of S-$\mathsf{MLWELE}_{m,n,k,\chi}$, and samples $r \overset{\$}{\leftarrow} \mathbb{Z}_q$. Eventually, it sends $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), r)$ to the solver of S-$\mathsf{MLWELE}_{m,n,k,\chi}$, and outputs what the solver outputs.

We now analyze the reduction. It's clear to see the components $\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k)$ are valid components of S-$\mathsf{MLWE\text{-}LS}_{m,n,k,\chi}$ instance. If $r = \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_1\boldsymbol{e})\rangle$, then the instance generated by the reduction is a valid instance for S-$\mathsf{MLWELE}_{m,n,k,\chi}$ solver. Otherwise, it is invalid.

It remains to analyze the probabilities that $r = \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_1\boldsymbol{e})\rangle$. As $\langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_1\boldsymbol{e})\rangle \in \mathbb{Z}_q$, we have $r = \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_1\boldsymbol{e})\rangle$ with probability at least $\frac{1}{q}$. The theorem then follows from a routine calculation. □

The hardness result of D-MLWELE as an important intermediate reduction of our main result can be summarized as the following Theorem.

Theorem 3.9 *Let $m, n, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, q be the prime modulus such that $q\mathcal{R}$ splits as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where $\ell = d/c$ for a constant $c \in \mathbb{Z}$ and $q \geq \ell^2$, χ be an error distribution that is invariant under all the automorphisms of $K = \mathbb{Q}[X]\backslash(X^d + 1)$. There exists a reduction from S-$\mathsf{MLWELE}_{\bar{m}^*,n,k,q,\chi}$ to D-$\mathsf{MLWELE}_{m,n,k,q,\chi}$, such that if ε is the advantage of D-$\mathsf{MLWE\text{-}LS}_{m,n,k,q,\chi}$ solver, then $\varepsilon' \geq 1 - \frac{\varepsilon}{8}$ is the advantage of S-$\mathsf{MLWELE}_{\bar{m}^*,n,k,q,\chi}$ solver, and $\bar{m}^* = \ell q^c mn \cdot \lceil 1/\varepsilon^2 \rceil$.*

Proof. We first summarize the reduction route as follows, and then explain the concrete steps later:

$$S\text{-}\mathsf{MLWELE}_{\bar{m}^*,n,k,q,\chi} \xrightarrow{(1)} (W)\text{-}\mathfrak{q}_i\text{-}\mathsf{MLWELE}_{m^*,n,k,q,\chi} \xrightarrow{(2)} (W)\text{-}D\text{-}\mathsf{MLWELE}^i_{m,n,k,q,\chi}$$
$$\xrightarrow{(3)} (A)\text{-}D\text{-}\mathsf{MLWELE}^i_{m,n,k,q,\chi} \xrightarrow{(4)} D\text{-}\mathsf{MLWELE}_{m,n,k,q,\chi}.$$

To start, we define the first intermediate assumption (W)-$\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ as follows.

Definition 3.10 ((W)-$\mathfrak{q}_i$-MLWELE$_{m^*,n,k,q,\chi}$) *Let $m^*, n, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$, q be the modulus such that $q\mathcal{R}$ splits as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where*

$\ell = d/c$ for a constant $c \in \mathbb{Z}$, χ be error distribution over $\mathcal{R}$. For any $\mathfrak{q}_i, i \in [\ell]$, the worst-case search problem $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ is defined as: given access to $(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$ for some arbitrary $\boldsymbol{s} \in \mathcal{R}_q^n$, where $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{m^* \times n}, \boldsymbol{e} \leftarrow \chi^{m^*}$, $\boldsymbol{z} \in \mathcal{R}_q^{k \cdot m^*}$ and $(c_1, \cdots, c_k) \in \mathcal{R}_q^k$ as defined in Definition 3.2, find $\boldsymbol{s}$ mod $\mathfrak{q}_i$.

Then, we have the following reduction.

Lemma 3.11 (S-MLWELE$_{\bar{m}^*,n,k,q,\chi}$ to (W)-$\mathfrak{q}_i$-MLWELE$_{m^*,n,k,q,\chi}$) *Let m^*, n, k, $d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, q be the modulus such that $q\mathcal{R}$ splits completely as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where $\ell = d/c$ for a constant $c \in \mathbb{Z}$, χ be error distribution over $\mathcal{R}$ and invariant under all the automorphisms of $K = \mathbb{Q}[X]\backslash(X^d + 1)$. Then for every $i \in \{1, \cdots, \ell\}$, there exists a deterministic poly-time reduction from S-$\mathsf{MLWELE}_{\bar{m}^*,n,k,q,\chi}$ to (W)-$\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$, such that if $1 - \varepsilon$ is the advantage of (W)-$\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ solver, then $1 - \ell\varepsilon$ is the advantage of S-$\mathsf{MLWELE}_{\bar{m}^*,n,k,q,\chi}$ solver, where $\varepsilon < \frac{1}{\ell}$, and $\bar{m}^* = \ell m^*$.*

Proof. To prove this theorem, we will work on an arbitrary $i \in \{1, \cdots, \ell\}$. And the same argument can be extended to any other i's. So, throughout the rest of the proof, we will view i as an arbitrary fixed index.

For $\tau \in \{1, \cdots, \ell\}$, let σ_τ be an automorphism that maps $\mathfrak{q}_\tau$ to $\mathfrak{q}_i$. We know that all these automorphisms exist as $K = \mathbb{Q}[X]\backslash(X^d + 1)$ is a Galois extension. Then the reduction proceeds as follow.

- For each $\tau \in \{1, \cdots, \ell\}$, the reduction runs through the following steps.
  - For given sample $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$, transform it to $(\sigma_\tau(\mathbf{A}), \sigma_\tau(\boldsymbol{b}), \sigma_\tau(\boldsymbol{z}), \sigma_\tau(c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$.
  - Send the transformed sample to the $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,\chi}$ oracle.
  - Upon receiving the answer $\boldsymbol{\eta} \in (\mathcal{R}/\mathfrak{q}_i\mathcal{R})^n$, store $\sigma_\tau^{-1}(\boldsymbol{\eta}) \in (\mathcal{R}/\mathfrak{q}_\tau\mathcal{R})^n$.
- Next, the reduction combines all $\{\sigma_\tau^{-1}(\boldsymbol{\eta})\}_{\tau \in \{1, \cdots, \ell\}}$ by the Chinese Remainder Theorem. Then it outputs the combined value $\boldsymbol{s}' \in \mathcal{R}_q^n$.

We now show that for each $\tau \in [\ell]$, $\sigma_\tau^{-1}(\boldsymbol{\eta}) = \boldsymbol{s}$ mod $\mathfrak{q}_\tau R$. To show this, we prove that the distribution of the transformed samples is correctly distributed as the $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,\chi}$ oracle requires. Particularly, for $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k)$, $\langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$, $\sigma_\tau(\mathbf{A})$ is uniformly random in $\sigma_\tau(\mathcal{R}_q^{m^* \times n}) = \mathcal{R}_q^{m^* \times n}$ as σ_τ is an automorphism. Next we have $\sigma_\tau(\boldsymbol{b}) = \sigma_\tau(\mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}) = \sigma_\tau(\mathbf{A}) \cdot \sigma_\tau(\boldsymbol{s}) + \sigma_\tau(\boldsymbol{e})$. It remains to show $\langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle = \langle \phi(\sigma_\tau(\boldsymbol{z})), \phi(\sigma_\tau(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e}))\rangle$. If this holds, then $(\sigma_\tau(\mathbf{A}), \sigma_\tau(\boldsymbol{b}), \sigma_\tau(\boldsymbol{z}), \sigma_\tau(c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$ would be the correct distribution that the $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,\chi}$ oracle expects, and then the oracle would return $\boldsymbol{\eta} = \sigma_\tau(\boldsymbol{s})$ mod $\mathfrak{q}_i R$ (with a non-negligible probability). Thus, we have $\sigma_\tau^{-1}(\boldsymbol{\eta}) = \boldsymbol{s}$ mod $\mathfrak{q}_k R$. Now we focus on proving $\langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle = \langle \phi(\sigma_\tau(\boldsymbol{z})), \phi(\sigma_\tau(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e}))\rangle$.

We write $\langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle = \sum_{i=1}^m \sum_{j=1}^k \langle \phi(z_{i,j}), \phi(c_j e_i)\rangle$, $\langle \phi(\sigma_\tau(\boldsymbol{z}))$, $\phi(\sigma_\tau(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e}))\rangle = \sum_{i=1}^m \sum_{j=1}^k \langle \phi(\sigma_\tau(z_{i,j})), \phi(\sigma_\tau(c_j e_i))\rangle$. If we can show that $\langle \phi(z_{i,j}), \phi(c_j e_i)\rangle = \langle \phi(\sigma_\tau(z_{i,j})), \phi(\sigma_\tau(c_j e_i))\rangle$ for any i, j, then the argument we need to prove follows. Without loss of generality, we just consider the case

$i = j = 1$, and other cases are similar. We write $z_{1,1} = a_0 + a_1X + \cdots + a_{d-1}X^{d-1}$ and $c_1e_1 = b_0 + b_1X + \cdots + b_{d-1}X^{d-1}$. Then $\langle\phi(z_{1,1}), \phi(c_1e_1)\rangle = \langle(a_0, \cdots, a_{d-1}), (b_0, \cdots, b_{d-1})\rangle$. On the other hand, $\sigma_\tau(X) = X^{\tau'}$, where $\tau' \in \mathbb{Z}_{2d}^*$ is an index corresponding to τ. Thus, $\sigma_\tau(z_{1,1}) = a_0+a_1X^{\tau'}+\cdots+a_{d-1}X^{(d-1)\tau'} \bmod (X^d+1) = a'_0+a'_1X+\cdots+a'_{d-1}X^{d-1}$. Therefore, $(|a_0|, \cdots, |a_{d-1}|)$ is equivalent to $(|a'_0|, \cdots, |a'_{d-1}|)$ up to a permutation. Similarly, let $\sigma_\tau(c_1e_1) = b'_0+b'_1X+\cdots+b'_{d-1}X^{d-1}$, then $(|b_0|, \cdots, |b_{d-1}|)$ is equivalent to $(|b'_0|, \cdots, |b'_{d-1}|)$ up to the same permutation. Furthermore, for $i, j \in \{0, \cdots, d-1\}$ if $a_i = \mathsf{Sign}(i, j)a'_j$ where $\mathsf{Sign}(i, j) = 1$ or -1, then $b_i = \mathsf{Sign}(i, j)b'_j$. As a result, $\sum_{i=0}^{d-1} a_ib_i = \sum_{i=0}^{d-1} a'_ib'_i$, and thus $\langle\phi(z_{1,1}), \phi(c_1e_1)\rangle = \langle\phi(\sigma_\tau(z_{1,1})), \phi(\sigma_\tau(c_1e_1))\rangle$.

Finally, by the Chinese Reminder Theorem, $\boldsymbol{s} \bmod q\mathcal{R}$ can be reconstructed from $\{\boldsymbol{s} \bmod \mathfrak{q}_\tau\mathcal{R}\}_{\tau=1}^{\ell}$. If the advantage of $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,\chi}$ solver is $1 - \varepsilon$, then by a union bound, the probability that all the $\{\boldsymbol{s} \bmod \mathfrak{q}_\tau\mathcal{R}\}_{\tau=1}^{\ell}$ are correct answers is $1 - \ell\varepsilon$. This completes the proof. □

In order to describe the second intermediate assumption, the following definition is needed.

Definition 3.12 (Hybrid MLWELE distribution) *For $i \in \{1, \cdots, \ell\}$, a distribution χ over $\mathcal{R}_q$ and $\boldsymbol{s} \xleftarrow{\$} \mathcal{R}^n$, we define the distribution $A^i_{m^*,k,\boldsymbol{s},\chi}$ over $\mathcal{R}_q^{m^*\times n}\times\mathcal{R}_q^{m^*}\times\mathcal{R}_q^{km^*}\times\mathcal{R}_q^{k}\times\mathbb{Z}_q$ as: sample $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle\phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$ as Definition 3.10 and output $(\mathbf{A}, \boldsymbol{b} + \boldsymbol{h}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle\phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$ where $\boldsymbol{h} \in \mathcal{R}_q^{m^*}$ are uniformly random mod $\mathfrak{q}_j\mathcal{R}$ for all $j \leq i$, and 0 over mod all the other ideals, i.e., $\mathfrak{q}_j\mathcal{R}$'s for $j > i$.*

We note that $A^0_{m^*,k,\boldsymbol{s},\chi}$ is the original distribution as Definition 3.10, $A^\ell_{m^*,k,\boldsymbol{s},\chi}$ is the distribution as the random case defined in Definition 3.3, and the other $A^i_{m^*,k,\boldsymbol{s},\chi}$'s are intermediate hybrids, which will be used via a hybrid argument later.

