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

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

[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

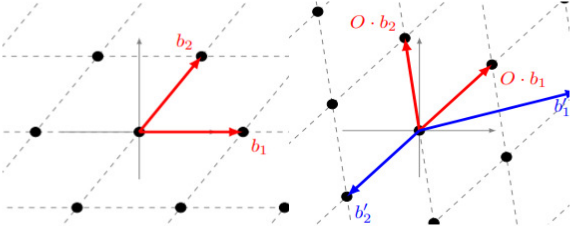
This talk : Heuristic polynomial time (in many cases) algorithm solving module-LIP for rank-2 modules when K is **totally real**.

Does not break Hawk!

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

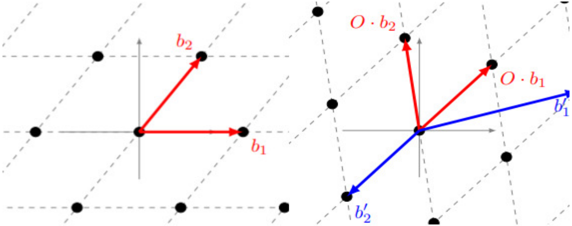
The module-Lattice Isomorphism Problem

- LIP : Find an isometry sending $\mathcal{L}_1$ on $\mathcal{L}_2$ (or determine if $\mathcal{L}_1$ and $\mathcal{L}_2$ are isometric).



The module-Lattice Isomorphism Problem

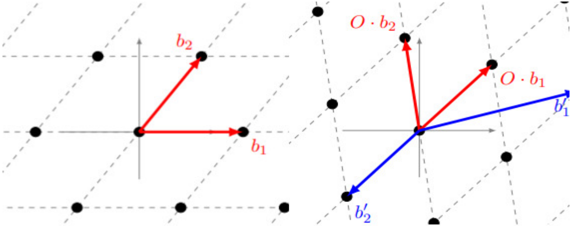
- LIP : Find an isometry sending $\mathcal{L}_1$ on $\mathcal{L}_2$ (or determine if $\mathcal{L}_1$ and $\mathcal{L}_2$ are isometric).



- module-LIP : Same but $\mathcal{L}_1$ and $\mathcal{L}_2$ are **module lattices** (finitely generated modules over $\mathcal{O}_K$, K a number field) and seek for isometry preserving the module structure.

The module-Lattice Isomorphism Problem

- LIP : Find an isometry sending $\mathcal{L}_1$ on $\mathcal{L}_2$ (or determine if $\mathcal{L}_1$ and $\mathcal{L}_2$ are isometric).

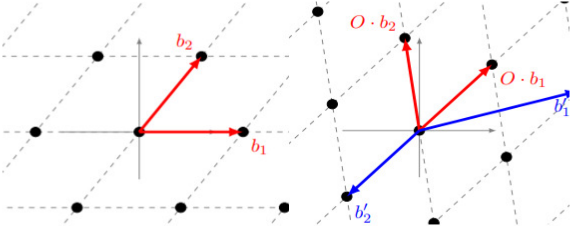


- module-LIP : Same but $\mathcal{L}_1$ and $\mathcal{L}_2$ are **module lattices** (finitely generated modules over $\mathcal{O}_K$, K a number field) and seek for isometry preserving the module structure.

Examples (of module lattices). *e.g.*, $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$

The module-Lattice Isomorphism Problem

- LIP : Find an isometry sending $\mathcal{L}_1$ on $\mathcal{L}_2$ (or determine if $\mathcal{L}_1$ and $\mathcal{L}_2$ are isometric).



- module-LIP : Same but $\mathcal{L}_1$ and $\mathcal{L}_2$ are **module lattices** (finitely generated modules over $\mathcal{O}_K$, K a number field) and seek for isometry preserving the module structure.

Examples (of module lattices). *e.g.*, $K = \mathbb{Q}[X]/(X^n + 1)$ and $\mathcal{O}_K = \mathbb{Z}[X]/(X^n + 1)$

1. rank one : fractional ideals of K
2. rank two : $\mathcal{O}_K \oplus \mathcal{O}_K$

are $\mathcal{O}_K$-modules which embed into an Euclidean lattice in $\mathbb{R}^{d\ell}$.

Example. Define module-LIP for $M = \mathcal{O}_K \oplus \mathcal{O}_K$.

The module-Lattice Isomorphism Problem

Example. Define module-LIP for $M = \mathcal{O}_K \oplus \mathcal{O}_K$.

M' is **isomorphic** to M iff $\exists\, O$ hermitian ($O^* O = Id$) such that $M' = O \cdot M$.
If B (resp. B') is a basis of M (resp. M'), then $O \cdot B \cdot U = B'$ for some $U \in \mathrm{GL}_2(\mathcal{O}_K)$.

The module-Lattice Isomorphism Problem

Example. Define module-LIP for $M = \mathcal{O}_K \oplus \mathcal{O}_K$.

M' is **isomorphic** to M iff $\exists\, O$ hermitian ($O^* O = Id$) such that $M' = O \cdot M$.
If B (resp. B') is a basis of M (resp. M'), then $O \cdot B \cdot U = B'$ for some $U \in \mathrm{GL}_2(\mathcal{O}_K)$.
Move to **quadratic forms** :

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

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

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. $G' = U^* U$.

Link with Hawk

Hawk : $K = \mathbb{Q}(\zeta_{2^k})$ **cyclotomic** number field
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (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
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (secret basis of $\mathcal{O}_K \oplus \mathcal{O}_K$)
$G = U^* U$ (public Gram matrix).

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

Link with Hawk

Hawk : $K = \mathbb{Q}(\zeta_{2^k})$ **cyclotomic** number field
$U \in \mathrm{GL}_2(\mathcal{O}_K)$ (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).

When K is totally real

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

When K is totally real

Suppose 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 $G = U^* U$.

When K is totally real

Suppose 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 $G = U^* U$.

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

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

When K is totally real

Suppose 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 $a + ib \in K(i) = L$.

When K is totally real

Suppose 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.

When K is totally real

- 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 $N_{L/K}(x + iy) = x^2 + y^2 = q$.

When K is totally real

- 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

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

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

When K is totally real

- 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

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

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

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

$\Rightarrow$ Get norm equations easy to solve.

When K is totally real

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$_K^{I_2}$ on input G in expected time

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

ρ_K residue at 1 of ζ_K (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 works 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 works 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 an invariant of the module $\mathcal{G}(M) = \langle\, ||v||^2 \mid v \in M \rangle$ "Gram ideal".

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 works 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 an invariant of the module $\mathcal{G}(M) = \langle \, ||v||^2 \mid v \in M \rangle$ "Gram ideal".

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 works 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 an invariant of the module $\mathcal{G}(M) = \langle\, \|v\|^2 \mid v \in M \rangle$ ”Gram ideal”.

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!