Cryptanalysis of Rank-2 Module-LIP with Symplectic Automorphisms

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

[1] Key Laboratory of Mathematics Mechanization, 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

Abstract. At Eurocrypt'24, Mureau et al. formally defined the Lattice Isomorphism Problem for module lattices (module-LIP) in a number field $\mathbb{K}$, and proposed a heuristic randomized algorithm solving module-LIP for modules of rank 2 in $\mathbb{K}^2$ with a totally real number field $\mathbb{K}$, which runs in classical polynomial time for a large class of modules and a large class of totally real number field under some reasonable number theoretic assumptions. In this paper, by introducing a (pseudo) symplectic automorphism of the module, we successfully reduce the problem of solving module-LIP over CM number field to the problem of finding certain symplectic automorphism. Furthermore, we show that a weak (pseudo) symplectic automorphism can be computed efficiently, which immediately turns out to be the desired automorphism when the module is in a totally real number field. This directly results in a provable deterministic polynomial-time algorithm solving module-LIP for rank-2 modules in $\mathbb{K}^2$ where $\mathbb{K}$ is a totally real number field, without any assumptions or restrictions on the modules and the totally real number fields. Moreover, the weak symplectic automorphism can also be utilized to invalidate the omSVP assumption employed in HAWK's forgery security analysis, although it does not yield any actual attacks against HAWK itself.

Keywords: Lattice automorphism · module-LIP · Symplectic matrix

1 Introduction

Lattices are discrete additive subgroups of $\mathbb{R}^m$, which provide rich geometric structures that can be used to define various computationally hard problems, such as the famous shortest vector problem (SVP) and the closest vector problem (CVP). Based on the hardness of these problems or their variants, lots of lattice-based cryptosystems have been constructed. It is widely believed that lattice-based cryptosystems are quantum-resistant, and some of them are selected as the standard algorithms in NIST's Post-Quantum Cryptography Standardization Project.

Lattice Isomorphism Problem (LIP) is another lattice-related computational problem. Two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are said to be isomorphic if there exists a bijective orthogonal transformation from $\mathcal{L}_1$ to $\mathcal{L}_2$. The search version of LIP refers to the question of finding such orthogonal transformation given the lattice bases of $\mathcal{L}_1$ and $\mathcal{L}_2$, and the decision version asks to determine whether the two given lattices are isomorphic or not. Research on LIP dates back to [22] in the 1990s, in which the LIP for low-dimensional lattices was considered. In [14], Haviv and Regev proposed an $n^{O(n)}$-time algorithm for solving the general LIP, which remains the fastest known algorithm for LIP. Since then, many more cryptanalytic works have been proposed [12,7,10,3,8,9,13,17,18,23,6].

Most of these works focus on a special case of LIP, namely, $\mathbb{Z}$LIP, in which $\mathcal{L}$ is the hypercubic lattice $\mathbb{Z}^n$, such as [12,4]. Recently, Ducas [6] explored a reduction from n-dimensional $\mathbb{Z}$LIP to $\frac{n}{2}$-dimensional SVP, which means that $\mathbb{Z}$LIP can be solved with $2^{n/2}$ time complexity due to the best provable algorithm [1] for SVP. A similar algorithmic result can be concluded by employing Bennett et al.'s reduction [3] from $\mathbb{Z}$SVP to $O(1)$-uSVP with the well-known reduction from $\mathbb{Z}$LIP to $\mathbb{Z}$SVP.

To improve the efficiency of LIP-based cryptosystems, an algebraic variant of LIP, called module-LIP problem, was introduced by Ducas et al. [8], where the module can be chosen as free module over a CM number field instead of just the ring of integers. By taking the module M as $\mathcal{O}_{\mathbb{L}}^2$ where the field $\mathbb{L}$ is a cyclotomic field with conductor being a power of 2, Ducas et al. [8] presented a signature scheme called HAWK, whose security relies on the hardness of $\mathcal{O}_{\mathbb{L}}^2$-LIP problem. HAWK is now a candidate algorithms in the first round of NIST's Post-Quantum Cryptography Standardization Project for additional digital signature proposals. However, in spite of the additional algebraic structure, we always treat $\mathcal{O}_{\mathbb{L}}^2$-LIP problem as an LIP on non-structured lattices when analyzing the security of HAWK. Hence, a natural problem is how to solve module-LIP more efficiently than LIP with its special algebraic structure.

For LIP with algebraic structures, the most well-known algorithm originates from the work of Gentry and Szydlo [13], which tries to recover the secret key of NTRUSign [15] by solving some special $\mathbb{Z}$LIP instance with algebraic structure. Later, Lenstra and Silverberg made [17,18] lots of in-depth analysis of the Gentry-Szydlo algorithm, and Lenstra and Silverberg [19] generalized it to check isomorphism of lattices over CM-orders. The essence of these algorithms lies in using sufficient lattice automorphisms to solve LIP. Note that the automorphisms provided by the algebraic structure of rank-1 module enable the Gentry-Szydlo algorithm and its variants to solve the corresponding rank-1 module-LIP problems over CM number fields.

At Eurocrypt'24, Mureau et al. [20] formalized the framework for module-LIP using the concept of pseudo-bases, allowing it to be defined on module lattices over general number fields. Furthermore, as their main technical contribution, they presented a heuristic algorithm for solving module-LIP when the module

$M \subset \mathbb{K}^2$ has rank 2 and when the number field $\mathbb{K}$ is a totally real number field. Roughly speaking, the strategy in [20] mainly utilizes the prime decomposition of the principal ideal generated by the sum of squares x^2+y^2 to guess the principal ideal generated by its factor $x+i\cdot y$. To avoid factoring a general integer during the process of prime ideal factorization, which is still hard on classical computers by now, the ideals should be selected carefully such that their norms are easy to be factored under some heuristic assumption. However, guessing the desired principal ideal by enumerating the possible combinations of prime ideals, still makes the time complexity of the final algorithm exponential in the number of distinct prime ideals factors (Theorem 4.6 in [20]). Therefore, the algorithm in [20] runs in polynomial time under some reasonable heuristic assumptions for a class of certain module-LIP, which relates to the arithmetic properties of the module and the field. It should be noted that their algorithm does not impact the security of HAWK, as pointed out in [20].

1.1 Our Contributions

In this paper, we present a provable deterministic polynomial-time algorithm to solve $\mathcal{O}_{\mathbb{L}}^2$-LIP where $\mathbb{L}$ is a CM number field, with the help of a new module lattice automorphism defined by a symplectic matrix with rank 2. Therefore, we reduce the problem of solving $\mathcal{O}_{\mathbb{L}}^2$-LIP to the problem of finding out the certain module lattice symplectic automorphism. Although it seems not easy to find the exact symplectic automorphism in general, we can compute another weak module lattice symplectic automorphism for $\mathcal{O}_{\mathbb{L}}^2$ efficiently when $\mathbb{L}$ is a CM number field. Specially, the weak symplectic automorphism will become a module lattice automorphism immediately when a totally real number field $\mathbb{K}$ is considered, which directly yields a provable deterministic polynomial-time algorithm solving $\mathcal{O}_{\mathbb{K}}^2$-LIP where $\mathbb{K}$ is a totally real number field.

Note that the forgery security of HAWK [8] is based on the hardness of the one more SVP (omSVP), which asks the adversary to find one more short enough non-trivial element in $\mathcal{O}_{\mathbb{L}}^2$ that is out of the trivial set $\{\alpha x\}_{\alpha \in \mu(\mathbb{L})}$ where x is a given short element. However, our weak symplectic automorphism φ, which can be computed efficiently, will yield another non-trivial short element $\varphi(x)$ directly, whose length is as the same as x's. This invalidates the omSVP assumption used in HAWK's forgery security analysis, although it does not yield any actual attacks against HAWK itself. An easy way to fix this issue is just adjusting the omSVP assumption by adding the new short elements we find into the trivial set.

We also generalize the algorithm to solve module-LIP for the rank-2 module $M \subset \mathbb{K}^2$ where the number field $\mathbb{K}$ is a totally real number field. By introducing a similar pseudo symplectic automorphism and utilizing eigenspaces to acquire isomorphism invariants, we have the following theorem.

Theorem 1.1 (informal) *Let $\mathbb{K}$ be a totally real number field. $M \subseteq \mathbb{K}^2$ is an module lattice of rank 2 with pseudobasis $\mathbf{B}$ and pseudo-Gram matrix $\mathbf{G}$. $\mathbf{G}'$ is the pseudo-Gram matrix of M' isomorphic to M. If $\mathbf{U}$ is congruence matrices* *between* $\mathbf{G}$ *and* $\mathbf{G}'$*, then there is a deterministic polynomial time algorithm to find* $\mathbf{U}$*, given a basis of* $\mathcal{O}_{\mathbb{K}}$*,* $\mathbf{B}$*, and* $\mathbf{G}'$*.*

The main contributions are summarized as below:

- We introduce a new tool called module lattice symplectic automorphism into designing algorithms solving module-LIP, and reduce the problem solving module-LIP to finding the certain symplectic automorphism. With this framework, we propose a provable deterministic polynomial-time algorithm that solves module-LIP for the rank-2 module $M \subset \mathbb{K}^2$ where $\mathbb{K}$ is a totally real number field.
- Compared with algorithms in [20], our algorithms are provable deterministic polynomial-time algorithm while the algorithms in [20] need some heuristic assumptions. Moreover, our algorithm, that solves module-LIP for the rank-2 module M in totally real number field, always runs in polynomial time regardless of the the arithmetic properties of the module, whereas the time complexity of algorithm in [20] relates to the arithmetic properties.
- We invalidates the omSVP assumption introduced by HAWK to prove its forgery security. Therefore, necessary adjustment about the omSVP assumption should be made to guarantee the validity of the security proof. We stress that our results haven't yielded any actual attack against HAWK.

1.2 Technical Overview

Isomophism and Automorphism. From a geometric perspective, LIP is to find the unitary matrix O such that $OM = M'$ for isomorphic module lattices M, M'. In this perspective, we call a unitary matrix A such that $AM = M$ an automorphism of M. If O is a unitary matrix such that $OM = M'$, then the automorphisms of M' all have the form OAO^{-1}, where A is an automorphism of M. From the algebraic perspective, LIP is to find the congruence matrix U such that $U^*GU = G'$, where $G = B^*B$ (B is a basis of the lattice). In this perspective, we call U such that $U^*GU = G$ an automorphism of G. The automorphisms of G all have the form $B^{-1}UB$, where U is a unitary matrix. There have been many works showing that automorphisms can play important roles in solving LIP [13,17,16,2].

The key to our technique is that, for the module lattice $\mathcal{O}_{\mathbb{L}}^2$, we find a lattice isomorphism that has not been considered before. In particular, if the base field is also considered to be a totally real number field, this lattice isomorphism will also be a module lattice isomorphism with more algebraic structures.

Specifically, we will exploit the symplectic property of rank 2 matrices (i.e. $B^TJ_2B = J_2, \forall B \in \mathrm{SL}_2(\mathbb{L})$, here $J_2 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$) and its variants to extract automorphism, such as computing $B^{-1}J_2B$ from B^TB (See Subsection 3.1).

Analysis of $\mathcal{O}_{\mathbb{L}}^2$-LIP. We reduce the problem of solving $\mathcal{O}_{\mathbb{L}}^2$-LIP to the problem of finding out the certain module lattice symplectic automorphism by using the Lenstra-Silverberg algorithm proposed in [19]. Informally speaking, this algorithm can find an isomorphism between a lattice and its certain canonical form using specific automorphisms of the lattice. The module lattice $\mathcal{O}_{\mathbb{L}}^2$ has the automorphisms $\{aI_2|a \in \mu(\mathbb{L})\}(\mu(\mathbb{L})$ denotes the roots of unity in $\mathbb{L})$ inherently, but these automorphisms are not enough for Lenstra-Silverberg algorithm. We discovered and carefully demonstrated that adding the certain module lattice symplectic automorphism to the above-mentioned automorphisms meets the requirements for the algorithm in [19].

It is worth mentioning that the main theorem (Theorem 2.1 in this paper) in [19] will be used over and over again as a powerful tool in our technique. However, before us it seems that people only focused on the original version [13].

As the discussion before, this sympletic automorphism can be computed if the considered field is a totally real number field. For HAWK, we can obtain a weak sympletic automorphism, and it will affect the existing omSVP assumptions. This invalidates the assumption used in their security analysis, although it does not yield attacks against the construction itself.

Algorithm for rank-2 module-LIP over totally real number field. We now explain how our algorithm for module-LIP works when the module $M \subset \mathbb{K}^2$ has rank-2 and when the number field $\mathbb{K}$ is a totally real number field. Firstly, we can still first obtain a pseudo-automorphism of M, which we call pseudo because it does not preserve M. To be specific, for $M' = OM$ where O is a unitary matrix, we can obtain OJ_2O^{-1}. If we look at the pseudo-automorphism from the perspective of matrix conjugation, then the eigenspace before conjugating differs from the eigenspace after conjugating by only one transition matrix for the same eigenvalue.

