# Cryptanalysis of the module-Lattice Isomorphism Problem

LACS Seminar, Luxembourg University

**Guilhem Mureau**

Inria de Bordeaux, IMB, France


May 27th 2026

# The Lattice Isomorphism Problem (LIP)

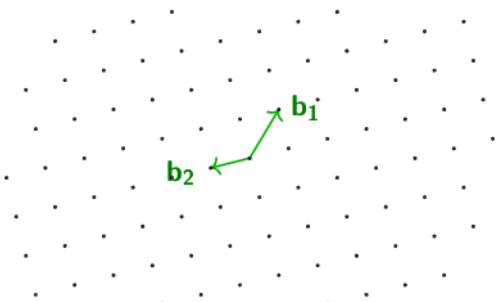
Let $\mathbf{b_1}, \ldots, \mathbf{b_n} \in \mathbb{R}^n$ be $\mathbb{R}$-linearly independent.

$\mathcal{L} := \{\sum_i x_i \mathbf{b_i} \mid x_i \in \mathbb{Z}\}$ is a **lattice** in $\mathbb{R}^n$ and $\mathbf{B} = (\mathbf{b_1} | \cdots | \mathbf{b_n})$ is a **basis** of $\mathcal{L}$.

# The Lattice Isomorphism Problem (LIP)

Let $\mathbf{b_1}, \ldots, \mathbf{b_n} \in \mathbb{R}^n$ be $\mathbb{R}$-linearly independent.

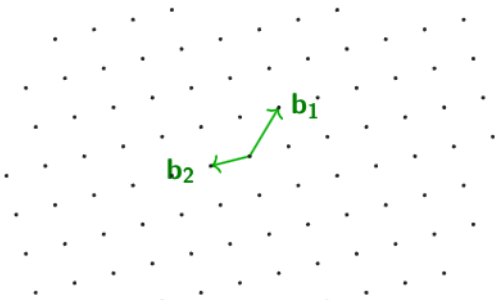
$\mathcal{L} := \{\sum_i x_i \mathbf{b_i} \mid x_i \in \mathbb{Z}\}$ is a **lattice** in $\mathbb{R}^n$ and $\mathbf{B} = (\mathbf{b_1} | \cdots | \mathbf{b_n})$ is a **basis** of $\mathcal{L}$.

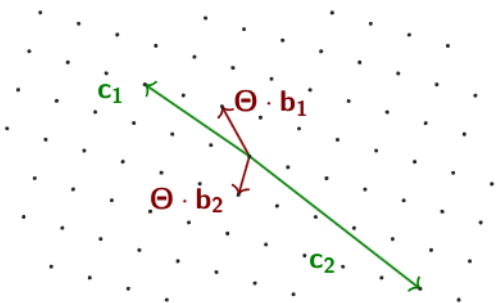


# The Lattice Isomorphism Problem (LIP)

Let $\mathbf{b_1}, \ldots, \mathbf{b_n} \in \mathbb{R}^n$ be $\mathbb{R}$-linearly independent.

$\mathcal{L} := \{\sum_i x_i \mathbf{b_i} \mid x_i \in \mathbb{Z}\}$ is a **lattice** in $\mathbb{R}^n$ and $\mathbf{B} = (\mathbf{b_1}|\cdots|\mathbf{b_n})$ is a **basis** of $\mathcal{L}$.




Lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ are **isomorphic** if $\mathcal{L}_2 = \mathbf{\Theta}(\mathcal{L}_1)$ for some $\mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R})$ orthogonal.

# Search and decision LIP

Given bases **B** and **C** of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

# Search and decision LIP

Given bases **B** and **C** of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

**Search LIP.** Assume that $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic. Compute an isometry

$$\mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \qquad \text{such that} \qquad \mathcal{L}_2 = \mathbf{\Theta}(\mathcal{L}_1).$$

# Search and decision LIP

Given bases $\mathbf{B}$ and $\mathbf{C}$ of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

**Search LIP.** Assume that $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic. Compute an isometry

$$\mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \qquad \text{such that} \qquad \mathcal{L}_2 = \mathbf{\Theta}(\mathcal{L}_1).$$

$$\Longleftrightarrow$$

$$\text{Compute } \mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \text{ or } \mathbf{U} \in \mathbf{GL}_n(\mathbb{Z}) \quad \text{s.t.} \quad \mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}.$$

# Search and decision LIP

Given bases $\mathbf{B}$ and $\mathbf{C}$ of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

**Search LIP.** Assume that $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic. Compute an isometry

$$\mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \qquad \text{such that} \qquad \mathcal{L}_2 = \mathbf{\Theta}(\mathcal{L}_1).$$

$$\Longleftrightarrow$$

$$\text{Compute } \mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \text{ or } \mathbf{U} \in \mathbf{GL}_n(\mathbb{Z}) \quad \text{s.t.} \quad \mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}.$$

**Decision LIP.** Decide whether $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic.

# Search and decision LIP

Given bases $\mathbf{B}$ and $\mathbf{C}$ of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively.

**Search LIP.** Assume that $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic. Compute an isometry

$$\mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \qquad \text{such that} \qquad \mathcal{L}_2 = \mathbf{\Theta}(\mathcal{L}_1).$$

$$\Longleftrightarrow$$

$$\text{Compute } \mathbf{\Theta} \in \mathcal{O}_n(\mathbb{R}) \text{ or } \mathbf{U} \in \mathbf{GL}_n(\mathbb{Z}) \quad \text{s.t.} \quad \mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}.$$

**Decision LIP.** Decide whether $\mathcal{L}_1$ and $\mathcal{L}_2$ are isomorphic.

**Remark:** Can state LIP with quadratic forms instead, useful for constructions.

# LIP in math and crypto

LIP is an old math problem: classifying lattices up to isometry. Related to old questions on quadratic forms, going back to Gauss.

# LIP in math and crypto

LIP is an old math problem: classifying lattices up to isometry. Related to old questions on quadratic forms, going back to Gauss.

