Cryptanalysis of rank-2 module-LIP in Totally Real Number Fields

G. Mureau, A. Pellet-Mary, H. Pliatsok, A. Wallet

Eurocrypt 2024, Zurich, May 30th

Hawk (Ducas, Postlethwaite, Pulles, van Woerden 2022)[1]

1. NIST submission (additional call for signatures)
2. based on module-LIP over cyclotomic fields
3. efficient / compact

[1] Hawk: Module LIP makes Lattice Signatures Fast, Compact and Simple

Introduction

Hawk (Ducas, Postlethwaite, Pulles, van Woerden 2022)[1]

1. NIST submission (additional call for signatures)
2. based on module-LIP over cyclotomic fields
3. efficient / compact

This talk: Heuristic polynomial time (in many cases) algorithm solving module-LIP for rank-2 modules when K is **totally real**.

[1] Hawk: Module LIP makes Lattice Signatures Fast, Compact and Simple

Introduction

Hawk (Ducas, Postlethwaite, Pulles, van Woerden 2022)[1]

1. NIST submission (additional call for signatures)
2. based on module-LIP over cyclotomic fields
3. efficient / compact

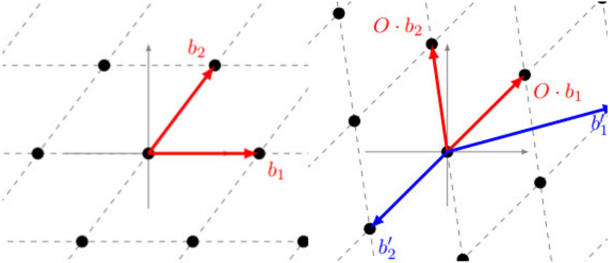
This talk: Heuristic polynomial time (in many cases) algorithm solving module-LIP for rank-2 modules when K is **totally real**.

Does not break Hawk!

[1] Hawk: Module LIP makes Lattice Signatures Fast, Compact and Simple

The module-Lattice Isomorphism Problem

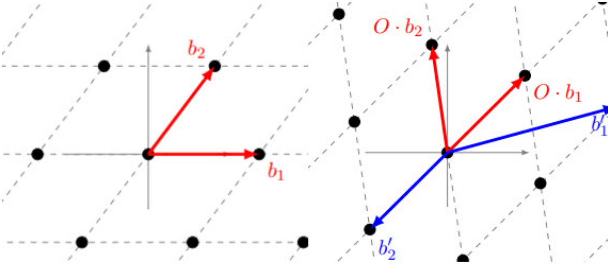
- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.



Given $(b_1 | b_2)$ and $(b'_1 | b'_2)$, find O.

The module-Lattice Isomorphism Problem

- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.



Given $(b_1|b_2)$ and $(b'_1|b'_2)$, find O.

State of the art: need to compute many short vectors in $\mathcal{L}_1$ and $\mathcal{L}_2$ (SVP, hard problem)

The module-Lattice Isomorphism Problem

- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.
- **module lattices** are finitely generated modules over $\mathcal{O}_K$ (K a number field).

Examples. $K = \mathbb{Q}[X]/(X^{2^k} + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^{2^k} + 1)$ (or $K = \mathbb{Q}$ and $\mathcal{O}_K = \mathbb{Z}$).

The module-Lattice Isomorphism Problem

- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.

- **module lattices** are finitely generated modules over $\mathcal{O}_K$ (K a number field).

Examples. $K = \mathbb{Q}[X]/(X^{2^k} + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^{2^k} + 1)$ (or $K = \mathbb{Q}$ and $\mathcal{O}_K = \mathbb{Z}$).

1. fractional ideals of K (rank one)
2. $\mathcal{O}_K \oplus \mathcal{O}_K$ (rank two)
3. in general: $M = \mathfrak{a}_1 v_1 \oplus \cdots \oplus \mathfrak{a}_\ell v_\ell$ (rank ℓ, $v_i \in K^\ell$, $\mathfrak{a}_i \subset K$)

The module-Lattice Isomorphism Problem

- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.

- **module lattices** are finitely generated modules over $\mathcal{O}_K$ (K a number field).

Examples. $K = \mathbb{Q}[X]/(X^{2^k} + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^{2^k} + 1)$ (or $K = \mathbb{Q}$ and $\mathcal{O}_K = \mathbb{Z}$).

1. fractional ideals of K (rank one)
2. $\mathcal{O}_K \oplus \mathcal{O}_K$ (rank two)
3. in general: $M = \mathfrak{a}_1 v_1 \oplus \cdots \oplus \mathfrak{a}_\ell v_\ell$ (rank ℓ, $v_i \in K^\ell$, $\mathfrak{a}_i \subset K$)

- module-LIP: Same as LIP but $\mathcal{L}_1$ and $\mathcal{L}_2$ are module lattices and seek for isometry preserving the **module structure**.