From a high level view, a fundamental reason why the module isomorphism problem for rank 2 is harder than for rank 1 lies in the fact that it is hard to find rank 1 submodule $N \subseteq M$, $N' \subseteq M'$ such that $N' = ON$. The intersection of the modules and eigenspaces of pseudo-automorphisms provides the submodules N, N' s.t. $N' = ON$, but previously we only knew automorphisms of pure quantities, that is, they only have trivial eigenspaces. This new (pseudo) automorphism fits our requirements nicely. And then rank 1 module-LIP can be solved by using algorithm in [19]. It should be pointed out that the eigenvalues and eigenvectors of this automorphism need to be lifted to be considered in $\mathcal{O}_{\mathbb{L}}$, and thus need to be argued more carefully.

In addition, for better intuition, we here give our technical overview from a geometric point of view. However, the geometric perspective and Gram matrix perspective can be transformed into each other. For computational reasons, the actual algorithm will be performed from the Gram matrix perspective. We will give a sketch of the actual algorithm at the beginning of Section 4.

Roadmap. The rest of the paper is organized as follows. Section 2 provides basic definitions and preliminaries. In Section 3, we present a provable deterministic polynomial-time algorithm to solve $\mathcal{O}_{\mathbb{L}}^2$-LIP where $\mathbb{L}$ is a CM number field, with the help of a new module lattice symplectic automorphism, and we also show how to find a weak module lattice symplectic automorphism efficiently, which can invalidate the omSVP assumption introduced by HAWK to prove its forgery security. In Section 4, we present the provable deterministic polynomial-time algorithm to solve module-LIP for the rank-2 module $M \subset \mathbb{K}^2$ where the number field $\mathbb{K}$ is a totally real number field. Section 5 concludes the paper shortly.

2 Notations and preliminaries

2.1 Notations

- The Euclidean norm of $a \in \mathbb{R}^n$ is denoted by $\|a\|$. The transpose of A is denoted by A^T, and $(A^{-1})^T$ is abbreviated as A^{-T}. 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 $\mathrm{diag}(r, r, \cdots, r)$, and sometimes use r to represent rI_n in matrix multiplications (such as $r\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} rx_1 \\ rx_2 \end{pmatrix}$). We will also emphasize this point from time to time in the proof.
- For a number field $\mathbb{K}$, the parameter $\mathbf{K}$ denotes degree of $\mathbb{K}$, $\log \Delta_{\mathbb{K}}$, and a basis of $\mathcal{O}_{\mathbb{K}}$.
- For x in $\underline{\text{a number}}$ field $\mathbb{K}$, we call x^* is the complex conjugation of x if $\sigma(x^*) = \overline{\sigma(x)}$, $\forall \sigma \in Hom_{\mathbb{Q}}(\mathbb{K}, \mathbb{C})$. For matrix $H = (h_{ij})$, let H^* denote $(h_{ij}^*)^T$ and $\overline{H}$ denote (h_{ij}^*) if all h_{ij}^* exist.
- For a ring A in a number field that are closed under complex conjugating, the unitary matrices 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 $\mathbb{F}$. Note $\mu(\mathbb{K}) \subset \mathcal{O}_{\mathbb{K}}$ and $\mu(\mathbb{K}) = \mathcal{U}_1(\mathcal{O}_{\mathbb{K}})$
- Assume G is an abelian group. For abelian groups A, B equiped a bilinear map $\varphi : A \times B \to G$, we define the group product $A \cdot B$ as the abelian group generated by $\{\varphi(a, b)\}$. We also usually use $a \cdot b$ to denote $\varphi(a, b)$. Further more, if there are canonical bilinear maps respectively between (A, B), (B, C), $(A \cdot B, C)$, $(A, B \cdot C)$ satisfying associative law i.e. $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, $\forall a \in A, b \in B, c \in C$, then the group product also have associative law, i.e. $(A \cdot B) \cdot C = A \cdot (B \cdot C)$. For example, matrix groups $A \subseteq \mathbb{K}^{n\times m}$, $B \subseteq \mathbb{K}^{m\times l}$, $C \subseteq \mathbb{K}^{l\times t}$ or A, B, C are ideals of a ring.

2.2 Lattices

Lattices are discrete additive subgroups of $\mathbb{R}^m$. A lattice is usually defined by a set of n linearly independent basis vectors $b_1, b_2, \ldots, b_n \in \mathbb{R}^m$. Any point in the lattice can be expressed as an integer linear combination of the basis vectors.

A lattice $\mathcal{L}$ of rank n and dimension m is a set of points in $\mathbb{R}^m$ that can be expressed as integer combinations of n linearly independent basis vectors $b_1, ..., b_n$. Denote $B = (b_1, ..., b_n)$ as the basis of the lattice $\mathcal{L}$, and then $\mathcal{L} = \{Bz : z \in \mathbb{Z}^n\}$.

2.3 Number Theory

A number field $\mathbb{K}$ is a finite extension of the rational numbers $\mathbb{Q}$. Any such $\mathbb{K}$ is isomorphic to $\mathbb{Q}[X]/(P)$ for an irreducible monic polynomial P. The degree of P matches the degree of the extension. For any extension $\mathbb{K}$ of degree d, there are exactly d embeddings $\sigma_1, ..., \sigma_d$ from $\mathbb{K}$ into the complex numbers $\mathbb{C}$. If an embedding sends $\mathbb{K}$ into the real numbers $\mathbb{R}$, it's called a real embedding. Otherwise, it's called complex. If an embedding is not real, it can be paired with its complex conjugate to give another distinct complex embedding. We use r_1 to denote the count of real embeddings and r_2 to denote the count of complex embeddings up to conjugation. Therefore, the total count of embeddings is $d = r_1 + 2r_2$. When all embeddings are real (i.e. $r_1 = d$), we say the extension $\mathbb{K}|\mathbb{Q}$ is totally real. Conversely, when all embeddings are complex (i.e. $2r_2 = d$), we call it totally imaginary.

CM number field A CM (number) field $\mathbb{L}$ is a number field if it's a quadratic extension $\mathbb{L}/\mathbb{K}$ where the base field $\mathbb{K}$ is totally real but $\mathbb{L}$ is totally imaginary. 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 $\exists \tau \in Gal(\mathbb{L}/\mathbb{K})$ s.t. $\forall x \in \mathbb{L}, \sigma_i(\tau(x)) = \overline{\sigma_i(x)}$. We usually denote $\tau(x)$ by x^*.

Canonical embedding We call this map $\sigma : x \in \mathbb{K} \mapsto (\sigma_1(x), \ldots, \sigma_d(x))^T \in \mathbb{C}^d$ canonical embedding of number field $\mathbb{K}$. We will often identify $\mathbb{K}$ with the image underlying its canonical embedding, then $\mathcal{O}_\mathbb{K}$ is a lattice. But note that we are not representing elements in $\mathbb{K}$ using the canonical embedding.

The norm map defined over $\mathbb{K}$ is $\mathcal{N}_\mathbb{K}(z) = \prod_i \sigma_i(z)$. Similarly, the trace map is $\mathrm{Tr}_\mathbb{K}(z) = \sum_i \sigma_i(z)$. Regard $z \in \mathbb{K}$ as $\mathbb{Q}$-linear map $m_z : x \in \mathbb{K} \mapsto zx \in \mathbb{K}$, then we have $\mathcal{N}_\mathbb{K}(z) = \det(m_z)$ and $\mathrm{Tr}_\mathbb{K}(z) = \mathrm{Tr}(m_z)$. Especially, if $z \in \mathbb{K}$, then $\mathcal{N}_\mathbb{K}(z)$, $\mathrm{Tr}_\mathbb{K}(z)$ belong to $\mathbb{Q}$. When there is no ambiguity, we drop the subscript.

The $\mathbb{R}$-algebra $\mathbb{K}_\mathbb{R} := \mathbb{K} \otimes_\mathbb{Q} \mathbb{R}$ is a real vector space of dimension d. If write $\mathbb{K}$ as $\mathbb{Q}[X]/(P)$, then we can use $\mathbb{R}[X]/(P)$ to denote $\mathbb{K} \otimes_\mathbb{Q} \mathbb{R}$. To keep the discussion concise, this paper will not delve deeply into the discussion about $\mathbb{K}_\mathbb{R}$.

Rings of integer Let $\mathcal{O}_\mathbb{K}$ denote the ring of integers of a number field $\mathbb{K}$. $\mathcal{O}_\mathbb{K}$ is a free $\mathbb{Z}$-module of rank d. The discriminant of $\mathbb{K}$, denoted $\Delta_\mathbb{K}$, is defined as $(\det(\sigma_i(\alpha_j))_{i,j})^2 \in \mathbb{Z}$, where $(\alpha_j)_{1 \leq j \leq d}$ is any basis of $\mathcal{O}_\mathbb{K}$. Specifically, there exists some absolute constant $c > 1$ such that $\Delta_\mathbb{K} \geq c^d$ for all number fields $\mathbb{K}$. In particular, we always have $d = \mathrm{poly}(\log \Delta_\mathbb{K})$.

When $\mathbb{K}$ is a totally real number field and $\mathbb{L} = \mathbb{K}[X]/(X^2 + 1)$, Mureau et.al. showed $\log \Delta_\mathbb{L} = \mathrm{poly}(\log \Delta_\mathbb{K})$ and the following lemma in [20, Section 2.2].

Lemma 2.1 (**[20, Lemma 2.6]**) *Let $\mathbb{K}$ be a totally real number field and $\mathbb{L} := \mathbb{K}[X]/(X^2+1)$. There exists a polynomial time algorithm A that, given as input a $\mathbb{Z}$-basis $B_\mathbb{K}$ of $\mathcal{O}_\mathbb{K}$, computes a $\mathbb{Z}$-basis $B_\mathbb{L}$ of $\mathcal{O}_\mathbb{L}$.*

Lemma 2.2 *Let $\mathbb{L}$ be a CM number field with degree n. Then $\forall x \in \mathcal{O}_\mathbb{L} \setminus \{0\}$, we have: $Tr_\mathbb{L}(x^*x) \geq n$, and $Tr_\mathbb{L}(x^*x) = n$ iff x is a root of unity.*

Proof. Note $\forall 1 \leq i \leq n$, $\sigma_i(x^*x) = (\sigma_i(x))^*\sigma_i(x) = |\sigma_i(x)|^2 \geq 0$. So $\mathrm{Tr}_\mathbb{L}(x^*x) = \sum_{i=1}^n \sigma_i(x^*x) \geq n(\prod_{i=1}^n |\sigma_i(x^*x)|)^{1/n} = n(\mathcal{N}(x^*x))^{1/n} \geq n$. The last inequality holds for $\forall r \in \mathcal{O}_\mathbb{L}$, $\mathcal{N}(r) \geq 1$. This means $\mathrm{Tr}_\mathbb{L}(x^*x) = n$ iff all the equals are taken iff $|\sigma_i(x)| = 1$ for all $1 \leq i \leq n$ iff[1] x is a root of unity. □

Ideals. An (fractional) ideal $\mathcal{I}$ is an finitely generated additive subgroup of $\mathbb{K}$ such that $x \cdot \mathcal{I} \subseteq \mathcal{I}$ for all $x \in \mathcal{O}_\mathbb{K}$. When an ideal is contained in $\mathcal{O}_\mathbb{K}$, we call it an integral ideal and usually use the fraktur lower-case letter to denote it (e.g. $\mathfrak{a}$). Principal ideal is an ideal generated by a single element $a \in \mathbb{K}$ i.e. $a\mathcal{O}_\mathbb{K}$. The product of two ideals $\mathcal{I}$ and $\mathcal{J}$ is their group product i.e. $\mathcal{I}\mathcal{J} = \{\sum_i x_i y_i | x_i \in \mathcal{I}, y_i \in \mathcal{J}\}$. An ideal $\mathcal{I}$ has the form $\frac{1}{d} \cdot \mathfrak{a}$, where $d \in \mathcal{O}_\mathbb{K} \setminus \{0\}, \mathfrak{a} \subset \mathcal{O}_\mathbb{K}$. Then we can define the algebraic norm $\mathcal{N}(\mathcal{I}) := \sharp(\mathcal{O}_\mathbb{K}/\mathfrak{a})/\sharp(\mathcal{O}_\mathbb{K}/(d\mathcal{O}_\mathbb{K}))$.

