Commitment Schemes Based on Module-LIP

Hengyi Luo[1,2], Kaijie Jiang[3], Renjie Jin[5], Yanbin Pan[1,2(✉)], and Anyu Wang[3,4]

[1] State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
luohengyi23@mails.ucas.ac.cn, panyanbin@amss.ac.cn
[2] School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China
[3] Institute for Advanced Study, BNRist, Tsinghua University, Beijing, China
jkj21@mails.tsinghua.edu.cn, anyuwang@tsinghua.edu.cn
[4] Zhongguancun Laboratory, Beijing, China
[5] College of Science, National University of Defense Technology, Changsha, China
jinrj97@163.com

Abstract. At EUROCRYPT 2025, Jiang et al. proposed a universal framework for constructing commitment schemes via group actions and instantiated it with the Lattice Isomorphism Problem (LIP) and the Lattice Automorphism Problem (LAP), marking the first application of LIP and LAP in cryptographic constructions beyond signature and encryption schemes. In this paper, we propose a more efficient module lattice version of Jiang et al.'s commitment schemes. More precisely, we adapt all group-action-based components from their framework to the module lattice setting, thereby constructing commitment schemes that rely on specific instances of the module lattice isomorphism problem (module-LIP). We also investigate the hardness of these module-LIP problems to provide security foundations for the schemes, while extending the theoretical foundations of lattice isomorphism problems in the context of module lattices.

Keywords: module-LIP · Group action · Commitment

1 Introduction

The Lattice Isomorphism Problem (LIP) asks to determine whether two given lattices are isomorphic, i.e., whether there exists an orthogonal bijective transformation between them (decision version), or to find such a transformation (search version). Since Ducas and van Woerden [13] constructed cryptographic schemes based on LIP, including a key encapsulation mechanism and a signature scheme, a series of related constructions have been proposed [13,12,2,5,24]. Among them, the efficient signature scheme based on LIP, Hawk, was submitted to NIST and entered the second round of the call for additional signatures. Roughly speaking, the idea of these approaches is to use LIP to mask a known lattice's favorable geometric structure, thereby allowing the constructor to leverage this structure while an adversary cannot do so unless the hardness assumption is broken.

Another natural approach is to view LIP as a matrix group acting on quadratic forms. In this direction, Jiang et al. proposed a universal framework for constructing commitment schemes using group actions and instantiated it with LIP [18]. Roughly speaking, the framework starts from a group action $G \curvearrowright \mathsf{X}$. One selects an element $x \in \mathsf{X}$ as the public commitment key, and uses $g \star x$ as the commitment to bind a group element g. To break the **binding** property, the committer would need to find a non-trivial element in the stabilizer subgroup of x, and this task is hard for certain group actions. To extract the message **hidden** in $g \star x$ from g, a component called the extractor is defined. Finally, to sample g so that its distribution is uniform conditioned on the orbit element $g \star x$, a component called the re-randomization algorithm is defined.

In this viewpoint, the re-randomization algorithm makes the hidden group transition look fresh even after its orbit endpoint is fixed, while the extractor turns that transition into the mask used to hide the message. We recall this protocol-level syntax in Section 2.

It is worth noting that, as mentioned earlier, when designing schemes using LIP, it is desirable that the specific lattice to be masked possesses sufficiently favorable geometric properties, while ensuring that it is sufficiently difficult for an adversary to remove this masking. Therefore, selecting a suitable specific lattice and analyzing the hardness of the corresponding LIP are crucial. Currently, the most prominent concerns are the LIP for the lattice $\mathbb{Z}^n$, i.e., $\mathbb{Z}$-LIP [2,10], and its algebraically structured counterpart, the $\mathcal{O}_{\mathbb{L}}^2$-LIP problem, where $\mathcal{O}_{\mathbb{L}}^2$ denotes the module lattice structure over a cyclotomic number field of conductor a power of two, denoted $\mathbb{L}$ [29,26,7,1]. Beyond these, the LIP corresponding to code-constructed lattices and Barnes-Wall (BW) lattices has also been studied [11,32,22].

Moreover, in both approaches, breaking the scheme typically requires solving the search version of LIP, whereas security reductions are usually constructed using the decision version. Unlike LWE, the search-to-decision reduction for LIP is far from trivial. However, it appears that beyond computing the genus invariant, there are no better algorithms for solving the decision LIP than solving the search LIP itself. Consequently, the current practice is to regard the decision LIP between two lattices as being as hard as determining whether they lie in the same genus. In recent years, the community has also attempted to introduce new invariants of a similar nature [25,27].

Thus, selecting a suitable pair of lattices within a given genus has become a focus. The computability of such a selection was addressed in [35,18]. However, corresponding results for structured lattices are still lacking. For the former approach, HAWK establishes its security proof by introducing a new hard problem (which, in a sense, has not yet been fully understood) to replace decision-LIP.

Previously, the only search-to-decision reduction for LIP was given in [17], which reduced $\mathbb{Z}$-LIP to a set of decision LIPs. It was not until the recent work [16] that this reduction was extended to reduce the LIP corresponding to direct sum lattices over general rings to certain decision LIPs. An important example

is the reduction of $\mathcal{O}_{\mathbb{L}}^2$-LIP to a series of decision LIPs between module lattices generated by

$$B_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & (1+X)^k \end{pmatrix}, \qquad B_k^{(1)} = \begin{pmatrix} 1 & 0 \\ (1+X)^{k-1} & (1+X)^k \end{pmatrix},$$

where X is a primitive root of unity in $\mathbb{L}$. This provides pairs of lattices for decision-LIP that inspire greater confidence than those merely lying in the same genus. Whether the lattices in one pair exhibit some intrinsic differences—whether geometric or algebraic—is a question that merits further study.

1.1 Our Contributions

Since algebraic structure usually improves efficiency, in this paper, we instantiate all components involved in Jiang et al.'s framework [18] using the module version of LIP and investigate the hardness of the underlying problems.

To construct the components with module lattices, a naive idea is to replace the group $\mathrm{GL}_n(\mathbb{Z})$ with $\mathrm{GL}_2(\mathcal{O}_{\mathbb{L}})$, a group related to module lattices. However, obstacles arise when constructing the re-randomization algorithm, since unlike general lattices, where efficient algorithms such as LLL can be used to find an integral basis, there is currently no efficient classical algorithm for finding an $\mathcal{O}_{\mathbb{L}}$-basis of module lattices. This issue is also reflected in the self-reduction of the module lattice isomorphism problem. To bypass this difficulty, the work [12] considers the group $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$. However, if we adopt $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$ directly, we face another obstacle when constructing deterministic extractors, since we can no longer use the determinant as the extractor as in [18]. To meet both requirements simultaneously, we exploit a new group action, together with the corresponding re-randomization algorithm and extractor.

By introducing the Generalized Special Linear group $\mathrm{GSL}_2(\mathcal{O}_{\mathbb{L}})$ over $\mathcal{O}_{\mathbb{L}}$ and its projective group $\mathrm{PGSL}_2(\mathcal{O}_{\mathbb{L}})$, along with selecting appropriate isomorphism classes of Hermitian forms, we define a new group action and successfully design the required re-randomization algorithm together with the extractor.

We would like to point out that it is challenging to prove that our designed algorithm satisfies the requirements of a re-randomization algorithm. We have to carefully switch between the new group action and the original one to leverage the properties brought by the original self-reduction algorithm. In fact, this switch may only hold for the equivalence class of diagonal Hermitian forms.

Another challenge is to find a pair of Hermitian forms such that the extractor can uniformly sample all possible information from the automorphism group of one form, while only extracting fixed information from the automorphism group of the other. Moreover, the distinguishing problem corresponding to this pair of Hermitian forms must be computationally hard. Even for $\mathbb{Z}^n$, [18] provides only the existence of such a pair of Hermitian forms (more precisely, quadratic forms) and some possible ways to find them, rather than an explicit description. In this paper, we describe the desired pair of Hermitian forms by adjusting the concrete hard decision LIP instances $B_k^{(0)}, B_k^{(1)}$ presented in [16].

Furthermore, the security of our commitment schemes is directly based on a special decision version of LIP and a type of lattice automorphism problem (LAP) rather than the more standard search version of LIP as in [18]. Regarding the decision version of LIP, recent work [16] presents a reduction from $\mathcal{O}_{\mathbb{L}}^2$-LIP to a family of decision LIP problems, offering more promising concrete hard decision LIP instances. Regarding LAP, a reduction from $\mathbb{Z}$-LIP to $\mathbb{Z}$-LAP, presented in [19], demonstrates the hardness of $\mathbb{Z}$-LAP. For module-LAP, we prove that it is at least as hard as $\mathcal{O}_{\mathbb{L}}^2$-LIP, which is one of the underlying hard problems of HAWK. Although a series of works [29,26,7,1] have attempted to study the hardness of $\mathcal{O}_{\mathbb{L}}^2$-LIP, there has been no substantial progress in the case where $\mathbb{L}$ is a cyclotomic field.

In general, reductions on structured lattices typically require more complex arguments than reductions on generic lattice problems. Benefiting from the simple structure brought by rank 2, we can adopt the technique of decomposing module lattices via eigenspaces, as used in [26], to achieve the reduction. The main challenge for the reduction lies in the fact that the eigenvalues of automorphisms may not lie within the originally considered field, making it impossible to compute eigenspaces over it. To overcome this, we need to operate over a field extension that includes these eigenvalues. To do these operations requires us to prove that the extended field remains a CM field and that the ring of integers over this extended field can be efficiently computed. For the cyclotomic fields, which are of our primary interest, there already exist abundant results. We also leverage some techniques from algebraic number theory to extend these results to the general case.

Overall, with these components and the framework in [18], we can obtain a series of commitment schemes. Our approach leverages the structured properties of module lattices to achieve improved efficiency, and the security guarantee is based on the hardness of $\mathcal{O}_{\mathbb{L}}^2$-LIP and certain decision module-LIPs. The primary contributions of our work are summarized as follows:

- We identify a suitable group action in the module lattice setting and develop a re-randomization algorithm and an extractor on it, resulting in a series of commitment schemes based on module-LIP within the framework of [18].
- We investigate the hardness assumptions underlying our schemes to provide security foundations, while extending the theoretical foundations of lattice isomorphism problems in the context of module lattices.

It should be noted that although our instantiation on structured lattices offers significant performance improvements over the instantiation on $\mathbb{Z}^n$ lattices in [18], it is still far from a practical scheme. In particular, the generation of commitments involves substantial computational overhead due to the high complexity of the computation of Hermite normal form (invoked in HermiteSolve). In practical implementations, we aim to leverage algebraic structure optimizations such as `TowerSolve` [30], though this would compromise the security reduction in the original framework. We briefly discuss this issue at the end of the paper.

Roadmap. The remainder of this paper is organized as follows. Sec. 2 provides the necessary background and preliminaries. Sec. 3 presents some hardness reductions for module-LIP and module-LAP over certain module lattices related to $\mathcal{O}_{\mathbb{L}}^2$-LIP, including the reduction from $\mathcal{O}_{\mathbb{L}}^2$-LIP to $\mathcal{O}_{\mathbb{L}}^2$-LAP. These reductions will serve as the security foundations for the commitment schemes constructed in the next section. Sec. 4 presents the components required by the construction framework, subsequently obtains the commitment schemes, and concludes with a brief analysis of the schemes' complexity. Finally, Sec. 5 concludes the paper.

2 Notation and Preliminaries

2.1 Notations

- We use $x \leftarrow \mathcal{D}$ to denote that x is sampled from a distribution $\mathcal{D}$. In this paper, we focus solely on discrete distributions. For a finite set S, we denote by $U(S)$ the uniform distribution over S, and write $s \leftarrow_{\$} S$ to indicate that s is drawn uniformly from S.
- The Euclidean norm of $a \in \mathbb{R}^n$ is denoted by $\|a\|$. Let $GL_n(\mathbb{R})$ and $GL_n(\mathbb{Z})$ be the general linear group of rank n over $\mathbb{R}$ and $\mathbb{Z}$, respectively.
- We use J_2 to represent the matrix $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, J_B to represent $B^{-1}J_2B$ for some 2×2 matrix B. We use rI_n to represent the matrix $\mathrm{diag}(r, r, \cdots, r)$.
- For a number field $\mathbb{K}$, the parameter $\mathbf{K}$ denotes the triplet consisting of the degree of $\mathbb{K}$, $\log \Delta_{\mathbb{K}}$, and a basis of $\mathcal{O}_{\mathbb{K}}$.
- For x in a number field $\mathbb{K}$, we call x^* the complex conjugate of x if $\sigma(x^*) = \overline{\sigma(x)}$, $\forall \sigma \in Hom_{\mathbb{Q}}(\mathbb{K}, \mathbb{C})$. For matrix $H = (h_{ij}) \in \mathbb{K}^{m \times n}$, let H^* denote $(h_{ij}^*)^T$ and $\overline{H}$ denote (h_{ij}^*) if all h_{ij}^* exist.
- $\mathcal{S}_n^{>0}(\mathbb{R})$ consists of all positive definite quadratic forms over $\mathbb{R}$. For a CM number field $\mathbb{L}$, the set of Hermitian forms $\mathcal{H}_\ell^{>0}(\mathbb{L})$ consists of all $\mathbf{Q} \in \mathbb{L}^{\ell \times \ell}$ such that $\mathbf{Q}^* = \mathbf{Q}$ and $\mathrm{Tr}_{\mathbb{L}}(v^*\mathbf{Q}v) > 0$ for all $v \in \mathbb{L}^\ell \setminus \{0\}$.
- For a ring A in a number field that is closed under complex conjugation, the unitary group over A is $\mathcal{U}_n(A) := \{T \in M_n(A) | T^*T = I_n\}$.
- For a number field $\mathbb{K}$, we use $\mu(\mathbb{K})$ to denote the roots of unity in $\mathbb{K}$. Note $\mu(\mathbb{K}) \subset \mathcal{O}_{\mathbb{K}}$ and $\mu(\mathbb{K}) = \mathcal{U}_1(\mathcal{O}_{\mathbb{K}})$
- For a number field $\mathbb{L}$, we use $(\mathcal{O}_{\mathbb{L}}^2)^*$ to denote the vectors $\begin{pmatrix} f \\ g \end{pmatrix}$ in $\mathcal{O}_{\mathbb{L}}^2$ such that $(f) + (g) = \mathcal{O}_{\mathbb{L}}$.
- For $\mathbb{L} = \mathbb{Q}[X]/(\Phi_{2^d}(X)) = \mathbb{Q}[X]/(X^{2^{d-1}} + 1)$, let $\Pi = 1 + X$, $\Pi_2 = 1 + X^2$, $\mathfrak{p} = \langle \Pi \rangle$. Define

$$B_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & \Pi^k \end{pmatrix}, \qquad B_k^{(1)} = \begin{pmatrix} 1 & 0 \\ \Pi^{k-1} & \Pi^k \end{pmatrix},$$

and

$$\mathbf{Q}_k^{(i)} = (B_k^{(i)})^* B_k^{(i)}, \qquad i = 0, 1.$$

We also define

$$\widetilde{B}_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & \Pi_2^k \end{pmatrix}, \qquad \widetilde{B}_k^{(1)} = \begin{pmatrix} 1 & 0 \\ \Pi_2^{k-1} & \Pi_2^k \end{pmatrix},$$

and

$$\widetilde{\mathbf{Q}}_k^{(i)} = (\widetilde{B}_k^{(i)})^* \widetilde{B}_k^{(i)}, \qquad i = 0, 1.$$

– $\lceil \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$ denotes the rounding function.

2.2 Lattices and quadratic forms

A lattice is a discrete additive subgroup of $\mathbb{R}^m$ (or less commonly $\mathbb{C}^m$). Typically, one constructs a lattice by selecting n linearly independent vectors $b_1, b_2, \ldots, b_n$ in $\mathbb{R}^m$, so that every lattice element can be expressed as an integer linear combination of these basis vectors. In other words, if we denote the basis by $B = (b_1, \ldots, b_n)$, then the lattice $\mathcal{L}(B)$ is given by $\{Bz : z \in \mathbb{Z}^n\}$, which defines a lattice of rank n in $\mathbb{R}^m$. Denote the i-th minima of $\mathcal{L}$ by $\lambda_i(\mathcal{L})$.

The inner product with respect to $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ is defined as $\langle x, y \rangle_{\mathbf{Q}} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, where $(x, y) \mapsto x^T \cdot \mathbf{Q} \cdot y$. The norm with respect to $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ is defined as $\|x\|_{\mathbf{Q}} = \sqrt{\langle x, x \rangle_{\mathbf{Q}}}$. Note that for a basis B and vectors $x, y \in \mathbb{R}^n$, we have $\langle Bx, By \rangle = x^T B^T B y = \langle x, y \rangle_{B^T B}$. Thus, the geometry of the lattice $\mathcal{L}(B)$ is fully described by $\mathbf{Q} = B^T B$. We can define $\lambda_i(\mathbf{Q})$ as $\lambda_i(\mathcal{L}(B))$.

2.3 Discrete gaussian sampling

Let σ be a positive real number. We define the Gaussian mass $\rho_\sigma : \mathbb{R}^n \to \mathbb{R}$ by $\rho_\sigma(x) = \exp(-\|x\|^2/\sigma^2)$ for any $x \in \mathbb{R}^n$. The discrete Gaussian distribution $D_{\mathcal{L},\sigma}$ over the lattice $\mathcal{L}$ assigns probability $\rho_\sigma(x)/\sum_{y \in \mathcal{L}} \rho_\sigma(y)$ to each $x \in \mathcal{L}$, and zero otherwise.

We also define the Gaussian mass with respect to a quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$ as $\rho_{\mathbf{Q},\sigma} : \mathbb{R}^n \to \mathbb{R}$, given by $\rho_{\mathbf{Q},\sigma}(x) = \exp(-\|x\|_{\mathbf{Q}}^2/\sigma^2)$ for any $x \in \mathbb{R}^n$. If $\mathbf{Q} = B^T B$ for some basis B of $\mathcal{L}$, then $D_{\mathbf{Q},\mathbb{Z}^n,\sigma}(x)$ is equivalent to $D_{\mathcal{L},\sigma}(B \cdot x)$, as $\rho_{\mathbf{Q},\sigma}(x) = \rho_\sigma(B \cdot x)$.

When σ is sufficiently large compared to the maximum length of the Gram–Schmidt basis vectors, efficient sampling from the discrete Gaussian distribution is possible. The following lemma provides the conditions under which this is achievable.

Lemma 2.1 ([12, Lemma 1]) *There exists a probabilistic polynomial-time algorithm that, given a quadratic form $\mathbf{Q} \in \mathcal{S}_n^{>0}(\mathbb{R})$, and a parameter $\sigma \geq \|\tilde{B_{\mathbf{Q}}}\|/\pi \cdot \sqrt{\log(2n+4)}$, outputs a sample according to the distribution $D_{\mathbf{Q},\mathbb{Z}^n,\sigma}$. Here $B_{\mathbf{Q}} \in M_n(\mathbb{R})$ such that $\mathbf{Q} = B_{\mathbf{Q}}^T B_{\mathbf{Q}}$ and $\tilde{B_{\mathbf{Q}}}$ denotes the Gram–Schmidt orthogonalization of $B_{\mathbf{Q}}$, and $\|\tilde{B_{\mathbf{Q}}}\|$ is the length of the longest vector in it. This is independent of the specific decomposition of $\mathbf{Q}$.*

Remark 1. By first LLL-reducing $\mathbf{Q}$, the condition on σ can be replaced by $\sigma \geq 2^{\Theta(n)}\lambda_n(\mathbf{Q})$. Indeed, LLL reduction gives $Q_1 = U^T Q U$ for some $U \in GL_n(\mathbb{Z})$, with a decomposition $Q_1 = B_{Q_1}^T B_{Q_1}$ satisfying

$$\|\widetilde{B}_{Q_1}\| \leq 2^{\Theta(n)}\lambda_n(\mathbf{Q}).$$

Applying the lemma to Q_1, we sample $v \leftarrow D_{Q_1,\mathbb{Z}^n,\sigma}$ and output Uv. Since U is a bijection of $\mathbb{Z}^n$ and

$$(U^{-1}z)^T Q_1 (U^{-1}z) = z^T Q z$$

for every $z \in \mathbb{Z}^n$, the output Uv is distributed as $D_{Q,\mathbb{Z}^n,\sigma}$.

2.4 Number Theory

A number field $\mathbb{K}$ is a finite extension of the rational numbers $\mathbb{Q}$. Equivalently, any such field may be expressed as $\mathbb{Q}[X]/(P)$, where P is a monic irreducible polynomial whose degree matches that of the extension. For a field $\mathbb{K}$ of degree n, there exist exactly n embeddings $\sigma_1, \ldots, \sigma_n$ into $\mathbb{C}$. Those mappings that send $\mathbb{K}$ into $\mathbb{R}$ are called real embeddings, while the others, which occur in complex conjugate pairs, are referred to as complex embeddings. If we denote by r_1 the number of real embeddings and by r_2 the number of pairs of complex embeddings, then $n = r_1 + 2r_2$. A field is said to be totally real if $r_1 = n$, and totally imaginary if $2r_2 = n$.

Canonical embedding. The **canonical embedding** of a number field $\mathbb{K}$ is defined by the mapping $\Sigma_{\mathbb{K}} : x \mapsto (\sigma_1(x), \ldots, \sigma_n(x))^T$, which sends elements into $\mathbb{C}^n$. Often, one identifies $\mathbb{K}$ with its image under this mapping so that the ring of integers $\mathcal{O}_{\mathbb{K}}$ acquires a lattice structure in $\mathbb{C}^n$. It is important to note, we are not representing elements in $\mathbb{K}$ using the canonical embedding. The norm on $\mathbb{K}$ is defined as $\mathcal{N}_{\mathbb{K}}(z) = \prod_{i=1}^n \sigma_i(z)$ and the trace as $\mathrm{Tr}_{\mathbb{K}}(z) = \sum_{i=1}^n \sigma_i(z)$. By considering the $\mathbb{Q}$-linear map $m_z : x \mapsto zx$, one sees that $\mathcal{N}_{\mathbb{K}}(z)$ equals the determinant of m_z and $\mathrm{Tr}_{\mathbb{K}}(z)$ equals its trace; notably, both quantities belong to $\mathbb{Q}$.

We also have the **coefficient embedding** $\mathrm{vec} : \mathbb{Q}[X]/(P) \to \mathbb{Q}^n, \quad a_0 + a_1 X + \cdots + a_{n-1}X^{n-1} \mapsto (a_0, a_1, \ldots, a_{n-1})^T$, which is an additive group isomorphism. In the 2-power cyclotomic setting, with the power basis $1, X, \ldots, X^{n-1}$, the canonical-embedding norm and the coefficient norm are related by

$$\|\Sigma_{\mathbb{L}}(f)\|^2 = n\|\mathrm{vec}(f)\|^2.$$

Thus norm estimates stated in the canonical embedding can be translated into the coefficient representation used by the concrete algorithms.

CM number field. A number field $\mathbb{L}$ is a CM number field if it is a quadratic extension $\mathbb{L}/\mathbb{K}$ where the base field $\mathbb{K}$ is totally real and $\mathbb{L}$ is totally imaginary. An equivalent condition is $\mathbb{L} = \mathbb{K}(\alpha)$ with $\mathbb{K}$ totally real, $\alpha^2 \in \mathbb{K}$ and $\phi(\alpha^2) < 0$ for any homomorphism $\phi : \mathbb{K} \to \mathbb{C}$. The extension $\mathbb{L}/\mathbb{K}$ is a Galois extension and we denote the Galois group by $Gal(\mathbb{L}/\mathbb{K})$. There is a complex conjugation in

$Gal(\mathbb{L}/\mathbb{K})$, i.e., there exists $\tau \in Gal(\mathbb{L}/\mathbb{K})$ s.t. $\forall x \in \mathbb{L}, \sigma_i(\tau(x)) = \overline{\sigma_i(x)}$. We usually denote $\tau(x)$ by x^*. Then the natural Hermitian inner product $\langle \Sigma(x), \Sigma(y) \rangle$ is equal to $\mathrm{Tr}_{\mathbb{L}}(x^* y)$. As an important example, the cyclotomic number fields are all CM number fields.

Rings of integer and ideals. Let $\mathcal{O}_{\mathbb{L}}$ denote the ring of integers of a number field $\mathbb{L}$. $\mathcal{O}_{\mathbb{L}}$ is a free $\mathbb{Z}$-module of rank n. The discriminant of $\mathbb{L}$, denoted $\Delta_{\mathbb{L}}$, is defined as $(\det(\sigma_i(\alpha_j))_{i,j})^2 \in \mathbb{Z}$, where $(\alpha_j)_{1 \le j \le n}$ is any basis of $\mathcal{O}_{\mathbb{L}}$. Specifically, there exists some absolute constant $c > 1$ such that $\Delta_{\mathbb{L}} \ge c^n$ for all number fields $\mathbb{L}$(due to Minkowski [31, pp. 261–264]). In particular, we always have $n = \mathrm{poly}(\log \Delta_{\mathbb{L}})$. A (fractional) ideal $\mathcal{I}$ is a finitely generated additive subgroup of $\mathbb{L}$ such that $x \cdot \mathcal{I} \subseteq \mathcal{I}$ for all $x \in \mathcal{O}_{\mathbb{L}}$. When an ideal is contained in $\mathcal{O}_{\mathbb{L}}$, we call it an integral ideal. We use $(x) := x \cdot \mathcal{O}_{\mathbb{L}}$ to denote the principal ideal generated by a single element $x \in \mathbb{L}$.