The module-Lattice Isomorphism Problem

- LIP: Find an isometry (distance preserving map) sending $\mathcal{L}_1$ on $\mathcal{L}_2$.

- **module lattices** are finitely generated modules over $\mathcal{O}_K$ (K a number field).

Examples. $K = \mathbb{Q}[X]/(X^{2^k} + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^{2^k} + 1)$ (or $K = \mathbb{Q}$ and $\mathcal{O}_K = \mathbb{Z}$).

1. fractional ideals of K (rank one)
2. $\mathcal{O}_K \oplus \mathcal{O}_K$ (rank two)
3. in general: $M = \mathfrak{a}_1 v_1 \oplus \cdots \oplus \mathfrak{a}_\ell v_\ell$ (rank ℓ, $v_i \in K^\ell$, $\mathfrak{a}_i \subset K$)

- module-LIP: Same as LIP but $\mathcal{L}_1$ and $\mathcal{L}_2$ are module lattices and seek for isometry preserving the **module structure**.

State of the art: embed module lattices to lattices $\subset \mathbb{R}^{d\ell}$ and solve LIP instance.

The module-Lattice Isomorphism Problem

Motivating example. K any number field and $M = \mathcal{O}_K \oplus \mathcal{O}_K$ (as in Hawk).

Notation: $X^* := \overline{X}^T$, for any $X \in M_2(K)$.

The module-Lattice Isomorphism Problem

Motivating example. K any number field and $M = \mathcal{O}_K \oplus \mathcal{O}_K$ (as in Hawk).
Notation: $X^* := \overline{X}^T$, for any $X \in M_2(K)$.

M' is **isomorphic** to M iff $\exists\, O : O^* O = Id$ and $M' = O \cdot M$.

The module-Lattice Isomorphism Problem

Motivating example. K any number field and $M = \mathcal{O}_K \oplus \mathcal{O}_K$ (as in Hawk).
Notation: $X^* := \overline{X}^T$, for any $X \in M_2(K)$.

M' is **isomorphic** to M iff $\exists\, O : O^* O = Id$ and $M' = O \cdot M$.
$M' = M'(B')$ is isomorphic to $M = M(B)$ iff $\exists\, U \in \mathrm{GL}_2(\mathcal{O}_K) : B' = OBU$.

The module-Lattice Isomorphism Problem

Motivating example. K any number field and $M = \mathcal{O}_K \oplus \mathcal{O}_K$ (as in Hawk).
Notation: $X^* := \overline{X}^T$, for any $X \in M_2(K)$.

M' is **isomorphic** to M iff $\exists\, O : O^* O = Id$ and $M' = O \cdot M$.
$M' = M'(B')$ is isomorphic to $M = M(B)$ iff $\exists\, U \in \mathrm{GL}_2(\mathcal{O}_K) : B' = OBU$.
Move to **quadratic forms**:

$$B \longmapsto G = B^* B \;\; ; \;\; B' \longmapsto G' = B'^* B', \quad \text{Gram matrix / Humbert form.}$$

$$B' = OBU \quad \Longrightarrow \quad G' = U^* G U, \quad \textbf{congruent} \text{ to } G.$$

The module-Lattice Isomorphism Problem

Motivating example. K any number field and $M = \mathcal{O}_K \oplus \mathcal{O}_K$ (as in Hawk).
Notation: $X^* := \overline{X}^T$, for any $X \in M_2(K)$.

M' is **isomorphic** to M iff $\exists\, O : O^* O = Id$ and $M' = O \cdot M$.
$M' = M'(B')$ is isomorphic to $M = M(B)$ iff $\exists\, U \in \mathrm{GL}_2(\mathcal{O}_K) : B' = OBU$.
Move to **quadratic forms**:

$$B \longmapsto G = B^* B \;\; ; \;\; B' \longmapsto G' = B'^* B', \quad \text{Gram matrix / Humbert form.}$$

$$B' = OBU \quad \Longrightarrow \quad G' = U^* G U, \quad \textbf{congruent} \text{ to } G.$$

Taking $B = G = I_2$, module-LIP with parameter K and I_2 is

module-LIP$_K^{I_2}$

Input: G' Gram matrix congruent to I_2
Goal: Compute **all** $U \in \mathrm{GL}_2(\mathcal{O}_K)$ s.t. $G' = U^* I_2 U = U^* U$.