Modules. Assume $\mathbb{K}$ is a number field. An $\mathcal{O}_\mathbb{K}$ module M is a subset of $\mathbb{K}^\ell$ of the form $b_1\mathcal{I}_1 + \quad + b_r\mathcal{I}_r$, where the $\mathcal{I}_i$'s are non-zero fractional ideals of $\mathbb{K}$ and $(b_1, ..., b_r)$ are $\mathbb{K}$-linearly independent vectors of $\mathbb{K}^\ell$, for some $\ell > 0$. We call $\mathbf{B} = (B, (\mathcal{I}_i)_{1 \leq i \leq r})$ a pseudo-basis for M, where B is the matrix whose columns are the b_i. The integer r is called the rank of the module. When $r = \ell$, we say that the module has full rank.

2.4 Module-LIP

Definition 2.1 *Let $\mathbf{B} = (B, (\mathcal{I}_i)_{1 \leq i \leq \ell})$ be a pseudo-basis of a rank-ℓ module M in $\mathbb{K}_\mathbb{R}^k$. The pseudo-Gram matrix associated with $\mathbf{B}$ is denoted by $\mathbf{G} := (G, (\mathcal{I}_i)_{1 \leq i \leq \ell})$, where $G = B^*B$.*

Definition 2.2 *Let $\mathbf{G} = (G, (\mathcal{I}_i)_{1 \leq i \leq \ell})$ and $\mathbf{G}' = (G', (\mathcal{J}_i)_{1 \leq i \leq \ell})$ be two pseudo-Gram matrices. They are said to be congruent if there exists $U = (u_{i,j})_{1 \leq i,j \leq \ell} \in GL_\ell(\mathbb{K})$ such that $G' = U^*GU$ and $u_{i,j} \in \mathcal{I}_i\mathcal{J}_j^{-1}$, $v_{i,j} \in \mathcal{J}_i\mathcal{I}_j^{-1}$, where $V = (v_{i,j})_{1 \leq i,j \leq \ell} := U^{-1}$. Such U is called a congruence matrix between $\mathbf{G}$ and $\mathbf{G}'$. This defines an equivalence relation $\sim$ on the set of pseudo-Gram matrices.*

In [20] Mureau et al. proposed three equivalent definitions of isomorphism between module lattices, and we only elaborate here the one we will use.

[1] This is a basic result in number theory. One can find a argument in [20, Lemma 2.14].

Definition 2.3 *Let $M, M' \subset \mathbb{K}_{\mathbb{R}}^{\ell}$ be two modules of rank ℓ with respective pseudo-bases $\mathbf{B} = (B, (\mathcal{I}_i)_{1\le i\le \ell})$ and $\mathbf{B}' = (B', (\mathcal{J}_i)_{1\le i\le \ell})$. Let $\mathbf{G}$ (resp. $\mathbf{G}'$) be the pseudo-Gram matrix associated with $\mathbf{B}$ (resp. $\mathbf{B}'$). We say that M, M' are isomorphic as module lattices if $\mathbf{G}$ and $\mathbf{G}'$ are congruent.*

Definition 2.4 (module-LIP$_{\mathbf{K}}^{\mathbf{B}}$) *For $\mathbf{B}$ a pseudo-basis of a module lattice $M \subset \mathbb{K}_{\mathbb{R}}^{\ell}$ with associated pseudo-Gram matrix $\mathbf{G}$, the (worst-case) search module lattice Isomorphism Problem with parameter $\mathbf{K}$ and $\mathbf{B}$, denoted by module-LIP$_{\mathbf{K}}^{\mathbf{B}}$, is, given as input any pseudo-Gram matrix $\mathbf{G}' \sim \mathbf{G}$ (see Definition 2.2), to find a congruence matrix between $\mathbf{G}$ and $\mathbf{G}'$.*

2.5 Algorithmic consideration

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

In fact, $\sigma(B_{\mathcal{O}_{\mathbb{K}}})$ being LLL-reduced implies that for any integral $x \in \mathcal{O}_{\mathbb{K}}$, $\mathrm{size}(x) = \mathrm{poly}(d, \|\sigma(x)\|)$. Inversely, $\|\sigma(x)\| \le \sum_i |x_i| \cdot \|\sigma(\alpha_i)\| \le d^{3/2} \cdot 2^d \cdot (\Delta_{\mathbb{K}}^{1/d}) \cdot \max_i |x_i|$ since $\lambda_d(\mathcal{O}_{\mathbb{K}}) \le \sqrt{d} \cdot (\Delta_{\mathbb{K}}^{1/d})$. This implies that the arithmetic operations on elements in $\mathcal{O}_{\mathbb{K}}$ are in polynomial time. And then the arithmetic operations on elements in $\mathbb{K}$ are in polynomial time.

A fractional ideal $\mathcal{I}$ is represented by a $\mathbb{Z}$-basis $(y_1, \dots, y_d)$ of the ideal, such that $(\sigma(y_i))_{1\le i\le d}$ is an LLL-reduced basis of $\sigma(\mathcal{I})$. In particular, we have $\|\sigma(y_i)\| \le 2^d \cdot \lambda_d(\mathcal{I}) \le \sqrt{d} \cdot 2^d \cdot \Delta_{\mathbb{K}}^{3/(2d)} \cdot \mathcal{N}(\mathcal{I})^{1/d}$(see e.g. [20, Section 2.3]). We define $\mathrm{size}(\mathcal{I}) := \sum_i \mathrm{size}(y_i)$.

Lemma 2.3 ([20, Lemma 2.9]) *Let $B = (B, (\mathcal{I}_i)_{1\le i\le r})$ be a pseudo-basis of a rank r module M in $\mathbb{K}^{\ell}$. Then, one can compute in polynomial time a basis $C \in \mathbb{C}^{d\ell \times dr}$ of $\sigma(M)$ such that the column vectors c_i of C satisfy $\|c_i\| \le \sqrt{d} \cdot 2^d \cdot (\Delta_{\mathbb{K}}^{3/(2d)}) \cdot \max_{1\le j\le r} \|\sigma(b_j)\| \cdot \mathcal{N}(\mathcal{I}_j)^{1/d}$, where b_j is the j-th column of B.*

Basic algorithms

Lemma 2.4 ([11, Lemma 2.8]) *With the representation of ideals as described above, one can sum up two ideals $\mathcal{I}$ and $\mathcal{J}$ in time poly(size($\mathcal{I}$), size($\mathcal{J}$)), multiply two ideals $\mathcal{I}$ and $\mathcal{J}$ in time poly(size($\mathcal{I}$), size($\mathcal{J}$), $\log \Delta_{\mathbb{K}}$), compute the inverse of an ideal $\mathcal{I}$ in time poly(size($\mathcal{I}$), $\log \Delta_{\mathbb{K}}$).*

As a generalization of the product of ideals, when the bilinear map between two abelian groups satisfies that it runs in polynomial time in the input size and outputs a lattice vector of polynomial size in the input size, we can compute the group product of these two abelian groups with the $\mathbb{Z}$-basis of them as input in polynomial time.

The following lemma guarantees that the computation of roots of unity in a given field is efficient.

Lemma 2.5 ([20, Corollary 2.11]) *Let $\mathbb{K}$ be a degree d number field. Then, $\mathbb{K}$ has at most $2d^2$ roots of unity, and there exists a polynomial-time algorithm that, given a basis of the ring of integers $\mathcal{O}_{\mathbb{K}}$, computes the roots of unity in $\mathbb{K}$.*

Lenstra-Silverberg Algorithm Gentry and Szydlo initially proposed an algorithm in [13] to recover x from x^*x and xR (where R is a certain type of polynomial ring). Later, Lenstra and Silverberg extended this in [17,18,19]. We describe here the main theorem presented in [19], which will be used later in Section 3 and Section 4.

Definition 2.5 *An order is a commutative ring of which the additive group is isomorphic to $\mathbb{Z}^n$ for some $n \in \mathbb{Z}_{\geq 0}$. A CM-order A is an order such that:*

1. *A has no non-zero nilpotent elements.*
2. *A is equipped with an conjugate automorphism $x \mapsto \overline{x}$ of A such that $\varphi(\overline{x}) = \overline{\varphi(x)}$ for all $x \in A$ and all ring homomorphisms $\varphi : A \to \mathbb{C}$.*

Definition 2.6 *Let A be a CM-order. A lattice L is an A-lattice if it's given an A-module structure with the property that for all $a \in A$ and $x, y \in L$ one has $\langle ax, y\rangle = \langle x, \bar{a}y\rangle$.*
An example of an A-lattice is the A-module A itself, with inner product $\langle a, b\rangle = Tr(\overline{a}b)$; here $Tr : A \to \mathbb{Z}$ is the trace function of A as a $\mathbb{Z}$-algebra. This A-lattice is called the standard A-lattice.

In algorithms, we can represent an order by a system $(b_{ijk})_{i,j,k=1}^n$ of integers with the property that, for some $\mathbb{Z}$-basis $\alpha_1, ..., \alpha_n$ of the order, one has $\alpha_i\alpha_j = \sum_{k=1}^n b_{ijk}\alpha_k$ for all $1 \leq i, j \leq n$. In other words, for $\mathbb{Z}$-basis $\alpha_1, ..., \alpha_n$ of the order, we specify the order by matrix representation of m_{α_i} under the basis $\alpha_1, ..., \alpha_n$, where m_{α_i} means the action of multiplying α_i. A lattice is specified by the Gram matrix of a $\mathbb{Z}$-basis $b_1, ..., b_m$. An A-lattice is specified as a lattice and a system of nm^2 integer coefficients that express $\alpha_i b_j$ on $b_1, ..., b_m$, where the $(\alpha_i)_{i=1}^n$ and $(b_j)_{j=1}^m$ are as above.

Definition 2.7 *An A-isomorphism $f : L \to M$ of A-lattices is an isomorphism of A-modules with $\langle f(x), f(y)\rangle = \langle x, y\rangle$ for all $x, y \in L$.*
One can see that if there is an A-isomorphism between an A-lattice L and A-module M which is also a lattice, then M is also an A-lattice.

Theorem 2.1 ([19, Theorem 1.5]) *There is a deterministic polynomial-time algorithm that, given a CM-order A and an A-lattice L, decides whether or not L is A-isomorphic with the standard A-lattice, and if so, computes such an A-isomorphism.*

3 Solving $\mathcal{O}_{\mathbb{L}}^2$-LIP with module symplectic automorphism

In this section, we focus on the $\mathcal{O}_{\mathbb{L}}^2$-LIP, in which $\mathbf{B}$ is taken to be $(I_2, (\mathcal{O}_{\mathbb{L}}))$ in module-LIP$_{\mathbf{L}}^{\mathbf{B}}$, and $\mathbb{L}$ is a CM number field.

We firstly present a deterministic polynomial-time algorithm solving $\mathcal{O}_{\mathbb{L}}^2$-LIP with the help of certain module lattice automorphism of $\mathcal{O}_{\mathbb{L}}^2$. It seems not easy to find such a module lattice automorphsim, but we can compute another weak module lattice automorphism. This weak module lattice automorphism will invalidate the omSVP assumption used in the security analysis of HAWK.

3.1 An algorithm for $\mathcal{O}_{\mathbb{L}}^2$-LIP with automorphism of $\mathcal{O}_{\mathbb{L}}^2$

In this subsection, we will mainly give a direct application of the Lenstra-Silverberg algorithm on $\mathcal{O}_{\mathbb{L}}^2$-LIP as Theorem 3.1. Essentially, this provides a reduction from $\mathcal{O}_{\mathbb{L}}^2$-LIP to finding certain module lattice automorphisms of $\mathcal{O}_{\mathbb{L}}^2$.

Recall $J_2 := \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$.

Theorem 3.1 *Let $\mathbb{L}$ be a CM number field with degree $2d$ and $B \in GL_2(\mathcal{O}_{\mathbb{L}})$. There is a deterministic polynomial-time algorithm that, given a basis of $\mathcal{O}_{\mathbb{L}}$, B^*B, and $B^{-1}J_2B$, outputs $\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})B$.*

Lemma 3.1 *Let $\mathbb{L}$ be a CM number field with degree $2d$. 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, $\sharp(\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})) \leq 2\sharp(\mu(\mathbb{L}))^2 \leq 128d^4$.