Module lattices and Hermitian forms. Assume $\mathbb{L}$ is a CM number field with degree n. A module lattice M (over $\mathbb{L}$) is an $\mathcal{O}_{\mathbb{L}}$ module in $\mathbb{L}^\ell$. Through canonical embedding $\Sigma_{\mathbb{L}}$, we can map M to a lattice in $\mathbb{C}^{n\ell}$. It has the form $b_1\mathcal{I}_1 + \cdots + b_r\mathcal{I}_r$, where the $\mathcal{I}_i$'s are non-zero fractional ideals of $\mathbb{L}$ and $(b_1, \ldots, b_r)$ are $\mathbb{L}$-linearly independent vectors of $\mathbb{L}^\ell$, for some $\ell > 0$. We call $\mathbf{B} = (B, (\mathcal{I}_i)_{1 \le i \le r})$ a pseudo-basis for M, where B is the matrix whose columns are the b_i. If there is a pseudo-basis of M such that all $\mathcal{I}_i$ are $\mathcal{O}_{\mathbb{L}}$, we call M a free module and B the free basis of M.

When $\mathbb{L}$ is a CM number field, the natural Hermitian inner product of $v, w \in \mathbb{L}^\ell$ under the embedding is equal to $\mathrm{Tr}_{\mathbb{L}}(v^* \cdot w)$. Similar to the definition over lattices, the inner product with respect to $\mathbf{Q} \in \mathcal{H}_\ell^{>0}(\mathbb{L})$ is defined as $\langle x, y \rangle_{\mathbf{Q}} : \mathbb{L}^\ell \times \mathbb{L}^\ell \to \mathbb{Q}$, where $(x, y) \mapsto \mathrm{Tr}(x^* \cdot \mathbf{Q} \cdot y)$. The norm with respect to $\mathbf{Q} \in \mathcal{H}_n^{>0}(\mathbb{L})$ is defined as $\|x\|_{\mathbf{Q}} = \sqrt{\langle x, x \rangle_{\mathbf{Q}}}$. We omit the subscript $\mathbf{Q}$ when $\mathbf{Q} = I_\ell$. Note that for a $B \in \mathbb{L}^{\ell \times \ell}$ and vectors $x, y \in \mathbb{L}^\ell$, we have $\langle Bx, By \rangle = \langle x, y \rangle_{B^* B}$.

We can therefore define the Gaussian mass $\rho_\sigma : \mathbb{L}^\ell \to \mathbb{R}$ by $\rho_\sigma(v) = \exp(-\mathrm{Tr}_{\mathbb{L}}(v^* v)/\sigma^2)$ for any $v \in \mathbb{L}^\ell$ and the Gaussian mass with respect to a Hermitian form $\mathbf{Q} \in \mathcal{H}_\ell^{>0}(\mathbb{L})$ as $\rho_{\mathbf{Q},\sigma} : \mathbb{L}^\ell \to \mathbb{R}$, given by $\rho_{\mathbf{Q},\sigma}(x) = \exp(-\|x\|_{\mathbf{Q}}^2/\sigma^2)$ for any $x \in \mathbb{L}^\ell$.

Cases on HAWK's setting. In HAWK, they consider $\mathbb{L} = \mathbb{Q}[X]/(\Phi_{2^d}(X)) = \mathbb{Q}[X]/(X^{2^{d-1}} + 1)$ to be a power of two cyclotomic number field and $\zeta_{2^d} := X$ to be a 2^d-th root of unity. Let $n = 2^{d-1} = \deg(\mathbb{L})$, $\Pi = 1 + X$, $\mathfrak{p} = \langle \Pi \rangle$. We have $\mathfrak{p}^{2^k} = \left\langle 1 + X^{2^k j} \right\rangle$ for any $0 \le k \le d-2$ and odd j, and $\langle 2 \rangle = \mathfrak{p}^n \subset \mathfrak{p}$. A corollary is that $\overline{\mathfrak{p}^s} = \overline{\mathfrak{p}}^s = \mathfrak{p}^s$ for any $s \in \mathbb{N}$.

In this case, we have $\mathcal{O}_{\mathbb{L}} = \mathbb{Z}[X]/(X^{2^{d-1}} + 1)$, and we directly represent elements in the field using their polynomial forms. This can also be understood as choosing $\{1, X, \ldots, X^{n-1}\}$ as a basis of $\mathcal{O}_{\mathbb{L}}$, where the resulting vector representation corresponds to the coefficient embedding. Under the coefficient embedding $\mathrm{vec} : a_0 + a_1 X + \cdots + a_{n-1} X^{n-1} \mapsto \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix}$, $\mathcal{O}_{\mathbb{L}}$ corresponds to $\mathbb{Z}^n$. Under this

basis, the linear transformation m_f corresponding to multiplication by f is represented by the matrix

$$\mathrm{rot}(f) = \big(\mathrm{vec}(f) | \, \mathrm{vec}(Xf) | \cdots | \, \mathrm{vec}(X^{n-1}f)\big) \in \mathbb{Q}^{n\times n}, \quad \text{for } f \in \mathbb{L},$$

and this naturally extends to matrices $B \in \mathbb{L}^{k\times\ell}$ as

$$\mathrm{rot}(B) = \begin{pmatrix} \mathrm{rot}(B_{11}) & \cdots & \mathrm{rot}(B_{1\ell}) \\ \vdots & & \vdots \\ \mathrm{rot}(B_{k1}) & \cdots & \mathrm{rot}(B_{k\ell}) \end{pmatrix} \in \mathbb{Q}^{nk\times n\ell}.$$

We have $\|f\|^2 = n\|\mathrm{vec}(f)\|^2$ and so $\rho_{\mathbf{Q},\sigma}(f/\sqrt{n}) = \rho_{\mathbf{Q},\sqrt{n}\sigma}(f) = \rho_{\mathrm{rot}(\mathbf{Q}),\sigma}(\mathrm{vec}(f))$.

The conjugation and multiplication-by-X operations are then expressed as $\tau : a_0 + a_1X + \cdots + a_{n-1}X^{n-1} \mapsto a_0 - (a_{n-1}X + \cdots + a_1X^{n-1})$, and $m_X : a_0 + a_1X + \cdots + a_{n-1}X^{n-1} \mapsto -a_{n-1} + a_0X + \cdots + a_{n-2}X^{n-1}$. They both belong to $\mathrm{Aut}(\mathbb{Z}^n) = S_n^{\pm}$.

In the setting considered by HAWK, it is required to sample matrices in SL_2 over the 2-power cyclotomic ring $\mathcal{O}_\mathbb{L}$ both in its key generation and in the self-reduction described in the paper. The approach is to first perform a sampling over $\mathcal{O}_\mathbb{L}^2$, and then complete it into a matrix in SL_2. The following lemma guarantees the existence of an efficient algorithm for the latter step.

Lemma 2.2 ([12, Algorithm 4]) *There is an efficient deterministic algorithm* `HermiteSolve` *that, given the field* $\mathbb{L}$ *and* $(f, g) \in \mathcal{O}_\mathbb{L}^2$*, outputs* $(F, G) \in \mathcal{O}_\mathbb{L}^2$ *such that* $fG - Fg = 1$ *when* $(f) + (g) = \mathcal{O}_\mathbb{L}$*, and outputs* $\perp$ *otherwise.*

The following Heuristic Assumption guarantees that when the first step of sampling is performed via discrete Gaussian sampling, the probability that the above algorithm fails is approximately 1/4. In Appendix A of the full version [14], they give a heuristic explanation and matching experimental evidence for $\mathbf{Q} = I_2$. Indeed, for this case, Lemma 4.4 in [33] establishes a constant lower bound. For $\mathbf{Q} = \mathbf{Q}_k^{(0)}$, our experimental verification for small k also aligns with this heuristic expectation.

Heuristic 1 (For $\mathbf{Q}_k^{(0)}$) *Suppose* $n \geq 4$ *and* $\sigma \geq \Omega(n)$*, the probability value*

$$\rho_{\mathbf{Q},\sigma}((\mathcal{O}_\mathbb{L}^2)^*)/\rho_{\mathbf{Q},\sigma}(\mathcal{O}_\mathbb{L}^2)$$

is around $\frac{3}{4}$*.*

A simple observation is that for any $U \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L})$, we have $U \cdot \mathcal{O}_\mathbb{L}^2 = \mathcal{O}_\mathbb{L}^2$ and $U \cdot (\mathcal{O}_\mathbb{L}^2)^* = (\mathcal{O}_\mathbb{L}^2)^*$. Hence, $\rho_{U^*\mathbf{Q}U,\sigma}(\mathcal{O}_\mathbb{L}^2) = \rho_{\mathbf{Q},\sigma}(\mathcal{O}_\mathbb{L}^2)$ and $\rho_{U^*\mathbf{Q}U,\sigma}((\mathcal{O}_\mathbb{L}^2)^*) = \rho_{\mathbf{Q},\sigma}((\mathcal{O}_\mathbb{L}^2)^*)$.

2.5 Module-LIP, module-LAP and module-ΔLIP

For simplicity, let $\mathbb{L}$ be a CM number field in this subsection.

Definition 2.1 *Two Hermitian forms* $\mathbf{Q}_0$ *and* $\mathbf{Q}_1$ *over* $\mathbb{L}$ *are said to be isomorphic if there exists* $U \in SL_n(\mathcal{O}_\mathbb{L})$ *(resp.* $GL_n(\mathcal{O}_\mathbb{L})$*) such that* $\mathbf{Q}_1 = U^*\mathbf{Q}_0U$*. The isomorphism class of* $\mathbf{Q}_0$ *is denoted by* $[\mathbf{Q}_0]_{sl(resp.\ gl)}$*. Sometimes we write* $[\mathbf{Q}_0]_{sl}$ *as* $[\mathbf{Q}_0]$ *for simplicity.*

Definition 2.2 (Module-Lattice isomorphism Problem, module-LIP) *Given Hermitian forms* $\mathbf{Q_1} \in [\mathbf{Q_0}]$ *and parameters of* $\mathbb{L}$*, the objective is to find*

$$U \in SL_\ell(\mathcal{O}_\mathbb{L}),$$
$$\mathbf{Q}_1 = U^*\mathbf{Q}_0U.$$

We denote this problem as

$$\textit{module-LIP}_{\mathbb{L},\mathbf{Q}_0,\mathbf{Q}_1},$$

and typically omit the subscript $\mathbb{L}$ *when no ambiguity arises.*

If $\mathbf{Q_0}$ *is fixed, denote by*

- $\textit{module-LIP}_{[\mathbf{Q_0}]}$ *the problem of solving* $\textit{module-LIP}_{\mathbf{Q_0},\mathbf{Q_1}}$ *for any* $\mathbf{Q_1} \in [\mathbf{Q_0}]$*,*
- $\textit{module-LIP}_{\mathcal{D}_{[\mathbf{Q_0}]}}$ *the problem of solving* $\textit{module-LIP}_{\mathbf{Q_0},\mathbf{Q_1}}$ *for* $\mathbf{Q_1}$ *sampled from the distribution* $\mathcal{D}_{[\mathbf{Q_0}]}$*.*

Remark 2. The choice of the group for the transition matrix U induces different isomorphism classes. Note that for $U \in \mathrm{GL}_n(\mathcal{O}_\mathbb{L})$, given $\mathbf{Q} = U^*\mathbf{Q_0}U$ in $[\mathbf{Q_0}]_{\mathrm{gl}}$, we can first apply the algorithm `LS1` (as in Proposition 2.2) to $\det(\mathbf{Q})/\det(\mathbf{Q_0})$ to compute $\det(U)$, and then perform a congruence transformation on $\mathbf{Q}$ using $\mathrm{diag}(\det(U)^{-1}, 1, \cdots, 1)$ to convert it into an instance in $[\mathbf{Q_0}]_{\mathrm{sl}}$. Therefore, the choice of isomorphism class does not affect the hardness of the problem. When the two Hermitian forms in the decision problem have the same determinant, the same reasoning applies to the decision version of module-LIP.

Definition 2.3 *For a Hermitian form* $\mathbf{Q}$ *over* $\mathbb{L}$*, we define its automorphism group as* $\{U \in GL_\ell(\mathcal{O}_\mathbb{L}) | U^*\mathbf{Q}U = \mathbf{Q}\} \triangleq Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q})$*. Note that* $\mu(\mathbb{L})I_\ell \subseteq \mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q})$ *always holds; we refer to these automorphisms as trivial.*

Remark 3. The automorphism group has such a property: $Aut_{\mathcal{O}_\mathbb{L}}(U^*\mathbf{Q}U) = U^{-1}Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q})U$. Indeed, we always have $\mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}) \subseteq \mathrm{GSL}_\ell(\mathcal{O}_\mathbb{L})$ (see its definition in Definition 4.1).

Definition 2.4 (module-Lattice Automorphism Problem, module-LAP) *Given a Hermitian form* $\mathbf{Q}$*, the objective is to find a non-trivial module lattice automorphism* $P \in Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}) \backslash \mu(\mathbb{L})I_\ell$*. We denote this problem as* $\textit{module-LAP}_{\mathbb{L},\mathbf{Q}}$*, and typically omit the subscript* $\mathbb{L}$ *when no ambiguity arises. Similarly to module-LIP, we define* $\textit{module-LAP}_{\mathbb{L},[\mathbf{Q}]_{gl}}$ *and* $\textit{module-LAP}_{\mathbb{L},\mathcal{D}_{[\mathbf{Q}]_{gl}}}$ *by sampling the form from the full isomorphism class and from the distribution on that class, respectively.*

Example 2.1 *Currently, the most prominent case is $\ell = 2$, $\mathbf{Q} = I_2$, corresponding to the hardness assumption used in HAWK. We denote these problems as $\mathcal{O}_{\mathbb{L}}^2$-LIP and $\mathcal{O}_{\mathbb{L}}^2$-LAP, and $Aut_{\mathcal{O}_{\mathbb{L}}}(I_2) = \mathcal{U}_2(\mathcal{O}_{\mathbb{L}})$.*

Lemma 2.3 ([26, Lemma 3.1]) *Let $\mathbb{L}$ be a CM number field with degree n. Then*

$$\mathcal{U}_2(\mathcal{O}_{\mathbb{L}}) = \{\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L})\} \bigcup \{\begin{pmatrix} 0 & \xi_1 \\ \xi_2 & 0 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L})\}.$$

Furthermore, $\sharp(\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})) \leq 2\sharp(\mu(\mathbb{L}))^2 \leq 8n^4$.

Definition 2.5 (module-ΔLIP) *For Hermitian forms $\mathbf{Q_1}$ and $\mathbf{Q_0}$, given $\mathbf{Q} \in [\mathbf{Q_1}] \cup [\mathbf{Q_0}]$ and parameters of $\mathbb{L}$, the objective is to determine whether $\mathbf{Q}$ belongs to $[\mathbf{Q}_0]$ or to $[\mathbf{Q}_1]$. We denote this problem as module-$\Delta LIP_{\mathbb{L},\mathbf{Q},\mathbf{Q_0},\mathbf{Q_1}}$, and typically omit the subscript $\mathbb{L}$ when no ambiguity arises. Denote by*

- *module-$\Delta LIP_{[\mathbf{Q_0}],[\mathbf{Q_1}]}$ the problem of solving module-$\Delta LIP_{\mathbf{Q},\mathbf{Q_0},\mathbf{Q_1}}$ for any $\mathbf{Q} \in [\mathbf{Q_1}] \cup [\mathbf{Q_0}]$,*
- *module-$\Delta LIP_{\mathcal{D}_{[\mathbf{Q_0}]},\mathcal{D}_{[\mathbf{Q_1}]}}$ the problem of solving module-$\Delta LIP_{\mathbf{Q},\mathbf{Q_0},\mathbf{Q_1}}$ for $\mathbf{Q}$ sampled from the distribution $\mathcal{D}_{[\mathbf{Q_0}]}$ or the $\mathcal{D}_{[\mathbf{Q_1}]}$.*

The following theorem is an important example under the general result in [16]. It presents a family of decision LIP and proves that if this family of decision LIP can be solved, then the $\mathcal{O}_{\mathbb{L}}^2$-LIP can be solved. The original text uses the geometric version of the module-LIP representation, which is here adapted to the representation of the Hermitian form under a free basis. The core reduction technique remains the same, but for completeness, its proof will be provided in the Appendix C.

Theorem 2.1 (Adapted from [16, Theorem 10.3]) *In line with HAWK's configuration, let $B_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & \Pi^k \end{pmatrix}$ and $B_k^{(1)} = \begin{pmatrix} 1 & 0 \\ \Pi^{k-1} & \Pi^k \end{pmatrix}$, and*

$$\mathbf{Q}_k^{(0)} = (B_k^{(0)})^* B_k^{(0)}, \qquad \mathbf{Q}_k^{(1)} = (B_k^{(1)})^* B_k^{(1)}.$$

The parameters of module-$\Delta LIP_{[\mathbf{Q}_k^{(0)}],[\mathbf{Q}_k^{(1)}]}$ depend only on k and the field $\mathbb{L}$. For convenience, we denote it as $\Delta_{k,\mathbb{L}}$. There is a (quantum) reduction from $\mathcal{O}_{\mathbb{L}}^2$-LIP to $\{\Delta_{k,\mathbb{L}},\ k = 2, \cdots, 2n^2\}$. In the reduction, each solver for $\Delta_{k,\mathbb{L}}$ is invoked at most once.

Remark 4. Although the original paper mentions that the first $\log n + 1$ $\Delta_{k,\mathbb{L}}$ oracles can be omitted, this appears to be achieved through a form of guessing rather than fundamentally eliminating the need for these oracles. In fact, since each oracle is invoked only once and a simple guess yields the correct oracle output with probability 1/2, we can always eliminate $O(\log n)$ oracles via guessing.

2.6 Algorithmic consideration

Representation of ideals and modules. Assume $B_{\mathcal{O}_\mathbb{L}} = (\alpha_j)_{j=1,\ldots,n}$ is a basis of $\mathcal{O}_\mathbb{L}$. We represent elements in $\mathbb{L}$ by their coordinates in the basis $B_{\mathcal{O}_\mathbb{L}}$, which is a vector in $\mathbb{Q}^n$. For $x \in \mathbb{L}$ represented by the vector $(x_1, \ldots, x_n)^T \in \mathbb{Q}^n$, we define $\text{size}(x) := \sum_i \text{size}(x_i)$, where $\text{size}(a/b) := \lceil \log_2 |a| \rceil + \lceil \log_2 |b| \rceil$ for $a, b \in \mathbb{Z}$ coprime. As is customary, we assume that in this paper the $B_{\mathcal{O}_\mathbb{L}}$ is always a LLL-reduced basis of $\mathcal{O}_\mathbb{L}$, i.e. $\sigma(B_{\mathcal{O}_\mathbb{L}})$ forms an LLL-reduced basis of $\mathcal{O}_\mathbb{L}$. This choice is made to ensure that the coefficients of $\alpha_i\alpha_j$ under $B_{\mathcal{O}_\mathbb{L}}$ representation do not blow up.

Basic algorithms. Here, we list some algorithms that will be used in the reduction presented in Section 3. The following lemma guarantees that the computation of roots of unity in a given field is efficient.

Lemma 2.4 ([28, Corollary 2.11]) *Let $\mathbb{K}$ be a degree n number field. Then, $\mathbb{K}$ has at most $2n^2$ roots of unity, and there is a polynomial-time algorithm that, given a basis of the ring of integers $\mathcal{O}_\mathbb{K}$, computes the roots of unity in $\mathbb{K}$.*

The following lemma guarantees that we can compute the intersection of a module lattice and a $\mathbb{L}$-linear space.

Lemma 2.5 ([26, Lemma 4.4]) *Let $\mathbb{L}$ be a number field, and $B_{\mathcal{O}_\mathbb{L}}$ be a $\mathbb{Z}$ basis of $\mathcal{O}_\mathbb{L}$. Then for $A \in \mathbb{L}^{2\times 2}$ and a lattice $\mathcal{L} \subseteq \mathbb{L}^2$, there is a deterministic polynomial-time algorithm that, given $B_{\mathcal{O}_\mathbb{L}}$, A, and a basis $B_\mathcal{L}$ of $\mathcal{L}$, outputs $\ker(A) \cap \mathcal{L}$.*

Proposition 2.1 ([26, Proposition 3.2]) *Let $B \in GL_2(\mathbb{L})$, and $r \in \mathbb{L}$. Define $t_* : \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{L}^2 \mapsto \begin{pmatrix} x^* \\ y^* \end{pmatrix} \in \mathbb{L}^2$. It's a $\mathbb{Q}$ linear map. Given as input a basis of $\mathcal{O}_\mathbb{L}$, $G = B^*B$, and $\det(B)$, we can compute $B^{-1}J_2t_*B$ and $m_r := B^{-1}(rI_2)B$ in polynomial time in the input size.*

From a constructive perspective, we will use the Hermitian form version of the free module-LIP as the hardness assumption. However, in some reductions, certain geometric versions of the problem are more convenient, because we need to perform operations such as intersection and summation, which can disrupt the free module representation (for example, the original version of Theorem 2.1 in [16]). In plain LIP, the conversion between the geometric and quadratic form versions is straightforward.

For module-LIP, the first challenge is to recover a free basis from a pseudo-basis of a free module, which is non-trivial. The folklore approach involves first computing an almost-free basis and then generating a generator for the remaining principal ideal of $\mathcal{O}_\mathbb{L}$, which can be done in polynomial time on a quantum computer (refer to the "Free modules" section in [9]). Secondly, operations such as module lattice addition and intersection can disrupt the pseudo-basis structure. The folklore method addresses this by computing the Hermite Normal Form

(HNF) (e.g., see the "Computing arithmetic operations with modules" section in [7]). Alternatively, a more direct approach is to first treat the module as a $\mathbb{Z}$-lattice, then compute the HNF using a $\mathbb{Z}$-basis as pseudo-generators to recover a pseudo-basis. Notably, [3] provides a polynomial-time algorithm for computing HNF in the full-rank case. Combining these techniques yields the following lemma. For completeness, we include a proof in Appendix C.

It will be employed in the proofs of Theorem 2.1 and Theorem 3.3. In all quantum reductions in this paper, the "quantum" requirement comes from this lemma, or more precisely, from recovering a free basis from a pseudo-basis. If we permitted the hard problem to be defined using pseudo-bases (a definition which is also meaningful from the perspective of hard problems), this requirement could be removed. However, for the constructive purposes of this paper, we employ the current definition.

Lemma 2.6 *Let $\mathbb{L}$ be a number field, and $B_{\mathcal{O}_\mathbb{L}}$ be a $\mathbb{Z}$ basis of $\mathcal{O}_\mathbb{L}$. Then there exists a quantum polynomial-time algorithm such that for any full rank free $\mathcal{O}_\mathbb{L}$-module $M \subseteq \mathbb{L}^m$, given a $\mathbb{Z}$-basis $B_{\mathcal{O}_\mathbb{L}}$ of $\mathcal{O}_\mathbb{L}$ and a $\mathbb{Z}$-basis of M, it outputs a free basis of M.*

Lenstra-Silverberg Algorithm. Gentry and Szydlo initially proposed an algorithm in [17] to recover x from x^*x and xR (where R is a certain type of polynomial ring). Later, Lenstra and Silverberg extended this in [20,21,23]. Luo et al. used it to show an algorithm for which, in a certain sense, is a high-dimensional version of the rank 1 module-LIP.

Proposition 2.2 ([26, Proposition 4.1]) *Let $\mathbb{F}$ be a CM-field or a totally real number field with degree n. Let A be the ring of integers of $\mathbb{F}$. [23, Examples 3.7 $(i)-(ii)$)] showed that A is a CM-order. The conjugate automorphism is just the complex conjugation $x \mapsto x^*$, and the trace function is just $Tr_\mathbb{F}$.*

- *For $\alpha \in \mathbb{F}$, there is a deterministic polynomial-time algorithm* `LS1` *that, given A, αA and $\alpha^*\alpha$, finds $\alpha\mu(A)$ in polynomial time, where $\mu(A)$ means roots of unity in A.*
- *For $v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \in \mathbb{F}^2$ and $B \in GL_2(\mathbb{F})$, there is a deterministic polynomial-time algorithm* `LS2` *that, given A, B^*B, $v^*v = v_1v_1^* + v_2v_2^*$, and $B^{-1}(A \cdot v)$, finds $B^{-1}(\mu(A) \cdot v)$ in polynomial time, where $\mu(A)$ means roots of unity in A.*

2.7 Commitment Scheme

The generic framework of [18] already gives commitment schemes once the required group-action components are available. Therefore, in this paper we focus on constructing these components in the module-LIP setting. We recall only the basic commitment syntax in the main text and move the stronger functional variants to the Appendix A.

Definition 2.6 *A commitment scheme Π_{com} consists of three PPT algorithms* (**Gen**, **Com**, **Open**) *parametrized by a security parameter λ and defined over message space $\mathcal{M}$, commitment space C, and opening space D.*

- ***Gen**: The algorithm* $\mathbf{Gen}(1^\lambda)$ *outputs the public commitment key Ck.*
- ***Com**: For any message $m \in \mathcal{M}$,* $\mathbf{Com}_{Ck}(m)$ *produces a tuple $(c, d) \in C \times D$, where c is the commitment value and d is the corresponding opening value.*
- ***Open**: The algorithm* $\mathbf{Open}_{Ck}(c, d)$ *returns the message $\tilde{m} \in \mathcal{M} \cup \{\perp\}$, with $\perp$ indicating an invalid commitment.*

The correctness of a commitment scheme requires that for any message $m \in \mathcal{M}$, $\mathbf{Open}_{Ck}(\mathbf{Com}_{Ck}(m)) = m$. The security of a commitment scheme is based on two properties: *hiding* and *binding*, which are defined as follows.