Link with Hawk

Hawk : $K = \mathbb{Q}(\zeta_{2^k})$ **cyclotomic** number field (**not** totally real)
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (secret basis of $\mathcal{O}_K \oplus \mathcal{O}_K$)
$G = U^* U$ (public Gram matrix).

Link with Hawk

Hawk : $K = \mathbb{Q}(\zeta_{2^k})$ **cyclotomic** number field (**not** totally real)
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (secret basis of $\mathcal{O}_K \oplus \mathcal{O}_K$)
$G = U^* U$ (public Gram matrix).

- Recovering U from G is a module-LIP$_K^{I_2}$ instance.

Link with Hawk

Hawk : $K = \mathbb{Q}(\zeta_{2^k})$ **cyclotomic** number field (**not** totally real)
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (secret basis of $\mathcal{O}_K \oplus \mathcal{O}_K$)
$G = U^* U$ (public Gram matrix).

- Recovering U from G is a module-LIP$_K^{I_2}$ instance.
- Any solution $V^* V = G$ is a **key recovering** (up to automorphism).

The attack over totally real fields

Attack: Main idea

From now on K is **totally real** (*e.g.*, $K = \mathbb{Q}(\zeta + \zeta^{-1})$) and $U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O}_K)$

Attack: Main idea

From now on K is **totally real** (*e.g.*, $K = \mathbb{Q}(\zeta + \zeta^{-1})$) and $U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O}_K)$

Goal : Recover U from $G = U^* U$.

Attack: Main idea

From now on K is **totally real** (*e.g.*, $K = \mathbb{Q}(\zeta + \zeta^{-1})$) and $U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O}_K)$

Goal : Recover U from $G = U^* U$.

$$G = U^* U = \begin{pmatrix} a\bar{a} + b\bar{b} & \star \\ \star & c\bar{c} + d\bar{d} \end{pmatrix} = \begin{pmatrix} a^2 + b^2 & \star \\ \star & c^2 + d^2 \end{pmatrix}$$

because K is **totally real**! Diagonal elements are **sums of two squares** in $\mathcal{O}_K$.

Attack: Main idea

From now on K is **totally real** ($e.g.$, $K = \mathbb{Q}(\zeta + \zeta^{-1})$) and $U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O}_K)$

Goal : Recover U from $G = U^* U$.

$$G = U^* U = \begin{pmatrix} a\bar{a} + b\bar{b} & \star \\ \star & c\bar{c} + d\bar{d} \end{pmatrix} = \begin{pmatrix} a^2 + b^2 & \star \\ \star & c^2 + d^2 \end{pmatrix}$$

because K is **totally real**! Diagonal elements are **sums of two squares** in $\mathcal{O}_K$.

$a^2 + b^2 = (a + ib)(a - ib) =: N_{L/K}(a + ib)$ **relative norm** of $a + ib \in K(i) = L$.

Attack: Main idea

From now on K is **totally real** (*e.g.*, $K = \mathbb{Q}(\zeta + \zeta^{-1})$) and $U = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \in \mathrm{GL}_2(\mathcal{O}_K)$

Goal : Recover U from $G = U^* U$.

$$G = U^* U = \begin{pmatrix} a\bar{a} + b\bar{b} & \star \\ \star & c\bar{c} + d\bar{d} \end{pmatrix} = \begin{pmatrix} a^2 + b^2 & \star \\ \star & c^2 + d^2 \end{pmatrix}$$

because K is **totally real**! Diagonal elements are **sums of two squares** in $\mathcal{O}_K$.

$a^2 + b^2 = (a + ib)(a - ib) =: N_{L/K}(a + ib)$ **relative norm** of $a + ib \in K(i) = L$.

Main idea: Solve relative norm equations to reconstruct U.

Attack: Solving norm equations

- Howgrave-Graham, Szydlo, "A Method to Solve Cyclotomic Norm Equations $f \star \bar{f}$"

NormEquation

Input: $q \in \mathcal{O}_K$, prime factorization of $|N_{K/\mathbb{Q}}(q)| \in \mathbb{N}$.
Output: all pairs $(x, y) \in \mathcal{O}_K \times \mathcal{O}_K$ such that $N_{L/K}(x + iy) = x^2 + y^2 = q$.

Attack: Solving norm equations

- Howgrave-Graham, Szydlo, "A Method to Solve Cyclotomic Norm Equations $f \star \bar{f}$"

NormEquation

Input: $q \in \mathcal{O}_K$, prime factorization of $|N_{K/\mathbb{Q}}(q)| \in \mathbb{N}$.
Output: all pairs $(x, y) \in \mathcal{O}_K \times \mathcal{O}_K$ such that $N_{L/K}(x + iy) = x^2 + y^2 = q$.