Proof. Assume $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathcal{U}_2(\mathcal{O}_{\mathbb{L}})$, then $2d = \mathrm{Tr}_{\mathbb{L}}(1) = \mathrm{Tr}_{\mathbb{L}}(a^*a + b^*b) = \mathrm{Tr}_{\mathbb{L}}(a^*a) + \mathrm{Tr}_{\mathbb{L}}(b^*b)$. By Lemma 2.2, one of a or b is zero, and the other is a root of unity. Do same discussion for c, d, one of c or d is zero, and the other is a root of unity. Similarity, one of a or c is zero, and the other is a root of unity. □

Lemma 3.2 *Use the notation in Theorem 3.1. Denote $B^{-1}J_2B$ by J_B, $\mathcal{O}_{\mathbb{L}}$ by R. Take $H := \langle J_B \rangle = \{I_2, J_B, -I_2, -J_B\}$. Then we can define modified group ring $R\langle H \rangle := R[H]/\langle I_2 + (-I_2) \rangle = R \cdot I_2 + R \cdot J_B$. Denote I_2 by e, J_B by σ, $R\langle H \rangle$ by $\widetilde{R}$.Then $\widetilde{R}$ is a CM-order. We usually use $ae + b\sigma$ to denote the element in $\widetilde{R}$, where $a, b \in R$. And then $Tr(ae + b\sigma)$ is just $2\,Tr_{\mathbb{L}}(a)$.*

Proof. R, H is communicative, so $R\langle H \rangle$ is communicative and the additive group is isomorphic to $\mathbb{Z}^{4d}$.

Assume $ae + b\sigma$ is nilpotent in $\widetilde{R}$, in which $a, b \in R$. Then $(ae + b\sigma)^m = 0$ for some $m \in \mathbb{Z}_+$. Consider $(bx + a)^m, x^2 + 1 \in R[x]$. Then $\exists r(x) \in R[x]\ s.t. \deg(r) \leq 1$, and $(bx + a)^m - r(x) \in \langle x^2 + 1 \rangle$. Assume $r(x) = cx + d$, then we have $(ae + b\sigma)^m - (c\sigma + de) = 0$ for $\sigma^2 + e = 0$. Therefore, $c\sigma + de = 0$ and this means

$c = d = 0$. If $bx + a \neq 0$ then $r(x) \neq 0$ since $x^2 + 1$ doesn't divide $(bx + a)^m$. It's a contradiction. So $b\sigma + ae = 0$.

The conjugate automorphsim is $ae + b\sigma \mapsto a^*e - b^*\sigma$. Then $\forall \varphi : \widetilde{R} \to \mathbb{C}$, $\varphi(\overline{ae}) = \overline{\varphi(ae)}$ since R is a CM-order and its conjugate automorphism is the complex conjugation. And $\varphi(\sigma)^2 = \varphi(\sigma^2) = -1 \Rightarrow \varphi(\sigma) \in \{\pm i\} \Rightarrow \overline{\varphi(\sigma)} = -\varphi(\sigma) = \varphi(\overline{\sigma})$. So $\varphi(\overline{r}) = \overline{\varphi(r)}$ for all $r \in \widetilde{R}$.

Thus $\widetilde{R}$ is a CM-order. Note $\widetilde{R} = Re \oplus R\sigma$ and Re, $R\sigma$ are invariant under $ae \Rightarrow \mathrm{Tr}(ae) = \mathrm{Tr}(ae|_{Re}) + \mathrm{Tr}(ae|_{R\sigma}) = 2\mathrm{Tr}_R(a) = 2\mathrm{Tr}_{\mathbb{L}}(a)$. Here Tr_R means trace on R. Similarly, since $b\sigma(Re) \subseteq R\sigma$, $b\sigma(R\sigma) \subseteq Re$, $\mathrm{Tr}(b\sigma) = 0$. □

Proof (proof of Theorem 3.1). Use the setting in Lemma 3.2. We take $M := \mathcal{O}_{\mathbb{L}}^2$ with the inner product $\langle , \rangle_M : (x, y) \in M^2 \mapsto 2\mathrm{Tr}_{\mathbb{L}}(x^* B^* B y) \in \mathbb{Z}$. $\widetilde{R}$ acts on M as $(ae+b\sigma, m) \in \widetilde{R} \times M \mapsto a \cdot m + b \cdot \sigma \cdot m$ (matrix multiplication). It makes M a $\widetilde{R}$-module (Note that the commutativity between rI_2 and σ matrices multiplication is utilized here to ensure the associativity of the ring operation.)

Assume $e_1 = B^{-1}\begin{pmatrix}1\\0\end{pmatrix} = e \cdot e_1, e_2 = B^{-1}\begin{pmatrix}0\\-1\end{pmatrix} = \sigma \cdot e_1 \in M$. Define $f : r \in \widetilde{R} \mapsto r \cdot e_1$. Then we have $\forall a_1 e + b_1\sigma, a_2 e + b_2 \sigma \in \widetilde{R}$,

$$\begin{aligned}
&\frac{1}{2}\left\langle f(a_1 e + b_1\sigma), f(a_2 e + b_2\sigma)\right\rangle_M \\
=&\mathrm{Tr}_{\mathbb{L}}((a_1 \cdot e_1 + b_1 \cdot e_2)^* B^* B (a_2 \cdot e_1 + b_2 \cdot e_2)) \\
=&\mathrm{Tr}_{\mathbb{L}}\left((a_1^*,\ -b_1^*)(B^*)^{-1} B^* B B^{-1}\begin{pmatrix}a_2\\-b_2\end{pmatrix}\right) \\
=&\mathrm{Tr}_{\mathbb{L}}\left((a_1^*,\ -b_1^*)\begin{pmatrix}a_2\\-b_2\end{pmatrix}\right) \\
=&\mathrm{Tr}_{\mathbb{L}}(a_1^* a_2 + b_1^* b_2) \\
=&\frac{1}{2}\mathrm{Tr}((a_1^* a_2 + b_1^* b_2)e + (a_1^* b_2 - b_1^* a_2)\sigma) \\
=&\frac{1}{2}\mathrm{Tr}\left(\overline{(a_1 e + b_1\sigma)}(a_2 e + b_2\sigma)\right) \\
=&\frac{1}{2}\left\langle a_1 e + b_1\sigma, a_2 e + b_2\sigma\right\rangle.
\end{aligned}$$

Obviously f is homomorphism of $\widetilde{R}$-module. Note $f(ae + b\sigma) = a \cdot e_1 + b \cdot e_2 = B^{-1}\begin{pmatrix}a\\-b\end{pmatrix}$. f is injective since $f(ae + b\sigma) = 0 \Rightarrow B^{-1}\begin{pmatrix}a\\-b\end{pmatrix} = 0 \Rightarrow \begin{pmatrix}a\\-b\end{pmatrix} = 0 \Rightarrow a = b = 0$. f is surjective since $\forall v \in M$, assume $Bv = \begin{pmatrix}a\\-b\end{pmatrix} \in \mathcal{O}_{\mathbb{L}}^2$, then $f(ae + b\sigma) = B^{-1}\begin{pmatrix}a\\-b\end{pmatrix} = v$.

In conclusion, we obtain M is A-isomorphic with the standard A-lattice, and then is a A-lattice. Using the polynomial-time algorithm in Theorem 2.1, we can get an A-isomorphsim ϕ between the standard A-lattice and M. Assume

$\phi(e) = B^{-1}\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}$ for some $a, b \in R$. Then $\mathrm{Tr}_{\mathbb{L}}(a_1^* a_1 + b_1^* b_1) = \frac{1}{2}\left\langle \phi(e), \phi(e) \right\rangle_M = \frac{1}{2}\left\langle e, e \right\rangle = n$. By Lemma 2.2, we have $a_1 \in \mu(R), b_1 = 0$ or $b_1 \in \mu(R), a_1 = 0$.

Assume $\phi(\sigma) = B^{-1}\begin{pmatrix} a_2 \\ b_2 \end{pmatrix}$. Similarly, we obtain $a_2 \in \mu(R), b_2 = 0$ or $b_2 \in \mu(R), a_2 = 0$. Note $M = R\phi(e) \bigoplus R\phi(\sigma)$, so a_1, a_2 are not both 0. This means $(\phi(e)|\phi(\sigma)) \in B^{-1}\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})$(by Lemma 3.1).

Then we compute $(\phi(e)|\phi(\sigma)) \cdot \mathcal{U}_2(\mathcal{O}_{\mathbb{L}}) = B^{-1}(\mu(A) \cdot v)\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})$ in polynomial time since $\sharp(\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})) \leq 128d^4$. □

3.2 New pseudo lattice automorphisms of rank 2 Module lattices over a CM number field

A very simple but important lemma is given below. It's actually the symplectic property of the 2 × 2 matrix.

Lemma 3.3 $\forall U \in GL_2(\mathbb{K})$, $U^T J_2 U = \det(U) \cdot J_2$.

Proof. Assume $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, then

$$U^T J_2 U = \begin{pmatrix} a & c \\ b & d \end{pmatrix}\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 0 & ad - bc \\ -(ad - bc) & 0 \end{pmatrix} = \det(U) \cdot J_2.$$

□

Proposition 3.1 *Let $\mathbb{K}$ be a number field, $B \in GL_2(\mathbb{K})$, and $r \in \mathbb{K}$. Given as input a basis of $\mathcal{O}_{\mathbb{K}}$, $G = B^T B$, and $\det(B)$, we can compute $J_B := B^{-1} J_2 B$ and $m_r := B^{-1}(rI_2)B$ in the time of polynomial of the input size.*

Proof. We claim that

$$J_B = (\det(B)I_2) \cdot G^{-1} \cdot J_2 \text{ and } m_r = (rI_2),$$

and then the time to compute J_B and m_r is polynomial.

It is obvious that $rI_2 = B^{-1}(rI_2)B$ since rI_2 is in center of $M_2(\mathbb{K})$, and we also have

$$\begin{aligned} &(\det(B)I_2) \cdot G^{-1} \cdot J_2 \\ =&(\det(B)I_2)B^{-1}(B^T)^{-1}J_2 B^{-1} B \\ =&B^{-1}(\det(B)I_2)\left((B^{-1})^T J_2 B^{-1}\right)B \\ =&B^{-1}(\det(B)I_2)(\det(B^{-1}))J_2 B \\ =&B^{-1}J_2 B, \end{aligned}$$

where the third equality holds by Lemma 3.3. □

Lemma 3.4 *Let $\mathbb{L}$ be a CM number field. 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 an $\mathbb{Q}$ linear map. We claim that $\forall U \in GL_2(\mathbb{L})$, $U^* J_2 t_* U = \det(U)^* \cdot J_2 t_*$.*

Proof. Note that for all $B \in M_2(\mathbb{L})$, $t_* \circ B = (B^*)^T \circ t_*$ as $\mathbb{Q}$ linear map. Assume $U = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, then

$$\begin{aligned} U^* J_2 t_* U &= \begin{pmatrix} a^* & c^* \\ b^* & d^* \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} a^* & b^* \\ c^* & d^* \end{pmatrix} t_* \\ &= \begin{pmatrix} 0 & (ad-bc)^* \\ -(ad-bc)^* & 0 \end{pmatrix} t_* \\ &= \det(U)^* \cdot J_2 t_*. \end{aligned}$$

□

Proposition 3.2 *Let $\mathbb{L}$ be a CM number field, $B \in GL_2(\mathbb{L})$, and $r \in \mathbb{L}$. Given as input a basis of $\mathcal{O}_{\mathbb{L}}$, $G = B^* B$, and $\det(B)$, we can compute $B^{-1} J_2 t_* B$ and $m_r := B^{-1}(rI_2)B$ in the time of polynomial of the input size.*

Proof. The computation of m_r is same as Proposition 3.1. Similarly we claim $B^{-1} J_2 t_* B = (\det(B)^* I_2) \cdot G^{-1} \cdot J_2 t_*$, and then the time to compute $B^{-1} J_2 t_* B$ is polynomial.

$$\begin{aligned} (\det(B)^* I_2) \cdot G^{-1} \cdot J_2 t_* &= (\det(B)^* I_2) B^{-1} (B^*)^{-1} J_2 t_* 2 B^{-1} B \\ &= B^{-1} (\det(B)^* I_2) \left((B^{-1})^* J_2 t_* B^{-1} \right) B \\ &= B^{-1} (\det(B)^* I_2) (\det(B^{-1})^*) J_2 t_* B \\ &= B^{-1} J_2 t_* B, \end{aligned}$$

where the third equality holds by Lemma 3.4. □

Remark 1. We refer to $B^{-1} J_2 B$ and $B^{-1} J_2 t_* B$ as pseudo lattice automorphisms. To further distinguish them, we refer to $B^{-1} J_2 B$ as a pseudo module lattice automorphism. We use the term 'pseudo' because when they act on the module lattice generated by the pseudo-basis **B**, the resulting elements may not necessarily still belong to the original module lattice. We call them automorphisms because the inner product (over the $\mathbb{Q}$-vector space) induced by $G = B^* \cdot B$ of a vector remains the same under the pseudo-automorphisms. More precisely, in the case of $B^{-1} J_2 t_* B$ (the case of $B^{-1} J_2 t_* B$ is similar), for any vectors $v_i = B^{-1}(x_i, y_i)^T, i = 1, 2$, their inner product induced by G is $\mathrm{tr}_{L/\mathbb{Q}}(v_1^* G v_2) = \mathrm{tr}_{L/\mathbb{Q}}(x_1^* x_2 + y_1^* y_2)$. Applying the pseudo-automorphism on v_i, the images become $(B^{-1} J_2 t_* B) B^{-1} (x_i, y_i)^T = B^{-1}(y_i^*, -x_i^*)^T$. Consequently, the inner product of the images is $\mathrm{tr}_{L/\mathbb{Q}}(x_1 x_2^* + y_1 y_2^*) = \mathrm{tr}_{L/\mathbb{Q}}(x_1^* x_2 + y_1^* y_2)$, the same as before. If considered on the $\mathbb{L}$-vector space, $B^{-1} J_2 t_* B$ cannot preserve the inner product, but $B^{-1} J_2 B$ can.

