Algorithms for the module lattice isomorphism problem in certain fields

I) The module lattice isomorphism problem

1) Remainders: LIP :

The lattice isomorphism problem (LIP) asks, given a (bad) basis B of $\mathcal{O}\mathbb{Z}^n$ for some (secret) $O \in \mathcal{O}_n(\mathbb{R})$ orthogonal, to recover O. Equivalently, we can recover $C = O^{-1}B$, which is a (bad) basis of $\mathbb{Z}^n$.

LIP (Gram matrix formulation):

Given $G = B^T B = C^T C$, recover the matrix C (or any matrix $\tilde{C}$ s.t. $G = \tilde{C}^T \tilde{C}$ and $\tilde{C}$ is a basis of $\mathbb{Z}^n$).

Remarks : • The Gram matrix formulation of LIP is convenient because C (the solution) and G (the input) have integer coefficients ⇝ no need to manipulate real numbers with O and B. ⟹ This is the formulation we will

use in this talk.

- More generally, one could define LIP for other lattices L than $\mathbb{Z}^n$. (but we will not need it for this talk)

2) Reminders: number theory:

- $K = \mathbb{Q}[X]/P(X)$ number field of degree d
- $\mathcal{O}_K$ ring of integers, for this talk $\mathcal{O}_K = \mathbb{Z}[X]/P(X)$
- σ canonical embedding (injective):

$$\sigma : K \hookrightarrow \mathbb{C}^d \quad (=: \sigma_1(a), \ldots, =: \sigma_d(a))$$

$$a = \sum_{i=0}^{d-1} a_i X^i \mapsto (a(\alpha_1), \ldots, a(\alpha_d))$$

where $(\alpha_1, \ldots, \alpha_d)$ are the complex roots of P.

Four running examples:

- $K = \mathbb{Q}$, $\mathcal{O}_K = \mathbb{Z}$, $\sigma = id$, $d = 2^r$, totally real field (all roots of P are in $\mathbb{R}$)
- $K = \mathbb{Q}(\sqrt{2}, \sqrt{3}, \ldots, \sqrt{p_r})$,
- $K = \mathbb{Q}[X]/X^d+1$, d power-of-two $\rightsquigarrow$ power-of-two cyclotomic field
- $K = \mathbb{Q}[X]/X^d-X-1$, d prime $\rightsquigarrow$ NTRUprime field

- $K_{\mathbb{R}} \subseteq \mathbb{C}^d$, $K_{\mathbb{R}} := \{(x_1,\dots,x_d) \mid x_i \in \mathbb{R} \text{ if } \alpha_i \in \mathbb{R}, \; x_i = \overline{x_j} \text{ if } \alpha_i = \overline{\alpha_j}\}$

Via σ, we see K as a subset of $K_{\mathbb{R}}$.

- In $K_{\mathbb{R}}$, sum and multiplications are performed coordinate-wise.

- The size of an element a of K is the hermition norme of its embedding $\sigma(a)$:

$$\|a\| := \|\sigma(a)\| = \langle \sigma(a), \sigma(a) \rangle = \sum_{i=1}^{d} \overline{\sigma(a)}_i \, \sigma(a)_i$$

- σ and $\|\ \|$ extend to vectors of K^k (and matrices of $K^{k\times k}$)

- Over $K_{\mathbb{R}}$, we can define a complex conjugation:

for $x \in K_{\mathbb{R}}$, $\overline{x} = (\overline{x_1},\dots,\overline{x_d}) \in K_{\mathbb{R}}$.

⚠ if $x \in \sigma(K)$, we do not always have $\overline{x} \in \sigma(K)$ (This is true for totally real and CM fields, but not for other fields)

3) Module-LIP :

In this talk, we will only consider modules of rank 2.

Module-LIP: Given $G = \overline{\sigma(C)}^T \sigma(C) \in K_{\mathbb{R}}^{2\times 2}$ where C is a basis of $\mathcal{O}_K^2$, recover C (or any $\tilde{C}$ basis of $\mathcal{O}_K^2$ s.t. $G = \overline{\sigma(\tilde{C})}^T \sigma(\tilde{C})$).

Remark:

We can generalise the definition to other modules than $\mathcal{O}_K^2$, and to modules with larger rank.

Hawk: The module-LIP problem defined above (with the module $\mathcal{O}_K^2$) is used with K a power-of-two cyclotomic field for the signature scheme Hawk [Ducas, Postlethwaite, Pulles and van Woerden. Asiacrypt '23] submitted to the NIST.