In high dimension, LIP is a hard algorithmic problem. Known algorithms typically need to compute short vectors: Plesken & Souvignier (1997), Haviv & Regev (2013).

# LIP in math and crypto

LIP is an old math problem: classifying lattices up to isometry. Related to old questions on quadratic forms, going back to Gauss.

In high dimension, LIP is a hard algorithmic problem. Known algorithms typically need to compute short vectors: Plesken & Souvignier (1997), Haviv & Regev (2013).

In cryptography:

- Ducas, van Woerden (2021): primitives based on **decision LIP**;
- Ducas *et. al.* (2022): signature scheme **HAWK**. Its security is related to the hardness of **search module-LIP**.

# module-LIP?

A **module-lattice** is a lattice with extra algebraic structure.

# module-LIP?

A **module-lattice** is a lattice with extra algebraic structure.

This structure is useful for cryptography:

- compact representation;
- fast arithmetic;
- smaller keys and signatures.

# module-LIP?

A **module-lattice** is a lattice with extra algebraic structure.

This structure is useful for cryptography:

- compact representation;
- fast arithmetic;
- smaller keys and signatures.

**module-LIP** is LIP restricted to module-lattices, where the isomorphism must preserve the module structure.

# module-LIP?

A **module-lattice** is a lattice with extra algebraic structure.

This structure is useful for cryptography:

- compact representation;
- fast arithmetic;
- smaller keys and signatures.

**module-LIP** is LIP restricted to module-lattices, where the isomorphism must preserve the module structure. As for LIP, two versions: search & decision module-LIP.

# module-LIP?

A **module-lattice** is a lattice with extra algebraic structure.

This structure is useful for cryptography:

- compact representation;
- fast arithmetic;
- smaller keys and signatures.

**module-LIP** is LIP restricted to module-lattices, where the isomorphism must preserve the module structure. As for LIP, two versions: search & decision module-LIP.

*Can we use the extra structure of module-LIP to solve it more efficiently than LIP?*

# Plan of the talk

1. **Background on number theory**

   We introduce only what is needed to define module-LIP, don't run away please.

# Plan of the talk

1. **Background on number theory**

   We introduce only what is needed to define module-LIP, don't run away please.

2. **Results on search module-LIP**

   We target rank two module lattices, as in HAWK.

# Plan of the talk

1. **Background on number theory**

   We introduce only what is needed to define module-LIP, don't run away please.

2. **Results on search module-LIP**

   We target rank two module lattices, as in HAWK.

3. **Focus on decision module-LIP**

   We introduce computable invariants and discuss the impact on HAWK.

# Background on number theory

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

**Example.** For $d = 2^r$, consider the cyclotomic field $K = \mathbb{Q}[X]/(X^d + 1)$.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

**Example.** For $d = 2^r$, consider the cyclotomic field $K = \mathbb{Q}[X]/(X^d + 1)$.

Its ring of integers is $\mathbb{Z}_K = \mathbb{Z}[X]/(X^d + 1)$.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

**Example.** For $d = 2^r$, consider the cyclotomic field $K = \mathbb{Q}[X]/(X^d + 1)$.

Its ring of integers is $\mathbb{Z}_K = \mathbb{Z}[X]/(X^d + 1)$.

Elements are polynomials with rational (resp. integral) coeff in $\zeta$, where $\zeta^d = -1$.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

**Example.** For $d = 2^r$, consider the cyclotomic field $K = \mathbb{Q}[X]/(X^d + 1)$.

Its ring of integers is $\mathbb{Z}_K = \mathbb{Z}[X]/(X^d + 1)$.

Elements are polynomials with rational (resp. integral) coeff in $\zeta$, where $\zeta^d = -1$.

There is a **complex conjugation**: $\overline{\zeta} = \zeta^{-1}$.

# Background on number theory (1/4)

A **number field** of degree $d$ is a field of the form

$$K = \mathbb{Q}[X]/P(X),$$

where $P(X) \in \mathbb{Q}[X]$ is irreducible of degree $d$. Denote by $\mathbb{Z}_K$ its **ring of integers**.

**Example.** For $d = 2^r$, consider the cyclotomic field $K = \mathbb{Q}[X]/(X^d + 1)$.

Its ring of integers is $\mathbb{Z}_K = \mathbb{Z}[X]/(X^d + 1)$.

Elements are polynomials with rational (resp. integral) coeff in $\zeta$, where $\zeta^d = -1$.

There is a **complex conjugation**: $\overline{\zeta} = \zeta^{-1}$.

$\longrightarrow$ These fields are among the most common in lattice-based cryptography.

# Background on number theory (2/4)

Let $K = \mathbb{Q}[X]/P(X)$, where $P$ has degree $d$. Let $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ be the roots of $P$.

# Background on number theory (2/4)

Let $K = \mathbb{Q}[X]/P(X)$, where $P$ has degree $d$. Let $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ be the roots of $P$.

Each root gives an embedding

$$\sigma_i : K \hookrightarrow \mathbb{C}, \qquad \sigma_i(X) = \alpha_i.$$

# Background on number theory (2/4)

Let $K = \mathbb{Q}[X]/P(X)$, where $P$ has degree $d$. Let $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ be the roots of $P$.

Each root gives an embedding

$$\sigma_i : K \hookrightarrow \mathbb{C}, \qquad \sigma_i(X) = \alpha_i.$$

**Fact:** $d = d_1 + 2d_2$, where

- $d_1$ is the number of real embeddings;
- $d_2$ is the number of pairs of complex conjugate embeddings.

# Background on number theory (2/4)

Let $K = \mathbb{Q}[X]/P(X)$, where $P$ has degree $d$. Let $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ be the roots of $P$.

Each root gives an embedding

$$\sigma_i : K \hookrightarrow \mathbb{C}, \qquad \sigma_i(X) = \alpha_i.$$

**Fact:** $d = d_1 + 2d_2$, where

- $d_1$ is the number of real embeddings;
- $d_2$ is the number of pairs of complex conjugate embeddings.