3.3 Impact of additional automorphism on HAWK

In this section, we will show the impact of additional automorphism on HAWK [8]. HAWK[2] is one of the brightest prospects at round one of the NIST for additional digital signatures [21]. HAWK is defined over a degree n, which is a power of two (equal to 256, 512 or 1024). A HAWK private key is a randomly generated basis for the lattice $\mathbb{Z}^{2n}$, consisting of four polynomials $f, g, F, G \in R_n = \mathbb{Z}[X]/(X^n+1)$, where f and g have small coefficients and together they satisfy the NTRU equation

$$fG - gF = 1 \, (\bmod X^n + 1)$$

The lattice secret basis B is $\begin{pmatrix} f & F \\ g & G \end{pmatrix}$ and the public key is

$$Q = B^*B = \begin{pmatrix} f^*f + g^*g & f^*F + g^*G \\ F^*f + G^*g & F^*F + G^*G \end{pmatrix}$$

In order to provide formal justification for the strong unforgeability under chosen message attack of HAWK, they formally introduce omSVP and they provide reductions in the (quantum) random oracle model from HAWK to omSVP[3], i.e. if there exists an adversary $\mathcal{A}$ against the (Q)ROM-SUF-CMA game of HAWK, then there exists an adversary $\mathcal{B}$ against the SAMPLE game in Figure 1. The omSVP is defined as follows.

Definition 3.1 *(Average case omSVP [8]). An average case* ***omSVP*** *instance is the pair* ***ac-omSVP = (Init, samp)****. On input 1^n,* ***Init*** *returns a form Q sampled from some distribution over $\mathcal{H}_{\ell_n}(\mathbb{K}_n)$, the roots of unity $\mu(\mathbb{K}_n)$ for $\mathbb{K}_n$, a length bound L_n, and a Gaussian parameter σ_n. On input Q,* ***samp*** *returns a sample from D_{Q,σ_n}. The adversary in Figure 1 wins whenever it can utilize the form Q and the samples it receives from* ***samp*** *to produce a non-trivial new element of $\mathcal{O}_{\mathbb{K}}^{\ell}$ that is sufficiently short.*

```
SAMPLE$_{\text{ac-omSVP},\mathcal{A}}$ $(1^n)$
$\mathcal{L} \leftarrow \{0\}$
$(Q, \mu(\mathbb{K}), L, \sigma) \leftarrow \text{Init}(1^n)$
$x^\star \leftarrow \mathcal{A}^{\text{samp}(Q)}(Q)$
return $[\![ \|x^\star\|_Q \leq L \wedge x^\star \notin \mathcal{L}_{\text{samples}} ]\!]$
```

```
samp$(Q)$
$x \leftarrow D_{Q,\sigma}$
$\mathcal{L}_{\text{samples}} \leftarrow \mathcal{L}_{\text{samples}} \cup \{\alpha x\}_{\alpha \in \mu(\mathbb{K})}$
return $x$
```

Fig. 1. The SAMPLE game

Here the "non-trivial new" depends on the definition of $\mathcal{L}_{\text{samples}}$ in samp(Q) in the SAMPLE game. In the SAMPLE game of HAWK, $Q = B^*B$, we can

[2] see https://hawk-sign,info

[3] See chapter 6 of the HAWK specification document for details.

think that the sample x we get has the form $x = B^{-1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $\| \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \| < L$. However, Proposition 3.2 tells us that in addition to $\{\alpha x\}_{\alpha \in \mu(\mathbb{K})}$, there is another type of trivial new vector that we can obtain efficiently. More specifically, we can compute the automorphism $B^{-1}J_2 t_* B$ by Proposition 3.2 and for a given sample $x = B^{-1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, applying this automorphism to x we will obtain an element $x^\star = B^{-1}J_2 t_* B \cdot B^{-1}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = B^{-1}\begin{pmatrix} x_2^* \\ -x_1^* \end{pmatrix}$ in Figure 1. Note that $\|x^\star\| = \|x\| < L$ and $x^\star \notin \{\alpha x\}_{\alpha \in \mu(\mathbb{K})}$, thus the $\{\alpha x^\star\}_{\alpha \in \mu(\mathbb{K})}$ are non-trivial new elements. In the case of HAWK, $\mu(\mathbb{K}) = \{X^i : i = 0, \ldots, 2n-1\}$, combined with automorphism $B^{-1}J_2 t_* B$, we get a subgroup G of $\mathrm{Aut}(Q)$ and G is isomorphic to the dihedral group D_{2n}[4]. At present, it seems that this automorphism has little impact on HAWK, but whether this automorphism will have a greater impact on HAWK requires further research in the future.

It is worth noting that omSVP and forging signatures on Hawk are not completely equivalent. Intuitively even if one can find an $x^\star$ that wins the SAMPLE game, one must also find a message and salt that hashes into a particular coset to make this a successful signature forgery. For more information see HAWK [8].

What's more, Theorem 3.1 tells us that in secret key recovery of HAWK, given $Q = B^* B$, if we can find $B^{-1}J_2 B$, then we can get the secret key B efficiently by Theorem 3.1, or equivalently, we have the following corollary, namely, if we have additional information $B^T B$, then we can find the secret key B. This provide new perspectives for cryptanalysis of HAWK.

Corollary 3.1 *Let $\mathbb{L}$ be a CM number field and $B \in GL_2(\mathcal{O}_{\mathbb{L}})$. There is a deterministic polynomial-time algorithm that, given a basis of $\mathcal{O}_{\mathbb{L}}$, $B^* B$, and $B^T B$, computes $\mathcal{U}_2(\mathcal{O}_{\mathbb{L}})B$.*

Proof. Recall $t_* : \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{L}^2 \mapsto \begin{pmatrix} x^* \\ y^* \end{pmatrix} \in \mathbb{L}^2$ and $t_* \circ B = (B^*)^T \circ t_*$ as $\mathbb{Q}$ linear map. We can compute $(B^T B)^{-1}(B^* B)^T t_*$ as $\mathbb{Q}$-linear map. Then note $(B^T B)^{-1}(B^* B)^T t_* = B^{-1}(B^*)^T t_* = B^{-1} t_* B$. By [20, Theorem 2.15] or Proposition 4.1, we can find $\det(B)$ from $\det(G)$ in polynomial time. Using Proposition 3.2, then we can compute $B^{-1}J_2 t_* B$. In conclusion, we can compute $B^{-1} t_* B \cdot B^{-1} J_2 t_* B = B^{-1} J_2 B$ in polynomial time. Then we use Theorem 3.1. □

4 An algorithm for module-LIP in rank 2 over totally real number fields

In this section, let $\mathbb{K}$ be a totally real number field and $\mathbb{L} = \mathbb{K}[X]/(X^2+1)$. By Lemma 2.1 and the discussion of size, we can always assume that the input parameter $\mathbf{K}$ and the input parameter $\mathbf{L}$ are equivalent. We can think of X as

[4] D_{2n} refers to the symmetries of the $2n$-gon, a group of order $4n$

the imaginary unit $\imath$, but sometime we use X again. We'll use this notation a lot.

In last section, all operations performed on $\mathbb{L}$ in Corollary 3.1 can be directly applied to $\mathbb{K}$, then we can directly obtain a algorithm for solving $\mathcal{O}_{\mathbb{K}}^2$-LIP. This is a deterministic polynomial-time algorithm for $\mathcal{O}_{\mathbb{K}}^2$-LIP, where $\mathbb{K}$ is a totally real number field. In contrast, the result in [20] only offers a heuristic polynomial-time algorithm for it.

In this section, with different approach and more elaborate processing, we present a deterministic polynomial-time algorithm for the module-LIP of rank 2 over a totally real number field as this following theorem.

Theorem 4.1 *Let $\mathbb{K}$ be a totally real number field and $\mathbb{L} = \mathbb{K}[X]/(X^2+1)$. $M \subseteq \mathbb{K}^2$ is an module lattice of rank 2 with pseudobasis parameter $\mathbf{B}$. Algorithm 7 takes as input parameter $\mathbf{K}$, $\mathbf{B}$, and $\mathbf{G}'$ an instance of module-LIP$_{\mathbf{K}}^{\mathbf{B}}$, runs in the polynomial time in the size if the input and finds all congruence matrices between $\mathbf{G}$ and $\mathbf{G}'$.*

To illustrate the structure of our algorithm, we first present an informal version of Algorithm 7 as below.

Algorithm 1: FindU(Informal)

```
Require: Parameter $\mathbf{B} = (B, (\mathcal{I}_i)_i)$ and $\mathbf{K}$. An module-LIP$_{\mathbf{K}}^{\mathbf{B}}$ instance
    $\mathbf{G}' = (G', (\mathcal{J}_i)_i)$.
Ensure: A congruence matrix $U$ between $\mathbf{G}$(the pseudo-Gram matrix associated
    with $\mathbf{B}$) and $\mathbf{G}'$.
  $\pm J_{BU} \leftarrow$ ConjOfJ$(\mathbf{B}, \mathbf{K}, \mathbf{G}')$ (Proposition 4.2)
  $(\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}) \leftarrow$ EigenSubLat$(J_{BU}, \mathbf{B}, \mathbf{K}, \mathbf{G}')$ (Proposition 4.3)
  $\mu(\mathbb{L}) \cdot (BU)^{-1}v_{\mathbf{B}} \leftarrow$ UseLS$(\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}, \mathbf{B}, \mathbf{K}, \mathbf{G}')$ (Proposition 4.4)
  $BU \leftarrow (v_{\mathbf{B}}|\overline{v_{\mathbf{B}}})((BU)^{-1}v_{\mathbf{B}}|\overline{(BU)^{-1}v_{\mathbf{B}}})^{-1}$
  $U \leftarrow B^{-1}BU$
  return $U$.
```

Roughly speaking, our algorithm can be divided into three steps.

Firstly, we extract a 'automorphism' of $\mathbf{G}'$ from the information of parameters $\mathbf{B}$ and $\mathbf{G}'$, denoted by $J_{BU} = (BU)^{-1}J_2(BU)$. We call this step as ConjOfJ that means finding conjugation of J_2.

Secondly, we identify the intersection of the eigenspace of this automorphism and the direct product of ideals in G, showing that it differs from the eigenspace of J_2 intersected with the lattice defined by B only by a factor of BU. This intersection is essentially a rank 1 module lattice, and with the BU factor accounted for, we can easily compute a pseudo-basis for it. We call this step as EigenSubLat that means computing sublattice composed by eigenvector.

Finally, we multiply the lattice obtained from the intersection by the inverse of the ideal derived from the previously computed pseudo-basis, resulting in a cyclic module lattice $\mathcal{O}_{\mathbb{L}} \cdot v$. Again, we will also obtain the value BUv. So we can use the Lenstra-Silverberg algorithm to recover v from $\mathcal{O}_{\mathbb{L}} \cdot v$, in the sense of a difference of one root of unity. We call this step as UseLS that simply means using Lenstra-Silverberg algorithm. With BUv and v known, we can easily recover BU.

4.1 Application of Lenstra-Silverberg algorithm

First we use the Theorem 2.1 (Lenstra-Silverberg algorithm) to give two specific algorithms, which will play an important role in the subsequent proofs. The first algorithm in the proposition is just Theorem 2.15 in [20], while the second algorithm can be seen as a high-dimensional generalization of the first algorithm. It is worth noting that the two algorithms in Proposition 4.1 are actually algorithms for solving rank-1 module-LIP in $\mathbb{K}$ and $\mathbb{K}^2$, respectively. We can generalize them into a unified form for solving rank-1 module-LIP, but we have chosen this current representation for easier understanding. So far, it seems that only the first algorithm has received attention, both in terms of applications and implementations.