Now, the second intermediate assumption is as follows.

Definition 3.13 ((W)-$\boldsymbol{D}$-$\mathsf{MLWELE}^{\boldsymbol{i}}_{\boldsymbol{m,n,k,q,\chi}}$) *The worst-case D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ problem is defined as follows: given access to an oracle sampling from $A^i_{m,k,\boldsymbol{s},\chi}$ for arbitrary $\boldsymbol{s} \in \mathsf{Supp}(\chi^n)$ and $j \in \{i-1, i\}$, find j.*

The following lemma states a reduction from (W)-$\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ to (W)-D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$.

Lemma 3.14 ((W)-$\mathfrak{q}_{\boldsymbol{i}}$-$\mathsf{MLWELE}_{\boldsymbol{m^*,n,k,q,\chi}}$ to (W)-$\boldsymbol{D}$-$\mathsf{MLWELE}^{\boldsymbol{i}}_{\boldsymbol{m,n,k,q,\chi}}$) *For any $i \in \{1, \cdots, \ell\}$, and ideal $\mathfrak{q}_i$ with $N(\mathfrak{q}_i) = q^{d/\ell} = q^c$ where $c \geq 1$ is a constant integer, there exists a probabilistic polynomial time reduction from $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ to (W)-D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$, such that if ε is the advantage of (W)-D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ solver, then $\varepsilon' \geq 1 - \frac{\varepsilon}{8}$ is the advantage of $\mathfrak{q}_i$-$\mathsf{MLWELE}_{m^*,n,k,q,\chi}$ solver, where $m^* = q^cmn \cdot \lceil\frac{1}{\varepsilon^2}\rceil$.*

Proof. At a high level, the reduction recovers $\boldsymbol{s} \bmod \mathfrak{q}_i\mathcal{R}$ by finding $s_1 \bmod \mathfrak{q}_i\mathcal{R}$, $\cdots, s_n \bmod \mathfrak{q}_i\mathcal{R}$, sequentially. For each $s_t \bmod \mathfrak{q}_i\mathcal{R}$ with $t \in [n]$, the reduction tries each of its possible values, and uses the (W)-D-$\mathsf{MLWELE}^{i}_{m,n,k,q,\chi}$ oracle to determine which trial is correct.

Without loss of generality, we consider the case of $s_1 \bmod \mathfrak{q}_i\mathcal{R}$, and other cases are similar. For each trial, the reduction transforms sample from $A^{i}_{m,k,\boldsymbol{s},\chi}$ so that the resulting sample is distributed according to $A^{i-1}_{m,k,\boldsymbol{s},\chi}$ if the trial equal to the value of $s_1 \bmod \mathfrak{q}_i\mathcal{R}^{\vee}$, or otherwise, $A^{i}_{m,k,\boldsymbol{s},\chi}$. Then the (W)-$D$-$\mathsf{MLWELE}^{i}_{m,n,k,q,\chi}$ oracle can be used to distinguish the two cases, and thus the reduction can determine whether this trial is correct. Since there are $N(\mathfrak{q}_i) = q^c = \mathsf{poly}(n)$ possible values, the reduction's running time is upper bounded by a polynomial. Moreover, the reduction needs to call (W)-D-$\mathsf{MLWELE}^{i}_{m,n,k,q,\chi}$ oracle $\lceil \frac{1}{\varepsilon^2} \rceil$ times in order to get a sufficient confidence (at least $1 - \frac{\varepsilon}{8}$, this can be argued by the classical Chernoff bound, and similar analysis can refer to [44]), and each call takes m samples, where ε is the advantage of (W)-D-$\mathsf{MLWELE}^{i}_{m,n,k,q,\chi}$ oracle. Thus, in total the reduction needs up to $m^* = q^c mn \cdot \lceil \frac{1}{\varepsilon^2} \rceil$ samples.

Below we just describe the transformation, and note that the other steps of the reduction are trivial (ref. [37]). Given a sample $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle)$ and a trial value $g \in R_q$, the reduction computes a sample

$$\begin{aligned}&(\mathbf{A}', \boldsymbol{b}', \boldsymbol{z}', (c'_1, \cdots, c'_k), \langle \phi(\boldsymbol{z}'), \phi(c'_1\boldsymbol{e}', \cdots, c'_k\boldsymbol{e}')\rangle)\\ =&(\mathbf{A} + \mathbf{Y}, \boldsymbol{b} + \boldsymbol{h} + g\boldsymbol{y}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle),\end{aligned}$$

where $\mathbf{Y} = \begin{bmatrix} y_1 & 0 \cdots 0 \\ \vdots & \vdots \ddots \vdots \\ y_m & 0 \cdots 0 \end{bmatrix} \in \mathcal{R}_q^{m \times n}$, $\boldsymbol{y} = (y_1, \cdots, y_m)^\top$ are sampled according to the distributions that are uniformly random mod $\mathfrak{q}_i\mathcal{R}$ and 0 mod all the other $\mathfrak{q}_j\mathcal{R}$'s. $\boldsymbol{h} \in \mathcal{R}_q^m$ is uniformly random mod $\mathfrak{q}_j\mathcal{R}$ for all $j < i$, and is 0 mod $\mathfrak{q}_j\mathcal{R}$'s for $j \geq i$.

It is clear that $\mathbf{A}'$ is uniformly random over $\mathcal{R}_q^{m \times n}$, because $\mathbf{A}$ is uniformly random over $\mathcal{R}_q^{m \times n}$. On the other hand, $\boldsymbol{b}'$ can be written as

$$\begin{aligned}\boldsymbol{b}' &= \boldsymbol{b} + \boldsymbol{h} + g\boldsymbol{y}\\ &= \mathbf{A}\boldsymbol{s} + \boldsymbol{e} + \boldsymbol{h} + g\boldsymbol{y}\\ &= \mathbf{A}'\boldsymbol{s} + \boldsymbol{e} + \boldsymbol{h} + (g - s_1)\boldsymbol{y}.\end{aligned}$$

If $s_1 \equiv g \bmod \mathfrak{q}_i\mathcal{R}$, then by the Chinese Remainder Theorem (Lemma 2.2), $(s_1 - g)\boldsymbol{y} = 0 \bmod q\mathcal{R}$. In this case, $(\mathbf{A}', \boldsymbol{b}', \boldsymbol{z}', (c'_1, \cdots, c'_k), \langle \phi(\boldsymbol{z}'), \phi(c'_1\boldsymbol{e}', \cdots, c'_k\boldsymbol{e}')\rangle)$ is distributed according to $A^{i-1}_{m,k,\boldsymbol{s},\chi}$. Otherwise if $s_1 \neq g \bmod \mathfrak{q}_i\mathcal{R}$, we claim that $(s_1 - g)\boldsymbol{y} \bmod \mathfrak{q}_i\mathcal{R}$ is uniformly random mod $\mathfrak{q}_i\mathcal{R}$ and is 0 mod all the other ideals $\mathfrak{q}_j\mathcal{R}$'s for $j \neq i$: as $\mathcal{R}/\mathfrak{q}_i$ is a field, $(s_1 - g)\boldsymbol{y} \bmod \mathfrak{q}_i\mathcal{R}$ is uniformly random for a random $\boldsymbol{y} \bmod \mathfrak{q}_i\mathcal{R}$, and any $(s_1 - g) \neq 0 \bmod \mathfrak{q}_i\mathcal{R}$. Therefore, $(g - s_1)\boldsymbol{y} + \boldsymbol{h}$ is uniformly random mod $\mathfrak{q}_j\mathcal{R}$ for all $j \leq i$, and is 0 mod all the remaining $\mathfrak{q}_j\mathcal{R}$'s. Thus, the distribution of $(\mathbf{A}', \boldsymbol{b}', \boldsymbol{z}', (c'_1, \cdots, c'_k), \langle \phi(\boldsymbol{z}'), \phi(c'_1\boldsymbol{e}', \cdots, c'_k\boldsymbol{e}')\rangle)$ follows $A^{i}_{m,k,\boldsymbol{s},\chi}$ in this case, completing the proof. □

The third intermediate assumption in the reduction route is as follows.

Definition 3.15 (Average-case Decision LWE relative to $\mathfrak{q}_i$) *For $i \in \{1, \cdots, \ell\}$ and a distribution χ over error $\mathcal{R}_q$, we say that an algorithm solves the D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ problem if with a non-negligible probability over the choice of a random $\boldsymbol{s} \leftarrow U(\mathcal{R}_q^n)$, it has a non-negligible difference in acceptance probability on inputs from $A^{i-1}_{m,k,\boldsymbol{s},\chi}$ versus inputs from $A^i_{m,k,\boldsymbol{s},\chi}$.*

We have the worst-case to average-case reduction as follows.

Lemma 3.16 (Worst-case to Average-case) *There exists a randomized poly-time reduction from worst-case (W)-D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ to average-case D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$, such that if ε is the advantage of D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ distinguisher, then ε is the advantage of (W)-D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ distinguisher.*

Proof. Given sample $(\mathbf{A}, \boldsymbol{b}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle) \leftarrow A^i_{m,k,\boldsymbol{s},\chi}$ for arbitrary $\boldsymbol{s} \in \mathcal{R}_q^n$, the reduction transforms it into

$$(\mathbf{A}, \boldsymbol{b} + \mathbf{A}\boldsymbol{s}' + \boldsymbol{h}, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1\boldsymbol{e}, \cdots, c_k\boldsymbol{e})\rangle),$$

where $\boldsymbol{s}' \xleftarrow{\$} \mathcal{R}_q^n$, and $\boldsymbol{h} \in \mathcal{R}_q^m$ is uniformly random mod $\mathfrak{q}_j\mathcal{R}$ for all $j \leq \nu$ (where $\nu \leq i$), and 0 over mod all the other ideals. It is easy to see that for all $\boldsymbol{s} \in \mathcal{R}_q^n$ and $i \in \{1, \cdots, \ell\}$, this transformation maps $A^i_{m,k,\boldsymbol{s},\chi}$ to $A^{\max\{\nu,i\}}_{m,k,U(\mathcal{R}_q^n),\chi}$.

Formally, the reduction is executed by repeating the following steps a polynomial number of times: Choose a $\boldsymbol{s}'$ from $\mathcal{R}_q^n$ uniformly at random, and then estimate the acceptance probability of the oracle on the following two input distributions: the first is obtained from our input by applying the above transformation with parameters $\boldsymbol{s}'$, and $i-1$; the second is obtained similarly using parameters $\boldsymbol{s}'$, and i. If in any of these polynomial number of attempts a non-negligible difference is observed between the two acceptance probabilities, output "$i-1$"; otherwise output "i".

If the input distribution is $A^i_{m,k,\boldsymbol{s},\chi}$, then in each of the attempts, the two distributions on which we estimate the oracle's acceptance probability are exactly the same, and we output "i" with overwhelming probability. If the input distribution is $A^{i-1}_{m,k,\boldsymbol{s},\chi}$, we estimate the oracle's acceptance probability on $A^{i-1}_{m,k,\boldsymbol{s}+U(\mathcal{R}_q^n),\chi}$ and $A^i_{m,k,\boldsymbol{s}+U(\mathcal{R}_q^n),\chi}$.

Let $B^{i-1}(\boldsymbol{s}')$ and $B^i(\boldsymbol{s}')$ be the two distributions on the vector which our reduction uses as input to the oracle. The average of $B^{i-1}(\boldsymbol{s}')$ over $\boldsymbol{s}'$ chosen independently from $U(\mathcal{R}_q^n)$, is $A^{i-1}_{m,k,\boldsymbol{s}+U(\mathcal{R}_q^n),\chi}$ and similarly with B^i and A^i.

Let S be the set of all vectors $\boldsymbol{s}$ for which the oracle has a non-negligible difference in acceptance probability on $B^{i-1}(\boldsymbol{s}')$ and $B^i(\boldsymbol{s}')$. By assumption, the measure of S under $U(\mathcal{R}_q^n)$ is non-negligible, then the measure of S under $\boldsymbol{s} + \boldsymbol{s}'$ is also non-negligible. This finishes the proof. □

The following lemma states the step (4) of the reduction route.