- **Hiding.** Given a commitment value c, it is computationally infeasible for an adversary $\mathcal{A}$ to determine any partial information about the committed message m. Formally, for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that:

$$\Pr\left[b = \tilde{b} \,\middle|\, \begin{array}{c} Ck \leftarrow \mathbf{Gen}(1^\lambda), (m_0, m_1, \alpha) \leftarrow \mathcal{A}_1(Ck) \\ b \xleftarrow{\$} \{0,1\}, (c, d) \leftarrow \mathbf{Com}_{Ck}(m_b), \tilde{b} \leftarrow \mathcal{A}_2(c; \alpha) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

- **Binding.** It is computationally infeasible for an adversary $\mathcal{A}$ to produce a collision, i.e., a triple (c, d, d') where (c, d) and (c, d') are valid commitments for distinct messages $m \neq m'$. Formally, for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that:

$$\Pr\left[\begin{array}{c} m \neq m', \\ m, m' \neq \perp \end{array} \,\middle|\, \begin{array}{c} Ck \leftarrow \mathbf{Gen}(1^\lambda), (c, d, d') \leftarrow \mathcal{A}(Ck) \\ m \leftarrow \mathbf{Open}_{Ck}(c, d), m' \leftarrow \mathbf{Open}_{Ck}(c, d') \end{array}\right] \leq \mathsf{negl}(\lambda).$$

2.8 Group action

In this subsection, we introduce some definitions related to the group action that are used in [18] to construct the commitment schemes.[1]

Definition 2.7 (Group Action) *Let G be a group with identity element e, and let X be a set. A group action of G on X is an operator $\star : G \times \mathsf{X} \to \mathsf{X}$ that satisfies the following conditions for all $g, h \in G$ and $x \in \mathsf{X}$:*

- *The identity element e of G acts trivially on X, i.e., $e \star x = x$.*
- *The group action is compatible with the group operation, i.e., $g \star (h \star x) = (gh) \star x$.*

We denote such a group action by $(G, \mathsf{X}, \star)$.

[1] Our definitions differ slightly from those in [18]. Specifically, we let X be a single-orbit set and consider a distribution $\mathcal{D}_{\mathsf{X}}$ over it, rather than $\mathcal{D}_{G,\mathsf{X}}$. Our distribution $\mathcal{D}_{\mathsf{X}}$ is induced by first sampling (g, x) from $\mathcal{D}_{G,\mathsf{X}}$ and then generating $g \star x$.

For a group action $(G, \mathsf{X}, \star)$, several important concepts are defined as follows:

- The *orbit* of an element $x \in \mathsf{X}$, denoted by $\mathcal{O}(x)$, is the set of all elements in X that can be obtained by acting on x with elements of G. Formally, $\mathcal{O}(x) \coloneqq \{g \star x : g \in G\}$. In this paper, we assume X is a single orbit under the group action for simplicity.
- The *stabilizer* of an element $x \in \mathsf{X}$, denoted by $\mathrm{Stab}(x)$, is the subgroup of G consisting of all elements that act trivially on x. Formally, $\mathrm{Stab}(x) \coloneqq \{g \in G : g \star x = x\}$. In this paper, we only consider the case where the stabilizer is finite and even polynomial of parameter.
- The set of elements in G that map x to another element $y \in \mathsf{X}$ is denoted by $\mathcal{I}(x, y)$. Formally, $\mathcal{I}(x, y) \coloneqq \{g \in G : g \star x = y\}$. It is worth noting that for any $g \in \mathcal{I}(x, y)$, the set $\mathcal{I}(x, y)$ can be expressed as the coset $g \cdot \mathrm{Stab}(x)$.
- We assume a group action $(G, \mathsf{X}, \star)$ with an associated efficiently samplable distribution $\mathcal{D}_{\mathsf{X}}$ on X, which we sometimes denote as $(G, \mathsf{X}, \star, \mathcal{D}_{\mathsf{X}})$ for simplicity.

One of the core hard problems in the construction framework is the *search Group Action Stabilizer Problem* (s-GASP), which involves finding a non-trivial element in a stabilizer. Specifically, given an element $x \in \mathsf{X}$ such that $\mathrm{Stab}(x) \neq \{e\}$, the objective is to find an element $h \in G$ such that $x = h \star x$ and $h \neq e$. The formal definition of this problem is provided below.

Definition 2.8 (s-GASP) *Let $\mathcal{F}$ be a family of group actions such that for a security parameter λ, $\mathcal{F}(1^\lambda)$ returns a group action $(G, \mathsf{X}, \star)$ with distribution $\mathcal{D}_{\mathsf{X}}$ over X. The s-GASP assumption requires that for all PPT adversaries $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Pr[\mathcal{A}(x) \star x = x, \mathcal{A}(x) \neq e \mid x \leftarrow \mathcal{D}_{\mathsf{X}}, \mathit{Stab}(x) \neq \{e\}] \leq \mathsf{negl}(\lambda).$$

Another important challenging problem is the *decisional Group Action Orbit Problem* (d-GAOP), which requires distinguishing between two classes of orbits. This problem can be viewed as a generalization of the decisional Group Action Inversion Problem (d-GAIP) [15].

Definition 2.9 (d-GAOP) *Let $\mathcal{F}$ be a family of group actions such that for a security parameter λ, $\mathcal{F}(1^\lambda)$ returns group actions $(G, \mathsf{X}, \star)$ and $(G, \mathsf{X}', \star)$ with distributions $\mathcal{D}_{\mathsf{X}}$ over X and $\mathcal{D}_{\mathsf{X}'}$ over X'. The d-GAOP assumption requires that for all PPT adversaries $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\lambda)$ such that*

$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{array}{c} y_0 \leftarrow \mathcal{D}_{\mathsf{X}}, y_1 \leftarrow \mathcal{D}_{\mathsf{X}'} \\ b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(y_b) \end{array}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

We now introduce the definition of a **weak re-randomized algorithm** in addition to the original re-randomized algorithm. The only difference between them is that we relax X to $\mathrm{Supp}(\mathcal{D}_{\mathsf{X}})$. A standard re-randomized algorithm immediately implies a weak re-randomized algorithm. The weak notion is useful

for the concrete construction below because it allows the input to remain in the sampled support and avoids the exponential Gaussian variance that would arise from demanding the same output-size bound on all of X. The standard notion still gives a worst-case to average-case self-reduction, which cannot be guaranteed by the weak notion.

Definition 2.10 (Re-Randomized Algorithm) *For a group action $(G, \mathsf{X}, \star, \mathcal{D}_{\mathsf{X}})$, a (resp. weak) re-randomized algorithm R takes as input $x \in \mathsf{X}$ and outputs a pair $(g \star x, g) \in \mathsf{X} \times G$ according to a distribution, denoted as $\mathsf{R}(x)$, such that for any $x, x' \in \mathsf{X}$ (resp. $x, x' \in \mathrm{Supp}(\mathcal{D}_{\mathsf{X}})$), and $(x'', g) \leftarrow \mathsf{R}(x)$*

- *the marginal distributions of the first variable are identical for $\mathsf{R}(x)$ and $\mathsf{R}(x')$,*
- *g is uniformly distributed on $\mathcal{I}(x, x'')$.*

We note that there exists a sufficient condition for a better use of the re-randomized algorithm. Its proof is provided in the Appendix C.

Lemma 2.7 *For a group action $(G, \mathsf{X}, \star, \mathcal{D}_{\mathsf{X}})$, an algorithm R is a (resp. weak) re-randomized algorithm if it takes as input $x \in \mathsf{X}$ and outputs a pair $(g \star x, g) \in \mathsf{X} \times G$ according to a distribution, denoted as $\mathsf{R}(x)$, such that:*

- *For any $x, x' \in \mathsf{X}$ (resp. $x, x' \in \mathrm{Supp}(\mathcal{D}_{\mathsf{X}})$) and $h \in \mathcal{I}(x, x')$, the distributions of $\mathsf{R}(h \star x)[2]$ and $\mathsf{R}(x)[2] \cdot h^{-1}$ are identical.*

Another key component is the Extractor E, whose function is to extract the hard core from the isomorphisms $\mathrm{Iso}(x, x'')$ between two elements. Combined with the aforementioned Re-Randomized Algorithm, we obtain a paradigm for constructing commitment schemes as in Protocol 2.1.

At the level of the basic protocol, a group-action commitment can be viewed as a one-time pad whose pad is extracted from a hidden transition in the group action. The public key is an element $y \in \mathsf{X}$. To commit to a message $m \in M$, the committer samples a transition $g \in G$ and a randomized representative $y' = g \star y$, and publishes

$$c = (\mathsf{E}(g) \cdot m,\ y')$$

while keeping g as the opening. Verification checks that $g \star y = y'$ and then unmasks the message using $\mathsf{E}(g)^{-1}$.

Thus hiding requires that, even after y' is fixed, the value $\mathsf{E}(g)$ is uniform in the message group. This is exactly the role of the re-randomization algorithm together with the uniformly extracting property. Binding, on the other hand, follows because two different openings to the same y' would give a non-trivial element of the stabilizer of y, which is precisely the stabilizer problem associated with the group action. Additional extractor properties then provide the stronger functional variants used later.

Commitment Scheme 2.1 : $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ *for a security parameter* λ *and message space* M*.*

- $\mathbf{Gen}(1^\lambda)$: Sample $y \leftarrow \mathcal{D}_{\mathsf{X}}$(resp. $\mathcal{D}_{\mathsf{X}'}$) and output $Ck = y$.
- $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ such that $g \star y = y'$. Compute $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y')$ and set $d = g$. Output (c, d).
- $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d)^{-1} \cdot c_1$; otherwise, output $\perp$.

Definition 2.11 (Extractor) *Let* $(G, \mathsf{X}, \star)$ *and* $(G, \mathsf{X}', \star)$ *be group actions, and let* M *be an abelian group. An extractor is a deterministic and efficient algorithm* $\mathsf{E} : G \to M$*, where* $|M| \geq 2$*. It may optionally possess the following properties:*

- *Uniformly extracting property (on* X*): for any* $y', y'' \in \mathsf{X}$*, it holds that* $\mathsf{E}(g)$ *is uniformly distributed on* M *for* $g \leftarrow_\$ \mathcal{I}(y', y'')$*.*
- *Homomorphic property:* $\mathsf{E} : G \to M$ *is a surjective group homomorphism, i.e., for any* $g_0, g_1 \in G$*,* $\mathsf{E}(g_0) \cdot \mathsf{E}(g_1)^{-1} = \mathsf{E}(g_0 \cdot g_1^{-1})$*.*
- *Local constant property (on* X'*): for any* $y', y'' \in \mathsf{X}'$*, it holds that* $\mathsf{E}(g)$ *is a constant for* $g \in \mathcal{I}(y', y'')$*.*
- *Trapdoor property (on* X*): there exists a PPT algorithm* F *that takes*

$$(u, g, x, Stab(x)) \in M \times G \times \mathsf{X} \times \mathcal{P}(G)$$

as input and outputs $g' \in G$*, such that: for any* $u \in M$*,* $y', y'' \in \mathsf{X}$*,* $g \in \mathcal{I}(y', y'')$*, the output* g' *of* $\mathsf{F}(u, g, y', Stab(y'))$ *satisfies* $g' \in \mathcal{I}(y', y'')$ *and* $\mathsf{E}(g') = u$*; the distributions of* $(\mathsf{E}(g), g)$ *and* (u, g') *are identical for any* $y', y'' \in \mathsf{X}$*, where* $g \leftarrow_\$ \mathcal{I}(y', y'')$*,* $u \leftarrow_\$ M$*,* $g' \leftarrow \mathsf{F}(u, g, y', Stab(y'))$*.*

Remark 5. The formulation differs slightly from [18, Definition 3.2] only in presentation. In [18], the extractor properties needed for different commitment functionalities are introduced across several theorems. Since Theorem 2.2 below summarizes these construction results in one statement, we collect here the uniformly extracting, homomorphic, local constant, and trapdoor properties in a single definition.

Indeed, the uniformly extracting property and homomorphic property imply the Trapdoor property.

Lemma 2.8 *For the group action* $(G, \mathsf{X}, \star)$*, if an extractor* E *satisfies the uniformly extracting property (on* X*) and the homomorphic property, then it has the Trapdoor property. Assume moreover that for every* $y \in \mathsf{X}$*, the stabilizer* $\mathrm{Stab}(y)$ *has polynomial size and can be enumerated efficiently.*

Proof. It suffices to construct F as follows: on input $(u, g, y', \mathrm{Stab}(y'))$, first enumerate $\mathrm{Stab}(y')$, then sample uniformly at random an element h from the fiber $\{h \in \mathrm{Stab}(y') \mid \mathsf{E}(h) = \mathsf{E}(g)^{-1}u\}$, and output $g' = gh$. By the additional assumption on stabilizers, F is a PPT algorithm. This satisfies $\mathsf{E}(g') = u$.

When u is uniformly distributed over M, $\mathsf{E}(g)^{-1}u$ is also uniformly distributed over M. Moreover, since $\mathsf{E}(U(\mathcal{I}(y', y'')))$ is uniform over M, the sampled g' is uniformly distributed over $\mathcal{I}(y', y'')$. Consequently, the distributions of $(\mathsf{E}(g), g)$ and (u, g') are identical. □

For orientation, the generic assumptions and components in the following theorem will later be instantiated by module-LIP objects as follows. This table is only a dictionary for the reader; the precise group action, sampler, and extractor are constructed and verified in Section 4.

Generic object in [18]	Module-LIP instantiation	Cryptographic role
$G \curvearrowright \mathsf{X}$	$\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L})$ acting on rank-2 Hermitian forms	commitment key and orbit space
s-GASP	module-LAP	binding / non-trivial stabilizer search
d-GAOP	module-ΔLIP	hiding between two orbit families
re-randomizer R	HAWK-style orbit sampler plus determinant randomization	conditional uniformity of openings
extractor E	determinant class modulo $\langle X^2 \rangle$	message mask

Theorem 2.2 ([18, Theorem 3.1,3.2,4.1,4.2]) *Suppose that the group action $(G, \mathsf{X}, \star, \mathcal{D}_\mathsf{X})$ satisfies the s-GASP assumption, $(G, \mathsf{X}, \star, \mathcal{D}_\mathsf{X})$ and $(G, \mathsf{X}', \star, \mathcal{D}_{\mathsf{X}'})$ satisfy the d-GAOP assumption, R is a weak re-randomized algorithm, and E is an extractor. Then*

- *If E has the uniformly extracting property on X, then Protocol 2.1 for X is a perfectly hiding and computationally binding commitment scheme based on the s-GASP assumption over $(G, \mathsf{X}, \star, \mathcal{D}_\mathsf{X})$. We refer to this as the* basic scheme.
- *If E has the uniformly extracting property on X and local constant property on X', then Protocol 2.1 for X' is a computationally hiding commitment scheme based on the d-GAOP assumption over $(G, \mathsf{X}, \star, \mathcal{D}_\mathsf{X})$ and $(G, \mathsf{X}', \star, \mathcal{D}_{\mathsf{X}'})$, and it is perfectly binding. We refer to this as the* dual scheme.
- *If E has the uniformly extracting property and homomorphic property on X, then Protocol A.1 is an enhanced linkable commitment.*
- *Assume that for some representative element x in X, its stabilizer is known, and the distribution $\mathcal{D}_\mathsf{X}$ is generated by sampling an element g from G and outputting $g \star x$. If E has the uniformly extracting property and trapdoor property on X and local constant property on X', then Protocol A.2 is a dual mode commitment.*

3 Module-LIP and Module-LAP on Certain Modules

In this section and the next, unless otherwise specified, we follow the setup in HAWK by setting $\mathbb{L} = \mathbb{Q}[X]/(\Phi_{2^d}(X)) = \mathbb{Q}[X]/(X^{2^{d-1}} + 1)$, with $n = \deg(\mathbb{L})$.

3.1 Aut of $\mathbf{Q}_k^{(0)}$ and $\widetilde{\mathbf{Q}}_k^{(1)}$

Due to space constraints, the proofs in this subsection are provided in the Appendix C.

Lemma 3.1 *For $k > 0$, both*

$$Aut_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q}_k^{(0)}) \quad \text{and} \quad Aut_{\mathcal{O}_{\mathbb{L}}}(\widetilde{\mathbf{Q}}_k^{(0)})$$

are equal to

$$\left\{ \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} : \xi_1, \xi_2 \in \mu(\mathbb{L}) \right\}.$$

Lemma 3.2 *For $k > 2$, let*

$$M = \begin{pmatrix} 1 & 1+X^2 \\ 0 & -1 \end{pmatrix}.$$

Then

$$Aut_{\mathcal{O}_{\mathbb{L}}}(\widetilde{\mathbf{Q}}_k^{(1)}) = M^{-1} \left\{ \begin{pmatrix} \xi_1 & 0 \\ \frac{\xi_2 - \xi_1}{1+X^2} & \xi_2 \end{pmatrix} : \xi_1, \xi_2 \in \mu(\mathbb{L}),\ \xi_1/\xi_2 = X^{2j},\ j \in \mathbb{Z} \right\} M.$$

3.2 LIP to LAP reduction on $[\mathbf{Q}_k^{(0)}]$

Theorem 3.1 *There is a reduction from module-LIP$_{[\mathbf{Q}_k^{(0)}]}$ to module-LAP$_{[\mathbf{Q}_k^{(0)}]}$. In particular, our reduction preserves the problem instance, i.e., for any $\mathbf{Q} \in [\mathbf{Q}_k^{(0)}]$ we can reduce module-LIP$_{\mathbf{Q},\mathbf{Q}_k^{(0)}}$ to module-LAP$_{\mathbf{Q}}$.*

For $\mathbf{Q} = U^* \mathbf{Q}_k^{(0)} U$, any module automorphism can be written as $U^{-1}PU$ for some $P \in Aut_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q}_k^{(0)})$. By Lemma 2.3 and Lemma 3.1, the possible candidates for P are known and polynomially many. Therefore, by enumeration, we can assume that, given the input module lattice automorphism $U^{-1}PU$, we have guessed P.

The technique used in the reduction can be viewed as an adaptation of the approach of [26] to our setting. Following the same guiding idea, we replace the pair $(J_2, (BU)^{-1}J_2(BU))$ with $(P, U^{-1}PU)$, so that the techniques from [26] can be applied. A sketch of this step can be found in Algorithm 1. The main subtlety is that the automorphisms defined here and the pseudo-automorphisms considered in [26] do not directly convert into one another, and therefore require some additional adjustments.

By Lemma 2.3 and Lemma 3.1, the structure of P can be classified into two types: diagonal and anti-diagonal. When P is a diagonal matrix, this operation can be easily accomplished. The process is described in the following proposition.

Proposition 3.1 *Assume $U \in GL_2(\mathcal{O}_{\mathbb{L}})$. Given $\mathbf{Q} = U^* \mathbf{Q}_k^{(0)} U$, $\xi_1 \neq \xi_2 \in \mu(\mathbb{L})$, and $P_U := U^{-1}PU$ where $P := \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix}$, we can find U in polynomial time of the size of input.*

Proof. Due to space constraints, its proof is provided in the Appendix C.

Algorithm 1: FindU(Informal)

Require: $P \in Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q_k^{(0)}}) \setminus \mu(\mathbb{L})I_2$; $U^{-1}PU$ and $\mathbf{Q} = U^*\mathbf{Q_k^{(0)}}U$ for which $U \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L})$

Ensure: A congruence matrix U between $\mathbf{Q_k^{(0)}}$ and $\mathbf{Q}$

1: $\lambda \leftarrow$ one of the eigenvalues of P
2: $M \leftarrow \ker(P - \lambda I_2) \cap \mathcal{O}_\mathbb{L}^2$, $U^{-1}M \leftarrow \ker(U^{-1}PU - \lambda I_2) \cap \mathcal{O}_\mathbb{L}^2$
3: Write $B_k^{(0)}M$ as $\mathcal{I} \cdot v$
4: $(B_k^{(0)}U)^{-1}(\mathcal{O}_\mathbb{L} \cdot v) \leftarrow \mathcal{I}^{-1} \cdot U^{-1}M$
5: $\mu(\mathbb{L}) \cdot (B_k^{(0)}U)^{-1}v \leftarrow$
6: LS2$(\mathcal{O}_\mathbb{L}, v^*v, U^*\mathbf{Q_k^{(0)}}U, (B_k^{(0)}U)^{-1}(\mathcal{O}_\mathbb{L} \cdot v))$
7: $B_k^{(0)}U \leftarrow (v|t_*(v))((B_k^{(0)}U)^{-1}v|(B_k^{(0)}U)^{-1}t_*(v))^{-1}$
8: **return** $(B_k^{(0)})^{-1}(B_k^{(0)}U)$.

However, when $k = 0$, P might be an anti-diagonal matrix, and consequently, there might be no eigenvalues of P in $\mathbb{L}$. In order to use the eigenspaces to decompose the $\mathcal{O}_\mathbb{L}^2$, it is necessary to extend the field L by adjoining the eigenvalues of P which is the square root of a root of unity in L. In this case, we must also prove that the extended field remains a CM field and that an integral basis of its ring of integers can still be efficiently computed. When $\mathbb{L}$ is a cyclotomic field, this follows easily from the structure of cyclotomic fields since the extended field is also a cyclotomic field with bounded degree.

Proposition 3.2 *Assume $U \in GL_2(\mathcal{O}_\mathbb{L})$. Given U^*U, $\xi_1, \xi_2 \in \mu(\mathbb{L})$, and $P_U := U^{-1}PU$ where $P := \begin{pmatrix} 0 & \xi_1 \\ \xi_2 & 0 \end{pmatrix}$, we can find U in polynomial time of the size of input.*

Proof. Due to space constraints, its proof is provided in the Appendix C.

Proof (of Theorem 3.1). Given $\mathbf{Q} = U^*\mathbf{Q_k^{(0)}}U$ for some $U \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L})$, and a non-trivial automorphism $P_U \in Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q})$, we know P_U has the form $U^{-1}PU$ for some $P \in Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q_k^{(0)}}) \setminus \mu(\mathbb{L})I_2$.

Iterate over $\widetilde{P} \in \mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)}) \setminus \mu(\mathbb{L})I_2$. By Lemma 3.1, when $k > 0$,

$$\mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)}) = \left\{ \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L}) \right\},$$

and when $k = 0$, $\mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)}) = \mathcal{U}_2(\mathcal{O}_\mathbb{L})$. If $\widetilde{P}$ is a diagonal matrix, input $\mathbf{Q}$, P_U, and $\widetilde{P}$ into Proposition 3.1; if $\widetilde{P}$ is an anti-diagonal matrix (which implies $k = 0$), input them into Proposition 3.2, and verify whether the inverse of the output matrix is an isomorphism from $\mathbf{Q}$ to $\mathbf{Q}_k^{(0)}$.

Correctness: When $P = \widetilde{P}$, by Proposition 3.1, Proposition 3.2, it will output the right U. U^{-1} is an isomorphism from $\mathbf{Q}$ to $\mathbf{Q}_k^{(0)}$.

Complexity: The number of iterations is less than the cardinality of $\mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)})$, which is bounded by $O(n^4)$. Each loop runs in polynomial time by Proposition 3.1 and Proposition 3.2.
Therefore, the overall time complexity is polynomial. □

When $\mathbb{L}$ is a general CM field, we also have a reduction from $\mathcal{O}_\mathbb{L}^2$-LIP to $\mathcal{O}_\mathbb{L}^2$-LAP. This is not immediately obvious, precisely due to the reason mentioned earlier: when P is an anti-diagonal matrix, we need to consider the structure of the extended field. Due to space constraints, we have placed the proofs in the Appendix B.

Theorem 3.2 *Let $\mathbb{L}$ be a CM number field. Given the parameter $\mathbf{L}$, there is a reduction from $\mathcal{O}_\mathbb{L}^2$-LIP to $\mathcal{O}_\mathbb{L}^2$-LAP. In particular, our reduction preserves the instance of the problems i.e. for any $\mathbf{Q} \in [I_2]$ we can reduce module-LIP$_{\mathbf{Q},I_2}$ to module-LAP$_\mathbf{Q}$.*

3.3 Hardness of module-LIP$_{[\mathbf{Q}_k^{(0)}]}$ for $k = O(\log n)$

In the previous subsection, we proved that module-LAP$_{[\mathbf{Q}_k^{(0)}]} \geq$ module-LIP$_{[\mathbf{Q}_k^{(0)}]}$. In this subsection, we further prove that module-LIP$_{[\mathbf{Q}_k^{(0)}]} =$ module-LIP$_{[\mathbf{I}_2]}$, i.e., $\mathcal{O}_\mathbb{L}^2$-LIP, for $k = O(\log n)$.

Theorem 3.3 *For $k = O(\log n)$, there is a (quantum) reduction from $\mathcal{O}_\mathbb{L}^2$-LIP to module-LIP$_{[\mathbf{Q}_k^{(0)}]}$.*

Due to space constraints, we have placed the proofs in the Appendix C.

Remark 6. The use of quantum computation arises in the step when showing that module-LIP$_{[\mathbf{Q}_k^{(0)}]} \geq$ module-LIP$_{[\mathbf{Q}_\ell^{(0)}]}$, fundamentally because we cannot compute $\mathcal{O}_\mathbb{L}\begin{pmatrix}1\\0\end{pmatrix} + \mathcal{O}_\mathbb{L}\begin{pmatrix}0\\\Pi^{k+1}\end{pmatrix}$ from $\mathcal{O}_\mathbb{L}\begin{pmatrix}\Pi\\0\end{pmatrix} + \mathcal{O}_\mathbb{L}\begin{pmatrix}0\\\Pi^{k+1}\end{pmatrix}$ while preserving the free basis representation. This requires invoking Lemma 2.6. Therefore, if we only require the reduction for the geometric version of module-LIP, the quantum step can be omitted.

Remark 7. Note that for any $\mathrm{poly}(n) \geq \ell \geq k$, the reduction module-LIP$_{[\mathbf{Q}_k^{(0)}]} \geq$ module-LIP$_{[\mathbf{Q}_\ell^{(0)}]}$ implies that $\mathcal{O}_\mathbb{L}^2$-LIP $\geq$ module-LIP$_{[\mathbf{Q}_k^{(0)}]}$. Combined with the above theorem, we have module-LIP$_{[\mathbf{Q}_k^{(0)}]} = \mathcal{O}_\mathbb{L}^2$-LIP, for $k = O(\log n)$.

3.4 Hardness of module-ΔLIP$_{\mathbb{L},[\tilde{\mathbf{Q}}_k^{(0)}],[\tilde{\mathbf{Q}}_k^{(1)}]}$

Lemma 3.3 *For $\mathbb{L} = \mathbb{Q}[X]/(X^{2^{d-1}}+1)$, let $\mathbb{K} = \mathbb{Q}(X^2) \subseteq \mathbb{L}$. Here $\Delta_{k,\mathbb{K}}$ denotes the decision module-ΔLIP family from Theorem 2.1, instantiated over $\mathbb{K}$. Then module-ΔLIP$_{\mathbb{L},[\tilde{\mathbf{Q}}_k^{(0)}],[\tilde{\mathbf{Q}}_k^{(1)}]}$ is at least as hard as $\Delta_{k,\mathbb{K}}$.*

Proof. Due to space constraints, its proof is provided in the Appendix C.