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 $\mathbb{F}$. [19, Examples 3.7(i)(ii))] showed that A is a CM-order. The conjugate automorphsim is just the complex conjugation $x \mapsto x^*$, and the trace function is just $Tr_{\mathbb{F}}$.*

1. *For $\alpha \in \mathbb{F}$, there is a deterministic polynomial-time algorithm* `LS1` *that, given A, αA and $\alpha^*\alpha$, then we can find $\alpha\mu(A)$ in polynomial time, where $\mu(A)$ means roots of unity in A.*
2. *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 $B^{-1}(A \cdot v)$, then we can find $B^{-1}(\mu(A) \cdot v)$ in polynomial time, where $\mu(A)$ means roots of unity in A.*

Algorithm 2: LS1(informal)

```
Require: $\mathcal{O}_{\mathbb{F}}$; lattice $\alpha\mathcal{O}_{\mathbb{F}}$ for some $\alpha \in \mathbb{F}$;$\alpha^*\alpha$
Ensure: $\alpha\mu(\mathcal{O}_{\mathbb{F}})$.
$\langle,\rangle_M \leftarrow ((x,y) \in (\alpha\mathcal{O}_{\mathbb{F}})^2 \mapsto \mathrm{Tr}_{\mathbb{F}}(\frac{x^*y}{\alpha^*\alpha}))$
$M \leftarrow (\alpha\mathcal{O}_{\mathbb{F}}, \langle,\rangle_M)$
$W \leftarrow$ LS$(\mathcal{O}_{\mathbb{F}}, M)$
return $W$.
```

Proof. 1. Let $M = \alpha A$. It's an (fractional) ideal of A, so is an A-module. Define the inner product in M as $\langle,\rangle_M : (x,y) \in M^2 \mapsto \mathrm{Tr}_{\mathbb{F}}(\frac{x^*y}{\alpha^*\alpha}) \in \mathrm{Tr}_{\mathbb{F}}(A) \subseteq \mathbb{Z}$.

Algorithm 3: LS2(informal)

```
Require: $\mathcal{O}_{\mathbb{F}}$; lattice $B^{-1}(\mathcal{O}_{\mathbb{F}} \cdot v)$ for some $v \in \mathbb{F}^2$;$v^*v$,$B^*B$
Ensure: $B^{-1}(\mu(\mathcal{O}_{\mathbb{F}}) \cdot v)$.
$\langle , \rangle_M \leftarrow ((x,y) \in (B^{-1}(\mathcal{O}_{\mathbb{F}} \cdot v))^2 \mapsto \mathrm{Tr}_{\mathbb{F}}(\frac{x^*(B^*B)y}{v^*v}))$
$M \leftarrow (B^{-1}(\mathcal{O}_{\mathbb{F}} \cdot v), \langle , \rangle_M)$
$W \leftarrow$ LS$(\mathcal{O}_{\mathbb{F}}, M)$
return $W$.
```

Then M is a integral lattice. Consider $f : a \in A \mapsto \alpha a \in M$. Obviously it's an isomophsim of A-module, and one can see $\langle f(x), f(y) \rangle_M = \langle x, y \rangle$ for all $x, y \in A$ by the definition of $\langle , \rangle_M$. So f is an A-isomorphsim, and M is an A-lattice isomorphic to standard A-lattice.

Using the polynomial-time algorithm in Theorem 2.1, we can get an A-isomorphsim ϕ between the standard A-lattice and M. Assume $\phi(1) = \alpha \cdot a$ for some $a \in A$. Then $\mathrm{Tr}_{\mathbb{F}}(aa^*) = \langle \phi(1), \phi(1) \rangle_M = \langle 1, 1 \rangle = n$. By Lemma 2.2, we have $a \in \mu(A)$. Then we compute $\phi(1) \cdot \mu(A) = \alpha\mu(A)$ in polynomial time since $\sharp(\mu(A)) \leq 2n^2$.

2. Let $M = B^{-1}(A \cdot v) = A \cdot (B^{-1}v)$. It's an A-module. Define the inner product in M as $\langle , \rangle_M : (x, y) \in M^2 \mapsto \mathrm{Tr}_{\mathbb{F}}(\frac{x^*(B^*B)y}{v^*v}) \in \mathrm{Tr}_{\mathbb{F}}(A) \subseteq \mathbb{Z}$. Then M is a integral lattice.

   Consider $f : a \in A \mapsto B^{-1}(av) = a(B^{-1}v) \in M$. Obviously it's an isomophsim of A-module, and one can see $\langle f(x), f(y) \rangle_M = \langle x, y \rangle$ for all $x, y \in A$ by the definition of $\langle , \rangle_M$. So f is an A-isomorphsim, and M is an A-lattice isomorphic to standard A-lattice.

   Using the polynomial-time algorithm in Theorem 2.1, we can get an A-isomorphsim ϕ between the standard A-lattice and M. Assume $\phi(1) = B^{-1}(av)$ for some $a \in A$. Then $\mathrm{Tr}_{\mathbb{F}}(aa^*) = \langle \phi(1), \phi(1) \rangle_M = \langle 1, 1 \rangle = n$. By Lemma 2.2, we have $a \in \mu(A)$. Then we compute $\mu(A) \cdot \phi(1) = B^{-1}(\mu(A) \cdot v)$ in polynomial time since $\sharp(\mu(A)) \leq 2n^2$. □

4.2 Find J_{BU}

Using the Proposition 3.1, we only need to know $\det(BU)$ to obtain J_{BU} using G'. We have known B and thus $\det(B)$. It remains to find $\det(U)$. This can be done by Gentry's algorithm i.e. LS1. To do so, we first need to extract $\det(U) \cdot \mathcal{O}_{\mathbb{K}}$ from $(\mathcal{I}_i)$ and $(\mathcal{J}_i)$, which is not unexpected since U actually gives an isomorphism from $\bigoplus \mathcal{I}_i$ to $\bigoplus \mathcal{J}_i$.

Lemma 4.1 ([5, Proposition 1.4.2]) *Follow the setup as in Definition 2.2. We have* $\det(U) \cdot \mathcal{O}_{\mathbb{K}} = \prod_{k=1}^{\ell} \mathcal{I}_k \prod_{k=1}^{\ell} \mathcal{J}_k^{-1}$.

Proof. For any permutation $\sigma \in S_\ell$,

$$\prod_{k=1}^{\ell} u_{k\sigma(k)} \in \prod_{k=1}^{\ell} \mathcal{I}_k \mathcal{J}_{\sigma(k)}^{-1} = \prod_{k=1}^{\ell} \mathcal{I}_k \prod_{k=1}^{\ell} \mathcal{J}_k^{-1}.$$

By the definition of determinant, $\det(U) \in \prod_{k=1}^{\ell} \mathcal{I}_k \prod_{k=1}^{\ell} \mathcal{J}_k^{-1}$ i.e.

$$\det(U) \cdot \mathcal{O}_{\mathbb{K}} \subseteq \prod_{k=1}^{\ell} \mathcal{I}_k \prod_{k=1}^{\ell} \mathcal{J}_k^{-1}.$$

Symmetrically, we have $\det(V) \cdot \mathcal{O}_{\mathbb{K}} \subseteq \prod_{k=1}^{\ell} \mathcal{J}_k \prod_{k=1}^{\ell} \mathcal{I}_k^{-1}$. Note that $\det(V) = \det(U)^{-1}$, so

$$\det(U) \cdot \mathcal{O}_{\mathbb{K}} = \prod_{k=1}^{\ell} \mathcal{I}_k \prod_{k=1}^{\ell} \mathcal{J}_k^{-1}.$$

□

Proposition 4.2 *Follow the setup provided in Theorem 4.1. U is a congruence matrix. There is a deterministic polynomial-time algorithm that, given parameter $\mathbf{B}$, $\mathbf{K}$, $\mathbf{G}'$, computes $\pm J_{BU}$.*

Proof. We show Algorithm 4 satisfies the requirements.

```
Algorithm 4: ConjOfJ
Require: Parameter $\mathbf{B} = (B, (\mathcal{I}_i)_i)$ and $\mathbf{K}$. An module-LIP$^{\mathbf{B}}_{\mathbf{K}}$ instance
    $\mathbf{G}' = (G', (\mathcal{J}_i)_i)$.
Ensure: $\pm J_{BU}$ for a congruence matrix $U$ between $\mathbf{G}$(the pseudo-Gram matrix
    associated with $\mathbf{B}$) and $\mathbf{G}'$.
1: $\det(U) \cdot \mathcal{O}_{\mathbb{K}} \leftarrow \prod_{k=1}^{l} \mathcal{I}_k \prod_{k=1}^{l} \mathcal{J}_k^{-1}$
2: $\det(U)^2 \leftarrow \frac{\det(G')}{\det(B)^2}$
3: $W_1 \leftarrow \texttt{LS1}(\mathbf{K}, \det(U) \cdot \mathcal{O}_{\mathbb{K}}, \det(U)^2)$
4: $W_2 \leftarrow \det(B) \cdot W_1 \cdot G'^{-1} \cdot J_2$
5: return $W_2$.
```

Correctness: Step 1 is right by Lemma 4.1. Step 2 is right since $\det(G') = \det(U^T B^T BU) = \det(U^T)\det(B^T)\det(B)\det(U) = \det(B)^2 \det(U)^2$. Step 3 is right by Proposition 4.1.

Complexity: At Step 1, both ideals' multiplication and inversion run in polynomial time. At Step 2, both elements' multiplication and inversion run in polynomial time. Step 3 also runs in polynomial time by Proposition 4.1. □

Remark 2. Here, Algorithm LS1 can be replaced by an algorithm for computing square roots over algebraic number fields.

Remark 3. If we consider B as a matrix over $\mathbb{K}_{\mathbb{R}}$, where $\det(U)$ is still an element of $\mathbb{K}$, we can therefore obtain an approximate value of $\det(U)^2$ by calculating $\frac{\det(G')}{\det(B)^2}$, and then recover the exact value of $\det(U)^2$.

4.3 Find sub module lattice composed by eigenvectors of $\boldsymbol{J_{BU}}$

For the remainder of this section, we need to transfer the whole setting from **K** to **L**, since the eigenvalues $\pm\imath$ of J_2 are not in $\mathbb{K}$. Specifically, we will consider $(B, (\mathcal{I}_i\mathcal{O}_{\mathbb{L}}))$ instead of $(B, (\mathcal{I}_i))$, $(G', (\mathcal{J}_i\mathcal{O}_{\mathbb{L}}))$ instead of $(G', (\mathcal{J}_i))$. We can compute $\mathcal{I}_i\mathcal{O}_{\mathbb{L}}$ by calculating $\sum_{j=1}^{d} y_{ij}\mathcal{O}_{\mathbb{L}}$, where $\{y_{ij}\}$ are a LLL-reduced $\mathbb{Z}$-basis for $\mathcal{I}_i$. It can be done in time $\text{poly}(\text{size}(\mathcal{I}_i), \log \Delta_{\mathbb{L}})$. We do the same thing for J_i. So we can assume we input $(B, (\mathcal{I}_i\mathcal{O}_{\mathbb{L}})), (G', (\mathcal{J}_i\mathcal{O}_{\mathbb{L}}))$ when we input $\mathbf{B}, \mathbf{G}'$. The following lemma tells us that the congruence matrix U does not change.

Lemma 4.2 *Follow the setup provided in Definition 2.2. Define* $\mathcal{I}'_k := \mathcal{I}_k \cdot \mathcal{O}_{\mathbb{L}}$, $\mathcal{J}'_k := \mathcal{J}_k \cdot \mathcal{O}_{\mathbb{L}}\, 1 \leq k \leq \ell$. *Then* $U(\bigoplus_{k=1}^{\ell} \mathcal{J}'_k) = \bigoplus_{k=1}^{l} \mathcal{I}'_k$. *Here, the term "direct sum" refers to the Cartesian product.*

Proof. Assume $e_i \in \mathbb{L}^{\ell}$ and its i-th component is 1, while the rest are 0. Then $\bigoplus_{k=1}^{\ell} \mathcal{I}'_k = \sum_{k=1}^{\ell} \mathcal{O}_{\mathbb{L}} \cdot \mathcal{I}_k \cdot e_k = \mathcal{O}_{\mathbb{L}} \sum_{k=1}^{\ell} \mathcal{I}_k \cdot e_k = \mathcal{O}_{\mathbb{L}} \bigoplus_{k=1}^{\ell} \mathcal{I}_k$.(Note that the products here can all be viewed as group products of Abelian groups, and it is easy to verify the second equality sign from the point of view of Abelian groups.) Similarly, $\bigoplus_{k=1}^{\ell} \mathcal{J}'_k = \mathcal{O}_{\mathbb{L}} \bigoplus_{k=1}^{\ell} \mathcal{J}_k$. Since U and elements in $\mathcal{O}_{\mathbb{L}}$ commute under matrix multiplication, it's enough to show $U(\bigoplus_{k=1}^{\ell} \mathcal{J}_k) = \bigoplus_{k=1}^{\ell} \mathcal{I}_k$. In the setting in Definition 2.2, $U = (u_{ij})$ and $u_{ij} \in \mathcal{I}_i \cdot \mathcal{J}_j^{-1}$. So $\forall v \in \bigoplus_{k=1}^{\ell} \mathcal{J}_k$, Uv's i-th component is in $\mathcal{I}_i$ i.e. $Uv \in \bigoplus_{k=1}^{\ell} \mathcal{I}_k$. This means $U(\bigoplus_{k=1}^{\ell} \mathcal{J}_k) \subseteq \bigoplus_{k=1}^{\ell} \mathcal{I}_k$. Symmetrically, we have $U^{-1}(\bigoplus_{k=1}^{\ell} \mathcal{I}_k) \subseteq \bigoplus_{k=1}^{\ell} \mathcal{J}_k$. Thus $U(\bigoplus_{k=1}^{\ell} \mathcal{J}_k) = \bigoplus_{k=1}^{\ell} \mathcal{I}_k$. □

Remark 4. Lemma 4.2 essentially states the following: if $(B, (\mathcal{I}_i)_i)$ and $(B', (\mathcal{J}_i)_i)$ are two pseudo-bases of the same module lattice $M \subset \mathbb{K}^{\ell}$, then $(B, (\mathcal{I}_i\mathcal{O}_{\mathbb{L}})_i)$ and $(B', (\mathcal{J}_i\mathcal{O}_{\mathbb{L}})_i)$ are two pseudo-bases of the same module lattice $\mathcal{O}_{\mathbb{L}}M \subset \mathbb{L}^{\ell}$.