Lemma 3.17 (D-$\mathsf{MLWELE}^i_{m,n,k,q,\chi}$ to D-$\mathsf{MLWELE}_{m,n,k,q,\chi}$) *For any oracle solving the D-$\mathsf{MLWELE}_{m,n,k,q,\chi}$ problem with advantage ε, there exists an*

$i \in \{1, \cdots, \ell\}$ and an efficient algorithm that solves $D\text{-}\mathsf{MLWELE}^{i}_{m,n,k,q,\chi}$ with advantage ε/ℓ using this oracle.

Proof (Sketch). This lemma can be proved by a simple hybrid argument. As the hybrid argument is standard, we just sketch the main idea: suppose there exists an algorithm that solves $D\text{-}\mathsf{MLWELE}_{m,n,k,q,\chi}$ with advantage ε, i.e., it distinguishes the two distributions defined as Definition 3.3. Then the algorithm must be able to distinguish some neighboring hybrids, i.e., $A^{i}_{m,k,\boldsymbol{s},\chi}$ and $A^{i-1}_{m,k,\boldsymbol{s},\chi}$, with advantage ε/ℓ, as there are ℓ intermediate hybrids. □

The proof of Theorem 3.9 follows from Lemmas 3.11, 3.14, 3.16 and 3.17. □

Finally, we show a reduction from $D\text{-}\mathsf{MLWELE}_{m,n,k,q,\chi}$ to $D\text{-}\mathsf{MLWE\text{-}LS}_{m,n,k,q,\chi}$ as follows.

Lemma 3.18 (D-MLWELE$_{m(nd)^2,n,k,q,\chi}$ to D-MLWE-LS$_{m,n,k,q,\chi}$) *There exists a probabilistic poly-time reduction from $D\text{-}\mathsf{MLWELE}_{m(nd)^2,n,k,q,\chi}$ to $D\text{-}\mathsf{MLWE\text{-}LS}_{m,n,k,q,\chi}$, such that if ε is the advantage of $D\text{-}\mathsf{MLWE\text{-}LS}_{m,n,k,q,\chi}$ distinguisher, then ε is the advantage of $D\text{-}\mathsf{MLWELE}_{m(nd)^2,n,k,q,\chi}$ distinguisher.*

Proof. Given a sample $(\boldsymbol{a}, b, \boldsymbol{z}, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}), \phi(c_1 e, \cdots, c_k e)\rangle)$ from the instance of $D\text{-}\mathsf{MLWELE}_{1,n,k,q,\chi}$, where $\boldsymbol{a} \leftarrow \mathcal{R}_q^n$, $b = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + e$ or b is a random vector over $\mathcal{R}_q$. Our target is to transform it into a valid instance of $D\text{-}\mathsf{MLWE\text{-}LS}_{1,n,k,q,\chi}$.

In a first stage, we additionally take several samples $\{(\boldsymbol{a}_i, b_i, \boldsymbol{z}_i, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}_i), \phi(c_1 e_i, \cdots, c_k e_i)\rangle)\}$[8] from the instance of $D\text{-}\mathsf{MLWELE}_{1,n,k,q,\chi}$ and construct a set of n tuples $\{(\boldsymbol{a}_i, b_i, \boldsymbol{z}_i, (c_1, \cdots, c_k), \langle \phi(\boldsymbol{z}_i), \phi(c_1 e_i, \cdots, c_k e_i)\rangle)\}$ such that the $\boldsymbol{a}_i$'s are linearly independent over $\mathcal{R}_q$ and generate $\mathcal{R}_q^n$. It's easy to check, after drawing $(nd)^2$ different samples, we can obtain such a set of tuples with overwhelming probability. We define $\bar{\mathbf{A}} = (\boldsymbol{a}_1^\top, \cdots, \boldsymbol{a}_n^\top), \bar{\boldsymbol{b}} = (b_1, \cdots, b_n)$. Then, if $b_i = \langle \boldsymbol{a}_i, \boldsymbol{s}\rangle + e_i$ for all $i \in \{1, \cdots, n\}$, we have $\bar{\boldsymbol{b}} = \bar{\mathbf{A}}^\top \cdot \boldsymbol{s} + \boldsymbol{e}$, where $\boldsymbol{e}$ is sampled from χ^n.

In a second stage, we compute

$$(\boldsymbol{a}' = -(\bar{\mathbf{A}})^{-1} \cdot \boldsymbol{a}, b' = b + \langle \boldsymbol{a}', \bar{\boldsymbol{b}}\rangle, \boldsymbol{z}' = (\boldsymbol{z}_1^*, \cdots, \boldsymbol{z}_k^*), (c_1, \cdots, c_k), \delta + \sum_{i}^{n} \delta_i), \qquad (1)$$

where $\delta = \langle \phi(\boldsymbol{z}), \phi(c_1 e, \cdots, c_k e)\rangle$, $\delta_i = \langle \phi(\boldsymbol{z}_i), \phi(c_1 e_i, \cdots, c_k e_i)\rangle$ for every $i \in \{1, \cdots, k\}$, $\boldsymbol{z}_i^* \in \mathcal{R}_q^{n+1}$ is defined as follows: for each $i \in [n]$, let $\boldsymbol{z}_i = (z_{i,1}, \cdots, z_{i,k})$, then $z^*_{i,j} = z_{j,i}$ for every $j \in \{1, \cdots, n\}$ and $z^*_{i,n+1} = z_i$. As $\boldsymbol{a}$ is uniformly at random, we have $\boldsymbol{a}'$ is also uniformly at random. Now, for $\bar{\boldsymbol{b}}$, we consider two cases:

- If $\bar{\boldsymbol{b}} \in \mathcal{R}_q^n$ is uniformly at random, then $(\boldsymbol{a}', b')$ is also uniformly at random.

[8] Here we choose the instances with different $\boldsymbol{z}_i$ and the same hint tuple $(c_1, \cdots, c_k)$, this sampling procedure is reasonable, as the hint vectors can be specified by the adversary.

– If $\bar{\boldsymbol{b}} = \bar{\mathbf{A}} \cdot \boldsymbol{s} + \boldsymbol{e}$, then

$$b' = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e - \langle (\bar{\mathbf{A}})^{-\top} \cdot \boldsymbol{a}, \bar{\mathbf{A}} \cdot \boldsymbol{s} \rangle + \langle \boldsymbol{a}', \boldsymbol{e} \rangle = \langle \boldsymbol{a}', \boldsymbol{e} \rangle + e.$$

It remains to show that the hints $\boldsymbol{z}', (c_1, \cdots, c_k)$ and the leakage $\delta + \sum_i^n \delta_i$ are consistent with the secret and error pair $(\boldsymbol{e}, e)$. In other words, we need to show that

$$\langle \phi(\boldsymbol{z}'), \phi(c_1(\boldsymbol{e}, e), \cdots, c_k(\boldsymbol{e}, e)) \rangle = \delta + \sum_{i=1}^{n} \delta_i.$$

It’s clear that $\delta + \sum_i^n \delta_i = \langle (\phi(\boldsymbol{z}_1), \cdots, \phi(\boldsymbol{z}_n), \phi(\boldsymbol{z})), \phi((c_1e_1, \cdots, c_ke_1), \cdots, (c_1e_n, \cdots, c_ke_n), (c_1e, \cdots, c_ke)) \rangle$. Moreover, for each i, we have $\delta_i = \sum_{j=1}^{k} \langle \phi(z_{i,j}), \phi(c_je_i) \rangle = \sum_{j=1}^{k} \langle \phi(z_{j,i}^*), \phi(c_je_i) \rangle$, and $\delta = \sum_{j=1}^{k} \langle \phi(z_j), \phi(c_je) \rangle = \sum_{j=1}^{k} \langle \phi(z_{j,n+1}^*), \phi(c_je) \rangle$. Therefore,

$$\begin{aligned} \delta + \sum_{i}^{n} \delta_i &= \sum_{i=1}^{n} \sum_{j=1}^{k} \langle \phi(z_{j,i}^*), \phi(c_je_i) \rangle + \sum_{j=1}^{k} \langle \phi(z_{j,n+1}^*), \phi(c_je) \rangle \\ &= \sum_{i=1}^{k} \sum_{j=1}^{n} \langle \phi(z_{i,j}^*), \phi(c_ie_i) \rangle + \sum_{i=1}^{k} \langle \phi(z_{i,n+1}^*), \phi(c_ie) \rangle \\ &= \sum_{i=1}^{k} (\sum_{j=1}^{n} \langle \phi(z_{i,j}^*), \phi(c_ie_i) \rangle + \langle \phi(z_{i,n+1}^*), \phi(c_ie) \rangle) \\ &= \sum_{i=1}^{k} \langle \phi(\boldsymbol{z}_i^*), c_i(\boldsymbol{e}, e) \rangle = \langle \phi(\boldsymbol{z}'), \phi(c_1(\boldsymbol{e}, e), \cdots, c_k(\boldsymbol{e}, e)) \rangle. \end{aligned}$$

According the argument above, we have that the tuple defined by (1) is valid instance for D-MLWE-$\mathsf{LS}_{1,n,k,q,\chi}$ distinguisher, as desired. □

Combine Theorem 3.9 ,Theorem 3.8 and Lemma 3.18, the hardness of D-MLWE-LS can be reduced to the hardness of the fundamental problem S-MLWE by the following Corollary.

Corollary 3.19 *Let $m, n, k, d > 0$ be integers, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, q be the modulus such that $q\mathcal{R}$ splits as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where $\ell = d/c$ for a constant $c \in \mathbb{Z}$, χ be an error distributions over $\mathcal{R}$ that is invariant under all the automorphisms of $K = \mathbb{Q}[X]/(X^d + 1)$. There exists a reduction from S-$\mathsf{MLWE}_{m^*,n,q,\chi}$ to D-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$, such that if ε is the advantage of D-MLWE-$\mathsf{LS}_{m,n,k,q,\chi}$ solver, then $\varepsilon' \geq \frac{1}{q} \cdot (1 - \frac{\varepsilon}{8})$ is the advantage of S-$\mathsf{MLWE}_{m^*,n,q,\chi}$ solver, and $m^* = \ell q^c m n^3 d^2 \cdot \lceil 1/\varepsilon^2 \rceil$.*

Remark 3.20 *In our reduction, we consider the ring $\mathbb{Z}[X]/(X^d + 1)$ which is frequently used in many applications. It should be noted that we can generalize the ring to the more general cyclotomic setting by representing a ring element as integer linear combinations of a certain $\mathbb{Z}$-basis of the ring. Then, the map ϕ and the automorphism are defined according to the $\mathbb{Z}$-basis.*

4 Application: More Efficient Opening Proof for One-Time BDLOP Commitment

In this section, we present an important application of MLWE with linear leakage, leading to more efficient opening proofs for one-time BDLOP commitments under the paradigm [36]. Our particular contribution is to derive a more fine-grained tradeoff between efficiency and leakage of the paradigm [36], which can potentially lead to even more efficient proofs.

The section is organized as follow. We first recall the classical opening proof for BDLOP commitment in [9], together with two rejection sampling algorithms [32, 36] in Section 4.1. Then in Section 4.2, we further generalize the *subset rejection sampling* algorithm proposed by [36] in two ways: (1) we use a smaller subset S_v for the accepting condition; (2) we extends the constant value M to a real-valued function $\mathcal{M}$ of (v, z), whose output can vary based on the input. These two ideas can improve efficiency of the opening proof for the setting of one-time BDLOP commitment. Finally, in Section 4.4, we compare in detail the efficiency differences of the opening protocol under four different rejection sampling algorithms in Figs. 3 and 4. Below we first present the parameters used in this section in Table 3.