Theorem 2.1 reduces $\mathcal{O}_{\mathbb{K}}^2$-LIP to $\{\Delta_{k,\mathbb{K}}, k = 2, \cdots, n^2/2\}$. Combined with the lemma above, we can reduce $\mathcal{O}_{\mathbb{K}}^2$-LIP to $\{\text{module-}\Delta\text{LIP}_{\mathbb{L},[\widetilde{\mathbf{Q}}_k^{(0)}],[\widetilde{\mathbf{Q}}_k^{(1)}]}, k = 2, \cdots, n^2/2\}$.

For any cyclotomic field $\mathbb{F}$ of power-of-two degree, $\mathcal{O}_{\mathbb{F}}^2$-LIP can be viewed as a problem parameterized by $\deg(\mathbb{F})$. From a practical solving perspective, for any $k > 1$, $\{\widetilde{\mathbf{Q}}_k^{(0)}, \widetilde{\mathbf{Q}}_k^{(1)}\}$ lie in the same genus and even in the same special genus, and there are currently no known algorithms for solving them other than via SVP. From a hardness perspective, if the problem parameterized by n is considered hard, then the problem parameterized by $n/2$ can still be regarded as hard. Therefore, under the assumption that $\mathcal{O}_{\mathbb{F}}^2$-LIP is hard, $\{\text{module-}\Delta\text{LIP}_{\mathbb{L},[\widetilde{\mathbf{Q}}_k^{(0)}],[\widetilde{\mathbf{Q}}_k^{(1)}]}$, $k = 2, \cdots, n^2/2\}$ provides potential hard instances of module-ΔLIP. Although we lack theoretical results to definitively establish which problems achieve the hardness of $\mathcal{O}_{\mathbb{F}}^2$-LIP, based on the reduction for the search version, we heuristically argue that problems corresponding to smaller values of k are more representative of the problem's inherent hardness.

4 Commitment Schemes Based on Module-LIP

4.1 Choice of instances

Group Action Based on Module Lattice. A naive idea for constructing a structured version in [18] is to replace $\mathrm{GL}_n(\mathbb{Z})$ with $\mathrm{GL}_2(\mathcal{O}_{\mathbb{L}})$. However, when constructing the re-randomized algorithm, obstacles arise because, unlike general lattices, where efficient algorithms such as LLL can be used to find an integral basis, there is currently no efficient classical algorithm for finding an $\mathcal{O}_{\mathbb{L}}$-basis for module lattices. This issue is also reflected in the self-reduction of the module lattice isomorphism problem. To bypass this difficulty, [12] considers $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$. If we adopt their strategy, we face another obstacle when constructing deterministic extractors, since we can no longer use det as an extractor as in [18]. To meet both requirements simultaneously, we consider such a subgroup.

Definition 4.1 *We define the Generalized Special Linear group over $\mathcal{O}_{\mathbb{L}}$ as $GSL_2(\mathcal{O}_{\mathbb{L}}) := \{M \in GL_2(\mathcal{O}_{\mathbb{L}}) | \det(M) \in \mu(\mathbb{L})\}$ and the projective group of it is $PGSL_2(\mathcal{O}_{\mathbb{L}}) := GSL_2(\mathcal{O}_{\mathbb{L}})/\mu(\mathbb{L})I_2$.*

Let $\mathbf{Q} \in \mathcal{H}_n^{>0}(\mathcal{O}_{\mathbb{L}})$ be a positive definite Hermitian form over $\mathcal{O}_{\mathbb{L}}$, and let $[\mathbf{Q}]$ denote the set of Hermitian forms equivalent to $\mathbf{Q}$, i.e., $[\mathbf{Q}]_{\mathrm{sl}} := \{\mathbf{V}^*\mathbf{Q}\mathbf{V} : \mathbf{V} \in \mathrm{SL}_n(\mathcal{O}_{\mathbb{L}})\}$, $[\mathbf{Q}]_{\mathrm{gsl}} := \{\mathbf{V}^*\mathbf{Q}\mathbf{V} : \mathbf{V} \in \mathrm{GSL}_n(\mathcal{O}_{\mathbb{L}})\}$. We can define the group action

$(\mathrm{PGSL}_n(\mathcal{O}_\mathbb{L}), [\mathbf{Q}]_{\mathrm{gsl}}, \star)$ as [2]:

$$\mathbf{Q}' \star \mathbf{V} = \mathbf{V}^* \mathbf{Q}' \mathbf{V} \quad \text{for any } \mathbf{V} \in \mathrm{PGSL}_n(\mathcal{O}_\mathbb{L}), \mathbf{Q}' \in [\mathbf{Q}]_{\mathrm{gsl}}. \tag{1}$$

This group action is closely related to free module lattice isomorphisms and automorphisms. Note that any automorphism of a rank-2 Hermitian form over $\mathcal{O}_\mathbb{L}$ is contained in $\mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$. The purpose of considering the projective group is to exclude trivial automorphisms of the form $\mu(\mathbb{L})I_n$ from the stabilizer of the group action. Consequently, the s-GASP for this group action on $[\mathbf{Q}]_{\mathrm{gsl}}$ is equivalent to module-LAP. Similarly, the d-GAOP for $[\mathbf{Q}_1]_{\mathrm{gsl}}$ and $[\mathbf{Q}_2]_{\mathrm{gsl}}$ is equivalent to module-ΔLIP$_{\mathcal{D}_{[\mathbf{Q}_1]},\mathcal{D}_{[\mathbf{Q}_2]}}$, where the conversion between isomorphism classes can be handled analogously to Remark 2.

Here, we focus on the case of rank-2 Hermitian forms. In the notation of Theorem 2.2, for the enhanced linkable commitment below we take

$$\mathsf{X} = [\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}]_{\mathrm{gsl}}.$$

For the dual-mode commitment, we take

$$\mathsf{X} = [\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}, \qquad \mathsf{X}' = [\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}.$$

The form $\widetilde{\mathbf{Q}}_k^{(0)}$ lies in the same $\mathrm{GL}_2(\mathcal{O}_\mathbb{L})$-class as $\mathbf{Q}_{2k}^{(0)}$, so they have the same hardness for module-LIP and module-LAP. However, their determinants differ, and hence their SL- or GSL-classes need not agree; this distinction matters for the decision problem, so we keep the tilde pair separate. The subsequent sections verify the required re-randomization and extractor properties for these concrete orbit sets. We assume that the distribution $\mathcal{D}_\mathsf{X}$ (resp. $\mathcal{D}_{\mathsf{X}'}$) is generated by sampling an element g from $\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L})$ and outputting $g \star \mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}$ for the enhanced linkable case, $g \star \widetilde{\mathbf{Q}}_k^{(0)}$ for the dual-mode X case, and $g \star \widetilde{\mathbf{Q}}_k^{(1)}$ for X'.

4.2 Re-randomized algorithm for $(\mathbf{PGSL_2}(\mathcal{O}_\mathbb{L}), [\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}], \star)$

As mentioned earlier, the main difficulty in structuring the instantiation in [18] lies in constructing a suitable re-randomized algorithm. We make adaptations to the construction in [12] for self-reduction to obtain the required re-randomization algorithm for $(\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L}), [\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}]_{\mathrm{gsl}}, \star)$. Note that $\mathbf{Q}_k^{(0)}$ is a diagonal matrix, for which we have $[\mathbf{Q}_k^{(0)}]_{\mathrm{gsl}} = [\mathbf{Q}_k^{(0)}]_{\mathrm{sl}} \triangleq [\mathbf{Q}_k^{(0)}]$.

We rewrite [12, Section 6, Lemmas 4,5] to better highlight the algorithmic properties involved, and provide the proofs in the Appendix C.

[2] The choice of representative in $\mathrm{GSL}_n(\mathcal{O}_\mathbb{L})/\mu(\mathbb{L})I_n$ does not matter for this group action since $\mathbf{V}^*\mathbf{Q}\mathbf{V} = (\xi\mathbf{V})^*\mathbf{Q}(\xi\mathbf{V})$ for any $\xi \in \mu(\mathbb{L})$. Here, in order to match the standard notation for isomorphisms of Hermitian forms and to simplify the presentation, we use a right-action notation for this action. This differs from the left-action convention introduced in Section 2, but the two conventions are equivalent after passing to inverses.

Lemma 4.1 (Adapted [12, Section 6 Lemma 4]) *Assume* $\mathbf{Q_0} \in \mathcal{H}_2^{>0}(\mathbb{L})$ *is diagonal and* $\mathsf{S} \subseteq [\mathbf{Q_0}]$*, and let* $T = \sup\{\|\widetilde{B_{\mathrm{rot}(\mathbf{Q})}}\|/\pi \cdot \sqrt{\log(2n+4)} : \mathbf{Q} \in \mathsf{S}\}$*, where* T *may be* ∞*. Under Heuristic 1 for* $\mathbf{Q_0}$*, for* $\sigma \geq \min\{2^{\Theta(n)}\lambda_{2n}(\mathbf{Q_0}), T\}$*, there exists an efficient randomized algorithm* $\mathsf{R}_0^{\mathsf{S}}$ *that takes any* $\mathbf{Q} \in \mathsf{S}$ *as input and outputs* $(\mathbf{R}, \mathbf{U})$ *such that* $(\mathbf{R} = \mathbf{U}^*\mathbf{Q}\mathbf{U}, \mathbf{U}) \in [\mathbf{Q}]_{sl} \times SL_2(\mathcal{O}_\mathbb{L})$*. We omit the superscript* S *whenever it does not cause ambiguity. The distribution of* $\mathsf{R}_0(\mathbf{Q})[2]$ *is given by:*

- *For any* $\mathbf{U} = (u_1|u_2) \in SL_2(\mathcal{O}_\mathbb{L})$*,*

$$\Pr[\mathsf{R}_0(\mathbf{Q})[2] = \mathbf{U}] = \begin{cases} \rho_{\mathbf{Q},\sigma}(u_1)/\rho_{\mathbf{Q_0},\sigma}((\mathcal{O}_\mathbb{L}^2)^*), & \text{if } u_2 \text{ is reduced with respect} \\ & \text{to } u_1 \text{ and } \mathbf{Q}; \\ 0, \text{ otherwise,} & \end{cases}$$

with the following property:

- *For any* $\mathbf{V} \in SL_2(\mathcal{O}_\mathbb{L})$ *s.t.* $\mathbf{V}^*\mathbf{Q}\mathbf{V} \in \mathsf{S}$*, the distribution of* $\mathbf{V}^{-1} \cdot \mathsf{R}_0(\mathbf{Q})[2]$ *is the same as that of* $\mathsf{R}_0(\mathbf{V}^*\mathbf{Q}\mathbf{V})[2]$*.*

Here $\mathtt{Reduce}(\mathbf{Q}, x, y) := y - \left\lceil \frac{x^* \cdot \mathbf{Q} \cdot y}{x^* \cdot \mathbf{Q} \cdot x} \right\rfloor \cdot x$ *and we say* y *is reduced with respect to* x *and* $\mathbf{Q}$ *if* $\left\lceil \frac{x^* \cdot \mathbf{Q} \cdot y}{x^* \cdot \mathbf{Q} \cdot x} \right\rfloor = 0$ *in which* $\lceil \cdot \rfloor : a_0 + a_1 X + \cdots + a_{n-1}X^{n-1} \mapsto \lceil a_0 \rfloor + \lceil a_1 \rfloor X + \cdots + \lceil a_{n-1} \rfloor X^{n-1}$*.*

Algorithm 2: $\mathsf{R}_0^{\mathsf{S}}$(adapted from [12, Algorithm 6]):

Require: Conductor $m = 2^\kappa$ cyclotomic $\mathbb{L}$, $\mathbf{Q} \in \mathsf{S}$, $\sigma \geq \min\{2^{\Theta(n)}\lambda_{2n}(\mathbf{Q_0}), T\}$
Ensure: $\mathbf{R} \in [\mathbf{Q}]_{sl}$ and $\mathbf{U} \in \mathrm{SL}_2(\mathcal{O}_\mathbb{L})$ such that $\mathbf{R} = \mathbf{U}^* \cdot \mathbf{Q} \cdot \mathbf{U}$
1: Parse $u_1 \leftarrow D_{\mathbf{Q},\sigma}$ as $u_1 = (f, g)^T \in \mathcal{O}_\mathbb{L}^2$
2: **if** $\mathtt{HermiteSolve}(\mathbb{L}, u_1)$ returns $\perp$ **then**
3: Restart
4: **else**
5: $\widetilde{u_2} \leftarrow \mathtt{HermiteSolve}(\mathbb{L}, u_1)$
6: **end if**
7: Let $u_2 = (F_\mathbf{Q}, G_\mathbf{Q})^T \leftarrow \mathtt{Reduce}(\mathbf{Q}, u_1, \widetilde{u_2})$
8: Let $\mathbf{U} = (u_1|\ u_2)$ and $\mathbf{R} = \mathbf{U}^* \cdot \mathbf{Q} \cdot \mathbf{U}$
9: **return** $(\mathbf{R}, \mathbf{U})$

There are some observations regarding the randomized algorithm R_0 when considered on a diagonal Hermitian form.

Lemma 4.2 *Assume* $\mathbf{Q_0} \in \mathcal{H}_2^{>0}(\mathbb{L})$ *is diagonal. For any* $\xi \in \mu(\mathbb{L})$*, let* $\mathbf{X} = \begin{pmatrix} 1 & 0 \\ 0 & \xi \end{pmatrix}$ *and* $\mathbf{Y} = \begin{pmatrix} \xi & 0 \\ 0 & \xi^{-1} \end{pmatrix}$*. Then for any* $\mathbf{U} \in SL_2(\mathcal{O}_\mathbb{L})$*, we have*

$$\Pr[\mathsf{R}_0(\mathbf{Q_0})[2] = \mathbf{X^{-1}UX}] = \Pr[\mathsf{R}_0(\mathbf{Q_0})[2] = \mathbf{U}] = \Pr[\mathsf{R}_0(\mathbf{Q_0})[2] = \mathbf{UY}].$$

Proof. Due to space constraints, its proof is provided in the Appendix C.

Observe that every $V \in \mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$ is uniquely written as

$$V = V_0 V_1, \qquad V_0 \in \mathrm{SL}_2(\mathcal{O}_\mathbb{L}), \quad V_1 = \mathrm{diag}(1, \det(V)).$$

A natural construction for a re-randomized algorithm on $(\mathrm{GSL}_2(\mathcal{O}_\mathbb{L}), [Q_0])$ is to sample the part of $\mathrm{SL}_2(\mathcal{O}_\mathbb{L})$ using R_0 and uniformly randomly sample the part of the form $\mathrm{diag}(1, \xi)$. When $\mathbf{Q_0}$ is diagonal, we consider the equivalence class $[Q_0]$, so it can be directly used as the input for R_0. We next prove that such a construction is indeed a re-randomized algorithm that satisfies the requirement.

Lemma 4.3 (Re-Randomized algorithm) *Assume $\mathbf{Q_0} \in \mathcal{H}_2^{>0}(\mathbb{L})$ is diagonal and $\mathcal{D}_{[\mathbf{Q_0}]}$ is a samplable distribution on $[\mathbf{Q_0}]$ such that $\mathbf{Q_0} \in \mathrm{Supp}(\mathcal{D}_{[\mathbf{Q_0}]})$. If $\mathsf{R}_0^{[\mathbf{Q_0}]}$ (resp. $\mathsf{R}_0^{\mathrm{Supp}(\mathcal{D}_{[\mathbf{Q_0}]})}$) is efficient, then the randomized algorithm R described in Algorithm Algorithm 3 is efficient, with the following property:*

- *For any*
$$\mathbf{Q} \in [\mathbf{Q_0}] \text{ (resp. } \mathrm{Supp}(\mathcal{D}_{[\mathbf{Q_0}]})), \mathbf{V} \in GSL_2(\mathcal{O}_\mathbb{L})$$
such that $\mathbf{V}^*\mathbf{Q}\mathbf{V} \in [\mathbf{Q_0}]$ (resp. $\mathrm{Supp}(\mathcal{D}_{[\mathbf{Q_0}]})$), the distribution of $\mathbf{V}^{-1} \cdot \mathsf{R}(\mathbf{Q})[2]$ is the same as that of $\mathsf{R}(\mathbf{V}^*\mathbf{Q}\mathbf{V})[2]$.

The same conclusion holds for the induced group action for $PGSL_2(\mathcal{O}_\mathbb{L})$.

Proof. The efficiency follows from the efficiency of R_0. We only need to prove the properties. In the current 2-power cyclotomic setting, $|\mu(\mathbb{L})| = 2n$; we keep the factor as $|\mu(\mathbb{L})|$ in the calculation below.

Let $\mathbf{V} \in \mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$ and $\mathbf{Q}, \mathbf{V}^*\mathbf{Q}\mathbf{V} \in [\mathbf{Q_0}]$ (resp. $\mathrm{Supp}(\mathcal{D}_{[\mathbf{Q_0}]})$). For the first claim, we want to show that for any $\mathbf{U} \in \mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$,

$$\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{Q}\mathbf{V})[2] = \mathbf{U}] = \Pr[\mathbf{V}^{-1}\mathsf{R}(\mathbf{Q})[2] = \mathbf{U}].$$

Write $\mathbf{Q} = \mathbf{B}^*\mathbf{Q_0}\mathbf{B}$, $\mathbf{V} = \mathbf{V_0}\mathbf{V_1}$, $\mathbf{U} = \mathbf{U_0}\mathbf{U_1}$ where $\mathbf{B}, \mathbf{V_0}, \mathbf{U_0} \in \mathrm{SL}_2(\mathcal{O}_\mathbb{L})$ and $\mathbf{V_1}(\text{resp. } \mathbf{U_1}) = \begin{pmatrix} 1 & 0 \\ 0 & \det(\mathbf{V}) \end{pmatrix}$(resp. $\begin{pmatrix} 1 & 0 \\ 0 & \det(\mathbf{U}) \end{pmatrix}$). Since $\mathbf{Q_0}$ is diagonal and $\mathbf{V_1} = \mathrm{diag}(1, \det(\mathbf{V}))$ with $\det(\mathbf{V}) \in \mu(\mathbb{L})$, we have $\mathbf{V_1}^*\mathbf{Q_0}\mathbf{V_1} = \mathbf{Q_0}$. Hence

$$\mathbf{V}^*\mathbf{Q}\mathbf{V} = (\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1})^*\mathbf{Q_0}(\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1}).$$

Note that $\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1} \in \mathrm{SL}_2(\mathcal{O}_\mathbb{L})$. From the construction of R, we can see

$$\begin{aligned}\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{Q}\mathbf{V})[2] = \mathbf{U}] &= \frac{1}{|\mu(\mathbb{L})|} \cdot \Pr[\mathsf{R}_0(\mathbf{V}^*\mathbf{Q}\mathbf{V})[2] = \mathbf{U_0}] \\ &= \frac{1}{|\mu(\mathbb{L})|} \cdot \Pr[\mathsf{R}_0(\mathbf{Q_0})[2] = (\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1})\mathbf{U_0}].\end{aligned}$$

The last equality holds by Lemma 4.1 since $\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1} \in \mathrm{SL}_2(\mathcal{O}_\mathbb{L})$ and

$$(\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1})^*\mathbf{Q_0}(\mathbf{V_1^{-1}}\mathbf{B}\mathbf{V_0}\mathbf{V_1}) = \mathbf{V}^*\mathbf{Q}\mathbf{V}.$$

Similarly, we have

$$\begin{aligned}\Pr[\mathbf{V}^{-1}\mathsf{R}(\mathbf{Q})[2]=\mathbf{U}] &= \Pr[\mathsf{R}(\mathbf{Q})[2]=\mathbf{VU}]\\ &= \frac{1}{|\mu(\mathbb{L})|}\cdot\Pr[\mathsf{R}_0(\mathbf{Q})[2]=\mathbf{V_0V_1U_0V_1^{-1}}]\\ &= \frac{1}{|\mu(\mathbb{L})|}\cdot\Pr[\mathsf{R}_0(\mathbf{Q_0})[2]=\mathbf{B}(\mathbf{V_0V_1U_0V_1^{-1}})].\end{aligned}$$

Denote $\mathbf{B}(\mathbf{V_0V_1U_0V_1^{-1}})$ by $\mathbf{M}\in \mathrm{SL}_2(\mathcal{O}_\mathbb{L})$, then

$$(\mathbf{V_1^{-1}BV_0V_1})\mathbf{U_0}=\mathbf{V_1^{-1}MV_1}.$$

By Lemma 4.2,

$$\Pr[\mathsf{R}_0(\mathbf{Q_0})[2]=\mathbf{M}]=\Pr[\mathsf{R}_0(\mathbf{Q_0})[2]=\mathbf{V_1^{-1}MV_1}],$$

so

$$\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{QV})[2]=\mathbf{U}]=\Pr[\mathbf{V}^{-1}\mathsf{R}(\mathbf{Q})[2]=\mathbf{U}].$$

For the second claim, we just need to show that for any $\mathbf{U}\in\mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$,

$$\sum_{\xi\in\mu(\mathbb{L})}\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{QV})[2]=\xi I_2\cdot\mathbf{U}]=\sum_{\xi\in\mu(\mathbb{L})}\Pr[\mathbf{V}^{-1}\mathsf{R}(\mathbf{Q})[2]=\xi I_2\cdot\mathbf{U}],$$

which can be directly derived from the previous conclusion. □

Algorithm 3: R: Re-Randomization for $\mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$

Require: Conductor $m=2^\kappa$ cyclotomic $\mathbb{L}$, $\mathbf{Q}\in\mathcal{H}_2^{\succeq 0}(\mathbb{L})$
Ensure: $\mathbf{R}\in[\mathbf{Q}]_{\mathrm{gsl}}$ and $\mathbf{U}\in\mathrm{GSL}_2(\mathcal{O}_\mathbb{L})$ such that $\mathbf{R}=\mathbf{U}^*\cdot\mathbf{Q}\cdot\mathbf{U}$
1: Let $(\mathbf{R_0},\mathbf{U_0})\leftarrow\mathsf{R}_0(\mathbf{Q})$
2: Parse $\xi\leftarrow_\$\mu(\mathbb{L})$
3: Let $Y=\begin{pmatrix}1&0\\0&\xi\end{pmatrix}$, $\mathbf{U}=\mathbf{U_0}Y$, and $\mathbf{R}=\mathbf{U}^*\mathbf{QU}$
4: **return** $(\mathbf{R},\mathbf{U})$

Remark 8. We can prove $\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{QV})[2]=\mathbf{U}]=\Pr[\mathsf{R}(\mathbf{V}^*\mathbf{QV})[2]=\xi I_2\cdot\mathbf{U}]$ for any $\xi\in\mu(\mathbb{L})$ by the other equality in Lemma 4.2.

4.3 Extractor on $\mathbf{PGSL_2}(\boldsymbol{\mathcal{O}_\mathbb{L}})$

Lemma 4.4 *For the group actions*

$$(PGSL_2(\mathcal{O}_\mathbb{L}),[\mathbf{Q_k^{(0)}}]_{gsl},\star);$$
$$(PGSL_2(\mathcal{O}_\mathbb{L}),[\widetilde{\mathbf{Q}}_k^{(0)}]_{gsl},\star),(PGSL_2(\mathcal{O}_\mathbb{L}),[\widetilde{\mathbf{Q}}_\ell^{(1)}]_{gsl},\star),\quad k\geq 0,\ \ell>2,$$

and the group $M = (\langle X \rangle / \langle X^2 \rangle, \times) \simeq (\{\pm 1\}, \times)$*, define* $\mathsf{E} : PGSL_2(\mathcal{O}_\mathbb{L}) \to M$ *such that* $\mathsf{E}(\mathbf{U}/\mu(\mathbb{L})I_2) = \det(\mathbf{U})/\langle X^2 \rangle$*. Then,* E *is an extractor satisfying all properties listed in Definition 2.11.*

Proof. Due to space constraints, its proof is provided in the Appendix C.

A Discussion on the extractor. Previously, we only briefly mentioned that because the det cannot serve as an extractor on $\mathrm{SL}_2(\mathcal{O}_\mathbb{L})$, additional efforts are required to adjust the group and the re-randomized algorithm. In fact, if we wish to preserve *the uniformly extracting property and homomorphic property*, it can be proven that for the equivalence classes we consider, there exists no extractor meeting the requirements if we select $\mathrm{SL}_2(\mathcal{O}_\mathbb{L})$ for the group action. This proof will be provided in the Appendix D.

4.4 Commitments

The Enhanced Linkable Commitment We take the group action

$$(\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L}), [\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}]_{\mathrm{gsl}}, \star)$$

with constant k, and a samplable distribution $\mathcal{D}_{[\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}]_{\mathrm{gsl}}}$. Under Heuristic 1 for $\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}$, by Lemma 4.3, Lemma 4.4, Theorem 2.2 and Theorem 3.1, there is an Enhanced Linkable Commitment based on the hardness of module-LIP$_{\mathcal{D}_{[\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}]_{\mathrm{gsl}}}}$. A natural choice of $\mathcal{D}_{[\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}]_{\mathrm{gsl}}}$ is $\mathsf{R}(\mathbf{Q}_\mathbf{k}^{(\mathbf{0})})[1]$ with $\sigma = 2^{\Theta(n)}$, in which case the security of the resulting commitment is based on the worst-case hardness of module-LIP$_{[\mathbf{Q}_\mathbf{k}^{(\mathbf{0})}]}$. Moreover, module-LIP$_{[\mathbf{Q}_k^{(0)}]}$ is as hard as $\mathcal{O}_\mathbb{L}^2$-LIP by Theorem 3.3.

When $k = 0$, another choice for the distribution $\mathcal{D}_{[I_2]}$ is to use the public-key distribution from HAWK; the resulting commitment scheme inherits the secret-key security of HAWK.