In the following we show by computation the intersection of the eigenspace of J_2 with the module lattice defined by **B**. And then consider the analogue after conjugating.

Definition 4.1 *Let* $\mathbb{K}$ *be a totally real number field and* $\mathbb{L} = \mathbb{K}[X]/(X^2+1)$. *For parameter* **B** *and* **K**, *assume* $B = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathbb{K}^{2\times 2}$. *Define* $\mathcal{I}_{\mathbf{B}} := (b - \imath a)\mathcal{I}_1\mathcal{O}_{\mathbb{L}} \cap (\imath c - d)\mathcal{I}_2\mathcal{O}_{\mathbb{L}}$, *and* $v_{\mathbf{B}} := \frac{1}{b-\imath a}\begin{pmatrix} a \\ b \end{pmatrix} + \frac{1}{\imath c - d}\begin{pmatrix} c \\ d \end{pmatrix}$. *We can see* $\mathcal{I}_{\mathbf{B}}$ *is an (fractional) ideal of* $\mathcal{O}_{\mathbb{L}}$, $v_{\mathbf{B}} \in \mathbb{L}^2$, *and* $\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, \mathcal{I}_{\mathbf{B}}^{-1}$ *can all be computed in polynomial time with inputs of the parameters* $\mathbf{B}, \mathbf{K}$.

Lemma 4.3 *Follow the setup provided in Theorem 4.1. U is a congruence matrix. Denote BU by* $\widetilde{B}$. *Assume* $J_{\widetilde{B}} = \widetilde{B}^{-1}J_2\widetilde{B}$ *and* $m_\imath = \imath I_2 = \widetilde{B}^{-1}\imath I_2\widetilde{B}$. *Then* $\ker(J_2 - \imath I_2) \cap B(\mathcal{I}_1\mathcal{O}_{\mathbb{L}} \bigoplus \mathcal{I}_2\mathcal{O}_{\mathbb{L}}) = \mathcal{I}_{\mathbf{B}} \cdot v_{\mathbf{B}}$ *and* $\ker(J_{\widetilde{B}} - m_\imath) \cap (\mathcal{J}_1\mathcal{O}_{\mathbb{L}} \bigoplus \mathcal{J}_2\mathcal{O}_{\mathbb{L}}) = \widetilde{B}^{-1}(\mathcal{I}_{\mathbf{B}} \cdot v_{\mathbf{B}})$.

Proof. Firstly,

$$\begin{aligned}
&\ker(J_2-\imath I_2)\cap B(\mathcal{I}_1\mathcal{O}_{\mathbb{L}}\oplus\mathcal{I}_2\mathcal{O}_{\mathbb{L}})\\
=&\{r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix}|r_j\in\mathcal{I}_j\mathcal{O}_{\mathbb{L}},\ J_2(r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix})=\imath(r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix})\}\\
=&\{r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix}|r_j\in\mathcal{I}_j\mathcal{O}_{\mathbb{L}},\ \begin{pmatrix}r_1b+r_2d\\-r_1a-r_2c\end{pmatrix}=\begin{pmatrix}\imath(r_1a+r_2c)\\\imath(r_1b+r_2d)\end{pmatrix}\}\\
=&\{r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix}|r_j\in\mathcal{I}_j\mathcal{O}_{\mathbb{L}},\ r_1b+r_2d=\imath(r_1a+r_2c)\}\\
=&\{r_1\begin{pmatrix}a\\b\end{pmatrix}+r_2\begin{pmatrix}c\\d\end{pmatrix}|r_j\in\mathcal{I}_j\mathcal{O}_{\mathbb{L}},\ r_1(b-\imath a)=r_2(\imath c-d)\}\\
=&\{\frac{r}{b-\imath a}\begin{pmatrix}a\\b\end{pmatrix}+\frac{r}{\imath c-d}\begin{pmatrix}c\\d\end{pmatrix}|r\in(b-\imath a)\mathcal{I}_1\mathcal{O}_{\mathbb{L}}\cap(\imath c-d)\mathcal{I}_2\mathcal{O}_{\mathbb{L}}\}\\
=&(b-\imath a)\mathcal{I}_1\mathcal{O}_{\mathbb{L}}\cap(\imath c-d)\mathcal{I}_2\mathcal{O}_{\mathbb{L}}\cdot\{\frac{1}{b-\imath a}\begin{pmatrix}a\\b\end{pmatrix}+\frac{1}{\imath c-d}\begin{pmatrix}c\\d\end{pmatrix}\}\\
=&\mathcal{I}_{\mathbf{B}}\cdot v_{\mathbf{B}},
\end{aligned}$$

and then

$$\begin{aligned}
&\ker(J_{\widetilde{B}}-m_\imath)\cap(\mathcal{J}_1\mathcal{O}_{\mathbb{L}}\oplus\mathcal{J}_2\mathcal{O}_{\mathbb{L}})\\
=&\ker(\widetilde{B}^{-1}(J_2-\imath I_2)\widetilde{B})\cap(\mathcal{J}_1\mathcal{O}_{\mathbb{L}}\oplus\mathcal{J}_2\mathcal{O}_{\mathbb{L}})\\
=&\widetilde{B}^{-1}(\ker(J_2-\imath I_2)\cap\widetilde{B}(\mathcal{J}_1\mathcal{O}_{\mathbb{L}}\oplus\mathcal{J}_2\mathcal{O}_{\mathbb{L}}))\\
=&\widetilde{B}^{-1}(\ker(J_2-\imath I_2)\cap B(\mathcal{I}_1\mathcal{O}_{\mathbb{L}}\oplus\mathcal{I}_2\mathcal{O}_{\mathbb{L}}))(\text{use } \textit{Lemma } 4.2)\\
=&\widetilde{B}^{-1}(\mathcal{I}_{\mathbf{B}}\cdot v_{\mathbf{B}})(\text{by above}).
\end{aligned}$$

□

The following lemma guarantees that we can compute the intersection above efficiently (without knowing $\widetilde{B}$). In fact, the integer version of intersection is already well known, we just need to modify it slightly.

Lemma 4.4 *Let $\mathbb{L}$ be a number field with degree n, and $B_{\mathcal{O}_{\mathbb{L}}}$ be a 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* $\ker(A)\cap\mathcal{L}$.

Proof. Assume $\mathrm{rank}(\mathcal{L})=r$. Note that $\ker(A)\cap\mathcal{L}=B_{\mathcal{L}}(\ker(A\cdot B_{\mathcal{L}})\cap\mathbb{Z}^r)$. Since $A\cdot B_{\mathcal{L}}\in\mathbb{L}^{2\times r}$, we can compute some $U\in\mathbb{Q}^{2n\times r}$ in polynomial time such that $A\cdot B_{\mathcal{L}}=\begin{pmatrix}B_{\mathcal{O}_{\mathbb{L}}}&0\\0&B_{\mathcal{O}_{\mathbb{L}}}\end{pmatrix}U$. Then $\ker(A)\cap\mathcal{L}=B_{\mathcal{L}}(\ker(U)\cap\mathbb{Z}^r)$, and $\ker(U)\cap\mathbb{Z}^r$ can be computed by using Hermit Normal Form (or Smith Normal Form) in polynomial time. □

Combining the lemmas and definitions in this subsection, we can directly obtain the following proposition.

Proposition 4.3 *Follow the setup provided in Theorem 4.1. U is a congruence matrix. There is a deterministic polynomial-time Algorithm 5 that, given J_{BU} and parameters $\mathbf{B}$, $\mathbf{K}$, $\mathbf{G}'$, output the ideal $\mathcal{I}_{\mathbf{B}}$, the vector $v_{\mathbf{B}}$, and the rank 1 module lattice $(BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$.*

```
Algorithm 5: EigenSubLat
Require: Parameter $\mathbf{B} = (B, (\mathcal{I}_i)_i)$ and $\mathbf{K}$. An module-LIP$^{\mathbf{B}}_{\mathbf{K}}$ instance
    $\mathbf{G}' = (G', (\mathcal{J}_i)_i)$. $J_{BU}$ for a congruence matrix $U$ between $\mathbf{G}$(the pseudo-Gram
    matrix associated with $\mathbf{B}$) and $\mathbf{G}'$. Write $B = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$.
Ensure: $\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$.
 1: $\mathcal{I}_{\mathbf{B}} \leftarrow (b - \imath a)\mathcal{I}_1\mathcal{O}_{\mathbb{L}} \cap (\imath c - d)\mathcal{I}_2\mathcal{O}_{\mathbb{L}}$
 2: $v_{\mathbf{B}} \leftarrow \frac{1}{b-\imath a}\begin{pmatrix} a \\ b \end{pmatrix} + \frac{1}{\imath c - d}\begin{pmatrix} c \\ d \end{pmatrix}$
 3: $(BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}} \leftarrow \ker(J_{BU} - m_\imath) \cap (\mathcal{J}_1\mathcal{O}_{\mathbb{L}} \oplus \mathcal{J}_2\mathcal{O}_{\mathbb{L}})$
 4: return $\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$.
```

Remark 5. Here we have computed a concrete pseudo-basis of $\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$ directly. In fact, if we first figure out $\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$ by finding $\ker(J_2 - \imath I_2) \cap B(\mathcal{I}_1\mathcal{O}_{\mathbb{L}} \oplus \mathcal{I}_2\mathcal{O}_{\mathbb{L}})$, and then find any pseudo-basis of $\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$, it will not affect the following operations. From this point of view it is better to generalize our algorithm to the case on $\mathbb{K}_{\mathbb{R}}$ (in this case our $\mathcal{I}_{\mathbf{B}}$ is not a fractional ideal of $\mathcal{O}_{\mathbb{K}}$).

4.4 Use Lenstra-Silverberg algorithm.

In the last subsection we ended up with a rank 1 module lattice. Just turn it into a cyclic module and we can use the Proposition 4.1.

Proposition 4.4 *Follow the setup provided in Theorem 4.1. U is a congruence matrix. There is a deterministic polynomial-time Algorithm 6 that, given $\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$ and parameters $\mathbf{B}$, $\mathbf{K}$, $\mathbf{G}'$, output $\mu(\mathbb{L}) \cdot (BU)^{-1}v_{\mathbf{B}}$, where $\mu(\mathbb{L})$ is the roots of unity contained in $\mathbb{L}$.*

Proof. Denote BU by $\widetilde{B}$ and $(BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$ by $\mathcal{L}'$.

Correctness: We have $\mathcal{L} = \mathcal{I}_{\mathbf{B}}^{-1} \cdot (\widetilde{B})^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}} = \widetilde{B}^{-1}\mathcal{I}_{\mathbf{B}}^{-1} \cdot \mathcal{I}_{\mathbf{B}} \cdot v_{\mathbf{B}} = \widetilde{B}^{-1}(\mathcal{O}_{\mathbb{L}} \cdot v_{\mathbf{B}})$. By Proposition 4.1, we can use $\mathbf{L}$, $G' = \widetilde{B}^*\widetilde{B}$, $\mathcal{L} = \widetilde{B}^{-1}(\mathcal{O}_{\mathbb{L}} \cdot v_{\mathbf{B}})$, $\omega = v_{\mathbf{B}}^* v_{\mathbf{B}}$ to find $\widetilde{B}^{-1}\mu(\mathbb{L}) \cdot v_{\mathbf{B}}$.