Parameters	Description
$\mathcal{R}$	Cyclotomic Ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ used in this section
d	ring dimension of $\mathcal{R}$
S_β	Set of all elements in $\mathcal{R}$ with ℓ_∞ norm at most β
q	modulus of BDLOP commitment
n, l, η	dimension parameters of BDLOP commitment
$\mathcal{C}$ κ	Challenge set of the opening ZKP system for BDLOP commitment $\mathcal{C} = \{c \in \mathcal{R} : \|c\|_1 = \kappa, \|c\|_\infty = 1\}$
m	dimension parameters of rejection sampling
$\hat{k}$	the parameter with respect to boosting soundness
$\mathcal{M}$	function from $(V, \mathbb{Z}^m)$ to $\mathbb{R}$
α	derivation of discrete Gaussian distribution for rejection sampling
$\hat{S}_v$	The subset of $\mathbb{Z}^m$ used for subset rejection sampling
$M, \mathfrak{c}$	constant parameters for subset rejection sampling
rep.	prover's expected repetition times for one non-abort
ℓ	the number of irreducible ideal modulo q, i.e., $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$
$\mathfrak{l}$	the bit-length of randomness leakage during the opening proof

Table 3. Notation of parameters in this section

4.1 Classical Opening Proof of BDLOP Commitment and Rejection Sampling Algorithms

Let us first recall the standard opening proof for BDLOP commitment scheme in [9]. Particularly, for a BDLOP commitment scheme with public parameters $\mathbf{A}_1 \in \mathcal{R}_q^{n\times\eta}, \mathbf{A}_2 \in \mathcal{R}_q^{l\times\eta}$, a message vector $\boldsymbol{m} \in \mathcal{R}_q^l$ is committed as $\mathsf{comm} :=$

$\begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{r} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{m} \end{bmatrix}$, where $\boldsymbol{r} \xleftarrow{\$} S_\beta^\eta$. Without loss of generality, we assume that $q\mathcal{R}$ splits as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where $\ell = d/c$ for a constant $c \in \mathbb{Z}$, and $q - 1 = 2\ell(\mathrm{mod} 4\ell)$. Clearly, if $\ell = d$, we say the ring $\mathcal{R}$ is full-splitting.

According to [7, 9], in order to prove knowledge of an opening to comm, one just needs to give an approximate proof for the first equation $\boldsymbol{t}_1 = \mathbf{A}_1 \cdot \boldsymbol{r}$ in the form of a three-round Schnorr-type Σ-protocol. Particularly in the first step, the prover first chooses a random vector $\boldsymbol{y}$, and then sends $\boldsymbol{w} = \mathbf{A}_1 \boldsymbol{y}$ to the verifier. Then, the verifier sends a short polynomial $c \in \mathcal{C} \subset R$ as a challenge. Finally, the prover replies with the vector $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{r}$. To achieve zero-knowledge, intuitively the masking vector $\boldsymbol{y}$ is used to hide the private randomness $\boldsymbol{r}$ of the commitment comm. Trivially one can set $\boldsymbol{y}$ to be super-polynomially larger than $c\boldsymbol{r}$ as some smudging noise, yet this would incur a large overhead in the proof size. To improve efficiency, [32] introduced the technique of *rejection sampling* that outputs $\perp$ instead of $\boldsymbol{z}$ with an appropriate probability, effectively wiping out the dependency of $c\boldsymbol{r}$ in $\boldsymbol{z}$.

Furthermore, in some settings such as proving the infinity norm of a vector as in [7,23,36], we need to set the underlying ring $\mathcal{R}$ to be full-splitting. In this case, the above mentioned initial Σ-protocol can only provide $1/q$ soundness, which is far away from negligible. In order to boost soundness, the work [7] applys Galois automorphisms. At a high level, given $\boldsymbol{r}, \boldsymbol{t}_1, \boldsymbol{t}_2$ as before, the prover $\mathcal{P}$ first generates $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{\hat{k}} \leftarrow \mathcal{D}_\alpha^\eta$. Then it outputs $(\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{\hat{k}})$, where $\boldsymbol{w}_i = \mathbf{A}_1 \cdot \boldsymbol{y}_i$. After receiving a challenge $c \leftarrow \mathcal{C}$ from the verifier, $\mathcal{P}$ computes

$$\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^{i-1}(c) \cdot \boldsymbol{r} \text{ for } i = 1, \cdots, \hat{k}$$

where $\sigma := \sigma_{2d/\hat{k}+1} \in \mathrm{Aut}(\mathcal{R}_q)$ is the automorphism of order $\hat{k}d/\ell$ and $\hat{k}$ is a divisor of d. After this, the prover applies rejection sampling $\mathsf{Rej}(\boldsymbol{z}, \boldsymbol{v}, \sigma)$ where $\boldsymbol{z} = \boldsymbol{z}_1 \| \cdots \| \boldsymbol{z}_{\hat{k}}$ and $\boldsymbol{v} = \sigma^0(c) \cdot \boldsymbol{r} \| \cdots \| \sigma^{\hat{k}-1}(c) \cdot \boldsymbol{r}$. If it does not abort, then $\mathcal{P}$ outputs $\boldsymbol{z}$. Finally, the verifier checks that $\boldsymbol{z}$ is small and

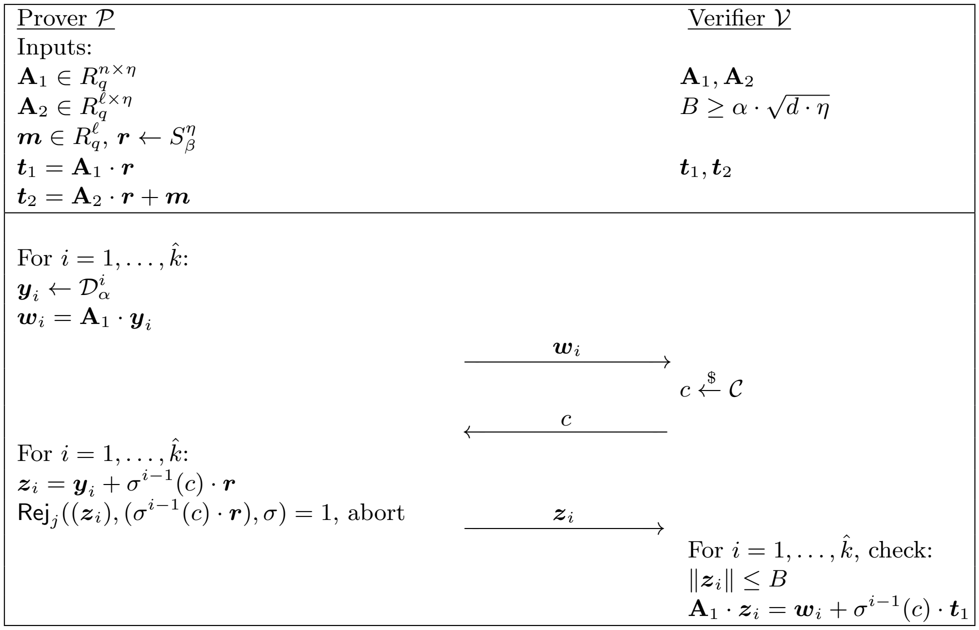
$$\mathbf{A}_1 \boldsymbol{z}_i = \boldsymbol{w}_i + \sigma^{i-1}(c) \cdot \boldsymbol{t}_1$$

for $i = 1, \cdots, \hat{k}$. As argued by [7], this protocol has soundness around $q^{-\hat{k}}$.

More formally, the protocol is described in Fig. 2, and the used rejection sampling algorithm is described as Rej_0 in Fig. 3.

Particularly, if we sample $\boldsymbol{y}_i$ from the discrete Gaussian distribution with derivation α, i.e., $\boldsymbol{y}_i \leftarrow D_\alpha^\eta$, then the vector $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^{i-1}(c)\boldsymbol{r}$ follows the shifted discrete Gaussian distribution $D_{\boldsymbol{v},\alpha}^\eta$ centered at $\boldsymbol{v} = \sigma^0(c) \cdot \boldsymbol{r} \| \cdots \| \sigma^{\hat{k}-1}(c) \cdot \boldsymbol{r}$. According to [32], we can "transform" the distribution $D_{\boldsymbol{v},\alpha}^\eta$ into the distribution D_α^η, by outputting $\boldsymbol{z} = \boldsymbol{z}_1 \| \cdots \| \boldsymbol{z}_{\hat{k}}$ with probability $\frac{D_\alpha^\eta}{M \cdot D_{\boldsymbol{v},\alpha}^\eta}$ (or otherwise $\perp$), where M is some positive integer so that this ratio is always smaller than 1. To further determine the concrete value for M, we need to compute an upper bound of $\frac{D_\alpha^\eta}{D_{\boldsymbol{v},\alpha}^\eta}$ as

$$\frac{D_\alpha^m}{D_{\boldsymbol{v},\alpha}^m} = \exp\left(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) \leq \exp\left(\frac{24\alpha\|\boldsymbol{v}\| + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) = M, \tag{2}$$



Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$\mathbf{A}_1 \in R_q^{n\times\eta}$		$\mathbf{A}_1, \mathbf{A}_2$
$\mathbf{A}_2 \in R_q^{\ell\times\eta}$		$B \geq \alpha \cdot \sqrt{d \cdot \eta}$
$\boldsymbol{m} \in R_q^{\ell}$, $\boldsymbol{r} \leftarrow S_\beta^\eta$		
$\boldsymbol{t}_1 = \mathbf{A}_1 \cdot \boldsymbol{r}$		$\boldsymbol{t}_1, \boldsymbol{t}_2$
$\boldsymbol{t}_2 = \mathbf{A}_2 \cdot \boldsymbol{r} + \boldsymbol{m}$		
For $i = 1, \ldots, \hat{k}$: $\boldsymbol{y}_i \leftarrow \mathcal{D}_\alpha^i$ $\boldsymbol{w}_i = \mathbf{A}_1 \cdot \boldsymbol{y}_i$		
	$\xrightarrow{\boldsymbol{w}_i}$	
		$c \overset{\$}{\leftarrow} \mathcal{C}$
	$\xleftarrow{c}$	
For $i = 1, \ldots, \hat{k}$: $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^{i-1}(c) \cdot \boldsymbol{r}$ $\mathsf{Rej}_j((\boldsymbol{z}_i), (\sigma^{i-1}(c) \cdot \boldsymbol{r}), \sigma) = 1$, abort	$\xrightarrow{\boldsymbol{z}_i}$	
		For $i = 1, \ldots, \hat{k}$, check: $\|\boldsymbol{z}_i\| \leq B$ $\mathbf{A}_1 \cdot \boldsymbol{z}_i = \boldsymbol{w}_i + \sigma^{i-1}(c) \cdot \boldsymbol{t}_1$

Fig. 2. Opening proof of BDLOP commitment through using our generalized rejection sampling, where $j = 0, \ldots, 3$.

where the above inequality is obtained through using a standard one-dimensional tail bound for the inner product of a discrete Gaussian with arbitrary vector. Clearly, if we want to set $M = \exp(1)$, then we need to set $\alpha = 12\|\boldsymbol{v}\|$. In this case, the size of $\boldsymbol{z}$ is about $\hat{k}\eta d \log(12\alpha) = \hat{k}\eta d \log(144\|\boldsymbol{v}\|)$, which depends on the value of α. This is essentially the intuition of [32].

In a recent work, Lyubashevsky et al. [36] observed that a much tighter upper bound for the ratio $D_\alpha^\eta / D_{\boldsymbol{v},\alpha}^\eta$ would imply a much smaller α, further lowering the size of $\boldsymbol{z}$. Particularly, if we assume that $\underline{\langle \boldsymbol{z}, \boldsymbol{v} \rangle \geq 0}$, then we have

$$\frac{D_\alpha^\eta}{D_{\boldsymbol{v},\alpha}^\eta} = \exp\left(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) \leq \exp\left(\frac{\|\boldsymbol{v}\|^2}{2\alpha^2}\right) = M. \tag{3}$$

In this case, if we want to set $M = \exp(1)$ in the following rejection sampling procedure again, we can set $\alpha = \|\boldsymbol{v}\|/\sqrt{2}$, which results in a decrease of around a factor of 17. This will clearly reduce the size of $\boldsymbol{z}$ to $\hat{k}\eta d \log(12\alpha) = \hat{k}\eta d \log(8.487\|\boldsymbol{v}\|)$. More formally, Lyubashevsky et al. [36] call such more efficient rejection sampling as *subset rejection sampling*, which is described as Rej_1 in Fig. 3. Clearly, Rej_1 can improve the size of the proof protocol in Fig. 2.