Putting them together gives the **canonical embedding** $\sigma : K \hookrightarrow \mathbb{R}^{d_1} \times \mathbb{C}^{d_2} \simeq \mathbb{R}^d$.
$\longrightarrow$ an element of $K$ can be seen as a vector in $\mathbb{R}^d$.

# Background on number theory (2/4)

Let $K = \mathbb{Q}[X]/P(X)$, where $P$ has degree $d$. Let $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ be the roots of $P$.

Each root gives an embedding

$$\sigma_i : K \hookrightarrow \mathbb{C}, \qquad \sigma_i(X) = \alpha_i.$$

**Fact:** $d = d_1 + 2d_2$, where

- $d_1$ is the number of real embeddings;
- $d_2$ is the number of pairs of complex conjugate embeddings.

Putting them together gives the **canonical embedding** $\sigma : K \hookrightarrow \mathbb{R}^{d_1} \times \mathbb{C}^{d_2} \simeq \mathbb{R}^d$.
$\longrightarrow$ an element of $K$ can be seen as a vector in $\mathbb{R}^d$.

**Example:** For power-of-two cyclotomic fields, $K \hookrightarrow \mathbb{C}^{d/2} \simeq \mathbb{R}^d$.

# Background on number theory (3/4)

An **ideal** of $\mathbb{Z}_K$ is a subgroup $\mathfrak{a} \subseteq \mathbb{Z}_K$ s.t. $\mathbb{Z}_K \cdot \mathfrak{a} \subseteq \mathfrak{a}$.

# Background on number theory (3/4)

An **ideal** of $\mathbb{Z}_K$ is a subgroup $\mathfrak{a} \subseteq \mathbb{Z}_K$ s.t. $\mathbb{Z}_K \cdot \mathfrak{a} \subseteq \mathfrak{a}$. If for all $x, y \in \mathbb{Z}_K$,

$$xy \in \mathfrak{a} \Rightarrow x \in \mathfrak{a} \text{ or } y \in \mathfrak{a},$$

it is a **prime ideal**, usually denoted by $\mathfrak{p} = \mathfrak{a}$.

# Background on number theory (3/4)

An **ideal** of $\mathbb{Z}_K$ is a subgroup $\mathfrak{a} \subseteq \mathbb{Z}_K$ s.t. $\mathbb{Z}_K \cdot \mathfrak{a} \subseteq \mathfrak{a}$. If for all $x, y \in \mathbb{Z}_K$,

$$xy \in \mathfrak{a} \Rightarrow x \in \mathfrak{a} \text{ or } y \in \mathfrak{a},$$

it is a **prime ideal**, usually denoted by $\mathfrak{p} = \mathfrak{a}$.

**Example:** $K = \mathbb{Q}$, $\mathbb{Z}_K = \mathbb{Z}$ and $\mathfrak{p} = 2\mathbb{Z}$ is a prime ideal.

# Background on number theory (3/4)

An **ideal** of $\mathbb{Z}_K$ is a subgroup $\mathfrak{a} \subseteq \mathbb{Z}_K$ s.t. $\mathbb{Z}_K \cdot \mathfrak{a} \subseteq \mathfrak{a}$. If for all $x, y \in \mathbb{Z}_K$,

$$xy \in \mathfrak{a} \Rightarrow x \in \mathfrak{a} \text{ or } y \in \mathfrak{a},$$

it is a **prime ideal**, usually denoted by $\mathfrak{p} = \mathfrak{a}$.

**Example:** $K = \mathbb{Q}$, $\mathbb{Z}_K = \mathbb{Z}$ and $\mathfrak{p} = 2\mathbb{Z}$ is a prime ideal.

Let $\ell > 0$. An **isometry** of $K^\ell$ is a linear map $\mathbf{\Theta} : K^\ell \to K^\ell$ such that

$$\mathbf{\Theta}^* \mathbf{\Theta} = \mathbf{Id},$$

where $\mathbf{\Theta}^* = \overline{\Theta^T}$ is the transpose-conjugate.

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1 \mathbf{b_1} + \cdots + \mathfrak{a}_\ell \mathbf{b}_\ell,$$

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1 \mathbf{b_1} + \cdots + \mathfrak{a}_\ell \mathbf{b}_\ell,$$

where $\mathbf{b_1}, \ldots, \mathbf{b}_\ell$ are $K$-linearly independent and $\mathfrak{a}_1, \ldots, \mathfrak{a}_\ell \subseteq \mathbb{Z}_K$ are ideals.

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1\mathbf{b_1} + \cdots + \mathfrak{a}_\ell\mathbf{b}_\ell,$$

where $\mathbf{b_1}, \ldots, \mathbf{b}_\ell$ are $K$-linearly independent and $\mathfrak{a}_1, \ldots, \mathfrak{a}_\ell \subseteq \mathbb{Z}_K$ are ideals.

In this talk we consider $\mathfrak{a}_1 = \cdots = \mathfrak{a}_\ell = \mathbb{Z}_K$. We say $\mathbf{B} = (\mathbf{b_1}| \ldots |\mathbf{b}_\ell)$ is a **basis** of $\mathcal{L}$.

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1\mathbf{b_1} + \cdots + \mathfrak{a}_\ell\mathbf{b}_\ell,$$

where $\mathbf{b_1}, \ldots, \mathbf{b}_\ell$ are $K$-linearly independent and $\mathfrak{a}_1, \ldots, \mathfrak{a}_\ell \subseteq \mathbb{Z}_K$ are ideals.

In this talk we consider $\mathfrak{a}_1 = \cdots = \mathfrak{a}_\ell = \mathbb{Z}_K$. We say $\mathbf{B} = (\mathbf{b_1}| \ldots |\mathbf{b}_\ell)$ is a **basis** of $\mathcal{L}$.

Through the canonical embedding $\sigma$ defined earlier, $\mathcal{L}$ is a lattice in $\mathbb{R}^{d\ell}$.

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1 \mathbf{b_1} + \cdots + \mathfrak{a}_\ell \mathbf{b}_\ell,$$

where $\mathbf{b_1}, \ldots, \mathbf{b}_\ell$ are $K$-linearly independent and $\mathfrak{a}_1, \ldots, \mathfrak{a}_\ell \subseteq \mathbb{Z}_K$ are ideals.