II) Module-LIP over totally real fields:

1) Totally real fields:

Def: K is totally real if all the roots $\alpha_1, \dots, \alpha_d$ of P are in $\mathbb{R}$ (equivalently, $\sigma(K) \subseteq \mathbb{R}^d$).

Example: $K = \mathbb{Q}[X]/X^2-2 = \mathbb{Q}(\sqrt{2})$

more generally $K = \mathbb{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \dots)$

If K is totally real, $\forall a \in K$, $\overline{\sigma(a)} = \sigma(a)$,
so the complex conjugation on K is the identity
and we can reformulate module-LIP.

Module-LIP (totally real case): Given $G = C^T C$ with C a basis of $\mathcal{O}_K^2$, find C.

2) Algorithm:

e.g. $C = \begin{pmatrix} X & 1 \\ 3 & 2X \end{pmatrix}$

Let us write $C = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$ with $a, b, c, d \in \mathcal{O}_K$ and $ad - bc = 1$

(more generally, $ad-bc$ is a unit of $\mathcal{O}_K$, but let's say for simplicity that it is 1).

We are given

$G = C^T C$ and want

to recover a, b, c and d.

$$G = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} a & c \\ b & d \end{pmatrix} = \begin{pmatrix} a^2+b^2 & ac+bd \\ ac+bd & c^2+d^2 \end{pmatrix} =: \begin{pmatrix} q_1 & q_2 \\ q_2 & q_3 \end{pmatrix}$$

e.g. $G = \begin{pmatrix} X & 3 \\ 1 & 2X \end{pmatrix}\begin{pmatrix} X & 1 \\ 3 & 2X \end{pmatrix} = \begin{pmatrix} X^2+9 & 7X \\ 7X & 1+4X^2 \end{pmatrix}$

Key observation: The diagonal elements are sums of 2 squares of (secret) integers ($\in \mathcal{O}_K$)

How to solve sum of 2 squares equations?

over $\mathbb{Z}$

Consider $\mathbb{Q}(i)$ and $\mathbb{Z}[i]$.

if $z = z_1 + z_2 i \in \mathbb{Q}(i)$

then $z\bar{z} = z_1^2 + z_2^2$

So $z_0 = a + bi \in \mathbb{Z}[i]$ is a solution to the

norm equation $z\bar{z} = q_1$ $(= a^2 + b^2)$

To solve norme equation:

- construct the ideal $I = z_0 \times \mathbb{Z}[i]$
- use I and q_1 to recover z_0

over K totally real:

Same as over $\mathbb{Z}$!

- $L = K(i)$
- $z_0 := a + bi \in \mathcal{O}_L$ is a solution to the norm equation $z\bar{z} = q_1$

To solve it:

(1) Construct the ideal $I = z_0 \mathcal{O}_L$.

(2) use $I = z_0 \mathcal{O}_L + q_1 = z_0 \bar{z_0}$ to recover z_0.

---

over K totally real:

(2) can be done in poly time with an algorithm due to Gentry-Szydlo (and Lenstra-Silverberg).

(1) First idea: factor $\underbrace{q_1}_{\text{known}} \mathcal{O}_L = (z_0 \mathcal{O}_L) \times (\bar{z_0} \mathcal{O}_L) = I\bar{I}$

↳ requires re-randomization and heuristics if we want an efficient classical algorithm.

Second idea: recall $G = \begin{pmatrix} a^2+b^2 & ac+bd \\ ac+bd & c^2+d^2 \end{pmatrix} = \begin{pmatrix} q_1 & q_2 \\ q_2 & q_3 \end{pmatrix}$

and $ad-bc = \underline{1}$

Then $\frac{a+ib}{c+id} = \frac{q_2 - i}{q_3}$ → known from public information!

Recall example:

$C = \begin{pmatrix} x & 1 \\ 3 & 2x \end{pmatrix}$ $G = \begin{pmatrix} 11 & 7x \\ 7x & 9 \end{pmatrix}$ $(a+ib)q_3 = (x+3i)\times 9 = 9x+27i$

$(c+id)(q_2 - i) = (1+2xi)(7x-i) = 7x - i + 14x^2 i + 2x$
$= 9x + 27i$

if $a+ib$ and $c+id$ are coprime

(this is the case, let's admit it)

Then $\left(\frac{a+ib}{c+id}\right) \times \mathcal{O}_L \cap \mathcal{O}_L = a+ib\,\mathcal{O}_L = \underline{I}$