Additional costs of [36]. It is not for free however for the improvement [36]. All the above analyses have a precondition – $\langle \boldsymbol{z}, \boldsymbol{v} \rangle \geq 0$. For randomly chosen $\boldsymbol{y}, \boldsymbol{r}$, this precondition happens with a probability $\approx 1/2$. This means that if we

```
$\mathsf{Rej}_0(\boldsymbol{z}, \boldsymbol{v}, \alpha)$
$u \xleftarrow{\$} [0,1)$
If $u > \frac{1}{M} \cdot \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2}{2\alpha^2})$
    return 1
Else
    return 0
```

```
$\mathsf{Rej}_1(\boldsymbol{z}, \boldsymbol{v}, \alpha)$
If $\langle \boldsymbol{z}, \boldsymbol{v}\rangle < 0$
    return 1
$u \xleftarrow{\$} [0,1)$
If $u > \frac{1}{M} \cdot \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2}{2\alpha^2})$
    return 1
Else
    return 0
```

Fig. 3. Rejection Sampling. Here, implicitly, the outcome 1 implies abort, and 0 implies non-abort.

want to leverage the above subset rejection sampling, the prover will first abort the protocol with a probability $\approx 1/2$ to ensure $\langle \boldsymbol{z}, \boldsymbol{v}\rangle \geq 0$, and then conduct the regular rejection sampling. So, for the same constant value M, even the output size of $\boldsymbol{z}$ is reduced, the running time of the prover inherently becomes almost 2 times longer than that of [9].

Of course, one can easily balance the prover's running time and the size of his output $\boldsymbol{z}$. Particularly, we can set the upper bound of probability ratio to be $M/2$, which will derive that the finally expected abort time is about M. But, this will result in a slightly larger α', i.e., $\alpha' = \alpha\sqrt{\frac{\ln M}{\ln M/2}}$.

Besides and more importantly, there is a security concern. After the prover outputting $\boldsymbol{z}$ successfully, it imposes the precondition $\langle \boldsymbol{z}, \boldsymbol{v}\rangle \geq 0$, which leaks almost one bit information of $\boldsymbol{r}$ to the adversary. In this case, we need to consider whether this would affect the security of the opening proof of the BDLOP commitment, and even the whole privacy-preserving protocols.

To analyze this, Lyubashevsky et al. [36] identified a new variant of extended MLWE, and prove security of the protocol based on the variant of extended MLWE. As noticed in the introduction, this extended MLWE can be captured by MLWE with linear leakage analyzed in Section 3 of this work, using a formal reduction argument. This strengthens the foundation of the paradigm, as the leakage variant is no easier than the standard MLWE asymptotically. Thus, we would be more confident in the practical parameters of [36] obtained by cryptanalysis arguments.

4.2 More Efficient One-Time Opening Proof through Using Generalized Subset Rejection Sampling Algorithms

Now we define our new *generalized subset rejection sampling algorithms* Rej_2 and Rej_3 as in Fig. 4. Then we show that the algorithms themselves can be simulated, and the opening protocol with Rej_2 or Rej_3 satisfies correctness, knowledge soundness and simulatability. This means we can replace Rej_0 or Rej_1 for the protocol in Fig. 2 in a black-box way, by using our generalized algorithms.

```
$\mathsf{Rej}_2(\boldsymbol{z}, \boldsymbol{v}, \alpha)$
If $\langle \boldsymbol{z}, \boldsymbol{v} \rangle < \mathfrak{c} \cdot \alpha \|\boldsymbol{v}\|$
    return 1
$u \xleftarrow{\$} [0, 1)$
If $u > \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\alpha^2})$
    return 1
Else
    return 0
```

```
$\mathsf{Rej}_3(\boldsymbol{z}, \boldsymbol{v}, \alpha)$
If $\langle \boldsymbol{z}, \boldsymbol{v} \rangle \notin [0, (\alpha^2 * \ln M)/3]$
    return 1
$u \xleftarrow{\$} [0, 1)$
If $u > \frac{1}{\exp(\frac{3\langle \boldsymbol{v}, \boldsymbol{z} \rangle}{\alpha^2})} \cdot \exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\alpha^2})$
    return 1
Else
    return 0
```

Fig. 4. Generalized Rejection Sampling.

Simulation of Generalized Subset Rejection Sampling

To argue that the algorithms Rej_2 and Rej_3 themselves can be simulated successfully, we first define a more general version of subset rejection sampling algorithm $\mathcal{A}$, i.e., Rej_2 and Rej_3 can be viewed as two special cases of $\mathcal{A}$. Then we show that $\mathcal{A}$ can be simulated successfully by another algorithm $\mathcal{F}$ in Theorem 4.1. Furthermore, by setting parameters appropriately, we can obtain two Theorems 4.3 and 4.4, which correspond to Rej_2 and Rej_3, respectively.

Theorem 4.1 (Generalized Subset Rejection Sampling) *Let V be an arbitrary set, and $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to \mathbb{R}$ be probability distributions. Define a family of set $\hat{S}_v \subset \mathbb{Z}^m$ for $v \in V$. Suppose $g_v : \mathbb{Z}^m \to \mathbb{R}$ is a family of probability distributions indexed by all $v \in V$ and there exist two constants $M \geq 1$, $1 \geq \gamma \geq 0$, and a function $\mathcal{M} : V \times \mathbb{Z}^m \to \mathbb{R}$, which satisfy:*

$$\forall\ v \in V, z \in \hat{S}_v : \mathcal{M}(v, z) \cdot g_v(z) \geq f(z)$$
$$\forall\ v \in V, z \in \hat{S}_v : 1 \leq \mathcal{M}(v, z) \leq M$$
$$\forall\ v \in V : \sum_{z \in \hat{S}_v} f(z) \geq \gamma.$$

then the output distribution of the following algorithm $\mathcal{A}$:

1. $v \xleftarrow{\$} h$
2. $z \xleftarrow{\$} g_v$
3. *if $z \notin \hat{S}_v$ then abort*
4. *output (z, v) with probability $\frac{f(z)}{\mathcal{M}(v,z) \cdot g_v(z)}$*

is identical to the distribution of the following algorithm $\mathcal{F}$:

1. $v \xleftarrow{\$} h$
2. $z \xleftarrow{\$} f$
3. *if $z \notin \hat{S}_v$ then abort*
4. *output (z, v) with probability $1/\mathcal{M}(v, z)$.*

Moreover, the probability of $\mathcal{A}$ and $\mathcal{F}$ outputting something is at least γ/M.

Proof. Given $v \in V$, if $z \in \hat{S}_v$, the probability of $\mathcal{A}$ outputting $z \in \mathbb{Z}^m$ is $g_v(z) \cdot \frac{f(z)}{\mathcal{M}(v,z) \cdot g_v(z)} = \frac{f(z)}{\mathcal{M}(v,z)}$. Otherwise, the probability that $\mathcal{A}$ outputs $z \notin \hat{S}_v$ is 0. As a result, it holds

$$\Pr[\mathcal{A} \text{ outputs something}] = \sum_{v \in V} h(v) \sum_{z \in \hat{S}_v} \frac{f(z)}{\mathcal{M}(v,z)} \geq \frac{\gamma}{M}.$$

Notice also that the probability of $\mathcal{F}$ outputting something is $\sum_{(v,z) \in V \times \hat{S}_v} \frac{h(v)f(z)}{\mathcal{M}(v,z)} \geq \frac{\gamma}{M}$. Besides, it holds

$$\begin{aligned}
\Delta(\mathcal{A}, \mathcal{F}) &= \frac{1}{2} \left(\sum_{(v,z) \in V \times \hat{S}_v} |\mathcal{A}(v,z) - \mathcal{F}(v,z)| \right) \\
&= \frac{1}{2} \sum_{v \in V} h(v) \left(\sum_{z \in \hat{S}_v} \left| g_v(z) \cdot \frac{f(z)}{\mathcal{M}(v,z) \cdot g_v(z)} - \frac{f(z)}{\mathcal{M}(v,z)} \right| \right) \\
&= \frac{1}{2} \sum_{v \in V} h(v) \left(\sum_{z \in \hat{S}_v} \left| \frac{f(z)}{\mathcal{M}(v,z)} - \frac{f(z)}{\mathcal{M}(v,z)} \right| \right) \\
&= 0.
\end{aligned}$$

□

Remark 4.2 *We note that compared with the original rejection sampling of Lemma 3.2 in [36], this generalized version just extends the constant value M to a real-valued function $\mathcal{M}(v,z)$, whose output may vary based on (v,z).*

Next, we consider the special case where $\boldsymbol{v} \in V \subseteq \mathbb{Z}^m$, $f := D_\alpha^m$, $g_{\boldsymbol{v}} := D_{\boldsymbol{v},\alpha}^m$, constant $M = 1$ and the constant function $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) = 1$. Thus, we have the following theorem for the rejection sampling algorithm Rej_2.

Theorem 4.3 *Let V be an arbitrary subset of $\mathbb{Z}^m$, and $h : V \to \mathbb{R}$ be probability distribution. Let $M = 1$. Given any $\boldsymbol{v} \in V$ and any constant $\mathfrak{c}$, define $\hat{S}_{\boldsymbol{v},\mathfrak{c}} = \{\boldsymbol{z} : \langle \boldsymbol{z}, \boldsymbol{v} \rangle \geq \mathfrak{c} \cdot \sigma \|\boldsymbol{v}\|\}$. Then it holds that the output distribution of $\mathcal{A}_2$:*

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D_{\boldsymbol{v},\alpha}^m$
3. *if* $\boldsymbol{z} \notin \hat{S}_{\boldsymbol{v},\mathfrak{c}}$ *then abort*
4. *output* $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* $\frac{D_\alpha^m(\boldsymbol{z})}{D_{\boldsymbol{v},\alpha}^m(\boldsymbol{z})}$.

is identical to the distribution of the following algorithm $\mathcal{F}_2$:

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D_\alpha^m$
3. *if* $\boldsymbol{z} \notin \hat{S}_{\boldsymbol{v},\mathfrak{c}}$ *then abort*

4. output $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* 1.

Moreover, the probability of $\mathcal{A}_2$ *and* $\mathcal{F}_2$ *outputting something is at least* α*, where* α *is the probability of a randomly chosen vector from* $D^m_{\boldsymbol{v},\alpha}$ *belonging to* $\hat{S}_{\boldsymbol{v},\mathfrak{c}}$.

Next, we consider the special case where $\boldsymbol{v} \in V \subseteq \mathbb{Z}^m$, $f := D^m_\alpha$, $g_{\boldsymbol{v}} := D^m_{\boldsymbol{v},\alpha}$, and $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) = \exp\left(\frac{3\langle \boldsymbol{v},\boldsymbol{z}\rangle}{\alpha^2}\right)$. Thus, we have the following theorem for the rejection sampling algorithm Rej_3.

Theorem 4.4 *Let* M *be a constant and* V *be an arbitrary subset of* $\mathbb{Z}^m$*, and* $h : V \to \mathbb{R}$ *be probability distribution. Given any* $\boldsymbol{v} \in V$*, define* $\hat{S}_{\boldsymbol{v},\mathfrak{c}} = \{\boldsymbol{z} : \langle \boldsymbol{z}, \boldsymbol{v}\rangle \geq \mathfrak{c}\cdot\alpha\|\boldsymbol{v}\|\}$*. Then there exists a function* $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) = \exp\left(\frac{3\langle \boldsymbol{v},\boldsymbol{z}\rangle}{\alpha^2}\right)$ *with* $1 \leq \mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) \leq M$ *and* $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) \cdot D^m_{\boldsymbol{v},\alpha}(\boldsymbol{z}) \geq D^m_\alpha(\boldsymbol{z})$*, such that the output distribution of* $\mathcal{A}_3$*:*[9]

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D^m_{\boldsymbol{v},\alpha}$
3. *if* $\boldsymbol{z} \notin \hat{S}_{\boldsymbol{v},\mathfrak{c}}$ *then abort*
4. *output* $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* $\frac{D^m_\alpha(\boldsymbol{z})}{\exp\left(\frac{3\langle \boldsymbol{v},\boldsymbol{z}\rangle}{\alpha^2}\right)\cdot D^m_{\boldsymbol{v},\alpha}(\boldsymbol{z})} = \exp\left(\frac{-8\langle \boldsymbol{z},\boldsymbol{v}\rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right)$.

is identical to the distribution of the following algorithm $\mathcal{F}_3$*:*

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D^m_\alpha$
3. *if* $\boldsymbol{z} \notin \hat{S}_{\boldsymbol{v},\mathfrak{c}}$ *then abort*
4. *output* $(\boldsymbol{z}, \boldsymbol{v})$ *with probability* $\frac{1}{\exp\left(\frac{3\langle \boldsymbol{v},\boldsymbol{z}\rangle}{\alpha^2}\right)}$.