Dual-Mode Commitment We take group actions $(\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L}), [\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}, \star)$ and $(\mathrm{PGSL}_2(\mathcal{O}_\mathbb{L}), [\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}, \star)$ with $k > 2$, $k = O(\log n)$, and samplable distributions $\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}}$ and $\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}}$. Under Heuristic 1 for $\widetilde{\mathbf{Q}}_k^{(0)}$, by Lemma 4.3, Lemma 4.4, and Theorem 2.2, there is a Dual-Mode Commitment based on the hardness of module-LIP$_{\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}}}$ and module-ΔLIP$_{\mathbb{L},\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}},\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}}}$.

When taking $\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}}$ as $\mathsf{R}(\widetilde{\mathbf{Q}}_k^{(0)})[1]$ and $\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}}$ as $\mathsf{R}(\widetilde{\mathbf{Q}}_k^{(1)})[1]$ with $\sigma = 2^{\Theta(n)}$, the hardness of module-ΔLIP$_{\mathbb{L},\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}},\mathcal{D}_{[\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}}}$ achieves that of the worst-case module-ΔLIP$_{\mathbb{L},[\widetilde{\mathbf{Q}}_k^{(0)}],[\widetilde{\mathbf{Q}}_k^{(1)}]}$, which was discussed in Section 3.4.

4.5 Discussion about Asymptotic Complexity

Plain version vs. Module version. Since the main body of all schemes is based on either the basic scheme or the dual scheme corresponding to Protocol 2.1, and the plain version did not provide concrete instance selection for the dual scheme, we focus our analysis on the asymptotic complexity of the basic scheme. Throughout this discussion, time complexities are bit-complexities and are stated in soft-O notation when they involve integer, matrix, or polynomial arithmetic. Size bounds refer to the total bit-size of the represented objects.

The process of generating Ck involves sampling from $\mathcal{D}_{[\mathbf{Q}_k^{(0)}]}$ and $\mathcal{D}_{[I_{2n}]}$, where k is a small constant. Assuming the hardness of module-LIP holds when the entry size is bounded by $B = O(\log n)$, the module version, following the HAWK-style key-generation approach in the 2-power cyclotomic ring, has quasi-linear running time $\widetilde{O}(n \log n)$ and key size $O(n \log n)$. The plain version requires sampling several vectors in $\mathbb{Z}^{2n}$, computing the HNF of a matrix in $\mathbb{Z}^{2n \times 2n}$, and performing matrix multiplications; the size of Ck is at least $O(n^2 \log n)$.

For commitment generation, the module version uses a constant number of discrete Gaussian samplings, `HermiteSolve`, `Reduce`, and a constant number of arithmetic operations in $\mathbb{L}$. The `Reduce` step includes an inversion and multiplications in the 2-power cyclotomic ring; with the conservative bit-complexity estimate used here, this post-solving arithmetic costs $\widetilde{O}(n^2 \log n)$ in the reduced-size regime. The plain version uses $O(n^2)$ discrete Gaussian samplings, one HNF computation, one determinant calculation, and two matrix multiplications. For opening, the module version performs a constant number of operations on reduced-size ring elements. After the size-control step described in the remark below, the relevant coefficients have bit-size $O(\log n)$, so fast cyclotomic arithmetic gives opening time $\widetilde{O}(n \log n)$. In the plain version, opening requires two matrix multiplications with entries of size $O(n \log n)$ and one determinant computation.

Table 1. Complexity and Size Comparison between Plain and Module Schemes

Schemes	Time Complexity			Size	
	$\mathbf{Gen}(1^\lambda)$	$\mathbf{Com}_y(m)$	$\mathbf{Open}_y(c, d)$	Ck	Commitment
Plain	T_{HNF}	$n^2 T_{\mathrm{DGS}} + T_{\mathrm{HNF}}$	$\widetilde{O}(n^{\omega+1} \log n)$	$O(n^2 \log n)$	$O(n^3 \log n)$
Module	$\widetilde{O}(n \log n)^\dagger$	$T_{\mathrm{DGS}} + T_{\mathrm{HNF}} + \widetilde{O}(n^2 \log n)$	$\widetilde{O}(n \log n)$	$O(n \log n)$	$O(n \log n)$

Note: T_{HNF} denotes the cost of computing the HNF for the relevant $2n \times 2n$ integer matrix, and T_{DGS} denotes the cost of one $2n$-dimensional discrete Gaussian sampling. The term $O(n^\omega)$ denotes matrix multiplication complexity. The $\widetilde{O}(n^2 \log n)$ term in module commitment generation is the conservative cost of the inversion and multiplications in the subsequent `Reduce` step. The notation $\widetilde{O}$ suppresses additional polylogarithmic factors arising from integer and polynomial arithmetic; displayed $\log n$ factors are kept when they represent coefficient bit-size. The † entry reports the HAWK-style practical generation approach; if generation is instantiated with the strictly HNF-based completion step, the corresponding HNF cost should be added.

Remark 9 (Size control after `Reduce`*).* The naive size accounting for the module commitment can be misleading: the HNF computation and subsequent inversions may inflate intermediate coefficients, which would lead to an $O(n^2 \log n)$ commitment-size bound before reduction. However, the output of `Reduce` allows one to reject sampled vectors satisfying

$$\left\langle (x^* \mathbf{Q} x)^{-1}, 1 \right\rangle > \mathrm{poly}(n)$$

during commitment generation. With this rejection step, the final reduced commitment has total bit-size $O(n \log n)$, and the coefficients used in the opening procedure remain of size $O(\log n)$. This rejection sampling does not change the required re-randomization properties; the detailed verification is deferred to Appendix Appendix E.

HermiteSolve vs. TowerSolve. The most computationally heavy component in the strictly analyzed version of our scheme is the `HermiteSolve` step used in the re-randomized algorithm to generate matrices in $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$. This step computes an integer Hermite Normal Form, as in the provable variant of HAWK [12]. In HAWK's practical variant, `HermiteSolve` is replaced by `TowerSolve` from [30], which is much faster but imposes additional conditions on the input.

The schemes covered by the theorems above use the HNF-based `HermiteSolve` routine and therefore satisfy the stated re-randomization property. Replacing this step by `TowerSolve` should be viewed only as a heuristic practical variant: it may break the strict uniformity required by the re-randomization proof, and therefore it is not included in the formal security claim.

Let B denote the relevant coefficient bit-size. The HNF-based `HermiteSolve` step costs roughly $\widetilde{O}(n^3 B)$ time and $O(n^2 B)$ space, whereas a `TowerSolve`-style routine costs roughly $\widetilde{O}(nB)$ time and $O(n(B + \log n))$ space. After replacing the solving step, the whole commitment-generation procedure is still not quasi-linear: the dominant cost shifts to the subsequent `Reduce` procedure and the remaining arithmetic in $\mathbb{L}$, whose inversion and multiplication operations cost conservatively $\widetilde{O}(n^2 \log n)$ in the reduced-size regime. Thus the TowerSolve-based **Com** phase is estimated as $T_{\mathrm{DGS}} + \widetilde{O}(n^2 \log n)$, even though the solving subroutine itself is only $\widetilde{O}(nB)$. Under the rejection step above, both variants have final commitment size $O(n \log n)$, and the opening phase costs $\widetilde{O}(n \log n)$.

For simplicity, consider the basic scheme constructed from $\mathbf{Q}_k^{(0)}$. If one substitutes the re-randomization algorithm R by the TowerSolve-based variant R', the resulting scheme should be treated as a computationally hiding heuristic variant rather than a perfectly hiding one. Although the output distribution of R' is not known to be uniform over the relevant isomorphism set, no attack is currently known that extracts useful information about $\mathrm{Aut}_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q})$ from this deviation. Nor is a reduction from this heuristic hiding statement to a standard assumption currently known.

When considering even $k > 2$, if we assume that module-ΔLIP$_{[\widetilde{\mathbf{Q}}_{k/2}^{(0)}],[\widetilde{\mathbf{Q}}_{k/2}^{(1)}]}$ under an appropriate distribution is hard, the hiding hardness can be viewed through the corresponding dual scheme based on $\widetilde{\mathbf{Q}}_{k/2}^{(1)}$. For a form $\mathbf{Q} \in [\widetilde{\mathbf{Q}}_{k/2}^{(1)}]$,

Table 2. Comparison between the HNF-Based and TowerSolve-Based Variants

Variant	Time Complexity			Size / Status	
	Solving step	$\mathbf{Com}_y(m)$	$\mathbf{Open}_y(c,d)$	Commitment	Claim
HNF-based	$\widetilde{O}(n^3B)$	$T_{\mathrm{DGS}} + T_{\mathrm{HNF}} + \widetilde{O}(n^2 \log n)$	$\widetilde{O}(n \log n)$	$O(n \log n)$	formal
TowerSolve-based	$\widetilde{O}(nB)$	$T_{\mathrm{DGS}} + \widetilde{O}(n^2 \log n)$	$\widetilde{O}(n \log n)$	$O(n \log n)$	heuristic

Note: The space bounds for the solving step are $O(n^2B)$ for the HNF-based routine and $O(n(B + \log n))$ for the TowerSolve-style routine. The $\widetilde{O}(n^2 \log n)$ term records the conservative bit-complexity of the inversion and multiplications in `Reduce`, which becomes the limiting asymptotic cost once the HNF-based solver is replaced. The TowerSolve row is a performance estimate only and is not part of the formal security theorem.

the extractor value on $\mathrm{Aut}_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q})$ is intrinsic to $\mathbf{Q}$, and the worst-case problem is hard under the discussion in Section 3.4. We leave a formal security treatment of the TowerSolve-based distribution as an open question.

5 Conclusion

This paper investigates the hardness of specific module-LIP problems and, building on this security foundation, uses the framework proposed in [18] to construct group-action-based commitment schemes in the module setting by selecting appropriate group actions. Compared to the original plain setting, these schemes achieve efficiency improvements. Although not yet practical, their module structure leaves room for further optimization in practical implementations.

References

1. Allombert, B., Pellet-Mary, A., van Woerden, W.P.J.: Cryptanalysis of rank-2 module-lip: A single real embedding is all it takes. In: Fehr, S., Fouque, P. (eds.) Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 184–212. Springer (2025). https://doi.org/10.1007/978-3-031-91124-8_7, https://doi.org/10.1007/978-3-031-91124-8_7
2. Bennett, H., Ganju, A., Peetathawatchai, P., Stephens-Davidowitz, N.: Just how hard are rotations of z n? algorithms and cryptography with the simplest lattice. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 252–281. Springer (2023)
3. Biasse, J.F., Fieker, C., Hofmann, T.: On the computation of the hnf of a module over the ring of integers of a number field. Journal of Symbolic Computation **80**, 581–615 (May 2017). https://doi.org/10.1016/j.jsc.2016.07.027, http://dx.doi.org/10.1016/j.jsc.2016.07.027
4. Biasse, J.F., Song, F.: Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields, pp.

893–902. SIAM (2016). https://doi.org/10.1137/1.9781611974331.ch64, https://epubs.siam.org/doi/abs/10.1137/1.9781611974331.ch64
5. Branco, P., Malavolta, G., Maradni, Z.: Fully-homomorphic encryption from lattice isomorphism. In: Theory of Cryptography: 23rd International Conference, TCC 2025, Aarhus, Denmark, December 1–5, 2025, Proceedings, Part I. p. 220–252. Springer-Verlag, Berlin, Heidelberg (2025). https://doi.org/10.1007/978-3-032-12287-2_8, https://doi.org/10.1007/978-3-032-12287-2_8
6. Buchmann, J.A., Lenstra, H.W.: Approximatting rings of integers in number fields. Journal de théorie des nombres de Bordeaux **6**(2), 221–260 (1994), https://jtnb.centre-mersenne.org/item/JTNB_1994__6_2_221_0/
7. Chevignard, C., Mureau, G., Espitau, T., Pellet-Mary, A., Pliatsok, H., Wallet, A.: A reduction from hawk to the principal ideal problem in a quaternion algebra. In: Fehr, S., Fouque, P. (eds.) Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part II. Lecture Notes in Computer Science, vol. 15602, pp. 154–183. Springer (2025). https://doi.org/10.1007/978-3-031-91124-8_6, https://doi.org/10.1007/978-3-031-91124-8_6
8. Cohen, H.: Advanced topics in computational number theory, chap. The Hermite Normal Form Algorithms in Dedekind Domains, p. 27. Springer (2000), https://api.semanticscholar.org/CorpusID:118279568
9. De Micheli, G., Micciancio, D., Pellet-Mary, A., Tran, N.: Reductions from module lattices to free module lattices, and application to dequantizing module-LLL. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology – CRYPTO 2023. pp. 836–865. Springer Nature Switzerland, Cham (2023)
10. Ducas, L.: Provable lattice reduction of z n with blocksize n/2. Designs, Codes and Cryptography pp. 1–8 (2023)
11. Ducas, L., Gibbons, S.: Hull attacks on the lattice isomorphism problem. In: IACR International Conference on Public-Key Cryptography. pp. 177–204. Springer (2023)
12. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.P.J.: Hawk: Module LIP makes lattice signatures fast, compact and simple. In: Agrawal, S., Lin, D. (eds.) Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13794, pp. 65–94. Springer (2022), https://doi.org/10.1007/978-3-031-22972-5_3
13. Ducas, L., van Woerden, W.P.J.: On the lattice isomorphism problem, quadratic forms, remarkable lattices, and cryptography. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology - EUROCRYPT 2022 - 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 643–673. Springer (2022), https://doi.org/10.1007/978-3-031-07082-2_23
14. Ducas, L., Postlethwaite, E.W., Pulles, L.N., van Woerden, W.: Hawk: Module LIP makes lattice signatures fast, compact and simple. Cryptology ePrint Archive, Paper 2022/1155 (2022), https://eprint.iacr.org/2022/1155
15. D'Alconzo, G., Flamini, A., Gangemi, A.: Non-interactive commitment from non-transitive group actions. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 222–252. Springer (2023)

16. van Gent, D., van Woerden, W.: A search to distinguish reduction for the isomorphism problem on direct sum lattices. In: Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part III. Springer-Verlag (2025), https://eprint.iacr.org/2025/1204
17. Gentry, C., Szydlo, M.: Cryptanalysis of the revised NTRU signature scheme. In: Knudsen, L.R. (ed.) Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2332, pp. 299–320. Springer (2002), https://doi.org/10.1007/3-540-46035-7_20
18. Jiang, K., Wang, A., Luo, H., Liu, G., Tang, G., Pan, Y., Wang, X.: Re-randomize and extract: A novel commitment construction framework based on group actions. In: Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part II. Springer-Verlag (2025), https://eprint.iacr.org/2025/400
19. Jiang, K., Wang, A., Luo, H., Liu, G., Yu, Y., Wang, X.: Exploiting the symmetry of $\mathbb{Z}^n$: Randomization and the automorphism problem. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023. pp. 167–200. Springer Nature Singapore, Singapore (2023)
20. Jr., H.W.L., Silverberg, A.: Revisiting the gentry-szydlo algorithm. In: Garay, J.A., Gennaro, R. (eds.) Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I. Lecture Notes in Computer Science, vol. 8616, pp. 280–296. Springer (2014), https://doi.org/10.1007/978-3-662-44371-2_16
21. Jr., H.W.L., Silverberg, A.: Lattices with symmetry. J. Cryptol. **30**(3), 760–804 (2017), https://doi.org/10.1007/s00145-016-9235-7
22. Kuninets, A., Leevik, A., Malygina, E., Melnichuk, E., Nabokov, D.: On the construction of barnes-wall lattices and their application in cryptography. Cryptology ePrint Archive, Paper 2025/1640 (2025), https://eprint.iacr.org/2025/1640
23. Lenstra Jr, H.W., Silverberg, A.: Testing isomorphism of lattices over cm-orders. SIAM Journal on Computing **48**(4), 1300–1334 (2019)
24. Leporati, A., Rovida, L., van Woerden, W.: Beyond LWE: a lattice framework for homomorphic encryption. Cryptology ePrint Archive, Paper 2025/1171 (2025), https://eprint.iacr.org/2025/1171
25. Ling, C., Liu, J., Mendelsohn, A.: On the spinor genus and the distinguishing lattice isomorphism problem. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024. pp. 329–358. Springer Nature Singapore, Singapore (2025)
26. Luo, H., Jiang, K., Pan, Y., Wang, A.: Cryptanalysis of rank-2 module-lip with symplectic automorphisms. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024. pp. 359–385. Springer Nature Singapore, Singapore (2025)
27. Mureau, G.: Special genera of hermitian lattices and applications to hawk. In: Applebaum, B., Lin, H.R. (eds.) Theory of Cryptography. pp. 289–317. Springer Nature Switzerland, Cham (2026)
28. Mureau, G., Pellet-Mary, A., Pliatsok, G., Wallet, A.: Cryptanalysis of rank-2 module-lip in totally real number fields. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 226–255. Springer (2024)

29. Mureau, G., Pellet-Mary, A., Pliatsok, H., Wallet, A.: Cryptanalysis of rank-2 module-lip in totally real number fields. Cryptology ePrint Archive, Paper 2024/441 (2024), https://eprint.iacr.org/2024/441, https://eprint.iacr.org/2024/441
30. Pornin, T., Prest, T.: More efficient algorithms for the NTRU key generation using the field norm. In: Lin, D., Sako, K. (eds.) Public-Key Cryptography – PKC 2019. pp. 504–533. Springer International Publishing, Cham (2019)
31. Schwerdtfeger, H.: Gesammelte abhandlungen by h. minkowski. chelsea publishing co., new york, 1967. xxxi +371 + iv + 465 pages and two portraits of the author. Canadian Mathematical Bulletin **12**(2), 241–241 (1969). https://doi.org/10.1017/S0008439500030265
32. Silva, F.C., Lie, M., Ling, C.: On hull attacks on the module lattice isomorphism problem. Cryptology ePrint Archive, Paper 2025/1376 (2025), https://eprint.iacr.org/2025/1376
33. Stehlé, D., Steinfeld, R.: Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. Cryptology ePrint Archive, Paper 2013/004 (2013), https://eprint.iacr.org/2013/004
34. Vaserštein, L.N.: On the group sl2 over dedekind rings of arithmetic type. Mathematics of the USSR-Sbornik **18**(2), 321 (feb 1972). https://doi.org/10.1070/SM1972v018n02ABEH001775, https://doi.org/10.1070/SM1972v018n02ABEH001775
35. van Woerden, W.: Dense and smooth lattices in any genus. In: Chung, K.M., Sasaki, Y. (eds.) Advances in Cryptology – ASIACRYPT 2024. pp. 386–417. Springer Nature Singapore, Singapore (2025)

Appendix A Supplementary Definitions for Commitment Schemes

Definition A.1 (Enhanced Linkable Commitment) *Let λ denote the security parameter, $(M, \cdot)$ represent a finite group serving as the message space, and (C, D) denote the commitment and opening spaces respectively. An enhanced linkable commitment scheme is a tuple*

$$\Pi_{elc} = (\boldsymbol{Gen}, \boldsymbol{Com}, \boldsymbol{Open}, (\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})),$$

consisting of a commitment scheme Π_{com} and a link component. The scheme satisfies the following properties:

1. *The link component comprises PPT algorithms $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$. For any two messages m_0 and m_1 in M with commitments $(c_0, d_0) = \boldsymbol{Com}(m_0)$ and $(c_1, d_1) = \boldsymbol{Com}(m_1)$:*
   - *$\mathsf{Link}(c_0, c_1, d_L)$ returns 0 or 1. If $\mathsf{Link}(c_0, c_1, d_L) = 1$, d_L is a valid linking value for the pair (c_0, c_1).*
   - *$\mathsf{LinkE}(d_0, d_1)$ generates a linking value d_L such that $\mathsf{Link}(c_0, c_1, d_L) = 1$.*
   - *$\mathsf{LinkC}(c_0, c_1, d_L)$ computes $m_0 \cdot m_1^{-1}$ if $\mathsf{Link}(c_0, c_1, d_L) = 1$, else returns $\perp$.*
2. ***Computationally Enhanced-Linkable-Hiding (ELH)**: No PPT adversary $\mathcal{A}$ can win the **ELH** game with probability exceeding $\frac{1}{2} + \mathsf{negl}(\lambda)$:*

$$\Pr[\mathcal{A} \text{ wins } \boldsymbol{ELH}(\Pi_{elc})] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

   Here, $\boldsymbol{ELH}(\Pi_{elc})$ represents the game described in Figure 1.
3. ***Computationally Enhanced-Linkable-Binding (ELB)**: No PPT adversary $\mathcal{A}$ can violate the binding property with probability higher than $\mathsf{negl}(\lambda)$:*

$$\Pr\left[\begin{array}{c} \mathbf{Open}(c_i, d_i) = m_i,\ i \in \{0,1\}, \\ \mathsf{Link}(c_0, c_1, d_L) = 1, \\ m_0 \cdot m_1^{-1} \neq \mathsf{LinkC}(c_0, c_1, d_L) \end{array} \middle| \begin{array}{c} Ck \leftarrow \mathbf{Gen}(1^\lambda), \\ \begin{pmatrix} m_0, m_1, c_0, c_1 \\ d_0, d_1, d_L \end{pmatrix} \leftarrow \mathcal{A}(Ck) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

4. ***Computationally Enhanced-Linkable-Unforgeable (ELU)**: No PPT adversary $\mathcal{A}$ can win the **ELU** game with probability exceeding $\mathsf{negl}(\lambda)$:*

$$\Pr[\mathcal{A} \text{ wins } \boldsymbol{ELU}(\Pi_{elc})] \leq \mathsf{negl}(\lambda).$$

   Here, $\boldsymbol{ELU}(\Pi_{elc})$ represents the game described in Figure 2.

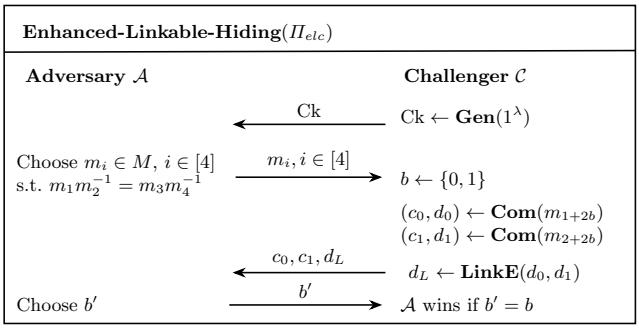


Fig. 1. The Enhanced Linkable Hiding Game

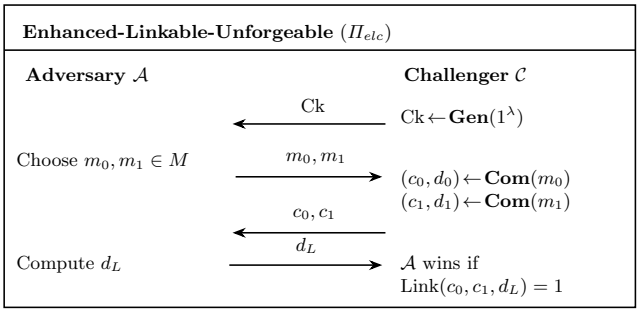


Fig. 2. The Enhanced Linkable Unforgeable Game

Commitment Scheme A.1 : $\Pi_{elc} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open}, (\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC}))$, *for a security parameter* λ.

- $\Pi_{com} = (\mathbf{Gen}, \mathbf{Com}, \mathbf{Open})$ is Protocol 2.1.

If $Ck = y$, for two commitments of m_0 and m_1 in Protocol 2.1:

$$\mathbf{Com}_y(m_0) \to (c_0, d_0) = ((c_{01}, c_{02}), d_0) = ((\mathsf{E}(g_0) \cdot m_0, y_0), g_0)$$
$$\mathbf{Com}_y(m_1) \to (c_1, d_1) = ((c_{11}, c_{12}), d_1) = ((\mathsf{E}(g_1) \cdot m_1, y_1), g_1)$$

The link component $(\mathsf{Link}, \mathsf{LinkE}, \mathsf{LinkC})$ as follows:

- $\mathsf{Link}(c_0, c_1, d_L) = \begin{cases} 1 & \text{if } d_L \star c_{12} = c_{02}. \\ 0 & else. \end{cases}$
- $\mathsf{LinkE}(d_0, d_1) = d_0 \cdot d_1^{-1}$.
- $\mathsf{LinkC}(c_0, c_1, d_L)$: If $d_L \star c_{12} = c_{02}$, it outputs $\mathsf{E}(d_L)^{-1} \cdot c_{01} \cdot c_{11}^{-1}$; otherwise, it outputs $\perp$.

Definition A.2 (Dual-Mode Commitment) A dual-mode commitment scheme is a tuple of PPT algorithms $\Pi_{dmc} = (\mathbf{Gen}, \mathbf{TGen}, \mathbf{Com}, \mathbf{TCom}, \mathbf{Open}, \mathbf{TCol})$, for a security parameter λ, message space M, commitment space C, and opening space D, such that

1. (**TGen**, **TCom**, **TCol**) satisfies:
   - $\mathbf{TGen}(1^\lambda)$ outputs the public commitment key Ck and corresponding trapdoor information Td.
   - $\mathbf{TCom}(Ck, Td)$ outputs a commitment value $c \in C$ and a state St.
   - $\mathbf{TCol}(Ck, Td, St, m)$ outputs an opening value $d \in D$.
2. **Perfectly Binding**: (**Gen**, **Com**, **Open**) is a perfectly binding commitment scheme.
3. **Completeness**: For $(Ck, Td) \leftarrow \mathbf{TGen}$ and $m \in M$, it holds that
$$\Pr[\mathbf{Open}_{Ck}(c, d) = m \mid (c, d) \leftarrow \mathbf{Com}_{Ck}(m)] = 1.$$
4. **Trapdoor Property**: For $(Ck, Td) \leftarrow \mathbf{TGen}$ and $m \in M$, the following distributions are identical:
$$\{(c, d, m) \mid (c, d) \leftarrow \mathbf{Com}_{Ck}(m)\} \quad \text{and}$$
$$\{(c, d, m) \mid (c, St) \leftarrow \mathbf{TCom}(Ck, Td),\ d \leftarrow \mathbf{TCol}(Ck, Td, St, m)\}.$$
5. **Key Indistinguishability**: For all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\lambda)$ such that
$$\Pr\left[b = \tilde{b} \;\middle|\; \begin{matrix} Ck_0 \leftarrow \mathbf{Gen}(1^\lambda), Ck_1 \leftarrow \mathbf{TGen}(1^\lambda) \\ b \leftarrow \{0,1\}, \tilde{b} \leftarrow \mathcal{A}(Ck_b) \end{matrix}\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Commitment Scheme A.2 : $\Pi_{dmc} = (\mathbf{Gen}, \mathbf{TGen}, \mathbf{Com}, \mathbf{TCom}, \mathbf{Open}, \mathbf{TCol})$, *for a security parameter* λ.