It runs in time

$$\text{poly}\left(deg(K), (\log |N_{K/\mathbb{Q}}(q)|)^{\mathbf{r}}\right),$$

where **r** is the number of distinct prime factors of $q \cdot \mathcal{O}_K$.

Attack: Solving norm equations

- Howgrave-Graham, Szydlo, "A Method to Solve Cyclotomic Norm Equations $f \star \bar{f}$"

NormEquation

Input: $q \in \mathcal{O}_K$, prime factorization of $|N_{K/\mathbb{Q}}(q)| \in \mathbb{N}$.
Output: all pairs $(x, y) \in \mathcal{O}_K \times \mathcal{O}_K$ such that $N_{L/K}(x + iy) = x^2 + y^2 = q$.

It runs in time

$$\text{poly}\left(deg(K), (\log |N_{K/\mathbb{Q}}(q)|)^{\mathbf{r}}\right),$$

where $\mathbf{r}$ is the number of distinct prime factors of $q \cdot \mathcal{O}_K$.

- Randomization of the input to guarantee $\mathbf{r} = 1$.

$|N_{K/\mathbb{Q}}(q)|$ prime power ✔ Heuristic ✖

Attack: Solving norm equations

- Howgrave-Graham, Szydlo, "A Method to Solve Cyclotomic Norm Equations $f \star \bar{f}$"

NormEquation

Input: $q \in \mathcal{O}_K$, prime factorization of $|N_{K/\mathbb{Q}}(q)| \in \mathbb{N}$.
Output: all pairs $(x, y) \in \mathcal{O}_K \times \mathcal{O}_K$ such that $N_{L/K}(x + iy) = x^2 + y^2 = q$.

It runs in time

$$\text{poly}\left(deg(K), (\log |N_{K/\mathbb{Q}}(q)|)^{\mathbf{r}}\right),$$

where **r** is the number of distinct prime factors of $q \cdot \mathcal{O}_K$.

- Randomization of the input to guarantee $\mathbf{r} = 1$.

$|N_{K/\mathbb{Q}}(q)|$ prime power ✔ Heuristic ✖

$\Rightarrow$ Get norm equations easy to solve.

Attack: Randomization step

GaussianGram

Input : G matrix, $s > 0$ sampling parameter.
Output : $(u, v) \in \mathcal{O}_K \oplus \mathcal{O}_K$ follows a discrete Gaussian distribution.

1. Sample $V \in M_2(\mathcal{O}_K)$ invertible, using two calls to `GaussianGram`(G, s)

Attack: Randomization step

GaussianGram

Input : G matrix, $s > 0$ sampling parameter.
Output : $(u, v) \in \mathcal{O}_K \oplus \mathcal{O}_K$ follows a discrete Gaussian distribution.

1. Sample $V \in M_2(\mathcal{O}_K)$ invertible, using two calls to `GaussianGram`(G, s)
2. Compute $G_V = V^T G V = \begin{pmatrix} q_1 & \star \\ \star & q_2 \end{pmatrix}$
3. Repeat untill $q_1 \cdot \mathcal{O}_K$ and $q_2 \cdot \mathcal{O}_K$ are **prime ideals**

Attack: Randomization step

GaussianGram

Input : G matrix, $s > 0$ sampling parameter.
Output : $(u, v) \in \mathcal{O}_K \oplus \mathcal{O}_K$ follows a discrete Gaussian distribution.

1. Sample $V \in M_2(\mathcal{O}_K)$ invertible, using two calls to $\texttt{GaussianGram}(G, s)$
2. Compute $G_V = V^T G V = \begin{pmatrix} q_1 & \star \\ \star & q_2 \end{pmatrix}$
3. Repeat untill $q_1 \cdot \mathcal{O}_K$ and $q_2 \cdot \mathcal{O}_K$ are **prime ideals**

Heuristic for the probability of success.
Numerical experiments + theoretical results (distribution of prime ideals, involves ρ_K, residue of ζ_K at 1).

Attack: Statement

Solving module-LIP for $\mathcal{O}_K \oplus \mathcal{O}_K$.

Suppose $K = \mathbb{Q}(\zeta_{2^k} + \zeta_{2^k}^{-1})$ and G a Gram matrix.
$\exists$ heuristic algorithm solving module-LIP$_K^{l_2}$ on input G in expected time

$$poly(\rho_K, deg(K), size(G)),$$

ρ_K residue of ζ_K at 1 (small in our experiments).