Moreover, the probability of $\mathcal{A}_3$ *and* $\mathcal{F}_3$ *outputting something is at least* $\frac{\gamma}{M}$*, where* γ *is the probability of a randomly chosen vector from* $D^m_{\boldsymbol{v},\alpha}$ *belonging to* $\hat{S}_{\boldsymbol{v},\mathfrak{c}}$.

Security of Opening Proof Protocol with Rej_2 and Rej_3

Here, we need to prove that the opening proof protocol with Rej_2 or Rej_3 satisfies correctness, knowledge soundness and simulatability, whose formal definitions are deferred to Appendix A.1. Similar to [36], we first represent the opening proof of BDLOP commitment as the commit-and-prove functionality $CP = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Prove}, \mathsf{Verify})$, and then show that CP satisfies simulatability, since the properties of correctness and knowledge soundness can be proven almost identically as in [7].

More formally, with random oracle $H : \{0,1\}^* \to \mathcal{C}$, the commit-and-prove functionality $CP = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Prove}, \mathsf{Verify})$ with respect to the language R_L is described as follows, where R_L is defined as $(\mathsf{params}, x, \boldsymbol{m}) \in R_L \Leftrightarrow \boldsymbol{m} \in \mathcal{R}_q$ for certain statement x.

[9] For such function $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) = \exp\left(\frac{3\langle \boldsymbol{v},\boldsymbol{z}\rangle}{\alpha^2}\right)$, the condition $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) \in [1, M]$ implies $\langle \boldsymbol{z}, \boldsymbol{v}\rangle \in [0, (\alpha^2 \cdot \ln M)/3]$.

- $\mathsf{Gen}(1^\lambda)$: Given a security parameter λ, the algorithm generates a commitment public parameter params, which specifies $\mathcal{R}_q^l$ as message space, $S_1^\eta \subset \mathcal{R}^\eta$ as randomness space, and $\mathcal{R}^{n+l}$ as the commitment space. Besides, it also generates $\mathbf{A}_1 \in \mathcal{R}_q^{n\times\eta}, \mathbf{A}_2 \in \mathcal{R}_q^{l\times\eta}$. Without loss of generality, for the underlying ring $\mathcal{R} = \mathbb{Z}[X]/\langle X^d+1\rangle$ and modulus q, we assume that $q\mathcal{R}$ splits as $q\mathcal{R} = \mathfrak{q}_1 \cdots \mathfrak{q}_\ell$, where $\ell = d/c$ for a constant $c \in \mathbb{Z}$, and $q-1 = 2\ell(\mathrm{mod} 4\ell)$. Clearly, if $\ell = d$, we say the ring $\mathcal{R}$ is full-splitting.
  Besides, the algorithm further chooses $\hat{k}$ as the public boosting parameter,[10] such that $\hat{k}|d$ and $q^{-1/\hat{k}}$ is negligible in λ, and set $\sigma := \sigma_{2d/\hat{k}+1} \in \mathrm{Aut}(\mathcal{R}_q)$ is the automorphism of order $\hat{k}d/\ell$.
- $\mathsf{Com}(\mathsf{params}, \boldsymbol{m}; \boldsymbol{r})$: Given params, $\boldsymbol{m} \in \mathcal{R}_q^l$, and randomness $\boldsymbol{r} \in S_1^\eta$, the algorithm generates a commitment $\mathsf{comm} := \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \boldsymbol{r} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{m} \end{bmatrix}$.
- $\mathsf{Prove}(\mathsf{params}, x, \mathsf{comm}, \boldsymbol{m}, \boldsymbol{r})$: Given params, $\mathsf{comm} \in \mathcal{R}_q^{n+l}$, and randomness $\boldsymbol{r} \in S_1^\eta$, the algorithm first samples $\boldsymbol{y}_i \leftarrow D_\alpha^\eta$ and computes $c = H(\{\mathbf{A}_1 \cdot \boldsymbol{y}_i\})$ for $i \in [\hat{k}]$. Then, it computes $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^{i-1}(c) \cdot \boldsymbol{r}$ and gets $b \leftarrow \mathsf{Rej}_j((\boldsymbol{z}_i), (c \cdot \boldsymbol{r}), \alpha)$ for $j = 2$ or 3. If $b = 0$, it outputs $\pi = (c, \boldsymbol{z})$ with $\boldsymbol{z} := (\boldsymbol{z}_i)$. Otherwise abort.
- $\mathsf{Verify}(\mathsf{params}, x, \mathsf{comm}, \pi)$: given $\mathsf{params}, \mathsf{comm}, \pi$, the algorithm parse comm as $\boldsymbol{t}_1 \in \mathcal{R}^n, \boldsymbol{t}_2 \in \mathcal{R}^l$, and parse π as $(c, \boldsymbol{z})$ with $\boldsymbol{z} := (\boldsymbol{z}_i)$. If $\|\boldsymbol{z}_i\| \leq \alpha \cdot \sqrt{d \cdot \eta}$ and $c = H(\{\mathbf{A}_1 \cdot \boldsymbol{z}_i - \sigma^{i-1}(c)\boldsymbol{t}_1\})$, accept. Otherwise, reject.

Furthermore, we have the following theorem.

Theorem 4.5 *In the random oracle model, if D-MLWE-$\mathsf{LS}_{n+l,\eta,\hat{k},q,\chi}$ assumption holds, then the CP with Rej_2 or Rej_3 is simulatable.*

Proof. We complete this proof through using hybrid argument. More formally, we define several hybrids as follows.

Hybrid_0: In this hybrid, all algorithms runs the real CP.

Hybrid_1: This hybrid is identical to Hybrid_0, except that the Prove algorithm is replaced with the following Prove_1:

- $\mathsf{Prove}_1(\mathsf{params}, x, \boldsymbol{m}, \boldsymbol{r})$: the algorithm first samples $\boldsymbol{y}_i \leftarrow D_\alpha^\eta$ and $c \leftarrow \mathcal{C}$, for $i \in [\hat{k}]$. Then, it computes $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^{i-1}(c) \cdot \boldsymbol{r}$ and gets $b \leftarrow \mathsf{Rej}_j((\boldsymbol{z}_i), (\sigma^{i-1}(c) \cdot \boldsymbol{r}), \alpha)$ for $j = 2$ or 3. If $b = 0$, it outputs $\pi = (c, \boldsymbol{z})$ with $\boldsymbol{z} := (\boldsymbol{z}_i)$, and program $c = H(\{\mathbf{A}_1\boldsymbol{z}_i - \sigma^{i-1}(c)\boldsymbol{t}_1\})$.

Clearly, Hybrid_0 and Hybrid_1 are identical except that the random oracle H is programmed at $\{\mathbf{A}_1 \cdot \boldsymbol{y}_i\}$. Thus, Hybrid_0 and Hybrid_1 are indistinguishable (if such value has not been previously queried) for all PPT adversaries, which can be argued similarly as for zero-knowledge property in [9, 20]. Here, we omit this detailed proof for simplicity.

[10] Of course, the number of $\hat{k}$ will affect the proof size of opening proof. Thus, we try to set it as small as possible.

Hybrid$_2$: This hybrid is identical to Hybrid$_1$, except that the Prove_1 algorithm is replaced with the following Prove_2:

- $\mathsf{Prove}_2(\mathsf{params}, x, \boldsymbol{m}, \boldsymbol{r})$: the algorithm first sample $\boldsymbol{z} := (\boldsymbol{z}_i) \leftarrow D_{\alpha}^{\hat{k}\cdot\eta}$ and $c \leftarrow \mathcal{C}$. Then, it gets $b \leftarrow \mathcal{F}_j((\boldsymbol{z}_i), (\sigma^{i-1}(c) \cdot \boldsymbol{r}), \alpha)$ for $i = 2$ or 3, where $\mathcal{F}_j$ is the algorithms defined in Theorem 4.3 or Theorem 4.4. If $b = 0$, it outputs $\pi = (c, \boldsymbol{z})$ and program $c = H(\{\mathbf{A}_1\boldsymbol{z}_i - \sigma^{i-1}(c)\boldsymbol{t}_1\})$.

Clearly, Hybrid$_1$ and Hybrid$_2$ are identical for all adversaries, due to Theorem 4.3 or Theorem 4.4.

Hybrid$_3$: This hybrid is identical to Hybrid$_2$, except that the Com and Prove_2 algorithms are replaced with the following Com_1 and Prove_3:

- $\mathsf{Com}_1(\mathsf{params}, x)$: given params, the algorithm samples $\boldsymbol{t}_1 \xleftarrow{\$} \mathcal{R}_q^n, \boldsymbol{t}_2 \xleftarrow{\$} \mathcal{R}_q^l$ and outputs $\mathsf{comm}^* = (\boldsymbol{t}_1, \boldsymbol{t}_2)$.
- $\mathsf{Prove}_3(\mathsf{params}, x, \mathsf{comm}^*)$: the algorithm first sample $\boldsymbol{z} := (\boldsymbol{z}_i) \leftarrow D_{\alpha}^{\hat{k}\cdot\eta}$, $\boldsymbol{r}^* \xleftarrow{\$} S_1^{\eta}$ and $c \leftarrow \mathcal{C}$. Then, it gets $b \leftarrow \mathcal{F}_j(\boldsymbol{z}, (\sigma^{i-1}(c) \cdot \boldsymbol{r}^*), \alpha)$ for $j = 2$ or 3. If $b = 0$, it outputs $\pi = (c, \boldsymbol{z})$ and program $c = H(\{\mathbf{A}_1\boldsymbol{z}_i - \sigma^{i-1}(c)\boldsymbol{t}_1\})$.

Hybrid$_2$ and Hybrid$_3$ are computational indistinguishability for all PPT adversaries, based on the assumption of D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$, with $\chi = S_1$. Particularly, we have the following formal lemma.

Lemma 4.6 *If the assumption* D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$ *holds, then* Hybrid$_2$ *and* Hybrid$_3$ *are computationally indistinguishable.*

Proof. Notice that if the assumption D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$ holds, then so does the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$ assumption, just as analyzed in Section 1. Particularly, the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$ chooses $(\boldsymbol{z} := (\boldsymbol{z}_i), c)$ from certain distribution for $i \in [\hat{k}]$, yet the leakage version D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$ allows the adversary to specify $(\boldsymbol{z} := (\boldsymbol{z}_i), c)$ in the beginning of the experiment. Thus, D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$ is stronger than the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$, i.e., the D-$\mathsf{MLWE\text{-}LS}_{n+l,\eta,\hat{k},q,\chi}$ assumption implies the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$ assumption.

Then, we focus on establishing the following reduction process: if given an adversary $\hat{\mathcal{A}}$ who can distinguish Hybrid$_2$ and Hybrid$_3$ with probability ε, we can construct another algorithm $\mathcal{B}$ who can solve the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$ problem with probability ε.

Concretely, suppose that $\mathcal{B}$ is given a tuple $\big((\mathbf{A}_1, \mathbf{A}_2), (\boldsymbol{t}_1, \boldsymbol{u}), \boldsymbol{z}, c, \langle \boldsymbol{z}, (\sigma^{i-1}(c) \cdot \boldsymbol{r})\rangle\big)$, for $\boldsymbol{z} \leftarrow D_{\alpha}^{\hat{k}\cdot\eta}, c \leftarrow \mathcal{C}$, $\boldsymbol{r} \leftarrow S_1^{\eta}$ and $i \in [\hat{k}]$. Firstly, $\hat{\mathcal{A}}$ outputs $\boldsymbol{m}$. Then, $\mathcal{B}$ sets $\boldsymbol{t}_2 = \boldsymbol{u} + \boldsymbol{m}$. Finally, if $\langle \boldsymbol{z}, (\sigma^{i-1}(c) \cdot \boldsymbol{r})\rangle \geq \mathfrak{c}\alpha\|c \cdot \boldsymbol{r}\|$ (corresponding to Rej_2) or $\langle \boldsymbol{z}, (\sigma^{i-1}(c) \cdot \boldsymbol{r})\rangle \in [0, \alpha^2 * \ln M]$ (corresponding to Rej_3), then $\mathcal{B}$ sets and outputs $\pi = (c, \boldsymbol{z})$, and sends $(\boldsymbol{t}_1, \boldsymbol{t}_2, \pi)$ to the adversary. Otherwise, it aborts. At the end, $\mathcal{B}$ outputs the bit sent from the adversary $\hat{\mathcal{A}}$.