- $\mathbf{Gen}(1^\lambda)$: Sample $y \leftarrow \mathcal{D}_{\mathsf{X}'}$ and output $Ck = y$.
- $\mathbf{TGen}(1^\lambda)$: Sample $y = g \star x \leftarrow \mathcal{D}_{\mathsf{X}}$ and compute $\mathrm{Stab}(y) = g \cdot \mathrm{Stab}(x) \cdot g^{-1}$. Output $(Ck, Td) = (y, \mathrm{Stab}(y))$.
- $\mathbf{Com}_y(m)$: For a message $m \in M$, sample $(y', g) \leftarrow \mathsf{R}(y)$ such that $g \star y = y'$. Compute $c = (c_1, c_2) = (\mathsf{E}(g) \cdot m, y')$ and set $d = g$. Output (c, d).
- $\mathbf{TCom}(y, \mathrm{Stab}(y))$: Sample $(y', g) \leftarrow \mathsf{R}(y)$, $u \leftarrow M$ and set $c = (c_1, c_2) = (u, y')$, $St = (u, g)$. Output (c, St).
- $\mathbf{Open}_y(c, d)$: If $d \star y = c_2$, output $\tilde{m} = \mathsf{E}(d)^{-1} \cdot c_1$; otherwise, $\tilde{m} = \perp$.
- $\mathbf{TCol}(y, \mathrm{Stab}(y), (u, g), m)$: Compute $r = u \cdot m^{-1}$, then sample $\mathsf{F}(r, g, y, \mathrm{Stab}(y)) \to g'$ and set $d = g'$. Output d.

Appendix B Generalized Theorem 3.1 for CM number field

The main structure of the reduction is identical to Theorem 3.1. What requires additional proof is that for a general CM field $\mathbb{L}$, after considering its extension of

the form $\mathbb{L}[Y]/(Y^2-\xi)$ (where ξ is a root of unity), the extended field remains a CM field and an integral basis of its ring of integers can still be efficiently computed.

Lemma B.1 *Assume $\xi \in \mathbb{L}$ is a root of unity and $X^2-\xi$ has no roots in $\mathbb{L}$. Let $\mathbb{F} := \mathbb{L}[X]/(X^2-\xi)$. Then $\mathbb{F}$ is also a CM number field.*

Proof. Let $\mathbb{K}$ be the totally real number field such that $\mathbb{L}/\mathbb{K}$ is a totally imaginary quadratic extension. If $\xi \neq -1$, consider $\mathbb{M} := \mathbb{K}(X+X^{-1}) \subseteq \mathbb{F}$. $\mathbb{M}$ is totally real since $\mathbb{Q}(X+X^{-1})$ and $\mathbb{K}$ are both totally real. Moreover, $\mathbb{M} \neq \mathbb{K}$; otherwise

$$(\xi+1)/X = (X^2+1)/X = X + X^{-1} \in \mathbb{K} \subseteq \mathbb{L} \Rightarrow X \in \mathbb{L},$$

leading to a contradiction.

If $\xi = -1$, assume $\mathbb{L} = \mathbb{K}(\alpha)$, where $\alpha^2 \in \mathbb{K}$ and $\phi(\alpha^2) < 0$ for any embedding $\phi : \mathbb{K} \to \mathbb{C}$. Consider $\mathbb{M} := \mathbb{K}(X \cdot \alpha) \subseteq \mathbb{F}$. Then $\mathbb{M}$ is totally real since $\mathbb{K}$ is totally real and $\phi((X\cdot\alpha)^2) = -\phi(\alpha^2) > 0$ for any embedding $\phi : \mathbb{M} \to \mathbb{C}$. Moreover, $\mathbb{M} \neq \mathbb{K}$; otherwise $X \cdot \alpha \in \mathbb{K} \subseteq \mathbb{L} \Rightarrow X \in \mathbb{L}$, leading to a contradiction. In both cases, $\mathbb{F} = \mathbb{M}(X)$. Indeed, the construction gives $[\mathbb{M} : \mathbb{K}] = 2$ and hence $[\mathbb{F} : \mathbb{M}] = 2$, while $X \notin \mathbb{M}$ because $\mathbb{M}$ is totally real and X is not real. Moreover, X satisfies a quadratic equation over $\mathbb{M}$: namely $T^2 - (X+X^{-1})T + 1 = 0$ in the case $\xi \neq -1$, and $T^2+1=0$ in the case $\xi = -1$. For every real embedding of $\mathbb{M}$, the two conjugates of X are roots of unity and are not ± 1, so they are non-real. Therefore $\mathbb{F}/\mathbb{M}$ is a totally imaginary quadratic extension, and $\mathbb{F}$ is a CM field. □

When $\mathbb{L}$ is a totally real number field and $\mathbb{F} = \mathbb{L}[X]/(X^2+1)$, Mureau et al. showed $\log \Delta_{\mathbb{F}} = \mathrm{poly}(\log \Delta_{\mathbb{L}})$ and we can compute $\mathcal{O}_{\mathbb{F}}$ from $\mathcal{O}_{\mathbb{L}}$ in [28, Section 2.2]. We generalize this conclusion as follows.

Lemma B.2 ([6, Lemma 1.4]) *There are polynomial-time algorithms that, given an algebraic number field K and one of (a), (b), determine the other:*

(a) the ring of algebraic integers of K;
(b) the largest squarefree divisor of the discriminant of K.

Lemma B.3 *Assume $\xi \in \mathbb{L}$ is a root of unity and $X^2-\xi$ has no roots in $\mathbb{L}$. Let $\mathbb{F} := \mathbb{L}[X]/(X^2-\xi)$. There exists a polynomial-time algorithm A that, given as input a $\mathbb{Z}$-basis $B_{\mathbb{L}}$ of $\mathcal{O}_{\mathbb{L}}$, computes a $\mathbb{Z}$-basis $B_{\mathbb{F}}$ of $\mathcal{O}_{\mathbb{F}}$.*

Proof. The module $A := \mathcal{O}_{\mathbb{L}} + \mathcal{O}_{\mathbb{L}} \cdot X \subseteq \mathcal{O}_{\mathbb{F}}$ is an order of $\mathbb{F}$. Assume $B_{\mathbb{L}} = \{\beta_1, \ldots, \beta_n\}$; then $\Delta_{\mathbb{L}} = \det((\mathrm{Tr}_{\mathbb{L}}(\beta_i\beta_j)))$ and the discriminant of A over $\mathbb{Z}$ is

$$\Delta_A = \det\left(\begin{pmatrix} (\mathrm{Tr}_{\mathbb{F}}(\beta_i\beta_j)) & (\mathrm{Tr}_{\mathbb{F}}(X\beta_i\beta_j)) \\ (\mathrm{Tr}_{\mathbb{F}}(X\beta_j\beta_i)) & (\mathrm{Tr}_{\mathbb{F}}(\xi\beta_i\beta_j)) \end{pmatrix}\right).$$

On one hand, $A \subseteq \mathcal{O}_{\mathbb{F}}$ implies $\Delta_{\mathbb{F}} | \Delta_A$. On the other hand, the computation below shows that $\Delta_A = 2^{2n}\Delta_{\mathbb{L}}^2$. Thus $\Delta_{\mathbb{F}} | 2^{2n}\Delta_{\mathbb{L}}^2$. We also know $\Delta_{\mathbb{L}} | \Delta_{\mathbb{F}}$. Hence

the largest squarefree divisor of $\Delta_{\mathbb{F}}$ is the largest squarefree divisor of $\Delta_{\mathbb{L}}$ or the largest squarefree divisor of $\Delta_{\mathbb{L}}$ multiplied by 2. By Lemma B.2, we can find the largest squarefree divisor of $\Delta_{\mathbb{L}}$ from $B_{\mathbb{L}}$. Then we can guess the largest squarefree divisor of $\Delta_{\mathbb{F}}$ and compute the $\mathbb{Z}$-basis of $\mathcal{O}_{\mathbb{F}}$ by Lemma B.2.

The computation is as follows. Note that

$$\begin{aligned}\mathrm{Tr}_{\mathbb{F}}(X\beta_i\beta_j) = \mathrm{Tr}_{\mathbb{L}}(\mathrm{Tr}_{\mathbb{F}/\mathbb{L}}(X\beta_i\beta_j)) = \mathrm{Tr}_{\mathbb{L}}(0) = 0,\\ \mathrm{Tr}_{\mathbb{F}}(\beta_i\beta_j) = \mathrm{Tr}_{\mathbb{L}}(\mathrm{Tr}_{\mathbb{F}/\mathbb{L}}(\beta_i\beta_j)) = 2\mathrm{Tr}_{\mathbb{L}}(\beta_i\beta_j),\\ \mathrm{Tr}_{\mathbb{F}}(\xi\beta_i\beta_j) = \mathrm{Tr}_{\mathbb{L}}(\mathrm{Tr}_{\mathbb{F}/\mathbb{L}}(\xi\beta_i\beta_j)) = 2\mathrm{Tr}_{\mathbb{L}}(\xi\beta_i\beta_j).\end{aligned}$$

Thus we have

$$\Delta_A = \det((2\mathrm{Tr}_{\mathbb{L}}(\beta_i\beta_j))) \cdot \det((2\mathrm{Tr}_{\mathbb{L}}(\xi\beta_i\beta_j))) = 2^{2n}\Delta_{\mathbb{L}} \cdot \det((\mathrm{Tr}_{\mathbb{L}}(\xi\beta_i\beta_j))).$$

For $(\mathrm{Tr}_{\mathbb{L}}(\xi\beta_i\beta_j))$, write it as $(\sigma_j(\xi\beta_i))\cdot(\sigma_i(\beta_j))$, where $\{\sigma_i\}$ is $\mathrm{Hom}_{\mathbb{Q}}(\mathbb{L},\mathbb{C})$. Then

$$\begin{aligned}\det((\mathrm{Tr}_{\mathbb{L}}(\xi\beta_i\beta_j))) &= \det((\sigma_j(\xi\beta_i))) \cdot \det((\sigma_i(\beta_j)))\\ &= \mathrm{Nm}_{\mathbb{L}}(\xi) \cdot \det((\sigma_j(\beta_i))) \cdot \det((\sigma_i(\beta_j)))\\ &= \det((\sigma_j(\beta_i))) \cdot \det((\sigma_i(\beta_j)))\\ &= \det((\mathrm{Tr}_{\mathbb{L}}(\beta_i\beta_j))) = \Delta_{\mathbb{L}}.\end{aligned}$$

□

From the proof, we can see $\Delta_{\mathbb{F}} \le 2^{2n}\Delta_{\mathbb{L}}^2$. Note $n = \mathrm{poly}(\log \Delta_{\mathbb{L}})$, so $\log \Delta_{\mathbb{F}} = \mathrm{poly}(\log \Delta_{\mathbb{L}})$.

Appendix C Omitted Proofs

Lemma C.1 (Pseudo-HNF computation, [3, Theorem 34]) *Let $\mathbb{K}$ be a number field with ring of integers $\mathcal{O}_{\mathbb{K}}$. Given pseudo-generators of a full-rank $\mathcal{O}_{\mathbb{K}}$-module contained in $\mathcal{O}_{\mathbb{K}}^m$, one can compute a pseudo-HNF, and hence a pseudo-basis of the module, in classical polynomial time.*

Lemma C.2 (Almost-free representation, [9, Lemma 2.2]) *Given a pseudo-basis of a rank-m $\mathcal{O}_{\mathbb{K}}$-module $M \subseteq \mathbb{K}^m$, there is a probabilistic polynomial-time algorithm that outputs an almost-free representation*

$$M = \mathcal{O}_{\mathbb{K}}c_1 + \cdots + \mathcal{O}_{\mathbb{K}}c_{m-1} + \mathfrak{a}c_m,$$

where $c_1, \ldots, c_m \in \mathbb{K}^m$ are $\mathbb{K}$-linearly independent and $\mathfrak{a}$ is a fractional ideal of $\mathcal{O}_{\mathbb{K}}$.

Lemma C.3 (Change of pseudo-basis, [8, Proposition 1.4.2]) *Let $(w_i, \mathfrak{a}_i)_i$ and $(\eta_j, \mathfrak{b}_j)_j$ be two pseudo-bases of the same torsion-free module over a Dedekind domain, and let U be the transition matrix defined by*

$$(\eta_1, \ldots, \eta_m) = (w_1, \ldots, w_m)U.$$

Then, setting $\mathfrak{a} = \prod_i \mathfrak{a}_i$ and $\mathfrak{b} = \prod_j \mathfrak{b}_j$, one has

$$\mathfrak{a} = \det(U)\mathfrak{b}.$$

Lemma C.4 (Principal ideal problem, [4, Theorem 1.3]) *There is a quantum polynomial-time algorithm which decides whether a given ideal of an order in a number field is principal and, if it is principal, computes a generator of it. The same conclusion applies to fractional ideals after clearing denominators.*

Proof (of Lemma 2.6). Let $M \subseteq \mathbb{L}^m$ be a full-rank free $\mathcal{O}_\mathbb{L}$-module. We are given a $\mathbb{Z}$-basis of $\mathcal{O}_\mathbb{L}$ and a $\mathbb{Z}$-basis of M.

From the given $\mathbb{Z}$-basis of M, we first view M as a finitely generated $\mathcal{O}_\mathbb{L}$-module by closing the generators under multiplication by the basis elements of $\mathcal{O}_\mathbb{L}$. If necessary, we multiply by a common denominator $\delta \in \mathcal{O}_\mathbb{L} \setminus \{0\}$ so that $\delta M \subseteq \mathcal{O}_\mathbb{L}^m$. Applying Lemma C.1 to δM gives a pseudo-basis of δM in classical polynomial time; dividing the basis vectors by δ gives a pseudo-basis of M.

Next, applying Lemma C.2 to this pseudo-basis, we obtain in probabilistic polynomial time an almost-free representation

$$M = \mathcal{O}_\mathbb{L} c_1 + \cdots + \mathcal{O}_\mathbb{L} c_{m-1} + \mathfrak{a} c_m,$$

where $c_1, \ldots, c_m \in \mathbb{L}^m$ are $\mathbb{L}$-linearly independent and $\mathfrak{a}$ is a fractional ideal of $\mathcal{O}_\mathbb{L}$.

Since M is free of rank m, there exists a free basis $e_1, \ldots, e_m$ of M. Viewing it as a pseudo-basis, we have

$$M = \mathcal{O}_\mathbb{L} e_1 \oplus \cdots \oplus \mathcal{O}_\mathbb{L} e_m.$$

The almost-free representation above gives another pseudo-basis

$$M = \mathcal{O}_\mathbb{L} c_1 \oplus \cdots \oplus \mathcal{O}_\mathbb{L} c_{m-1} \oplus \mathfrak{a} c_m.$$

Let $U \in \mathrm{GL}_m(\mathbb{L})$ be the change-of-basis matrix such that

$$(c_1, \ldots, c_m) = (e_1, \ldots, e_m) U.$$

By Lemma C.3, the products of the coefficient ideals satisfy

$$\mathcal{O}_\mathbb{L} = \det(U)\mathfrak{a}.$$

Hence $\mathfrak{a} = \det(U)^{-1}\mathcal{O}_\mathbb{L}$, so $\mathfrak{a}$ is principal.

It remains to compute a generator of $\mathfrak{a}$. By Lemma C.4, after clearing denominators if necessary, this principal ideal generator can be computed in quantum polynomial time. Let $\alpha \in \mathbb{L}$ satisfy $\mathfrak{a} = (\alpha)$. Then

$$M = \mathcal{O}_\mathbb{L} c_1 + \cdots + \mathcal{O}_\mathbb{L} c_{m-1} + \mathcal{O}_\mathbb{L}(\alpha c_m).$$

Since $c_1, \ldots, c_m$ are $\mathbb{L}$-linearly independent, the vectors

$$c_1, \ldots, c_{m-1}, \alpha c_m$$

are also $\mathbb{L}$-linearly independent, and therefore form a free basis of M.

All steps are polynomial-time, and the only quantum step is the computation of a generator of the principal ideal $\mathfrak{a}$. This proves the lemma. □

Proof (of Lemma 2.7). By the hypothesis, for any $x, x' \in \mathsf{X}$ (resp. $x, x' \in \mathrm{Supp}(\mathcal{D}_{\mathsf{X}})$) and any $h \in \mathcal{I}(x, x')$, the distributions of $\mathsf{R}(h \star x)[2]$ and $\mathsf{R}(x)[2] \cdot h^{-1}$ are identical. Hence, for any fixed $g_0 \in G$,

$$\Pr[\mathsf{R}(h \star x)[2] = g_0 h^{-1}] = \Pr[\mathsf{R}(x)[2] \cdot h^{-1} = g_0 h^{-1}] = \Pr[\mathsf{R}(x)[2] = g_0].$$

For any $x, x' \in \mathsf{X}$ (resp. $x, x' \in \mathrm{Supp}(\mathcal{D}_{\mathsf{X}})$), and $(x'', \gamma) \leftarrow \mathsf{R}(x)$, assume $h \in \mathcal{I}(x, x')$. First,

$$\begin{aligned}
\Pr[\mathsf{R}(x)[1] = x''] &= \Pr[\mathsf{R}(x)[2] \in \mathcal{I}(x, x'')] \\
&= \sum_{f \in \mathcal{I}(x, x'')} \Pr[\mathsf{R}(x)[2] = f] \\
&= \sum_{f \in \mathcal{I}(x, x'')} \Pr[\mathsf{R}(h \star x)[2] = f \cdot h^{-1}] \\
&= \sum_{t \in \mathcal{I}(x', x'')} \Pr[\mathsf{R}(x')[2] = t] = \Pr[\mathsf{R}(x')[1] = x''].
\end{aligned}$$

The third equality uses the property, and the fourth equality uses $\mathcal{I}(x', x'') = \mathcal{I}(x, x'') \cdot h^{-1}$.
Second, for any $g, g' \in \mathcal{I}(x, x'')$, we have

$$\Pr[\mathsf{R}(x)[2] = g'] = \Pr[\mathsf{R}((g^{-1} \cdot g') \star x)[2] = g' \cdot (g^{-1} \cdot g')^{-1}] = \Pr[\mathsf{R}(x)[2] = g].$$

□

Proof (of Theorem 2.1). For an $\mathcal{O}_{\mathbb{L}}^2$-LIP instance $\mathbf{Q} = U^*U$, select v_1 from $\{(1,0)^T, (0,1)^T, (1,1)^T\}$ such that $U^{-1}(J_2 t_* u_1 - u_1) = U^{-1} J_2 t_* U v_1 - v_1$ does not belong to $\langle \Pi \rangle \mathcal{O}_{\mathbb{L}}^2$, where $U^{-1} J_2 t_* U$ is computed from $\mathbf{Q}$ via Proposition 2.1. Assuming $u_1 = U v_1$, we have $u_1 \not\equiv (0,0)^T, (1,1)^T \mod \langle \Pi \rangle \mathcal{O}_{\mathbb{L}}^2$; otherwise, $U^{-1}(J_2 t_* u_0 - u_0) \equiv (0,0) \mod \langle \Pi \rangle \mathcal{O}_{\mathbb{L}}^2$. Recall that $\mathfrak{p} = \langle \Pi \rangle$ has absolute norm 2, hence $\mathcal{O}_{\mathbb{L}}/\mathfrak{p} \simeq \mathbb{F}_2$. Thus there are only two residue classes modulo $\mathfrak{p}$. Without loss of generality, we can assume $u_1 = (\alpha_1, \beta_1)$ with $\alpha_1 \in 1 + \langle \Pi \rangle$ and $\beta_1 \in \langle \Pi \rangle$; otherwise, we replace U with $J_2 U$.

Suppose we have computed v_i such that $u_i := U v_i = (\alpha_i, \beta_i)$, where $\alpha_i \in 1 + \langle \Pi \rangle$ and $\beta_i \in \langle \Pi \rangle^i$ for some $i \geq 1$. Assume further that $\beta_i / \Pi^i \equiv c \mod \langle \Pi \rangle$ for some $c \in \{0, 1\}$. Consider the free module lattice M defined by

$$\begin{aligned}
M &= v_i \cdot \mathcal{O}_{\mathbb{L}} + \langle \Pi^{i+1} \rangle \mathcal{O}_{\mathbb{L}}^2 \\
&= U^{-1} \big(u_i \cdot \mathcal{O}_{\mathbb{L}} + \langle \Pi^{i+1} \rangle \mathcal{O}_{\mathbb{L}}^2 \big) \\
&= U^{-1} \big((1, c\Pi^i)^T \cdot \mathcal{O}_{\mathbb{L}} + (0, \Pi^{i+1})^T \cdot \mathcal{O}_{\mathbb{L}} \big) \\
&= U^{-1} B_{i+1}^{(c)} \cdot \mathcal{O}_{\mathbb{L}}^2.
\end{aligned}$$

Regarding the justification for the third equality. The left-hand side is contained in the right-hand side: since $c \cdot \alpha_i \Pi^i - \beta_i \in \langle \Pi \rangle^{i+1}$, there exists $t_i \in \mathcal{O}_{\mathbb{L}}$ such that

$$(1, c\Pi^i)^T \cdot \alpha_i - (0, \Pi^{i+1})^T \cdot t_i = u_i,$$

and

$$(\Pi^{i+1}, 0)^T = (1, c\Pi^i)^T \cdot \Pi^{i+1} - (0, \Pi^{i+1})^T \cdot c\Pi^i.$$

The right-hand side is contained in the left-hand side: it suffices to show that $(1, c\Pi^i)^T$ belongs to the left-hand side. Note that there exists $w_i \in 1 + \langle \Pi \rangle$ such that $w_i\alpha_i - 1 \in \langle \Pi^{i+1} \rangle$ (for example, take $w_i = \sum_{j=0}^{i}(1-\alpha_i)^j$). Moreover, there exists $s_i \in \mathcal{O}_{\mathbb{L}}$ such that $w_i\beta_i - c\Pi^i = s_i\Pi^{i+1}$. Hence,

$$(1, c\Pi^i)^T = u_i \cdot w_i - (0, \Pi^{i+1})^T \cdot s_i.$$

Using v_i, we can compute a $\mathbb{Z}$-basis of M. Then, by Lemma 2.6, we can compute a free basis $B = U^{-1}B_{i+1}^{(c)}V$ for some $V \in \mathrm{GL}_2(\mathcal{O}_{\mathbb{L}})$ in polynomial time on a quantum computer. Taking $\mathbf{Q}_{i+1} = B^*\mathbf{Q}B$ as the input to $\Delta_{i+1,\mathbb{L}}$, we can obtain the value of c.

Let $\widetilde{v}_i = U^{-1}J_2t_*Uv_i$. Define $v_{i+1} = v_i + c\Pi^i \cdot \widetilde{v}_i$. Assuming $Uv_{i+1} = (\alpha_{i+1}, \beta_{i+1})$, note that

$$Uv_{i+1} = Uv_i + c\Pi^i J_2 t_* Uv_i = (\alpha_i + c\Pi^i\beta_i^*,\ \beta_i - c\Pi^i\alpha_i^*).$$

Using the facts that $\alpha_i \in 1 + \langle \Pi \rangle$ and $\beta_i/\Pi^i \equiv c \mod \langle \Pi \rangle$, we obtain $\alpha_{i+1} \in 1 + \langle \Pi \rangle$ and $\beta_{i+1} \in \langle \Pi \rangle^{i+1}$.