In this talk we consider $\mathfrak{a}_1 = \cdots = \mathfrak{a}_\ell = \mathbb{Z}_K$. We say $\mathbf{B} = (\mathbf{b_1} | \ldots | \mathbf{b}_\ell)$ is a **basis** of $\mathcal{L}$.

Through the canonical embedding $\sigma$ defined earlier, $\mathcal{L}$ is a lattice in $\mathbb{R}^{d\ell}$.

**Example (HAWK):** Power-of-two cyclotomic $K = \mathbb{Q}[X]/(X^d + 1)$, and

$$\mathcal{L} = \mathbb{Z}_K \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \mathbb{Z}_K \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

# Background on number theory (4/4)

Let $\ell \in \mathbb{Z}_{>0}$. A **module-lattice** in $K^\ell$ is a $\mathbb{Z}_K$-module of the form:

$$\mathcal{L} = \mathfrak{a}_1 \mathbf{b_1} + \cdots + \mathfrak{a}_\ell \mathbf{b}_\ell,$$

where $\mathbf{b_1}, \ldots, \mathbf{b}_\ell$ are $K$-linearly independent and $\mathfrak{a}_1, \ldots, \mathfrak{a}_\ell \subseteq \mathbb{Z}_K$ are ideals.

In this talk we consider $\mathfrak{a}_1 = \cdots = \mathfrak{a}_\ell = \mathbb{Z}_K$. We say $\mathbf{B} = (\mathbf{b_1} | \ldots | \mathbf{b}_\ell)$ is a **basis** of $\mathcal{L}$.

Through the canonical embedding $\sigma$ defined earlier, $\mathcal{L}$ is a lattice in $\mathbb{R}^{d\ell}$.

**Example (HAWK):** Power-of-two cyclotomic $K = \mathbb{Q}[X]/(X^d + 1)$, and

$$\mathcal{L} = \mathbb{Z}_K \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \mathbb{Z}_K \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Moreover, $\{\text{Bases of } \mathcal{L}\} = \mathbf{GL}_2(\mathbb{Z}_K)$.

# Results on search module-LIP

# Results on search module-LIP (1/4)

Let $\mathcal{L}, \mathcal{M}$ be two module-lattices in $K^\ell$, with bases $\mathbf{B}, \mathbf{C} \in \mathbf{GL}_\ell(K)$.

# Results on search module-LIP (1/4)

Let $\mathcal{L}, \mathcal{M}$ be two module-lattices in $K^\ell$, with bases $\mathbf{B}, \mathbf{C} \in \mathbf{GL}_\ell(K)$.

**Search module-LIP.** Assume that $\mathcal{L}$ and $\mathcal{M}$ are isomorphic as module-lattices. Compute a $K$-linear isometry $\mathbf{\Theta} : K^\ell \longrightarrow K^\ell$ such that

$$\mathcal{M} = \mathbf{\Theta}(\mathcal{L}).$$

# Results on search module-LIP (1/4)

Let $\mathcal{L}, \mathcal{M}$ be two module-lattices in $K^\ell$, with bases $\mathbf{B}, \mathbf{C} \in \mathbf{GL}_\ell(K)$.

**Search module-LIP.** Assume that $\mathcal{L}$ and $\mathcal{M}$ are isomorphic as module-lattices. Compute a $K$-linear isometry $\mathbf{\Theta} : K^\ell \longrightarrow K^\ell$ such that

$$\mathcal{M} = \mathbf{\Theta}(\mathcal{L}).$$

Equivalently, compute $\mathbf{\Theta}$ or $\mathbf{U} \in \mathbf{GL}_\ell(\mathbb{Z}_K)$ such that $\mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}$.

# Results on search module-LIP (1/4)

Let $\mathcal{L}, \mathcal{M}$ be two module-lattices in $K^\ell$, with bases $\mathbf{B}, \mathbf{C} \in \mathbf{GL}_\ell(K)$.

**Search module-LIP.** Assume that $\mathcal{L}$ and $\mathcal{M}$ are isomorphic as module-lattices. Compute a $K$-linear isometry $\mathbf{\Theta} : K^\ell \longrightarrow K^\ell$ such that

$$\mathcal{M} = \mathbf{\Theta}(\mathcal{L}).$$

Equivalently, compute $\mathbf{\Theta}$ or $\mathbf{U} \in \mathbf{GL}_\ell(\mathbb{Z}_K)$ such that $\mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}$.

**Fact:** Security of HAWK related to module-LIP for $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$.

# Results on search module-LIP (1/4)

Let $\mathcal{L}, \mathcal{M}$ be two module-lattices in $K^\ell$, with bases $\mathbf{B}, \mathbf{C} \in \mathbf{GL}_\ell(K)$.

**Search module-LIP.** Assume that $\mathcal{L}$ and $\mathcal{M}$ are isomorphic as module-lattices. Compute a $K$-linear isometry $\mathbf{\Theta} : K^\ell \longrightarrow K^\ell$ such that

$$\mathcal{M} = \mathbf{\Theta}(\mathcal{L}).$$

Equivalently, compute $\mathbf{\Theta}$ or $\mathbf{U} \in \mathbf{GL}_\ell(\mathbb{Z}_K)$ such that $\mathbf{C} = \mathbf{\Theta}\,\mathbf{B}\,\mathbf{U}$.

**Fact:** Security of HAWK related to module-LIP for $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$. Precisely:

$$\text{Break module-LIP} \Longrightarrow \text{key recovering in HAWK.}$$

# Results on search module-LIP (2/4)

Multiple attempts to break module-LIP for rank two module-lattices (as in HAWK).

# Results on search module-LIP (2/4)

Multiple attempts to break module-LIP for rank two module-lattices (as in HAWK).

Results are highly dependent on the choice of parameter $K$:

# Results on search module-LIP (2/4)

Multiple attempts to break module-LIP for rank two module-lattices (as in HAWK).