It is easy to verified that if $\boldsymbol{t}_1 = \mathbf{A}_1\boldsymbol{r}$ and $\boldsymbol{u} = \mathbf{A}_2\boldsymbol{r}$, then the $(\boldsymbol{t}_1, \boldsymbol{t}_2, \pi)$ is followed the distribution of Hybrid$_2$. Otherwise, if $\boldsymbol{t}_1, \boldsymbol{u}$ are chosen uniformly at random and also independent from $\boldsymbol{r}$, the $(\boldsymbol{t}_1, \boldsymbol{t}_2, \pi)$ is followed the distribution of Hybrid_3. Hence, $\mathcal{B}$ can efficiently solve the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$ problem with probability ε. According to the assumption on the hardness of the extended $\mathsf{MLWE}_{n+l,\eta,\hat{k},q,\chi}$, ε is negligible. □

And in Hybrid_3, all vectors are uniformly at random, so the it suffices simulatability. Overall the statement of Theorem 4.5 holds, according to hybrid argument. □

4.3 Further Decreasing Standard Deviation

It should be noted that the boosting procedure in Figure 2 enlarges the norm of the vector $\boldsymbol{z}$ by a factor of $\hat{k}$, compared with the original proof with non-splitting underlying ring $\mathcal{R}$ in [9]. To deal with this issue, the work [35] proposed a simple modification of the protocol. Suppose $q\mathcal{R}$ splits completely, i.e., $\ell = d$. C is the distribution over $\mathcal{R}$, where each coefficient of a challenge $c \leftarrow C$ are independently identically distributed with $\Pr(0) = 1/2$ and $\Pr(1) = \Pr(1) = 1/4$.

Let us define the distribution $\hat{C}$ over $\mathcal{R}^{\hat{k}}$, through first sampling $c = c_0 + c_1 + \ldots + c_{d-1}X^{d-1} \leftarrow C$ and rewrite it as $c = c'_0 + c'_1 X + \cdots + c'_{\hat{k}-1}X^{\hat{k}-1}$ where

$$c'_i = \sum_{j=0}^{d/\hat{k}-1} c_{j\hat{k}+i} X^{j\hat{k}},$$

and then output $(c'_1, \ldots, c'_{\hat{k}})$. By definition of $\sigma := \sigma_{2d/\hat{k}+1} \in \mathrm{Aut}(\mathcal{R}_q)$, we have that $\sigma(c'_i) = c'_i$ for each i. Therefore, we have

$$\sigma^i(c) = \sum_{j=0}^{\hat{k}-1} \sigma^i(X^j) c_j.$$

Now we can sketch the modified protocol as follows. $\mathcal{P}$ samples $\boldsymbol{y}'_1, \cdots, \boldsymbol{y}'_\eta$ from $\mathcal{D}_\sigma^{\hat{k}}$, and then sends $\boldsymbol{w}'_i = \mathbf{A}_1 \cdot \boldsymbol{y}'_i$. Upon receiving a challenge $c \leftarrow \mathcal{C}$, the prover computes $c'_0, \cdots, c'_{\hat{k}-1}$ as above, and sets $\boldsymbol{z}'_i$ as

$$\begin{pmatrix} \boldsymbol{z}'_1 \\ \boldsymbol{z}'_2 \\ \vdots \\ \boldsymbol{z}'_{\hat{k}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{y}'_1 \\ \boldsymbol{y}'_2 \\ \vdots \\ \boldsymbol{y}'_{\hat{k}} \end{pmatrix} + \begin{pmatrix} c'_0 \cdot \boldsymbol{r} \\ c'_1 \cdot \boldsymbol{r} \\ \vdots \\ c'_{\hat{k}-1} \cdot \boldsymbol{r} \end{pmatrix}.$$

Since each c_i has only at most $d/\hat{k}$ non-zero coefficients, one can decrease the standard deviation possibly by a factor of $\hat{k}$. Finally, the prover applies rejection sampling $\mathsf{Rej}_j(\boldsymbol{z}, \boldsymbol{v}, \sigma)$ where $\boldsymbol{z} = \boldsymbol{z}'_1 \| \cdots \| \boldsymbol{z}'_{\hat{k}}$ and $\boldsymbol{v} = c'_0 \cdot \boldsymbol{r} \| \cdots \| c'_{\hat{k}-1} \cdot \boldsymbol{r}$. After receiving vectors $\boldsymbol{z}'_j$, the verifier first checks whether $\mathbf{A}_1 \cdot \boldsymbol{z}'_i = \boldsymbol{w}'_i + c'_{i-1} \cdot \boldsymbol{t}_1$ for $i = 1, \cdots, \hat{k}$ and that each $\boldsymbol{z}'_i$ is small or not.

4.4 Comparison of Efficiency

Intuitively, by using Rej_2 or Rej_3, we can get much better upper bounds for $\frac{D_\alpha^\eta}{D_{\boldsymbol{v},\alpha}^\eta}$ than Equations (2) and (3), allowing us to derive much smaller values for α. Particularly, for Rej_2 and the corresponding Theorem 4.3, if we use the condition that $\langle \boldsymbol{z}, \boldsymbol{v} \rangle \geq \mathfrak{c} \cdot \alpha \|\boldsymbol{v}\|$, then we have

$$\frac{D_\alpha^\eta(\boldsymbol{z})}{D_{\boldsymbol{v},\alpha}^\eta(\boldsymbol{z})} = \exp\left(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) \leq \exp\left(\frac{-2\mathfrak{c} \cdot \alpha \|\boldsymbol{v}\| + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) = 1. \quad (4)$$

Here, we set $-2\mathfrak{c} \cdot \alpha \|\boldsymbol{v}\| + \|\boldsymbol{v}\|^2 = (-2\mathfrak{c} \cdot \alpha + \|\boldsymbol{v}\|) \cdot \|\boldsymbol{v}\| = 0$. Thus, we just need to set $\alpha = \frac{\|\boldsymbol{v}\|}{2\mathfrak{c}}$. Besides, we notice that the event of Rej_2's abort only depends on whether the random vector $\boldsymbol{z}$ is in the subset $\hat{S}_{\boldsymbol{v},\mathfrak{c}}$, for any fixed $\boldsymbol{v}$ and $\mathfrak{c}$. Clearly, by careful balancing the parameter $\mathfrak{c}$, we can get a much smaller α, for the same expected repetition times. The detailed example data are listed in Fig. 4.

Then, for Rej_3 and the related Theorem 4.4, if we assume that $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) = \exp\left(\frac{3\langle \boldsymbol{v}, \boldsymbol{z} \rangle}{\alpha^2}\right)$, then we have

$$\frac{D_\alpha^\eta(\boldsymbol{z})}{\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) \cdot D_{\boldsymbol{v},\alpha}^\eta(\boldsymbol{z})} = \frac{\exp\left(\frac{-2\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right)}{\exp\left(\frac{3\langle \boldsymbol{v}, \boldsymbol{z} \rangle}{\alpha^2}\right)} = \exp\left(\frac{-8\langle \boldsymbol{z}, \boldsymbol{v} \rangle + \|\boldsymbol{v}\|^2}{2\alpha^2}\right) \leq 1. \quad (5)$$

Here, we set $-8\mathfrak{c} \cdot \alpha \|\boldsymbol{v}\| + \|\boldsymbol{v}\|^2 = (-8\mathfrak{c} \cdot \alpha + \|\boldsymbol{v}\|) \cdot \|\boldsymbol{v}\| = 0$. Thus, we just need to set $\alpha = \frac{\|\boldsymbol{v}\|}{8\mathfrak{c}}$. Besides, we notice that the probability of Rej_3's abort depends on $\hat{S}_{\boldsymbol{v},\mathfrak{c}}$ and function $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z})$, i.e., the probability of Rej_3's non-abort is $\Pr[\boldsymbol{z} \in \hat{S}_{\boldsymbol{v},\mathfrak{c}}] \cdot \frac{1}{\exp\left(\frac{3\langle \boldsymbol{v}, \boldsymbol{z} \rangle}{\alpha^2}\right)}$, for $\boldsymbol{z} \xleftarrow{\$} D_\alpha^m$. Clearly, for any fixed $\boldsymbol{v}, \alpha, \boldsymbol{z} \xleftarrow{\$} D_\alpha^m$, $\Pr[\boldsymbol{z} \in \hat{S}_{\boldsymbol{v},\mathfrak{c}}]$ depends on the choice of $\mathfrak{c}$. Notice also that, the condition $\mathcal{M}(\boldsymbol{v}, \boldsymbol{z}) \in [1, M]$ implies $\langle \boldsymbol{z}, \boldsymbol{v} \rangle \in [0, (\alpha^2 \cdot \ln M)/3]$. Thus, through setting different M, we can compute the prover's expected repetition numbers for one time non-abort, and the detailed data are listed in Fig. 4.

According to the above principles, we can determine the concrete proof sizes under various sets of parameters. The detailed numbers are presented in Tables 1 and 4.

Comparison with [28]. As mentioned in the introduction, a concurrent work [28] also improves the state-of-the-art proof of knowledge protocols for BDLOP commitment schemes. Particularly, they remove the additional computational overheads produced by the rejections in the framework of [36], and provide a comparison of the output size under their framework with that under [36]'s framework. We clarify that as a framework similar to [36], our framework also needs more computational overheads than [28]. For the output size, we can provide a fair comparison between our framework and theirs by utilize the benchmark introduced in Section 5.1 of [28].

Concretely, same to [28], 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. In this case, the parameters can be set as: $q \approx 2^{32}, d = 128, \kappa =$

	M	$\mathfrak{c}$	rep.	α	Size of $\boldsymbol{z}$	$\mathfrak{l}$
Rej_0	3	-	≈ 3	$11 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 11 \cdot \|\boldsymbol{v}\|)$	0
	6		≈ 6	$6.74 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 6.74 \cdot \|\boldsymbol{v}\|)$	
Rej_1	3	0	≈ 3	$1.11 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log(12 \cdot 1.11 \cdot \|\boldsymbol{v}\|)$	1
	4		≈ 4	$0.85 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.85 \cdot \|\boldsymbol{v}\|)$	
	6		≈ 6	$0.675 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.675 \cdot \|\boldsymbol{v}\|)$	
Rej_2	1	0.438	≈ 3	$1.142 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 1.155 \cdot \|\boldsymbol{v}\|)$	$\log_2 3$
		0.672	≈ 4	$0.744 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.744 \cdot \|\boldsymbol{v}\|)$	2
		0.97	≈ 6	$0.515 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.515 \cdot \|\boldsymbol{v}\|)$	$\log_2 6$
		1.149	≈ 8	$0.435 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.435 \cdot \|\boldsymbol{v}\|)$	3
Rej_3	1.8	0.5	≈ 5.8	$0.25 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.25 \cdot \|\boldsymbol{v}\|)$	$\log_2 q$
	2	0.5	≈ 6.48	$0.25 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.25 \cdot \|\boldsymbol{v}\|)$	
	2.5	0.5	≈ 8.1	$0.25 \cdot \|\boldsymbol{v}\|$	$\hat{k}\eta d \log_2(12 \cdot 0.25 \cdot \|\boldsymbol{v}\|)$	

Table 4. Comparison with the usage of different rejection sampling algorithms for the protocol in Figs. 2 and 4, where M and $\mathfrak{c}$ denote the parameters for each of four algorithms, rep. denotes prover's expected repetition times for one non-abort, η denotes the dimension of $\boldsymbol{z}$, d denotes the ring dimension of the underlying ring $\mathcal{R}$, $\hat{k}$ denotes the parameter for boosting the soundness. And $\mathfrak{l}$ denotes the number of leakage bits on random during the security proof. Moreover, $\boldsymbol{v} = (\sigma^0(c)\boldsymbol{r}\| \dots \|\sigma^{\hat{k}-1}(c)\boldsymbol{r})$, where $\boldsymbol{r}$ is the randomness vector for BDLOP commitment, and c is the challenge from the verifier in the opening proof protocol.