Inductively, we obtain v_{2n^2}. Consider the module lattice $M = v_{2n^2} \cdot \mathcal{O}_{\mathbb{L}} + \langle \Pi^{2n^2} \rangle \mathcal{O}_{\mathbb{L}}^2$. Similar to the previous argument,

$$M = U^{-1}B_{2n^2}^{(0)}\mathcal{O}_{\mathbb{L}}^2 = U^{-1}\big((1,0)^T\mathcal{O}_{\mathbb{L}} \oplus (0, \Pi^{2n^2})^T\mathcal{O}_{\mathbb{L}}\big).$$

Moreover, since

$$\langle 2 \rangle = \mathfrak{p}^n, \qquad \mathfrak{p} = \langle \Pi \rangle,$$

we have

$$\langle \Pi^{2n^2} \rangle = \mathfrak{p}^{2n^2} = (\mathfrak{p}^n)^{2n} = \langle 2^{2n} \rangle.$$

Therefore

$$UM = (1,0)^T\mathcal{O}_{\mathbb{L}} \oplus (0, 2^{2n})^T\mathcal{O}_{\mathbb{L}}.$$

Compute an LLL-reduced basis (b_i) of M under the norm induced by $\mathbf{Q}$; then b_1 satisfies $\|Ub_1\| \le 2^{(2n-1)/2}\lambda_1(U \cdot M) = \sqrt{n}\,2^{(2n-1)/2}$. Hence, if $b_1 \in M$, then $Ub_1 = (a, 2^{2n}b)^T$ for some $a, b \in \mathcal{O}_{\mathbb{L}}$. Then $\|Ub_1\|^2 = \mathrm{tr}(a^*a) + \mathrm{tr}(2^{4n}b^*b)$, and from the previous upper bound we deduce $b = 0$. We compute

$$M_1 = b_1 \cdot \mathbb{L} \cap \mathcal{O}_{\mathbb{L}}^2 = U^{-1}\big((a,0)^T \cdot \mathbb{L} \cap U \cdot \mathcal{O}_{\mathbb{L}}^2\big) = \big(U^{-1}(1,0)^T\big) \cdot \mathcal{O}_{\mathbb{L}}$$

(by computing $\mathrm{span}_{\mathbb{Q}}(\mathrm{rot}(b_1)) \cap \mathbb{Z}^{2n}$). Using Proposition 2.2, input $\mathcal{O}_{\mathbb{L}}$, $\mathbf{Q}$, 1, and M_1 into `LS2` to obtain $U^{-1}(1,0)^T \cdot \mu(\mathbb{L})$. Apply $U^{-1}J_2t_*U$ to get $U^{-1}(0,1)^T \cdot \mu(\mathbb{L})$. By enumerating $\mu(\mathbb{L})$ (which can be done in polynomial time according to Lemma 2.4), we can compute U^{-1}. □

Lemma C.5 *$\forall x \in \mathfrak{p}$, $\|x\|^2 \in 2n\mathbb{Z}$. $\forall x \in \mathcal{O}_{\mathbb{L}}$, $\|x\|^2 = 2n$ iff $x = (1 + X^s)X^t$ for some $s, t \in \mathbb{Z}$. If $x_1 = (1 + X^{s_1})X^{t_1}$ and $x_2 = (1 + X^{s_2})X^{t_2}$ s.t. $x_1^*x_1 = x_2^*x_2$, then $x_1/x_2 \in \mu(\mathbb{L})$.*

Proof. Note that

$$\frac{1}{n}\|x\|^2 = \|\mathrm{vec}(x)\|^2 \equiv \left\langle \mathrm{vec}(x), \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \right\rangle \pmod 2 = \frac{1}{n}\left\langle x, 1 + X + \cdots + X^{n-1} \right\rangle.$$

In the 2-power cyclotomic setting, we have

$$\langle 1 + X \rangle = \langle 1 - X^{-1} \rangle.$$

Indeed,

$$(1 - X^{-1})(1 + X^{-1} + \cdots + X^{1-n}) = 1 - X^{-n} = 2,$$

so $2 \in \langle 1 - X^{-1} \rangle$, and hence

$$1 + X = X(1 - X^{-1}) + 2 \in \langle 1 - X^{-1} \rangle.$$

Conversely,

$$2 = (1 + X)(-X^{n-1} + X^{n-2} - \cdots + 1) \in \langle 1 + X \rangle$$

and $1 + X^{-1} = X^{-1}(1 + X) \in \langle 1 + X \rangle$, so

$$1 - X^{-1} = 2(-X^{-1}) + (1 + X^{-1}) \in \langle 1 + X \rangle.$$

Therefore the two ideals are equal. Thus, since $x \in \mathfrak{p} = \langle 1 + X \rangle$, there exists $b \in \mathcal{O}_{\mathbb{L}}$ such that $x = b(1 - X^{-1})$. Then

$$\begin{aligned} \frac{1}{n}\left\langle x, 1 + X + \cdots + X^{n-1} \right\rangle &= \frac{1}{n}\mathrm{Tr}_{\mathbb{L}}(b^*(1 - X)(1 + X + \cdots + X^{n-1})) \\ &= 2\frac{1}{n}\mathrm{Tr}_{\mathbb{L}}(b^*) \in 2\mathbb{Z}. \end{aligned}$$

Thus we have $\|x\|^2 \in 2n\mathbb{Z}$ and $\|x\|^2 = 2n$ iff $\|\mathrm{vec}(x)\|^2 = 2$ iff $\mathrm{vec}(x)$ has only two non-zero components, and they belong to ± 1 iff $x = (1 + X^s)X^t$ for some $s, t \in \mathbb{Z}$. If $x_1 = (1 + X^{s_1})X^{t_1}$ and $x_2 = (1 + X^{s_2})X^{t_2}$ s.t. $x_1^* x_1 = x_2^* x_2$, we have

$$(1 + X^{s_1})^*(1 + X^{s_1}) = (1 + X^{s_2})^*(1 + X^{s_2}),$$

which implies that $X^{s_1} + X^{-s_1} = X^{s_2} + X^{-s_2}$. Therefore, we can deduce that $\{X^{s_1}, X^{-s_1}\} = \{X^{s_2}, X^{-s_2}\}$ since they are both the root sets of the equation $x^2 - (X^{s_1} + X^{-s_1})x + 1 = 0$. It follows that $1 + X^{s_1} = 1 + X^{s_2}$ or $1 + X^{-s_2} = X^{-s_2}(1 + X^{s_2})$, and thus $x_1/x_2 \in \mu(\mathbb{L})$. □

Proof (of Lemma 3.1). When

$$U = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in GSL_2(\mathcal{O}_{\mathbb{L}})$$

such that $(\mathbf{B}_k^{(0)}U)^*(\mathbf{B}_k^{(0)}U) = (\mathbf{B}_k^{(0)})^*\mathbf{B}_k^{(0)}$, we first have

$$\left\| \begin{pmatrix} a \\ c(1+X)^k \end{pmatrix} \right\|^2 = \left\| \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\|^2 = n.$$

By Lemma C.5, $\|c(1+X)^k\|^2 \in 2n\mathbb{Z}$. Note that $\|a\|^2 = n\langle \mathrm{vec}(a), \mathrm{vec}(a)\rangle \in n\mathbb{Z}$. Thus $c = 0$ and $a \in \mu(\mathbb{L})$. We also have

$$b^*a = \begin{pmatrix} b \\ d(1+X)^k \end{pmatrix}^* \begin{pmatrix} a \\ c(1+X)^k \end{pmatrix} = \begin{pmatrix} 0 \\ (1+X)^k \end{pmatrix}^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 0.$$

Obviously, $ad = ad - bc \in \mu(\mathbb{L})$ implies that $d \in \mu(\mathbb{L})$. Thus

$$Aut_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)}) \subseteq \left\{ \begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L}) \right\}.$$

The reverse inclusion follows by direct verification. The statement for $\widetilde{\mathbf{Q}}_k^{(0)}$ follows by the same argument, equivalently by conjugating the above computation with

$$D_k = \mathrm{diag}\left(1, \left(\frac{1+X^2}{(1+X)^2}\right)^k\right) \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L}),$$

since $\widetilde{\mathbf{Q}}_k^{(0)} = D_k^* \mathbf{Q}_{2k}^{(0)} D_k$. □

Proof (of Lemma 3.2). Note

$$\widetilde{B}_k^{(1)} = \begin{pmatrix} 1 & 0 \\ (1+X^2)^{k-1} & (1+X^2)^k \end{pmatrix} = C_k M,$$

where

$$C_k = \begin{pmatrix} 1 & 1+X^2 \\ (1+X^2)^{k-1} & 0 \end{pmatrix}, \qquad M = \begin{pmatrix} 1 & 1+X^2 \\ 0 & -1 \end{pmatrix}.$$

It suffices to show that

$$Aut_{\mathcal{O}_\mathbb{L}}(C_k^* C_k) = \left\{ \begin{pmatrix} \xi_1 & 0 \\ \frac{\xi_2 - \xi_1}{1+X^2} & \xi_2 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L}), \xi_1/\xi_2 = X^{2j}, j \in \mathbb{Z} \right\}.$$

If there exists a matrix $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in GSL_2(\mathcal{O}_\mathbb{L})$ such that

$$(C_k U)^*(C_k U) = C_k^* C_k,$$

we first have

$$\left\| \begin{pmatrix} b + d(1+X^2) \\ b(1+X^2)^{k-1} \end{pmatrix} \right\|^2 = \left\| \begin{pmatrix} 1+X^2 \\ 0 \end{pmatrix} \right\|^2 = 2n.$$

By Lemma C.5, $\|b(1+X^2)^{k-1}\|^2 \in 2n\mathbb{Z}$. Thus we have

$$\|b(1+X^2)^{k-1}\|^2 = 0, \|b + d(1+X^2)\| = 2n$$

or

$$\|b(1+X^2)^{k-1}\|^2 = 2n, \|b+d(1+X^2)\| = 0.$$

Note that we further have

$$\begin{pmatrix} b+d(1+X^2) \\ b(1+X^2)^{k-1} \end{pmatrix}^* \begin{pmatrix} b+d(1+X^2) \\ b(1+X^2)^{k-1} \end{pmatrix} = \begin{pmatrix} 1+X^2 \\ 0 \end{pmatrix}^* \begin{pmatrix} 1+X^2 \\ 0 \end{pmatrix} = (1+X^2)^*(1+X^2),$$

and thus by Lemma C.5, we have either

$$b(1+X^2)^{k-1} = 0, \quad b+d(1+X^2) = (1+X^2)\xi_2,$$

or

$$b(1+X^2)^{k-1} = (1+X^2)\xi_2, \quad b+d(1+X^2) = 0$$

for some $\xi_2 \in \mu(\mathbb{L})$.

For the second case, since $k > 2$ and $1+X^2$ is not a unit in $\mathcal{O}_\mathbb{L}$, a contradiction arises. Hence $b = 0$ and $d = \xi_2$. Moreover,

$$\begin{aligned} \xi_2^*(1+X^2)^*(a+c(1+X^2)) &= \begin{pmatrix} b+d(1+X^2) \\ b(1+X^2)^k \end{pmatrix}^* \begin{pmatrix} a+c(1+X^2) \\ a(1+X^2)^{k-1} \end{pmatrix} \\ &= \begin{pmatrix} 1+X^2 \\ 0 \end{pmatrix}^* \begin{pmatrix} 1 \\ (1+X^2)^{k-1} \end{pmatrix} \\ &= (1+X^2)^*. \end{aligned}$$

Thus, $a + c(1+X^2) = \xi_2$. Since $ad - bc \in \mu(\mathbb{L})$ and $b = 0$, it follows that $ad \in \mu(\mathbb{L})$, which implies $a \in \mu(\mathbb{L})$. Denote a as ξ_1. Then we have $c = \frac{\xi_2 - \xi_1}{1+X^2}$. Since $c \in \mathcal{O}_\mathbb{L}$, we have $1 - \xi_1/\xi_2 \in \langle 1+X^2 \rangle = \mathfrak{p}^2 \Rightarrow \xi_1/\xi_2 = X^{2j}$ for some $j \in \mathbb{Z}$ (otherwise $\langle 1 - \xi_1/\xi_2 \rangle = \mathfrak{p}$, there is a contradiction). Therefore

$$Aut_{\mathcal{O}_\mathbb{L}}(C_k^* C_k) \subseteq \left\{ \begin{pmatrix} \xi_1 & 0 \\ \frac{\xi_2 - \xi_1}{1+X^2} & \xi_2 \end{pmatrix} | \xi_1, \xi_2 \in \mu(\mathbb{L}), \xi_1/\xi_2 = X^{2j}, j \in \mathbb{Z} \right\}.$$

The reverse inclusion follows by direct verification. Conjugating by M gives the stated formula for $Aut_{\mathcal{O}_\mathbb{L}}(\widetilde{\mathbf{Q}}_k^{(1)})$. □

Proof (of Proposition 3.1). We first compute $\ker(P_U - \xi_1 I_2) \cap \mathcal{O}_\mathbb{L}^2$ by Lemma 2.5 in polynomial time; it is

$$\begin{aligned} \ker(U^{-1} \begin{pmatrix} 0 & 0 \\ 0 & \xi_2 - \xi_1 \end{pmatrix} U) \cap \mathcal{O}_\mathbb{L}^2 &= U^{-1}(e_1\mathbb{L}) \cap U^{-1}\mathcal{O}_\mathbb{L}^2 = U^{-1}(e_1\mathbb{L} \cap \mathcal{O}_\mathbb{L}^2) \\ &= U^{-1}e_1\mathcal{O}_\mathbb{L} = (B_k^{(0)}U)^{-1}(B_k^{(0)}e_1\mathcal{O}_\mathbb{L}). \end{aligned}$$

where $e_1 = (1,0)^T$. Also we can compute $\tau_{(B_k^{(0)}U)} = (B_k^{(0)}U)^{-1}J_2 t_*(B_k^{(0)}U)$ by Proposition 2.1 using $\mathbf{Q}$.

Then using `LS2` (Proposition 2.2) for input $\mathcal{O}_\mathbb{L}$, $(B_k^{(0)}e_1)^*(B_k^{(0)}e_1)$, $(B_k^{(0)}U)^{-1}(B_k^{(0)}e_1\mathcal{O}_\mathbb{L})$, $(B_k^{(0)}U)^*(B_k^{(0)}U) = \mathbf{Q}$, we can find $\mu(\mathbb{L}) \cdot (B_k^{(0)}U)^{-1}(B_k^{(0)}e_1)$.

Next, for every $w \in \mu(\mathbb{L}) \cdot (B_k^{(0)}U)^{-1}(B_k^{(0)}e_1)$, we have

$$(B_k^{(0)}e_1 | J_2 t_* B_k^{(0)} e_1)(w|\tau_{(B_k^{(0)}U)}w)^{-1}.$$

Then we can find $B_k^{(0)}U$ when $w = U^{-1}e_1$, and hence compute U. □

Proof (of Proposition 3.2). Let $\xi := \xi_1 \cdot \xi_2$, and

$$\mathbb{F} := \begin{cases} \mathbb{L}[Y]/(Y^2 - \xi) \text{ if } Y^2 - \xi \text{ has no roots in } \mathbb{L}; \\ \mathbb{L} \text{ otherwise, and take } Y \text{ s.t. } Y^2 = \xi. \end{cases}$$

For the former, we know that $\{Y^i \mid i = 0, \cdots, 2n-1\}$ is a $\mathbb{Z}$-basis of $\mathcal{O}_\mathbb{F}$. Thus we can compute $\ker(P_U - Y) \cap \mathcal{O}_\mathbb{F}^2$ by Lemma 2.5 in polynomial time. We now identify this kernel explicitly. For any $x \in \mathbb{F}^2$,

$$x \in \ker(U^{-1}(P - YI_2)U) \iff (P - YI_2)(Ux) = 0 \iff Ux \in \ker(P - YI_2).$$

Hence

$$\ker(U^{-1}(P - YI_2)U) = U^{-1}(\ker(P - YI_2)).$$

Let $v = (\xi_1, Y)^T$. Since $Y^2 = \xi_1\xi_2$, one checks that $Pv = Yv$. Similarly, $(\xi_1, -Y)^T$ is an eigenvector with eigenvalue $-Y$. Therefore $\ker(P - YI_2) = v\mathbb{F}$, and hence

$$\ker(U^{-1}(P - YI_2)U) = U^{-1}(v\mathbb{F}).$$

Intersecting with $\mathcal{O}_\mathbb{F}^2$, and using $U \in GL_2(\mathcal{O}_\mathbb{L}) \subseteq GL_2(\mathcal{O}_\mathbb{F})$, so that $U^{-1}\mathcal{O}_\mathbb{F}^2 = \mathcal{O}_\mathbb{F}^2$, gives

$$\ker(U^{-1}(P - YI_2)U) \cap \mathcal{O}_\mathbb{F}^2 = U^{-1}(v\mathbb{F}) \cap U^{-1}\mathcal{O}_\mathbb{F}^2 = U^{-1}(v\mathbb{F} \cap \mathcal{O}_\mathbb{F}^2) = U^{-1}v\mathcal{O}_\mathbb{F}.$$

Also we can compute $U^{-1}J_2t_*U$ by Proposition 2.1 using U^*U. Then using `LS2` for input $\mathcal{O}_\mathbb{F}$, v^*v, $U^{-1}v\mathcal{O}_\mathbb{F}$ and U^*U, we can find $\mu(\mathbb{F}) \cdot U^{-1}v$. Next, for every $w \in \mu(\mathbb{F}) \cdot U^{-1}v$, we compute $(v|J_2t_*v)(w|U^{-1}J_2t_*Uw)^{-1}$. Then we can find the U when $w = U^{-1}v$. □

Proof (of Theorem 3.3). We will complete the proof in the following order. First, we prove that for all $\mathrm{poly}(n) \geq \ell \geq k$,

$$\text{module-LIP}_{[\mathbf{Q}_k^{(0)}]} \geq \text{module-LIP}_{[\mathbf{Q}_\ell^{(0)}]} \geq \text{module-}\Delta\text{LIP}_{\mathcal{D}_{[\mathbf{Q}_\ell^{(0)}]}, \mathcal{D}_{[\mathbf{Q}_\ell^{(1)}]}}.$$

Then, for the decision oracles required to solve $\mathcal{O}_\mathbb{L}^2$-LIP in Theorem 2.1, we implement the solver for module-ΔLIP$_{\mathcal{D}_{[\mathbf{Q}_\ell^{(0)}]}, \mathcal{D}_{[\mathbf{Q}_\ell^{(1)}]}}$ with $\ell \geq k$ using the oracle for module-LIP$_{[\mathbf{Q}_k^{(0)}]}$, while for $\ell < k$ we simply guess.
Since $k = O(\log n)$, as long as we have access to the oracle for module-LIP$_{[\mathbf{Q}_k^{(0)}]}$, we can provide all oracle outputs required by Theorem 2.1 to solve $\mathcal{O}_\mathbb{L}^2$-LIP in polynomial time with probability $1/2^k$, thereby achieving a successful solution.

The reduction module-$\mathrm{LIP}_{[\mathbf{Q}_\ell^{(0)}]} \geq$ module-$\Delta\mathrm{LIP}_{\mathcal{D}_{[\mathbf{Q}_\ell^{(0)}]},\mathcal{D}_{[\mathbf{Q}_\ell^{(1)}]}}$ is straightforward. One simply needs to feed the instance of module-$\Delta\mathrm{LIP}_{\mathcal{D}_{[\mathbf{Q}_\ell^{(0)}]},\mathcal{D}_{[\mathbf{Q}_\ell^{(1)}]}}$ to the solver for module-$\mathrm{LIP}_{[\mathbf{Q}_\ell^{(0)}]}$. If the solver succeeds, output 0; otherwise, output 1. Thus, it remains only to prove that for any $\mathrm{poly}(n) \geq \ell \geq k$, module-$\mathrm{LIP}_{[\mathbf{Q}_k^{(0)}]} \geq$ module-$\mathrm{LIP}_{[\mathbf{Q}_\ell^{(0)}]}$.

Given $U_0^*\mathbf{Q}_0^{(\ell)}U_0$, we first iteratively compute matrices M_i such that there exists $U_{i+1} \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L})$ satisfying

$$M_i = U_i^{-1}\begin{pmatrix} \Pi & 0 \\ 0 & 1 \end{pmatrix} U_{i+1}, \quad i = 0, \cdots, \ell - k - 1.$$

Define $\mathbf{G}_0 = U_0^*\mathbf{Q}_\ell^{(0)}U_0$ and $\mathbf{G}_i = \Pi^{-2}M_i^*\mathbf{G}_{i-1}M_i$. Then we have $\mathbf{G}_i = U_i^*\mathbf{Q}_{\ell-i}^{(0)}U_i$. Suppose M_{i-1} has been computed (let it be nothing if $i = 0$) and $\mathbf{G}_i$ is also known. Consider the set

$$\begin{aligned}\mathcal{L}_i = \{x \in \mathcal{O}_\mathbb{L}^2 \mid x^*\mathbf{G}_i x \in \mathfrak{p}\} &= U_i^{-1}\{x \in \mathcal{O}_\mathbb{L}^2 | x^*\mathbf{Q}_{\ell-i}^{(0)}x \in \mathfrak{p}\} \\ &= U_i^{-1}\left(\mathcal{O}_\mathbb{L}\begin{pmatrix}\Pi \\ 0\end{pmatrix} + \mathcal{O}_\mathbb{L}\begin{pmatrix}0 \\ 1\end{pmatrix}\right)\end{aligned}$$

is a free $\mathcal{O}_\mathbb{L}$ module. We first compute a $\mathbb{Z}$-basis of $\mathcal{L}_i$, and then compute the $\mathcal{O}_\mathbb{L}$-basis M_i of $\mathcal{L}_i$ by Lemma 2.6 in polynomial time on a quantum computer. By the property of free bases, there exists $U_{i+1} \in \mathrm{GL}_2(\mathcal{O}_\mathbb{L})$ such that

$$M_i = U_i^{-1}\begin{pmatrix} \Pi & 0 \\ 0 & 1 \end{pmatrix} U_{i+1}.$$

After computing $M_i = U_i^{-1}\begin{pmatrix} \Pi & 0 \\ 0 & 1 \end{pmatrix} U_{i+1}$ for $i = 0, \cdots, \ell-k-1$ and obtaining $\mathbf{G}_{\ell-k}$, inputting $\mathbf{G}_{\ell-k}$ into the solver for module-$\mathrm{LIP}_{[\mathbf{Q}_k^{(0)}]}$ will output an element $T_{\ell-k}$ of $U_{\ell-k}^{-1} \cdot \mathrm{Aut}_{\mathcal{O}_\mathbb{L}}(\mathbf{Q}_k^{(0)})$. For $k > 0$, $T_{\ell-k} = U_{\ell-k}^{-1}\begin{pmatrix}\xi_1 & 0 \\ 0 & \xi_2\end{pmatrix}$ for some $\xi_1, \xi_2 \in \mu(\mathbb{L})$.

For $k = 0$, we consider both the case of $T_{\ell-k}$ and the case of replacing $T_{\ell-k}$ with $T_{\ell-k}J_2$. In summary, we may assume that $T_{\ell-k}$ is of the form $U_{\ell-k}^{-1}\begin{pmatrix}\xi_1 & 0 \\ 0 & \xi_2\end{pmatrix}$. By recursively computing $T_i = M_i T_{i+1}\begin{pmatrix} \Pi & 0 \\ 0 & 1 \end{pmatrix}^{-1}$, we can recursively prove that $T_i = U_i^{-1}\begin{pmatrix}\xi_1 & 0 \\ 0 & \xi_2\end{pmatrix}$ (utilizing the commutativity of diagonal matrix multiplication) for $i = \ell - k, \cdots, 0$. Then T_0 is an isomorphism from $\mathbf{G}_0$ to $\mathbf{Q}_\ell^{(0)}$.

Regarding the computation of $\mathcal{L}_i$, there are multiple approaches. One can utilize the identity

$$(x+y)^*\mathbf{G}_i(x+y) = x^*\mathbf{G}_i x + y^*\mathbf{G}_i y \mod \mathfrak{p}$$

(since for any $z \in \mathcal{O}_{\mathbb{L}}$, $z + z^* \in \mathfrak{p}$) to transform it into solving a linear equation over $\mathbb{Z}_2$. Alternatively, one can repeatedly perform Gaussian sampling until vectors falling in $\mathcal{L}_i$ can generate $\mathcal{L}_i$ (determined by checking the determinant). □

Proof (of Lemma 3.3). It suffices to prove that the embedded pair is contained in

$$[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gl}}, \qquad [\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gl}}.$$

After viewing the two matrices defining $\Delta_{k,\mathbb{K}}$ over $\mathbb{L}$, they are exactly

$$\widetilde{B}_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & (1+X^2)^k \end{pmatrix}, \qquad \widetilde{B}_k^{(1)} = \begin{pmatrix} 1 & 0 \\ (1+X^2)^{k-1} & (1+X^2)^k \end{pmatrix}.$$

Since $\mathrm{SL}_2(\mathcal{O}_{\mathbb{K}}) \subseteq \mathrm{GL}_2(\mathcal{O}_{\mathbb{L}})$, the claim follows. □

Proof (of Lemma 4.1). By Lemma 2.1, the first step can be performed efficiently. By Lemma 2.2, Algorithm `HermiteSolve` is efficient. By Heuristic 1, the expected number of iterations for Steps 2–6 is approximately 4/3. Therefore, this is an efficient randomized algorithm.

Next, we consider the distribution of $\mathsf{R}_0(\mathbf{Q})[2]$. For any $\mathbf{U} = (u_1|u_2) \in \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$, let $\widetilde{u_2} = \mathtt{HermiteSolve}(\mathbb{L}, u_1)$ and $u_2' = \mathtt{Reduce}(\mathbf{Q}, u_1, \widetilde{u_2})$. Then, from the steps of the algorithm, it follows that the probability of sampling $\mathbf{U}$ equals the probability of obtaining u_1 via rejection sampling in Steps 1–4, multiplied by the probability that u_2 equals the value u_2' computed from the inputs f, g in Steps 5–7, that is,

$$\Pr[\mathsf{R}_0(\mathbf{Q})[2] = \mathbf{U}] = D_{\mathbf{Q},\sigma}(u_1)/D_{\mathbf{Q},\sigma}((\mathcal{O}_{\mathbb{L}}^2)^*) \cdot \Pr[u_2 = u_2'].$$

The first item is equal to $\rho_{\mathbf{Q},\sigma}(u_1)/\rho_{\mathbf{Q},\sigma}((\mathcal{O}_{\mathbb{L}}^2)^*) = \rho_{\mathbf{Q},\sigma}(u_1)/\rho_{\sigma}((\mathcal{O}_{\mathbb{L}}^2)^*)$. It remains to show $u_2 = u_2'$ iff u_2 is reduced with respect to u_1 and $\mathbf{Q}$.

If $u_2 = u_2' = \mathtt{Reduce}(\mathbf{Q}, u_1, \widetilde{u_2}) = \widetilde{u_2} - \left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot \widetilde{u_2}}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor \cdot u_1$, then

$$\left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot u_2}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor = \left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot \widetilde{u_2}}{u_1^* \cdot \mathbf{Q} \cdot u_1} - \left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot \widetilde{u_2}}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor \right\rfloor = 0.$$

Conversely, assume that $\left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot u_2}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor = 0$. By previous argument, $\left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot u_2'}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor$ is also 0. Note $(u_1|u_2), (u_1|u_2')$ are both in $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$, so $\det((u_1|u_2 - u_2')) = 0$. Hence there exists some $\lambda \in \mathbb{L}$ s.t. $u_2 - u_2' = \lambda u_1$. Since $u_1 \in (\mathcal{O}_{\mathbb{L}}^2)^*$ and $u_2, u_2' \in \mathcal{O}_{\mathbb{L}}^2$, we have

$$(\lambda) = (\lambda u_1[1]) + (\lambda u_1[2]) = ((u_2 - u_2')[1]) + ((u_2 - u_2')[2]) \subseteq \mathcal{O}_{\mathbb{L}}.$$

This means $\lambda \in \mathcal{O}_{\mathbb{L}}$. Thus $0 = \left\lceil \frac{u_1^* \cdot \mathbf{Q} \cdot (u_2 - u_2')}{u_1^* \cdot \mathbf{Q} \cdot u_1} \right\rfloor = \lceil \lambda \rfloor = \lambda$, i.e. $u_2 = u_2'$.