Results are highly dependent on the choice of parameter $K$:

- When $K$ is **totally real** (all embeddings $K \hookrightarrow \mathbb{R}$), polynomial-time attack.
  Joint work with A. Pellet-Mary, H. Pliatsok and A. Wallet (EUROCRYPT 2024).

# Results on search module-LIP (2/4)

Multiple attempts to break module-LIP for rank two module-lattices (as in HAWK).

Results are highly dependent on the choice of parameter $K$:

- When $K$ is **totally real** (all embeddings $K \hookrightarrow \mathbb{R}$), polynomial-time attack.
  Joint work with A. Pellet-Mary, H. Pliatsok and A. Wallet (EUROCRYPT 2024).

- When $K$ is **totally complex** (no embedding $K \hookrightarrow \mathbb{R}$) as in HAWK, polynomial-time equivalence to a problem on ideals in a **quaternion algebra**.

  No practical attack on HAWK. Joint work with C. Chevignard, P-A. Fouque, A. Pellet-Mary, A. Wallet (EUROCRYPT 2025).

# Results on search module-LIP (2/4)

Multiple attempts to break module-LIP for rank two module-lattices (as in HAWK).

Results are highly dependent on the choice of parameter $K$:

- When $K$ is **totally real** (all embeddings $K \hookrightarrow \mathbb{R}$), polynomial-time attack.
  Joint work with A. Pellet-Mary, H. Pliatsok and A. Wallet (EUROCRYPT 2024).

- When $K$ is **totally complex** (no embedding $K \hookrightarrow \mathbb{R}$) as in HAWK, polynomial-time equivalence to a problem on ideals in a **quaternion algebra**.

  No practical attack on HAWK. Joint work with C. Chevignard, P-A. Fouque, A. Pellet-Mary, A. Wallet (EUROCRYPT 2025).

- Other choices of $K$: quantum polynomial-time attack. Classical poly-time for a large family of fields.

  B. Allombert, A. Pellet-Mary, W. van Woerden (EUROCRYPT 2025).

# Results on search module-LIP (3/4)

$\longrightarrow$ rank-2 module-LIP is broken for almost all fields $K$, but HAWK is **still safe**!

# Results on search module-LIP (3/4)

$\longrightarrow$ rank-2 module-LIP is broken for almost all fields $K$, but HAWK is **still safe**!

All papers follow the same principle:

Reduce rank-2 module-LIP over $K$ to rank-1 module-LIP over an extension $\mathcal{A}/K$.

# Results on search module-LIP (3/4)

$\longrightarrow$ rank-2 module-LIP is broken for almost all fields $K$, but HAWK is **still safe**!

All papers follow the same principle:

Reduce rank-2 module-LIP over $K$ to rank-1 module-LIP over an extension $\mathcal{A}/K$.

**Remark**: rank-1 module-lattices are ideal-lattices.

# Results on search module-LIP (3/4)

$\longrightarrow$ rank-2 module-LIP is broken for almost all fields $K$, but HAWK is **still safe**!

All papers follow the same principle:

Reduce rank-2 module-LIP over $K$ to rank-1 module-LIP over an extension $\mathcal{A}/K$.

**Remark**: rank-1 module-lattices are ideal-lattices.

- When $K$ is totally real, $\mathcal{A}$ is a number field. We use Gentry-Szydlo's algorithm.
- When $K$ is totally complex, $\mathcal{A}$ is a quaternion algebra. No similar tool is known.
- For other fields, $\mathcal{A}$ is a number field and $\exists$ extension of Gentry-Szydlo's algorithm.

# Results on search module-LIP (4/4)

D. van Gent, W. van Woerden (2025): A search-to-distinguish reduction for module-LIP. It holds for a specific family of module-lattices, including HAWK's lattice.

# Results on search module-LIP (4/4)

D. van Gent, W. van Woerden (2025): A search-to-distinguish reduction for module-LIP. It holds for a specific family of module-lattices, including HAWK's lattice.

To break HAWK's **search** module-LIP instance, it is enough to solve

"Are $\mathcal{L}$ and $\mathcal{M}$ isomorphic module-lattices?"

for a polynomially many pairs $(\mathcal{L}, \mathcal{M})$.

# Results on search module-LIP (4/4)

D. van Gent, W. van Woerden (2025): A search-to-distinguish reduction for module-LIP. It holds for a specific family of module-lattices, including HAWK's lattice.

To break HAWK's **search** module-LIP instance, it is enough to solve

"Are $\mathcal{L}$ and $\mathcal{M}$ isomorphic module-lattices?"

for a polynomially many pairs $(\mathcal{L}, \mathcal{M})$.

$\longrightarrow$ Gives a new direction to target HAWK!

# Focus on decision module-LIP

# Focus on decision module-LIP (1/10)

Let $\ell > 0$ (e.g. $\ell = 2$) and $\mathcal{L}, \mathcal{M} \subset K^\ell$ module-lattices.

# Focus on decision module-LIP (1/10)

Let $\ell > 0$ (e.g. $\ell = 2$) and $\mathcal{L}, \mathcal{M} \subset K^\ell$ module-lattices. We target:

**decision module-LIP.** Test whether $\mathcal{L}$ and $\mathcal{M}$ are isomorphic (as module-lattices):

$$\exists \mathbf{\Theta} : K^\ell \to K^\ell \qquad \text{s.t.} \qquad \begin{cases} \mathbf{\Theta}(\mathcal{L}) = \mathcal{M}; \\ \mathbf{\Theta}^* \mathbf{\Theta} = \mathbf{Id}. \end{cases}$$

# Focus on decision module-LIP (1/10)

Let $\ell > 0$ (e.g. $\ell = 2$) and $\mathcal{L}, \mathcal{M} \subset K^\ell$ module-lattices. We target:

**decision module-LIP.** Test whether $\mathcal{L}$ and $\mathcal{M}$ are isomorphic (as module-lattices):

$$\exists\, \mathbf{\Theta} : K^\ell \to K^\ell \qquad \text{s.t.} \qquad \begin{cases} \mathbf{\Theta}(\mathcal{L}) = \mathcal{M}; \\ \mathbf{\Theta}^* \mathbf{\Theta} = \mathbf{Id}. \end{cases}$$

When this holds, we write $\mathcal{M} \in \mathrm{cls}(\mathcal{L})$.