$32, \ell = 1$. As claimed in [28], one should set $n = 6, \eta = 10$ to achieve the 128-bit security in the framework of [36], and they can set $n = 5, \eta = 9$ to achieve the same security level. On the other hand, their output size is bounded by $(\kappa\alpha_1 + \alpha_2)\sqrt{(n + \eta + \ell)d/\pi}$, where $\alpha_1 \geq 2\sqrt{\frac{2\log(2d(1+1/\varepsilon))}{\pi}}, \alpha_2 \geq 2\sqrt{2}\kappa \cdot \sqrt{\frac{2\log(2d(1+1/\varepsilon))}{\pi}}$, and ε is a security parameter that should be set at most 2^{-128} to be consistent with the 128-bit security. Under these parameters, the output size of [28] is approximately 35490 (ℓ_2-norm of the output vector $\boldsymbol{z}$, following from the presentation of [28]). In our case, we set $n = 6, \eta = 10$, albeit our improvement of [36]'s framework. Meanwhile, the output size in our framework is bounded by $\tau \cdot \frac{\kappa(n+\ell+\eta)d}{\sqrt{\pi}}$, where $\tau = 0.25$ in Rej_3. Therefore, output size of our framework is approximately 9824.

To sum up, the output size in our framework is smaller than that in [28] (approximate 3.6x). Consequently, our framework and [28]'s framework provide a trade-off between computational overhead and communication overhead.

References

1. P. Abla, F.-H. Liu, H. Wang, and Z. Wang. Ring-based identity based encryption - asymptotically shorter MPK and tighter security. In K. Nissim and B. Waters, editors, *TCC 2021, Part III*, volume 13044 of *LNCS*, pages 157–187. Springer, Heidelberg, Nov. 2021.
2. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In Gilbert [26], pages 553–572.
3. S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 98–115. Springer, Heidelberg, Aug. 2010.
4. J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In M. Fischlin, J. Buchmann, and M. Manulis, editors, *PKC 2012*, volume 7293 of *LNCS*, pages 334–352. Springer, Heidelberg, May 2012.
5. J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs. Learning with rounding, revisited - new reduction, properties and applications. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 57–74. Springer, Heidelberg, Aug. 2013.
6. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 595–618. Springer, Heidelberg, Aug. 2009.
7. T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. In Micciancio and Ristenpart [40], pages 470–499.
8. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
9. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In D. Catalano and R. De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Heidelberg, Sept. 2018.
10. A. Boldyreva and D. Micciancio, editors. *CRYPTO 2019, Part I*, volume 11692 of *LNCS*. Springer, Heidelberg, Aug. 2019.
11. J. Bootle, V. Lyubashevsky, and G. Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In Boldyreva and Micciancio [10], pages 176–202.
12. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module-LWE with binary secret. In K. G. Paterson, editor, *CT-RSA 2021*, volume 12704 of *LNCS*, pages 503–526. Springer, Heidelberg, May 2021.
13. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module learning with errors with short distributions. Cryptology ePrint Archive, Paper 2022/472, 2022. `https://eprint.iacr.org/2022/472`.
14. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 868–886. Springer, Heidelberg, Aug. 2012.
15. Z. Brakerski and N. Döttling. Lossiness and entropic hardness for ring-LWE. In R. Pass and K. Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 1–27. Springer, Heidelberg, Nov. 2020.
16. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In S. Goldwasser, editor, *ITCS 2012*, pages 309–325. ACM, Jan. 2012.

17. J. H. Cheon and T. Takagi, editors. *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*. Springer, Heidelberg, Dec. 2016.
18. D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi. LWE with side information: Attacks and concrete security estimation. In Micciancio and Ristenpart [40], pages 329–358.
19. R. del Pino and S. Katsumata. A new framework for more efficient round-optimal lattice-based (partially) blind signature via trapdoor sampling. In Dodis and Shrimpton [21], pages 306–336.
20. R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *ACM CCS 2018*, pages 574–591. ACM Press, Oct. 2018.
21. Y. Dodis and T. Shrimpton, editors. *CRYPTO 2022, Part II*, volume 13508 of *LNCS*. Springer, Heidelberg, Aug. 2022.
22. N. Döttling, D. Kolonelos, R. W. F. Lai, C. Lin, G. Malavolta, and A. Rahimi. Efficient laconic cryptography from learning with errors. In C. Hazay and M. Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 417–446. Springer, Heidelberg, Apr. 2023.
23. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In S. Moriai and H. Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 259–288. Springer, Heidelberg, Dec. 2020.
24. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In Boldyreva and Micciancio [10], pages 115–146.
25. M. F. Esgin, R. K. Zhao, R. Steinfeld, J. K. Liu, and D. Liu. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In L. Cavallaro, J. Kinder, X. Wang, and J. Katz, editors, *ACM CCS 2019*, pages 567–584. ACM Press, Nov. 2019.
26. H. Gilbert, editor. *EUROCRYPT 2010*, volume 6110 of *LNCS*. Springer, Heidelberg, May / June 2010.
27. S. Katsumata and S. Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In Cheon and Takagi [17], pages 682–712.
28. D. Kim, D. Lee, J. Seo, and Y. Song. Toward practical lattice-based proof of knowledge from hint-mlwe. Cryptology ePrint Archive, Paper 2023/623, 2023. `https://eprint.iacr.org/2023/623`.
29. A. Langlois and D. Stehle. Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography, 2015.
30. B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In Cheon and Takagi [17], pages 373–403.
31. F.-H. Liu and Z. Wang. Rounding in the rings. In Micciancio and Ristenpart [40], pages 296–326.
32. V. Lyubashevsky. Lattice signatures without trapdoors. In D. Pointcheval and T. Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, Heidelberg, Apr. 2012.
33. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Dodis and Shrimpton [21], pages 71–101.

34. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In M. Tibouchi and H. Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 218–248. Springer, Heidelberg, Dec. 2021.
35. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In J. Ligatti, X. Ou, J. Katz, and G. Vigna, editors, *ACM CCS 2020*, pages 1051–1070. ACM Press, Nov. 2020.
36. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In J. Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 215–241. Springer, Heidelberg, May 2021.
37. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In Gilbert [26], pages 1–23.
38. V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for ring-LWE cryptography. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 35–54. Springer, Heidelberg, May 2013.
39. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 204–224. Springer, Heidelberg, Apr. / May 2018.
40. D. Micciancio and T. Ristenpart, editors. *CRYPTO 2020, Part II*, volume 12171 of *LNCS*. Springer, Heidelberg, Aug. 2020.
41. A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 525–542. Springer, Heidelberg, Aug. 2011.
42. C. Peikert. A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939, 2015. `https://eprint.iacr.org/2015/939`.
43. C. Peikert and S. Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Boldyreva and Micciancio [10], pages 89–114.
44. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In H. N. Gabow and R. Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
45. D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In M. Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 617–635. Springer, Heidelberg, Dec. 2009.
46. S. Yamada. Adaptively secure identity-based encryption from lattices with asymptotically shorter public parameters. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 32–62. Springer, Heidelberg, May 2016.
47. S. Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part III*, volume 10403 of *LNCS*, pages 161–193. Springer, Heidelberg, Aug. 2017.

A Supplementary Preliminaries

A.1 Commit-and-Prove

Let's recall the commit-and-prove paradigm. Let R_L be a polynomial-time verifiable relation containing (ck, x, w). We call ck the commitment key, x the statement and w the witness. Then, we can define the language L_{ck} as the set of statements x for which there exists a witness w such that $(ck, x, w) \in R_L$.

Now, the commit-and-prove functionality for a relation R_L can be defined as follows. Formally, a commit-and-prove functionality (CP) consists of four algorithms $CP = (\mathsf{Gen}, \mathsf{Com}, \mathsf{Prove}, \mathsf{Verify})$, where $\mathsf{Com}, \mathsf{Verify}$ should be deterministic whereas $\mathsf{Gen}, \mathsf{Prove}$ are probabilistic.

- $\mathsf{Gen}(1^\lambda)$: Given a security parameter λ, generates a commitment key ck. The commitment key specifies a message space $\mathcal{M}_{ck}$ a randomness space $\mathcal{R}_{ck}$ and commitment space $\mathcal{C}_{ck}$.
- $\mathsf{Com}_{ck}(m; r)$: Given a commitment key ck, a message $m \in \mathcal{M}_{ck}$ and randomness $r \in \mathcal{R}_{ck}$ returns a commitment $\mathsf{comm} \in \mathcal{C}_{ck}$.
- $\mathsf{Prove}_{ck}(x, ((m_1, r_1), \cdots, (m_n, r_n)))$: Given a commitment key ck, statement x and commitment openings $m_i \in \mathcal{M}_{ck}, r_i \in \mathcal{R}_{ck}$ and $(ck, x, (m_1, \cdots, m_n)) \in R_L$ returns a proof π.
- $\mathsf{Verify}_{ck}(x, \mathsf{comm}_1, \cdots, \mathsf{comm}_n, \pi)$: Given a commitment key ck, a statement x, a proof π and commitments $\mathsf{comm}_i \in \mathcal{C}_{ck}$, outputs 1 (accept) or 0 (reject).

This paradigm should satisfies the following properties:

Correctness . The commit-and-prove functionality CP has statistical correctness with correctness error $\rho : \mathbb{N} \to [0, 1]$ if for all adversaries $\mathcal{A}$:

$$\Pr\left[\begin{array}{l} ck \leftarrow \mathsf{Gen}(1^\lambda); (x, m_1, r_1, \cdots, m_n, r_n) \leftarrow \mathcal{A}(ck); c_i = \mathsf{Com}_{ck}(m_i; r_i); \\ \pi \leftarrow \mathsf{Prove}_{ck}(x, ((m_1, r_1), \cdots, (m_n, r_n))) : \mathsf{Verify}_{ck}(x, c_1, \cdots, c_n, \pi) = 0 \end{array}\right] \leq \rho(\lambda),$$

where $\mathcal{A}$ outputs $m_i \in \mathcal{M}_{ck}, r_i \in \mathcal{R}_{ck}$ so that $(ck, x, (m_1, \cdots, m_n)) \in R_L$.

Knowledge Soundness . The commit-and-prove functionality CP is knowledge sound with knowledge error $\epsilon : \mathbb{N} \to [0, 1]$ if for all PPT $\mathcal{A}$ there exists an expected polynomial time extractor $\mathcal{E}$ so that:

$$\Pr\left[\begin{array}{l} ck \leftarrow \mathsf{Gen}(1^\lambda); (x, c_1, \cdots, c_n, \pi) \leftarrow \mathcal{A}(ck); ((m_1^*, r_1^*), \cdots, (m_n^*, r_n^*)) \leftarrow \mathcal{E}(c_1, \cdots, c_n) : \\ \mathsf{Verify}_{ck}(x, c_1, \cdots, c_n, \pi) = 1 \wedge ((ck, x, (m_1^*, \cdots, m_n^*)) \notin R_L \vee \exists i, \mathsf{Com}(m_i^*; r_i^*) \neq c_i) \end{array}\right],$$

is less or equal to $\epsilon(\lambda)$, where $\mathcal{E}$ outputs $m_i^* \in \mathcal{M}_{ck}$ and $r_i^* \in \mathcal{R}_{ck}$.

Simulatability . The commit-and-prove functionality CP is simulatable if there exist PPT simulators SimCom and $\mathsf{SimProve}$ such that for all PPT adversaries $\mathcal{A}$:

$$\Pr\left[\begin{array}{l} ck \leftarrow \mathsf{Gen}(1^\lambda); (x, m_1, \cdots, m_n) \leftarrow \mathcal{A}(ck); r_1, \cdots r_n \leftarrow \xi; \forall i, c_i = \mathsf{Com}_{ck}(m_i, r_i); \\ \pi \leftarrow \mathsf{Prove}_{ck}(x, (m_1, r_1), \cdots, (m_n, r_n)) : (ck, x, (m_1, \cdots, m_n)) \in R_L \wedge \mathcal{A}(c_1, \cdots, c_n, \pi) = 1 \end{array}\right]$$

$$\approx \Pr\left[\begin{array}{l} ck \leftarrow \mathsf{Gen}(1^\lambda); (x, m_1, \cdots, m_n) \leftarrow \mathcal{A}(ck); c_1, \cdots, c_n \leftarrow \mathsf{SimCom}_{ck}(x); \\ \pi \leftarrow \mathsf{SimProve}_{ck}(x, c_1, \cdots, c_n) : (ck, x, (m_1, \cdots, m_n)) \in R_L \wedge \mathcal{A}(c_1, \cdots, c_n, \pi) = 1 \end{array}\right],$$

where ξ is a probability distribution on $\mathcal{R}_{ck}$.

The difference between simulatability and zero-knowledge is that randomness $r_1, \cdots, r_n$ is directly generated from ξ as it would in the real-world protocol rather than chosen from adversary.