Complexity: On Step 1, one can compute $\mathcal{I}_{\mathbf{B}}^{-1}\pi_1(\mathcal{L}')$ firstly, where π_1 is the projection of vectors onto their first component (WLOG, we can assume $\pi_1(\mathcal{L}') \neq$

Algorithm 6: UseLS

```
Require: Parameters $\mathbf{B} = (B, (\mathcal{I}_i)_i)$ and $\mathbf{K}$. An module-LIP$^{\mathbf{B}}_{\mathbf{K}}$ instance
    $\mathbf{G}' = (G', (\mathcal{J}_i)_i)$. The ideal $\mathcal{I}_{\mathbf{B}}$, vector $v_{\mathbf{B}}$. The rank 1 module lattice
    $(BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$ for a congruence matrix $U$ between $\mathbf{G}$(the pseudo-Gram matrix
    associated with $\mathbf{B}$) and $\mathbf{G}'$.
Ensure: $\mu(\mathbb{L}) \cdot (BU)^{-1}v_{\mathbf{B}}$.
 1: $\mathcal{L} \leftarrow \mathcal{I}_{\mathbf{B}}^{-1}(BU)^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}}$(regard $r \in \mathcal{I}_{\mathbf{B}}^{-1}$ as $rI_2$)
 2: $\omega \leftarrow v_{\mathbf{B}}^{*}v_{\mathbf{B}}$
 3: $W \leftarrow \texttt{LS2}(\mathbf{L}, G', \omega, \mathcal{L})$
 4: return $W$.
```

$\{0\}$ i.e. $\pi_1(\widetilde{B}^{-1}v_{\mathbf{B}}) \neq 0$). Note $\pi_1(\mathcal{L}')$ is fractional ideal of $\mathcal{O}_{\mathbb{L}}$.(we have shown $\mathcal{L}' = \mathcal{I}_{\mathbf{B}}(\widetilde{B}^{-1}v_B)$, then $\pi_1(\mathcal{L}') = \mathcal{I}_{\mathbf{B}}\pi_1(\widetilde{B}^{-1}v_B)$.) So it's a product of fraction ideals and can be computed in polynomial time. Then take one vector $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ of the reduced basis of $\mathcal{L}'$ and consider embedding $\iota : x_1 \in \mathcal{I}_{\mathbf{B}}^{-1}\pi_1(\mathcal{L}') \mapsto \begin{pmatrix} x_1 \\ x_1(x_0^{-1}y_0) \end{pmatrix} \in \mathbb{L}^2$. We have $\iota(\mathcal{I}_{\mathbf{B}}^{-1}\pi_1(\mathcal{L}')) = \mathcal{L}'$.($\mathcal{L}' = \mathcal{I}_B(\widetilde{B}^{-1}v_B) \Rightarrow \forall \begin{pmatrix} x \\ y \end{pmatrix} \in \mathcal{L}'$, $x^{-1}y$ is a constant.) Therefore the image of a basis of $\mathcal{I}_{\mathbf{B}}^{-1}\pi_1(\mathcal{L}')$ under ι is a basis of $\mathcal{L}'$ and can be computed in polynomial time. So step 7 runs in polynomial time. Step 2 is just conjugation and matrix multiplication. Step 3 runs in polynomial time by Proposition 4.1. □

Remark 6. If the reader can accept the language of group products, then Step 1 is just a group product that corresponds matrix multiplications as the bilinear map and lattice vectors in $\mathbb{L}^2$ as output.

4.5 The algorithm

Proof (proof of Theorem 4.1). We prove the correctness and the time complexity as below.

Correctness: Denote BU by $\widetilde{B}$. Assume U is a congruence matrix U between $\mathbf{G}$ and $\mathbf{G}'$. By Proposition 4.2 we can assume $\sigma = (\widetilde{B})^{-1}J_2(\widetilde{B})$ after step 3. This time we have $(\mathcal{I}, v, \mathcal{L}')$ is just $(\mathcal{I}_{\mathbf{B}}, v_{\mathbf{B}}, (\widetilde{B})^{-1}\mathcal{I}_{\mathbf{B}}v_{\mathbf{B}})$ by Proposition 4.3. Next, by Proposition 4.4 we have $W = \widetilde{B}^{-1}\mu(\mathbb{L}) \cdot v_{\mathbf{B}}$, where $\mu(\mathbb{L})$ is the roots of unity contained in $\mathbb{L}$.
Then we assume $w = \widetilde{B}^{-1}v$ after Step 6. In this time, $(\widetilde{B})^{-1}J_2t_*(\widetilde{B})w = (\widetilde{B})^{-1}J_2t_*(\widetilde{B})\widetilde{B}^{-1}v = \widetilde{B}^{-1}J_2t_*v$ (note $\widetilde{B} \in \mathbb{K}^{2\times 2}$) and v, J_2t_*v are $\mathbb{L}$-linear independent (note $\langle v, J_2t_*v\rangle = 0$). So $(w|(\widetilde{B})^{-1}J_2t_*(\widetilde{B})w) = \widetilde{B}^{-1}(v|J_2t_*v)$ and then $D = \widetilde{B}$. Finally, we get $V = B^{-1}\widetilde{B} = U$.

Algorithm 7: FindU

```
Require: Parameter $\mathbf{B} = (B, (\mathcal{I}_i)_i)$ and $\mathbf{K}$. An module-LIP$^{\mathbf{B}}_{\mathbf{K}}$ instance
    $\mathbf{G}' = (G', (\mathcal{J}_i)_i)$.
Ensure: A congruence matrix $U$ between $\mathbf{G}$ (the pseudo-Gram matrix associated
    with $\mathbf{B}$) and $\mathbf{G}'$.
$P \leftarrow$ ConjOfJ$(\mathbf{B}, \mathbf{K}, \mathbf{G}')$
$S \leftarrow \varnothing$
for $\sigma \in P$ do
    $(\mathcal{I}, v, \mathcal{L}') \leftarrow$ EigenSubLat$(\sigma, \mathbf{B}, \mathbf{K}, \mathbf{G}'))$
    $W \leftarrow$ UseLS$(I, v, \mathcal{L}', \mathbf{B}, \mathbf{K}, \mathbf{G}')$
    for $w \in W$ do
        $D \leftarrow (v|J_2 t_* v)(w|(\widetilde{B})^{-1} J_2 t_*(\widetilde{B})w)^{-1}$
        $V \leftarrow B^{-1} D$
        if $V$ is a congruence matrix between $\mathbf{G}$ and $\mathbf{G}'$ then
            $S \leftarrow S \cup \{V\}$
        end if
    end for
end for
return $S$.
```

Complexity: We can compute matrix products and inverses over $\mathbb{K}$ (resp. $\mathbb{L}$) in polynomial time. By Proposition 3.2, we can compute $(\widetilde{B})^{-1} J_2 t_*(\widetilde{B})$ in polynomial time. By Proposition 4.2, 4.3, 4.4, Step 1, 4, 5 all run in polynomial time of size$(\mathbf{B}, \mathbf{K}, \mathbf{G}')$. And $\sharp(P) = 2, \sharp(W) = \sharp(\mathcal{U}_2(\mathcal{O}_{\mathbb{L}}))$ are in poly(degree($\mathbb{K}$)). In summary, the whole algorithm runs in polynomial time. □

Remark 7. The algorithms in Sections 3 and 4 can be seen as a deterministic reduction from module-LIP to finding a specific module automorphism, i.e. the conjugate of J, in the case of a rank 2 free module and a CM number field. In fact, this specific automorphism can be relaxed to any non-trivial module automorphism, i.e. the conjugate of any element J' in $\mathcal{U}_2(\mathcal{O}_{\mathbb{L}}) \setminus \mu(\mathbb{L}) I_2$. This can be easily achieved by modifying the algorithm in Section 4.
The algorithm in Section 4 can be considered in two parts: first, elevating the problem from the totally real number field $\mathbb{K}$ to the field extension of $\mathbb{K}$ with the imaginary unit; second, solving the module-LIP problem over the CM number field after obtaining the conjugate of J. Assuming the obtained module automorphism is the conjugate of J', we only need to modify the first step to elevate from a certain CM number field $\mathbb{L}$ to the extension of $\mathbb{L}$ with certain root of unity, ensuring that J' has eigenvalues in the extension. Note that this extension remains a CM number field.

Remark 8. We could do argument similar to the proof of Corollary 3.1 to show that: under the additional condition that a hint $\widetilde{B}^T \widetilde{B}$ is given, Theorem 4.1 still holds for CM number fields.

5 Conclusion

In this paper, we introduce a new tool called (pseudo) symplectic automorphism of the module, with which we can solve $\mathcal{O}_{\mathbb{L}}^2$-LIP efficiently for a CM number field $\mathbb{L}$. Although we do not know how to find such automorphism efficiently in general, a weak one can always be computed in polynomial time, which is enough to invalidate the omSVP assumptions utilized in HAWK's security proof and directly results in a provable deterministic polynomial-time algorithm solving module-LIP for rank-2 modules in $\mathbb{K}^2$ where $\mathbb{K}$ is a totally real number field.

Acknowledgments

Thank the anonymous referees for their valuable suggestions on improving the presentation of this paper. Hengyi Luo and Yanbin Pan were supported in part by National Key Research and Development Project (No. 2020YFA0712300), National Natural Science Foundation of China (No. 62372445, 62032009, 12226006) and Innovation Program for Quantum Science and Technology under Grant 2021ZD0302902. Anyu Wang is partially supported by Tsinghua University Dushi Program.

References

1. Aggarwal, D., Dadush, D., Regev, O., Stephens-Davidowitz, N.: Solving the shortest vector problem in 2^{n} time using discrete gaussian sampling: Extended abstract. In: Servedio, R.A., Rubinfeld, R. (eds.) Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015. pp. 733–742. ACM (2015), https://doi.org/10.1145/2746539.2746606
2. Benčina, B., Budroni, A., Chi-Domínguez, J.J., Kulkarni, M.: Properties of lattice isomorphism as a cryptographic group action. In: Saarinen, M.J., Smith-Tone, D. (eds.) Post-Quantum Cryptography. pp. 170–201. Springer Nature Switzerland, Cham (2024)
3. 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)
4. Blanks, T.L., Miller, S.D.: Generating cryptographically-strong random lattice bases and recognizing rotations of Z^n. In: Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings 12. pp. 319–338. Springer (2021)
5. 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
6. Ducas, L.: Provable lattice reduction of z n with blocksize n/2. Designs, Codes and Cryptography pp. 1–8 (2023)
7. Ducas, L., Gibbons, S.: Hull attacks on the lattice isomorphism problem. In: IACR International Conference on Public-Key Cryptography. pp. 177–204. Springer (2023)
8. 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
9. 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
10. Dutour Sikirić, M., Haensch, A., Voight, J., van Woerden, W.P.: A canonical form for positive definite matrices. Open Book Series **4**(1), 179–195 (2020)
11. Felderhoff, J., Pellet-Mary, A., Stehlé, D., Wesolowski, B.: Ideal-SVP is Hard for Small-Norm Uniform Prime Ideals. In: Theory of Cryptography, TCC 2023. Lecture Notes in Computer Science, vol. 14372, pp. 63–92. Springer Nature Switzerland, Taipei (Taiwan), Taiwan (Dec 2023). https://doi.org/10.1007/978-3-031-48624-1_3, https://hal.science/hal-04326750

12. Geißler, K., Smart, N.P.: Computing the $\mathrm{M} = \mathrm{U}\ \mathrm{U}^{\mathrm{t}}$ integer matrix decomposition. In: Paterson, K.G. (ed.) Cryptography and Coding, 9th IMA International Conference, Cirencester, UK, December 16-18, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2898, pp. 223–233. Springer (2003), https://doi.org/10.1007/978-3-540-40974-8_18
13. 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
14. Haviv, I., Regev, O.: On the lattice isomorphism problem. In: Chekuri, C. (ed.) Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014. pp. 391–404. SIAM (2014), https://doi.org/10.1137/1.9781611973402.29
15. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: digital signatures using the NTRU lattice. In: Joye, M. (ed.) Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2612, pp. 122–140. Springer (2003), https://doi.org/10.1007/3-540-36563-X_9
16. 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)
17. 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
18. 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
19. Lenstra Jr, H.W., Silverberg, A.: Testing isomorphism of lattices over cm-orders. SIAM Journal on Computing **48**(4), 1300–1334 (2019)
20. 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)
21. NIST: Post-quantum cryptography: digital signature schemes. round 1 additional signatures (2023)
22. Plesken, W., Souvignier, B.: Computing isometries of lattices. Journal of Symbolic Computation **24**(3-4), 327–334 (1997)
23. Szydlo, M.: Hypercubic lattice reduction and analysis of GGH and NTRU signatures. In: Biham, E. (ed.) Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings. Lecture Notes in Computer Science, vol. 2656, pp. 433–448. Springer (2003), https://doi.org/10.1007/3-540-39200-9_27