# Focus on decision module-LIP (1/10)

Let $\ell > 0$ (e.g. $\ell = 2$) and $\mathcal{L}, \mathcal{M} \subset K^\ell$ module-lattices. We target:

**decision module-LIP.** Test whether $\mathcal{L}$ and $\mathcal{M}$ are isomorphic (as module-lattices):

$$\exists\, \mathbf{\Theta} : K^\ell \to K^\ell \qquad \text{s.t.} \qquad \begin{cases} \mathbf{\Theta}(\mathcal{L}) = \mathcal{M}; \\ \mathbf{\Theta}^* \mathbf{\Theta} = \mathbf{Id}. \end{cases}$$

When this holds, we write $\mathcal{M} \in \mathrm{cls}(\mathcal{L})$.

**Strategy.** This is very hard in high dimensions. Instead, we check for isomorphisms over **larger spaces**. Only a necessary condition $\to$ gives a **partial distinguisher**.

# Focus on decision module-LIP (2/10)

Over a larger spaces?

# Focus on decision module-LIP (2/10)

Over a larger spaces? To each prime ideal $\mathfrak{p} \subset \mathbb{Z}_K$, we associate a **local** field $K \hookrightarrow K_{\mathfrak{p}}$.

# Focus on decision module-LIP (2/10)

Over a larger spaces? To each prime ideal $\mathfrak{p} \subset \mathbb{Z}_K$, we associate a **local** field $K \hookrightarrow K_{\mathfrak{p}}$.

We test isomorphism **locally at** $\mathfrak{p}$:

$$\exists\, \mathbf{\Theta}_{\mathfrak{p}} : K_{\mathfrak{p}}^{\ell} \to K_{\mathfrak{p}}^{\ell} \qquad \text{s.t.} \qquad \begin{cases} \mathbf{\Theta}_{\mathfrak{p}}(\mathcal{L}_{\mathfrak{p}}) = \mathcal{M}_{\mathfrak{p}}; \\ \mathbf{\Theta}_{\mathfrak{p}}^{*}\mathbf{\Theta}_{\mathfrak{p}} = \mathbf{Id}. \end{cases}$$

# Focus on decision module-LIP (2/10)

Over a larger spaces? To each prime ideal $\mathfrak{p} \subset \mathbb{Z}_K$, we associate a **local** field $K \hookrightarrow K_{\mathfrak{p}}$.

We test isomorphism **locally at** $\mathfrak{p}$:

$$\exists\, \mathbf{\Theta}_{\mathfrak{p}} : K_{\mathfrak{p}}^{\ell} \to K_{\mathfrak{p}}^{\ell} \qquad \text{s.t.} \qquad \begin{cases} \mathbf{\Theta}_{\mathfrak{p}}(\mathcal{L}_{\mathfrak{p}}) = \mathcal{M}_{\mathfrak{p}}; \\ \mathbf{\Theta}_{\mathfrak{p}}^{*}\mathbf{\Theta}_{\mathfrak{p}} = \mathbf{Id}. \end{cases}$$

Two module-lattices $\mathcal{L}, \mathcal{M} \subset K^{\ell}$ are said to be in the same **genus** if they are locally isomorphic at **every** prime ideal $\mathfrak{p}$. Notation: $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$.

# Focus on decision module-LIP (3/10)

Isomorphic module-lattices are locally isomorphic at any $\mathfrak{p}$: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

# Focus on decision module-LIP (3/10)

Isomorphic module-lattices are locally isomorphic at any $\mathfrak{p}$: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

In general, $\mathrm{gen}(\mathcal{L})$ is the disjoint union of finitely many isomorphism classes.

# Focus on decision module-LIP (3/10)

Isomorphic module-lattices are locally isomorphic at any $\mathfrak{p}$: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

In general, $\mathrm{gen}(\mathcal{L})$ is the disjoint union of finitely many isomorphism classes.

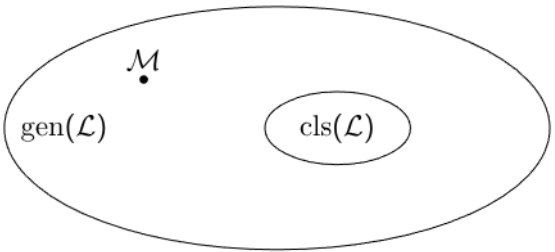
Testing $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$ is easy: $\exists$ poly-time algorithm $\longrightarrow$ Can assume $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$.

# Focus on decision module-LIP (3/10)

Isomorphic module-lattices are locally isomorphic at any $\mathfrak{p}$: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

In general, $\mathrm{gen}(\mathcal{L})$ is the disjoint union of finitely many isomorphism classes.

Testing $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$ is easy: $\exists$ poly-time algorithm $\longrightarrow$ Can assume $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$.



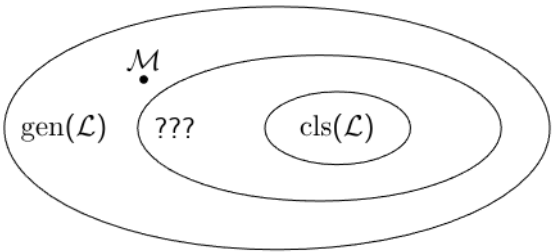

**Intermediate relation?** Find an equivalence relation $\sim$ on $\mathrm{gen}(\mathcal{L})$ s.t. the equivalence class of $\mathcal{L}$ is between $\mathrm{gen}(\mathcal{L})$ and $\mathrm{cls}(\mathcal{L})$. We want $\mathcal{M} \sim \mathcal{L}$ to be **efficiently testable**.

# Focus on decision module-LIP (3/10)

Isomorphic module-lattices are locally isomorphic at any $\mathfrak{p}$: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

In general, $\mathrm{gen}(\mathcal{L})$ is the disjoint union of finitely many isomorphism classes.