Example: imagine $a=3$, $c=5$, $b=d=0$, $\mathcal{O}_L = \mathbb{Z}[i]$

$\left(\frac{a+ib}{c+id} \middle| \mathcal{O}_L\right) \cap \mathcal{O}_L = \left\{\frac{3}{5}(x_1+x_2 i) \,\middle|\, x_1, x_2 \in \mathbb{Z}\right\} \cap \left\{y_1+y_2 i \,\middle|\, y_1, y_2 \in \mathbb{Z}\right\}$

$= \left\{3x_1+3x_2 i \,\middle|\, x_1, x_2 \in \mathbb{Z}\right\}$

We obtain a close formula for I, using only public information.

This solves (1). Combining with (2) solves the norm equation, so one recovers a, b. Similarly one recovers c and d.

We obtain a

poly time algorithm solving module-LIP (for module $\mathcal{O}_K^2$) when K is totally real

II) Module-LIP over other number fields:

1) Fields with one real embedding:

Imagine $\alpha_1 \in \mathbb{R}$ (the first root of P).

We are given $G = \begin{pmatrix} \overline{\sigma(a)}\sigma(a) + \overline{\sigma(b)}\sigma(b) & * \\ * & * \end{pmatrix}$

The first coordinate of $\overline{\sigma(a)}\sigma(a) + \overline{\sigma(b)}\sigma(b)$

is $\overline{a(\alpha_1)}\, a(\alpha_1) + \overline{b(\alpha_1)}\, b(\alpha_1)$

$= a(\alpha_1)^2 + b(\alpha_1)^2$ because $\alpha_1 \in \mathbb{R}$, $a, b \in \mathbb{Q}[X]$

$= (a^2+b^2)(\alpha_1)$

(admitted) Given (a^2+b^2) evaluated at α_1 with sufficiently many bits of precision, one can recover the coefficients (in $\mathbb{Q}$) of the polynomial a^2+b^2 exactly.

Doing the same for the other coefficients of G, we can recover

$$\tilde{G} = \begin{pmatrix} a^2+b^2 & ac+bd \\ ac+bd & c^2+d^2 \end{pmatrix} = C^T C$$

$\Rightarrow$ we are back to the same situation as for totally real fields.

We can apply the same norm equation algorithm.

(1) will work exactly the same ⇝ poly time

(2) is a bit more tricky. We can obtain

⇝ heuristic quantum poly time

⇝ heuristic classical poly time <u>if</u> the Galois group of K is <u>2-transitive</u>.

<u>Overall</u>: Heuristic poly time (classical) algo for module-LIP (in module $\mathcal{O}_K^2$) when K is an NTRU Prime field ($K = \mathbb{Q}[X]/X^p - X - 1$)

<u>2) CM fields</u> :

Let's look at a concrete example: $K = \mathbb{Q}(i)$, $\mathcal{O}_K = \mathbb{Z}[i]$.

$$C = \begin{pmatrix} a_1+ia_2 & c_1+ic_2 \\ b_1+ib_2 & d_1+id_2 \end{pmatrix}$$

$$G = \overline{C}^T C = \begin{pmatrix} a_1^2+a_2^2+b_1^2+b_2^2 & (a_1-ia_2)(c_1+ic_2)+(b_1-ib_2)(d_1+id_2) \\ (a_1+ia_2)(c_1-ic_2)+(b_1+ib_2)(d_1-id_2) & c_1^2+c_2^2+d_1^2+d_2^2 \end{pmatrix}$$

<u>Sum of 4 squares</u> on the diagonal now.

↳ Quaternion algebra? $A = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + ij\mathbb{Q}$ with $i^2 = j^2 = -1$ $ij = -ji$

$z_0 = a_1 + ia_2 + jb_1 + ijb_2$ is a solution to the

norm equation $z_0\overline{z_0} = q_1$ (where $\overline{z_0} = a_1 - ia_2 - jb_1 - ijb_2$)

↳ same 2 steps as before but in A now.

(1) works the same

(2) We don't know how to make it work...

(The non-commutativity is really annoying)

$\Longrightarrow$ We do not get an algorithm, but only a reduction.

Remark: This would generalize to any CM field, but then we consider A a quaternion algebra over a number field.

Overall: poly time reduction from module-LIP (with module $\mathcal{O}_K^2$) in cyclotomic fields (or CM fields) to the problem of computing the generator of a principal ideal in a quaternion algebra (over a nb field).
(i.e., a problem in a module of rk 1 in A).