Numerical experiments

Full attack here: https://gitlab.inria.fr/capsule/code-for-module-lip

$(m, 2d)$	$(64, 32)$	$(128, 64)$	$(256, 128)$
Time	2	25	850

$(m, 2d)$	$(228, 72)$	$(276, 88)$	$(260, 96)$	$(232, 112)$	$(340, 128)$	$(296, 144)$
Time (s)	74	195	434	652	2980	4205

Table: Times in seconds for attacks over various maximal totally real subfields K of cyclotomic fields with conductors $m = 4k$, averaged over 5 instances. The degree d of K is $\varphi(m)/2$, and the lattices involved have dimension $2d$. The upper table are powers-of-two. Experiments performed on a MacBook Pro (Apple M2), with Sagemath 10.2 and Pari/GP 2.15.5.

More generally

- module-LIP defined for any number field, any module lattice $M \subset K^\ell$.

$$\text{Find all "congruence matrices" } U \text{ s.t. } G' = U^* G U$$

More generally

- module-LIP defined for any number field, any module lattice $M \subset K^\ell$.

$$\text{Find all "congruence matrices" } U \text{ s.t. } G' = U^* G U$$

- Attack for **any totally real number field** K, **any module lattice** $M \subset K^2$.

More generally

- module-LIP defined for any number field, any module lattice $M \subset K^\ell$.

$$\text{Find all "congruence matrices" } U \text{ s.t. } G' = U^* G U$$

- Attack for **any totally real number field** K, **any module lattice** $M \subset K^2$.
- In general, can't hope for polynomial time complexity. Depends on the "Gram ideal" $\mathcal{G}(M) = \langle \, ||v||^2 \mid v \in M \rangle$. *e.g.*, $\mathcal{G}(\mathcal{O}_K \oplus \mathcal{O}_K) = \mathcal{O}_K$.

More generally

- module-LIP defined for any number field, any module lattice $M \subset K^\ell$.

$$\text{Find all "congruence matrices" } U \text{ s.t. } G' = U^* G U$$

- Attack for **any totally real number field** K, **any module lattice** $M \subset K^2$.
- In general, can't hope for polynomial time complexity. Depends on the "Gram ideal" $\mathcal{G}(M) = \langle \, ||v||^2 \mid v \in M \rangle$. $e.g., \mathcal{G}(\mathcal{O}_K \oplus \mathcal{O}_K) = \mathcal{O}_K$.

Solving module-LIP for rank-2 modules in totally real number fields.

Parameters: K totally real, $M \subset K^2$, with (pseudo-)basis B and $G = B^* B$.
Input: G' (pseudo-)Gram matrix congruent to G.

More generally

- module-LIP defined for any number field, any module lattice $M \subset K^\ell$.

$$\text{Find all "congruence matrices" } U \text{ s.t. } G' = U^* G U$$

- Attack for **any totally real number field** K, **any module lattice** $M \subset K^2$.
- In general, can't hope for polynomial time complexity. Depends on the "Gram ideal" $\mathcal{G}(M) = \langle \, ||v||^2 \mid v \in M \rangle$. *e.g.*, $\mathcal{G}(\mathcal{O}_K \oplus \mathcal{O}_K) = \mathcal{O}_K$.

Solving module-LIP for rank-2 modules in totally real number fields.

Parameters: K totally real, $M \subset K^2$, with (pseudo-)basis B and $G = B^* B$.
Input: G' (pseudo-)Gram matrix congruent to G.
$\exists$ heuristic algorithm finding all conguence matrices in expected time

$$\left(poly(\rho_K, \log \Delta_K, size(\mathbf{G'})) \right)^{\mathbf{r}+1} + T_{factor}(N_{K/\mathbb{Q}}(\mathcal{G}(M))),$$

where $\mathbf{r}$ is the number of distinct prime factors of $\mathcal{G}(M)$.

To sum up

- When K totally real and M has rank 2, module-LIP reduces to **norm equations** in number fields.

To sum up

- When K totally real and M has rank 2, module-LIP reduces to **norm equations** in number fields.

- Classical problem. We randomize to have easy instances.

To sum up

- When K totally real and M has rank 2, module-LIP reduces to **norm equations** in number fields.

- Classical problem. We randomize to have easy instances.

- Under some heuristic for the randomization, polynomial time (in many cases) algorithm solving module-LIP.

To sum up

• When K totally real and M has rank 2, module-LIP reduces to **norm equations** in number fields.

• Classical problem. We randomize to have easy instances.

• Under some heuristic for the randomization, polynomial time (in many cases) algorithm solving module-LIP.

Open questions. • For modules with rank $\ell > 2$?
• Rank 2 over K cyclotomic ?

Thanks for your attention!

Full article here!