Testing $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$ is easy: $\exists$ poly-time algorithm $\longrightarrow$ Can assume $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$.




**Intermediate relation?** Find an equivalence relation $\sim$ on $\mathrm{gen}(\mathcal{L})$ s.t. the equivalence class of $\mathcal{L}$ is between $\mathrm{gen}(\mathcal{L})$ and $\mathrm{cls}(\mathcal{L})$. We want $\mathcal{M} \sim \mathcal{L}$ to be **efficiently testable**.

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

Assume $K$ is **totally real**. $\exists$ a poly-time (quantum) algorithm to distinguish spinor genera of module-lattices in $K^2$. Gives an intermediate classification.

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

Assume $K$ is **totally real**. $\exists$ a poly-time (quantum) algorithm to distinguish spinor genera of module-lattices in $K^2$. Gives an intermediate classification.

**Limitations:** In HAWK, $K$ is totally complex $\neq$ totally real $\longrightarrow$ no practical impact.

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

Assume $K$ is **totally real**. $\exists$ a poly-time (quantum) algorithm to distinguish spinor genera of module-lattices in $K^2$. Gives an intermediate classification.

**Limitations:** In HAWK, $K$ is totally complex $\neq$ totally real $\longrightarrow$ no practical impact.

Analogue when $K$ is totally complex?

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

Assume $K$ is **totally real**. $\exists$ a poly-time (quantum) algorithm to distinguish spinor genera of module-lattices in $K^2$. Gives an intermediate classification.

**Limitations:** In HAWK, $K$ is totally complex $\neq$ totally real $\longrightarrow$ no practical impact.

Analogue when $K$ is totally complex?

**M.** (TCC 2025): Assume $K$ is **totally complex**.
$\exists$ algorithm to distinguish **special genera** of module-lattices in $K^\ell$. For several module-lattices (HAWK-like lattices), it runs in classical poly-time

# Focus on decision module-LIP (4/10)

C. Ling, J. Liu, A. Mendelsohn (ASIACRYPT 2024) considered the **spinor genus**.

Assume $K$ is **totally real**. $\exists$ a poly-time (quantum) algorithm to distinguish spinor genera of module-lattices in $K^2$. Gives an intermediate classification.

**Limitations:** In HAWK, $K$ is totally complex $\neq$ totally real $\longrightarrow$ no practical impact.

Analogue when $K$ is totally complex?

**M.** (TCC 2025): Assume $K$ is **totally complex**.
$\exists$ algorithm to distinguish **special genera** of module-lattices in $K^\ell$. For several module-lattices (HAWK-like lattices), it runs in classical poly-time

**Rest of this talk:** Define and distinguish special genera + discuss impact on HAWK.

# Focus on decision module-LIP (5/10)

Module-lattices $\mathcal{L}$, $\mathcal{M} \subset K^\ell$ belong to the same **special genus** if:

$$(\exists\, \mathbf{\Theta}, \forall\, \mathfrak{p}, \exists\, \mathbf{\Theta}_\mathfrak{p} \text{ with } \det \mathbf{\Theta}_\mathfrak{p} = 1) \qquad \text{s.t.} \qquad \mathcal{M}_\mathfrak{p} = \mathbf{\Theta} \circ \mathbf{\Theta}_\mathfrak{p}(\mathcal{L}_\mathfrak{p}).$$

# Focus on decision module-LIP (5/10)

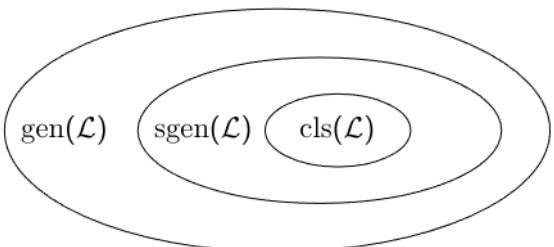
Module-lattices $\mathcal{L}, \mathcal{M} \subset K^\ell$ belong to the same **special genus** if:

$$(\exists\, \mathbf{\Theta}, \forall\, \mathfrak{p}, \exists\, \mathbf{\Theta}_\mathfrak{p} \text{ with } \det \mathbf{\Theta}_\mathfrak{p} = 1) \qquad \text{s.t.} \qquad \mathcal{M}_\mathfrak{p} = \mathbf{\Theta} \circ \mathbf{\Theta}_\mathfrak{p}(\mathcal{L}_\mathfrak{p}).$$

Equivalence relation $\sim$ on $\mathrm{gen}(\mathcal{L})$,
denote the class of $\mathcal{L}$ by $\mathrm{sgen}(\mathcal{L})$.

We have: $\mathrm{cls}(\mathcal{L}) \subseteq \mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

Gives an **intermediate classification** between $\mathrm{gen}(\mathcal{L})$ and $\mathrm{cls}(\mathcal{L})$.



# Focus on decision module-LIP (6/10)

Shimura (1964) studied how $\mathrm{gen}(\mathcal{L})$ splits into multiple special genera.

# Focus on decision module-LIP (6/10)

Shimura (1964) studied how gen($\mathcal{L}$) splits into multiple special genera.

Let $\ell > 0$ and module-lattices $\mathcal{L}$, $\mathcal{M} \subset K^\ell$.

# Focus on decision module-LIP (6/10)

Shimura (1964) studied how $\mathrm{gen}(\mathcal{L})$ splits into multiple special genera.

Let $\ell > 0$ and module-lattices $\mathcal{L}, \mathcal{M} \subset K^\ell$.

The **module index** of $\mathcal{M}$ in $\mathcal{L}$ is an ideal denoted by

$$[\mathcal{L} : \mathcal{M}]$$

# Focus on decision module-LIP (6/10)

Shimura (1964) studied how $\mathrm{gen}(\mathcal{L})$ splits into multiple special genera.

Let $\ell > 0$ and module-lattices $\mathcal{L}, \mathcal{M} \subset K^\ell$.