Finally, for any $\mathbf{U} = (u_1|u_2) \in \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$, we have

$$\Pr[\mathbf{V}^{-1} \cdot \mathsf{R}_0(\mathbf{Q})[2] = \mathbf{U}] = \Pr[\mathsf{R}_0(\mathbf{Q})[2] = \mathbf{V}\mathbf{U}].$$

Using the distribution expression derived above, we obtain

$$\Pr[\mathsf{R}_0(\mathbf{Q})[2] = \mathbf{VU}] = \begin{cases} \dfrac{\rho_{\mathbf{Q},\sigma}(\mathbf{V}u_1)}{\rho_{\mathbf{Q_0},\sigma}((\mathcal{O}_{\mathbb{L}}^2)^*)}, & \text{if } \mathbf{V}u_2 \text{ is reduced with respect} \\ & \text{to } \mathbf{V}u_1 \text{ and } \mathbf{Q}; \\ 0, \text{ otherwise,} \end{cases}$$

and

$$\Pr[\mathsf{R}_0(\mathbf{V}^*\mathbf{QV})[2] = \mathbf{U}] = \begin{cases} \dfrac{\rho_{\mathbf{V}^*\mathbf{QV},\sigma}(u_1)}{\rho_{\mathbf{Q_0},\sigma}((\mathcal{O}_{\mathbb{L}}^2)^*)}, & \text{if } u_2 \text{ is reduced with respect} \\ & \text{to } u_1 \text{ and } \mathbf{V}^*\mathbf{QV}; \\ 0, \text{ otherwise.} \end{cases}$$

Note $\rho_{\mathbf{Q},\sigma}(\mathbf{V}u_1) = \rho_{\mathbf{V}^*\mathbf{QV},\sigma}(u_1)$. We just need to show $\mathbf{V}u_2$ is reduced with respect to $\mathbf{V}u_1$ and $\mathbf{Q}$ iff u_2 is reduced with respect to u_1 and $\mathbf{V}^*\mathbf{QV}$. This follows directly from the definition. □

Proof (of Lemma 4.2). Write $\mathbf{U}$ as $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, then $\mathbf{X^{-1}UX} = \begin{pmatrix} a & \xi b \\ \xi^{-1}c & d \end{pmatrix}$ and $\mathbf{UY} = \begin{pmatrix} \xi a & \xi^{-1}b \\ \xi c & \xi^{-1}d \end{pmatrix}$.

Note

$$\begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} a \\ c \end{pmatrix} = \begin{pmatrix} \xi a \\ \xi c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} \xi a \\ \xi c \end{pmatrix}$$

and

$$\begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} \xi b \\ d \end{pmatrix} = \xi(\begin{pmatrix} a \\ c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} b \\ d \end{pmatrix}), \begin{pmatrix} \xi a \\ \xi c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} \xi^{-1}b \\ \xi^{-1}d \end{pmatrix} = \xi^{-2}(\begin{pmatrix} a \\ c \end{pmatrix}^* \mathbf{Q_0} \begin{pmatrix} b \\ d \end{pmatrix})$$

since $\mathbf{Q_0}$ is diagonal. Also note the action of multiplying by ξ can be viewed as a sign permutation on the coefficients and then $\forall f \in \mathbb{L}$, $\lceil f \rfloor = 0$ iff $\lceil \xi f \rfloor = 0$ iff $\lceil \xi^{-2} f \rfloor = 0$.

Hence $\rho_{\mathbf{Q_0},\sigma}(\begin{pmatrix} a \\ c \end{pmatrix}) = \rho_{\mathbf{Q_0},\sigma}(\begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix}) = \rho_{\mathbf{Q_0},\sigma}(\begin{pmatrix} \xi a \\ \xi c \end{pmatrix})$ and $\begin{pmatrix} \xi b \\ d \end{pmatrix}$ is reduced with respect to $\begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix}$ and $\mathbf{Q_0}$ iff $\begin{pmatrix} b \\ d \end{pmatrix}$ is reduced with respect to $\begin{pmatrix} a \\ c \end{pmatrix}$ and $\mathbf{Q_0}$ iff $\begin{pmatrix} \xi^{-1}b \\ \xi^{-1}d \end{pmatrix}$ is reduced with respect to $\begin{pmatrix} \xi a \\ \xi c \end{pmatrix}$ and $\mathbf{Q_0}$.

Thus, by Lemma 4.1, we obtain the desired conclusion. □

Proof (of Lemma 4.4). Homomorphic property: It is easy to see E is a well-defined surjective group homomorphism (note $\mathsf{E}(\mu(\mathbb{L})I_2) = \langle X^2 \rangle$). Uniformly extracting property on $[\mathbf{Q_k^{(0)}}]_{\text{gsl}}$ and $[\widetilde{\mathbf{Q}}_k^{(0)}]_{\text{gsl}}$: For any $\mathsf{S} \in \{[\mathbf{Q_k^{(0)}}]_{\text{gsl}}, [\widetilde{\mathbf{Q}}_k^{(0)}]_{\text{gsl}}\}$ and any $\mathbf{Q} \in \mathsf{S}$,

$$\mathsf{E}(\text{Stab}(\mathbf{Q})) = \det(Aut_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q}))/\langle X^2 \rangle = M$$

by Lemma 3.1. Thus, E is surjective even when restricted to the subgroup $\mathrm{Stab}(\mathbf{Q})$. Furthermore, $\mathsf{E}(U(\mathrm{Stab}(\mathbf{Q}))) = U(M)$ from the invariance of the $U(\mathrm{Stab}(\mathbf{Q}))$ to shifts by elements from that same group. Then for any $\mathsf{S} \in \{[\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}]_{\mathrm{gsl}}, [\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}\}$ and any $\mathbf{Q}, \mathbf{Q}' \in \mathsf{S}$, assume $\mathbf{Y} \in \mathcal{I}(\mathbf{Q}, \mathbf{Q}')$. We have

$$\mathsf{E}(U(\mathcal{I}(\mathbf{Q}, \mathbf{Q}'))) = \mathsf{E}(\mathbf{Y} \cdot U(\mathrm{Stab}(\mathbf{Q}))) = \mathsf{E}(\mathbf{Y}) \cdot U(M) = U(M).$$

Local constant property on $[\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}$: For any $\mathbf{Q} \in [\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}$, $\mathsf{E}(\mathrm{Stab}(\mathbf{Q})) = \det(Aut_{\mathcal{O}_{\mathbb{L}}}(\widetilde{\mathbf{Q}}_k^{(1)}))/\langle X^2 \rangle$ is trivial by Lemma 3.2. Since E is a homomorphism, for any $\mathbf{Q_1}, \mathbf{Q_2} \in [\widetilde{\mathbf{Q}}_k^{(1)}]_{\mathrm{gsl}}$, $\mathsf{E}(\mathcal{I}(\mathbf{Q_1}, \mathbf{Q_2}))$ is a constant.
Trapdoor property on $[\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}]_{\mathrm{gsl}}$ and $[\widetilde{\mathbf{Q}}_k^{(0)}]_{\mathrm{gsl}}$: By the argument presented in Lemma 2.8, the result follows. □

Appendix D Discussion about extractors over $\mathbf{SL_2}(\boldsymbol{\mathcal{O}}_{\mathbb{L}})$

In this section, we focus on proving the following theorem. Regarding the extractor as a group homomorphism is motivated by two factors: first, it is required for constructing an enhanced linkable commitment; second, it represents a natural methodology for extractor construction itself. Our proof reveals that the feasibility of finding such an extractor is intimately connected to the structure of the stabilizer subgroup of the group action—specifically, the automorphism group of a module lattice. In this section, we consider the number field $\mathbb{L} = \mathbb{Q}[X]/(X^{2^{d-1}} + 1)$, where $d > 3$.

Theorem D.1 *For the group action $(SL_2(\mathcal{O}_{\mathbb{L}}), [\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}]_{sl}, \star)$, $k \geq 0$, there exists no extractor satisfying* ***the uniformly extracting property and homomorphic property****.*

To complete the proof, we need to introduce some structural properties of $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$ and $Aut_{\mathcal{O}_{\mathbb{L}}}(\mathbf{Q}_k^{(0)})$.

Definition D.1 *For R a unital ring, we denote by $E_n(R)$ the subgroup of $GL_n(R)$ generated by all transvections $e_{ij}(r) = I_n + r\epsilon_{ij}$ (a.k.a. the elementary matrices over R) with $r \in R$, $1 \leq i \neq j \leq n$, where I_n is the identity matrix and ϵ_{ij} is the matrix whose (i, j) entry is 1 while all its other entries are zero.*

Lemma D.1 (induced by [34]) *$SL_2(\mathcal{O}_{\mathbb{L}}) = E_2(\mathcal{O}_{\mathbb{L}})$. Here we make use of the fact that $\mathcal{O}_{\mathbb{L}}^{\times}$ is infinite.*

From this, we easily obtain the following lemma.

Lemma D.2 *The derived subgroup of $SL_2(\mathcal{O}_{\mathbb{L}})$ is itself, i.e. $[SL_2(\mathcal{O}_{\mathbb{L}}), SL_2(\mathcal{O}_{\mathbb{L}})] = SL_2(\mathcal{O}_{\mathbb{L}})$.*

Proof. Since $e_{ik}(s) = [e_{ij}(1), e_{jk}(s)] \in [\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})]$, $\forall s \in \mathcal{O}_{\mathbb{L}}$, $E_2(\mathcal{O}_{\mathbb{L}}) \subseteq [\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})]$. By Lemma D.1, $\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}) = E_2(\mathcal{O}_{\mathbb{L}}) \subseteq [\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})]$. Hence $[\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})] = \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$. □

Lemma D.3 *Let* $g = \begin{pmatrix} \xi & 0 \\ 0 & \xi^{-1} \end{pmatrix}$. *For*

$$SU_2(\mathcal{O}_{\mathbb{L}}) := \left\{ \begin{pmatrix} \xi & 0 \\ 0 & \xi^{-1} \end{pmatrix} \;\middle|\; \xi \in \mu(\mathbb{L}) \right\} \cup \left\{ \begin{pmatrix} 0 & \xi \\ -\xi^{-1} & 0 \end{pmatrix} \;\middle|\; \xi \in \mu(\mathbb{L}) \right\} = \langle g, J_2 \rangle$$

, the normal subgroups of $SU_2(\mathcal{O}_{\mathbb{L}})$ *include* $\left\langle g^{2^i} \right\rangle$ *for* $i \in \mathbb{Z}$, *as well as* $\langle J_2, g^2 \rangle$, $\langle gJ_2, g^2 \rangle$ *and* $SU_2(\mathcal{O}_{\mathbb{L}})$ *itself. Moreover, the derived subgroup* $[SU_2(\mathcal{O}_{\mathbb{L}}), SU_2(\mathcal{O}_{\mathbb{L}})]$ *of* $SU_2(\mathcal{O}_{\mathbb{L}})$ *is* $\langle g^2 \rangle$.

Proof. The key observation is that $gJ_2 = J_2 g^{-1}$. Let H be a normal subgroup of $\mathrm{SU}_2(\mathcal{O}_{\mathbb{L}})$. If H contains no anti-diagonal matrix, then $H \subseteq \langle g \rangle$ is isomorphic to $C_{2^{d-1}}$, which is also cyclic and generated by some $\begin{pmatrix} X^{2^i} & 0 \\ 0 & X^{-2^i} \end{pmatrix}$. If H contains an anti-diagonal matrix, which can be written as $g^k J_2$, then $g^{k+2} J_2 = g g^k J_2 g^{-1} \in H$, and consequently $g^2 = g^{k+2} J_2 (g^k J_2)^{-1} \in H$. If H contains both J_2 and g, then $H = \mathrm{SU}_2(\mathcal{O}_{\mathbb{L}})$; if H contains J_2 but not g, then $H = \langle g^2, J_2 \rangle$; if H contains neither J_2 nor g, then k is odd, and consequently $H = \langle gJ_2, g^2 \rangle$. Among these normal subgroups, $\langle g^2 \rangle$ is the minimal normal subgroup that makes the quotient group abelian; therefore, $[\mathrm{SU}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SU}_2(\mathcal{O}_{\mathbb{L}})] = \langle g^2 \rangle$. □

Proof (of Theorem D.1). Suppose such an extractor $\mathsf{E} : \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}) \to M$ exists. It is a group homomorphism, and the uniformly extracting property is equivalent to the restriction of E to $\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})$ being surjective. Continuing with $g = \begin{pmatrix} X & 0 \\ 0 & X^{-1} \end{pmatrix}$, by Lemma 3.1, for $k > 0$, $\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}) = \langle g \rangle$; and by Lemma 2.3, $\mathrm{Stab}(\mathbf{Q}_{\mathbf{0}}^{(\mathbf{0})}) = \mathrm{SU}_2(\mathcal{O}_{\mathbb{L}})$.

We have the following equality:

$$\begin{aligned} M &= \mathsf{E}(\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})) = \mathsf{E}([\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}), \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})]) = [\mathsf{E}(\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})), \mathsf{E}(\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}))] \\ &= [\mathsf{E}(\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})), \mathsf{E}(\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}))] = \mathsf{E}([\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})]). \end{aligned}$$

For $k > 0$, the $[\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})]$ is trivial, while $|M| > 1$, a contradiction. For $k = 0$, by the previous lemma, note that any normal subgroup of $\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})$ either contains $[\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})]$ or is contained in $[\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})]$. Therefore, either $\ker(\mathsf{E}|_{\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})}) = \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})$ or $\mathsf{E}([\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})])$ is strictly contained in $\mathsf{E}(\mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}))$. The former implies that M is trivial, a contradiction; the latter contradicts the fact that $M = \mathsf{E}([\mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})}), \mathrm{Stab}(\mathbf{Q}_{\mathbf{k}}^{(\mathbf{0})})])$. □

Appendix E Size after `Reduce`

We justify the size claim for the second column produced by the reduction step. The goal is to show that

$$u_2 = \mathtt{Reduce}(\mathbf{Q}, u_1, \widetilde{u}_2)$$

has polynomially bounded coefficients, after adding only a mild rejection condition on $(u_1^*\mathbf{Q}u_1)^{-1}$.

Throughout this appendix, write

$$\alpha = \Pi\Pi^*, \qquad \mathbf{Q}_0 = \mathbf{Q}_k^{(0)} = \begin{pmatrix} 1 & 0 \\ 0 & \alpha^k \end{pmatrix}, \qquad \mathbf{Q} = B^*\mathbf{Q}_0 B,$$

where

$$B \in \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}).$$

We work in the size regime considered in the main text: the coefficients of B are polynomially bounded, and the Gaussian parameter used to sample u_1 is polynomially bounded. Since $B \in \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}})$, if

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

then

$$B^{-1} = \begin{pmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{pmatrix}.$$

Thus B^{-1} has the same coefficient-size bound as B.

Therefore, to bound u_2, it is enough to bound Bu_2. Indeed,

$$u_2 = B^{-1}Bu_2,$$

and multiplication by B^{-1} has polynomially bounded operator norm under the above size assumption on B. Hence a polynomial bound on $\|Bu_2\|$ implies a polynomial bound on $\|u_2\|$. Since $u_2 \in \mathcal{O}_{\mathbb{L}}^2$, this gives polynomially bounded coefficients, or equivalently $O(\log n)$ bits per coefficient in the polynomial-size parameter regime.

We now bound Bu_2. Write

$$Bu_1 = \begin{pmatrix} x \\ y \end{pmatrix}.$$

Then

$$q := u_1^*\mathbf{Q}u_1 = (Bu_1)^*\mathbf{Q}_0(Bu_1) = x^*x + \alpha^k y^*y.$$

First, we recall why Bu_1 itself is already controlled by the Gaussian sampling step. The map

$$u \longmapsto Bu$$

is a bijection of $\mathcal{O}_{\mathbb{L}}^2$, and also of $(\mathcal{O}_{\mathbb{L}}^2)^*$. Moreover,

$$u^*\mathbf{Q}u = (Bu)^*\mathbf{Q}_0(Bu).$$

Thus, up to the usual conditioning on primitivity, Bu_1 is sampled from the discrete Gaussian associated with $\mathbf{Q}_0$. Standard Gaussian tail bounds give, except with negligible probability,

$$\|Bu_1\|_{\mathbf{Q}_0} = O(\sigma\sqrt{n}),$$

where

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_{\mathbf{Q}_0}^2 = \mathrm{Tr}_{\mathbb{L}}(xx^* + \alpha^k yy^*).$$

This is a bound in the $\mathbf{Q}_0$-norm, so we convert it to the ordinary canonical norm. We have

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|^2 = \mathrm{Tr}_{\mathbb{L}}(xx^* + yy^*).$$

Let

$$\Lambda_k := \|\alpha^{-k}\|_\infty = \max_{\sigma:\mathbb{L}\hookrightarrow\mathbb{C}} \sigma(\alpha)^{-k}.$$

Then, embedding-wise,

$$yy^* \le \Lambda_k \, \alpha^k yy^*.$$

Consequently,

$$\left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|^2 \le \max\{1, \Lambda_k\} \mathrm{Tr}_{\mathbb{L}}(xx^* + \alpha^k yy^*) = \max\{1, \Lambda_k\} \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_{\mathbf{Q}_0}^2.$$

Hence

$$\|Bu_1\| = O\Big(\sigma\sqrt{n} \cdot \max\{1, \Lambda_k\}^{1/2}\Big).$$

For the power-of-two cyclotomic fields used here,

$$\|\alpha^{-1}\|_\infty = n^{O(1)}.$$

Therefore

$$\Lambda_k = \|\alpha^{-k}\|_\infty = n^{O(k)},$$

and so

$$\|Bu_1\| = O\Big(\sigma\sqrt{n} \cdot n^{O(k)}\Big).$$

Under the parameter choices of the main text, this is polynomial in n.

Next define

$$u_1^\perp := B^{-1} \begin{pmatrix} -\alpha^k y^* \\ x^* \end{pmatrix}.$$

Since $B^{-1} \in M_2(\mathcal{O}_{\mathbb{L}})$, we have

$$u_1^\perp \in \mathcal{O}_{\mathbb{L}}^2.$$

Moreover,

$$u_1^* \mathbf{Q} u_1^\perp = \begin{pmatrix} x \\ y \end{pmatrix}^* \mathbf{Q}_0 \begin{pmatrix} -\alpha^k y^* \\ x^* \end{pmatrix} = -x^*\alpha^k y^* + y^*\alpha^k x^* = 0.$$

Also, we have

$$\det(u_1, u_1^\perp) = \det \begin{pmatrix} x & -\alpha^k y^* \\ y & x^* \end{pmatrix} = xx^* + \alpha^k yy^* = q.$$

Now let

$$u_2 = \texttt{Reduce}(\mathbf{Q}, u_1, \widetilde{u}_2), \qquad \det(u_1, \widetilde{u}_2) = 1.$$

The reduction step only subtracts a multiple of u_1, hence it preserves the determinant condition:

$$\det(u_1, u_2) = 1.$$

Since $(u_1, u_1^{\perp})$ is an $\mathbb{L}$-basis of $\mathbb{L}^2$, we can write

$$u_2 = a u_1 + b u_1^{\perp} \qquad \text{for some } a, b \in \mathbb{L}.$$

Taking determinants with first column u_1, we obtain

$$1 = \det(u_1, u_2) = b \det(u_1, u_1^{\perp}) = bq.$$

Thus

$$b = q^{-1} = (u_1^* \mathbf{Q} u_1)^{-1}.$$

Multiplying the decomposition of u_2 by B, we get

$$Bu_2 = aBu_1 + bBu_1^{\perp} = a \begin{pmatrix} x \\ y \end{pmatrix} + b \begin{pmatrix} -\alpha^k y^* \\ x^* \end{pmatrix}.$$

We now bound the two terms separately.

First consider aBu_1. Since $u_1^* \mathbf{Q} u_1^{\perp} = 0$, the coefficient a is

$$a = \frac{u_1^* \mathbf{Q} u_2}{u_1^* \mathbf{Q} u_1}.$$

By the definition of `Reduce`, the output is reduced with respect to u_1 and $\mathbf{Q}$, so

$$\lceil a \rfloor = 0.$$

Hence a lies in the fixed rounding fundamental region. Multiplication by such an element has polynomially bounded operator norm in the power basis. Using the above bound on Bu_1, we obtain

$$\|aBu_1\| = O\Big(\sigma\sqrt{n} \cdot n^{O(k)}\Big),$$

up to an additional polynomial factor from the rounding region. This is polynomial in the parameter regime considered here.

It remains to bound $bBu_1^{\perp}$. Since $q = q^*$ is totally positive, also $b = b^*$. Therefore

$$\|bBu_1^{\perp}\|^2 = \mathrm{Tr}_{\mathbb{L}}\left(bb^*\left(xx^* + \alpha^{2k} yy^*\right)\right).$$

For every embedding, $0 \le \sigma(\alpha) \le 4$. Hence, embedding-wise,

$$xx^* + \alpha^{2k} yy^* \le 4^k \left(xx^* + \alpha^k yy^*\right) = 4^k q.$$

It follows that

$$\|bBu_1^{\perp}\|^2 \le 4^k \mathrm{Tr}_{\mathbb{L}}(bb^* q) = 4^k \mathrm{Tr}_{\mathbb{L}}(q^{-1}).$$

Thus the only quantity not directly controlled by the Gaussian variance is

$$q^{-1} = (u_1^* \mathbf{Q} u_1)^{-1}.$$

We therefore add the rejection condition

$$\mathrm{Tr}_{\mathbb{L}}\left((u_1^* \mathbf{Q} u_1)^{-1}\right) \le P(n)$$

for some fixed polynomial P. Under this condition,

$$\|bBu_1^{\perp}\| \le 2^k \sqrt{P(n)}.$$

Combining the two estimates, we get

$$\|Bu_2\| = O\Big(\sigma\sqrt{n} \cdot n^{O(k)} + 2^k \sqrt{P(n)}\Big),$$

up to polynomial factors coming from the rounding fundamental region. Under the parameter choices of the main text, this is $O(\mathrm{poly}(n))$. Since $u_2 = B^{-1}Bu_2$ and B^{-1} has polynomially bounded coefficients, we conclude that

$$\|u_2\| = O(\mathrm{poly}(n)).$$

Therefore $u_2 \in \mathcal{O}_{\mathbb{L}}^2$ has polynomially bounded coefficients. Equivalently, each coefficient has $O(\log n)$ bits in the polynomial-size parameter regime, and the total representation size of u_2 is $O(n \log n)$.

In practice, this additional rejection condition appears to be very mild. For the Gaussian samples used in the algorithm, the scalar

$$q = u_1^* \mathbf{Q} u_1$$

is typically not close to zero in any embedding, so $\mathrm{Tr}_{\mathbb{L}}(q^{-1})$ is usually small. Thus the rejection condition should be viewed mainly as a safeguard for worst-case size accounting, rather than as a condition expected to dominate the running time.

Compatibility with re-randomization. Let $\mathsf{R}_{0,\mathrm{rej}}$ denote the variant of R_0 which restarts unless the sampled first column u satisfies

$$\mathsf{Good}_{\mathbf{Q}}(u) \iff \mathrm{Tr}_{\mathbb{L}}\left((u^* \mathbf{Q} u)^{-1}\right) \le P(n).$$

This rejection predicate does not affect the equivariance properties used by the re-randomization algorithm.

First, it preserves the second item of Lemma 4.1. For every

$$\mathbf{V} \in \mathrm{SL}_2(\mathcal{O}_{\mathbb{L}}),$$

we have

$$(\mathbf{V}u)^* \mathbf{Q} (\mathbf{V}u) = u^* (\mathbf{V}^* \mathbf{Q} \mathbf{V}) u.$$

Therefore

$$\mathsf{Good}_{\mathbf{Q}}(\mathbf{V}u) \iff \mathsf{Good}_{\mathbf{V}^*\mathbf{Q}\mathbf{V}}(u).$$

Since $u \mapsto \mathbf{V}u$ is a bijection of $(\mathcal{O}^2_{\mathbb{L}})^*$, the conditioned normalizing mass is transported exactly as in the original proof. The Gaussian weights and the reducedness condition are transported by the same change of variables. Hence the second item of Lemma 4.1 remains valid for $\mathsf{R}_{0,\mathrm{rej}}$.

Second, the rejection predicate also preserves Lemma 4.2. In that lemma, one uses root-of-unity transformations of the form

$$\mathbf{X} = \begin{pmatrix} 1 & 0 \\ 0 & \xi \end{pmatrix}, \qquad \mathbf{Y} = \begin{pmatrix} \xi & 0 \\ 0 & \xi^{-1} \end{pmatrix}, \qquad \xi \in \mu(\mathbb{L}).$$

If the first column of $\mathbf{U}$ is $(a, c)^T$, then the first columns of $\mathbf{X}^{-1}\mathbf{U}\mathbf{X}$, $\mathbf{U}$, and $\mathbf{U}\mathbf{Y}$ are respectively

$$\begin{pmatrix} a \\ \xi^{-1}c \end{pmatrix}, \qquad \begin{pmatrix} a \\ c \end{pmatrix}, \qquad \begin{pmatrix} \xi a \\ \xi c \end{pmatrix}.$$

Since $\xi^*\xi = 1$ and $\mathbf{Q}_0$ is diagonal, these three columns give the same value of

$$u^*\mathbf{Q}_0 u.$$

Therefore $\mathsf{Good}_{\mathbf{Q}_0}(u)$ is invariant under the same transformations used in Lemma 4.2. The proof of that lemma goes through unchanged.

Consequently, the proof of the re-randomization algorithm is unaffected by adding this rejection step. The rejection changes only the conditioned marginal distribution of the sampled first column; it does not change the equivariance identities required for re-randomization.