The **module index** of $\mathcal{M}$ in $\mathcal{L}$ is an ideal denoted by

$$[\mathcal{L} : \mathcal{M}]$$

Easily computed from bases of $\mathcal{L}$ and $\mathcal{M}$. Behaves as the "covolume of $\mathcal{M}$ in $\mathcal{L}$".

# Focus on decision module-LIP (6/10)

Shimura (1964) studied how $\mathrm{gen}(\mathcal{L})$ splits into multiple special genera.

Let $\ell > 0$ and module-lattices $\mathcal{L}, \mathcal{M} \subset K^\ell$.

The **module index** of $\mathcal{M}$ in $\mathcal{L}$ is an ideal denoted by

$$[\mathcal{L} : \mathcal{M}]$$

Easily computed from bases of $\mathcal{L}$ and $\mathcal{M}$. Behaves as the "covolume of $\mathcal{M}$ in $\mathcal{L}$".

**Example:** If $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ and $\mathcal{M} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbf{2}\mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$, then

$$[\mathcal{L} : \mathcal{M}] = 2\mathbb{Z}_K.$$

# Focus on decision module-LIP (6/10)

Shimura (1964) studied how gen($\mathcal{L}$) splits into multiple special genera.

Let $\ell > 0$ and module-lattices $\mathcal{L}, \mathcal{M} \subset K^\ell$.

The **module index** of $\mathcal{M}$ in $\mathcal{L}$ is an ideal denoted by

$$[\mathcal{L} : \mathcal{M}]$$

Easily computed from bases of $\mathcal{L}$ and $\mathcal{M}$. Behaves as the "covolume of $\mathcal{M}$ in $\mathcal{L}$".

**Example:** If $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ and $\mathcal{M} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbf{2}\mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$, then

$$[\mathcal{L} : \mathcal{M}] = 2\mathbb{Z}_K.$$

**Remark:** $[\mathcal{L} : \mathcal{M}]$ is well defined even if there is no inclusion.

# Focus on decision module-LIP (7/10)

For simplicity, fix $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$ as in HAWK.

# Focus on decision module-LIP (7/10)

For simplicity, fix $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$ as in HAWK. Define:

$$J = \{\mathfrak{a}\text{ideal} \mid \mathfrak{a} \cdot \overline{\mathfrak{a}} = \mathbb{Z}_K\},$$
$$J_0 = \{g \cdot \mathbb{Z}_K \mid g \in K^\times \text{ and } g\overline{g} = 1\}.$$

$J_0$ is a subgroup of $J$ (principal ideals).

# Focus on decision module-LIP (7/10)

For simplicity, fix $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$ as in HAWK. Define:

$$J = \{\mathfrak{a}\text{ideal} \mid \mathfrak{a} \cdot \overline{\mathfrak{a}} = \mathbb{Z}_K\},$$
$$J_0 = \{g \cdot \mathbb{Z}_K \mid g \in K^\times \text{ and } g\overline{g} = 1\}.$$

$J_0$ is a subgroup of $J$ (principal ideals). Shimura's theorem implies:

$$\begin{aligned} \text{gen}(\mathcal{L}) &\longrightarrow J \\ \mathcal{M} &\longmapsto [\mathcal{L} : \mathcal{M}] \end{aligned}$$

is well-defined and induces a **bijection** between the special genera in $\text{gen}(\mathcal{L})$ and $J/J_0$.

# Focus on decision module-LIP (7/10)

For simplicity, fix $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ as in HAWK. Define:

$$J = \{\mathfrak{a}\text{ideal} \mid \mathfrak{a} \cdot \overline{\mathfrak{a}} = \mathbb{Z}_K\},$$
$$J_0 = \{g \cdot \mathbb{Z}_K \mid g \in K^\times \text{ and } g\overline{g} = 1\}.$$

$J_0$ is a subgroup of $J$ (principal ideals). Shimura's theorem implies:

$$\begin{aligned} \text{gen}(\mathcal{L}) &\longrightarrow J \\ \mathcal{M} &\longmapsto [\mathcal{L} : \mathcal{M}] \end{aligned}$$

is well-defined and induces a **bijection** between the special genera in $\text{gen}(\mathcal{L})$ and $J/J_0$.

**Consequence:** Assume $\mathcal{M} \in \text{gen}(\mathcal{L})$. Then $\mathcal{M} \in \text{sgen}(\mathcal{L})$ **iff**

$$[\mathcal{L} : \mathcal{M}] \text{ has the form } g \cdot \mathbb{Z}_K \text{ with } g\overline{g} = 1.$$

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

Testing if an ideal $\mathfrak{a}$ is principal is a **hard problem** for (classical) computers.

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

Testing if an ideal $\mathfrak{a}$ is principal is a **hard problem** for (classical) computers. Luckily:

**Gentry-Szydlo's algorithm**

There is a (classical) **poly-time** algorithm that given an ideal $\mathfrak{a}$ and $q \in K$ s.t. $\mathfrak{a} \cdot \overline{\mathfrak{a}} = q \cdot \mathbb{Z}_K$, computes $g \in K$ such that $\mathfrak{a} = g \cdot \mathbb{Z}_K$ and $g\overline{g} = q$ (if it exists).

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

Testing if an ideal $\mathfrak{a}$ is principal is a **hard problem** for (classical) computers. Luckily:

**Gentry-Szydlo's algorithm**

There is a (classical) **poly-time** algorithm that given an ideal $\mathfrak{a}$ and $q \in K$ s.t. $\mathfrak{a} \cdot \overline{\mathfrak{a}} = q \cdot \mathbb{Z}_K$, computes $g \in K$ such that $\mathfrak{a} = g \cdot \mathbb{Z}_K$ and $g\overline{g} = q$ (if it exists).

- **Fundamental tool** for the cryptanalysis of (rank-2) search module-LIP.

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

Testing if an ideal $\mathfrak{a}$ is principal is a **hard problem** for (classical) computers. Luckily:

**Gentry-Szydlo's algorithm**

There is a (classical) **poly-time** algorithm that given an ideal $\mathfrak{a}$ and $q \in K$ s.t. $\mathfrak{a} \cdot \overline{\mathfrak{a}} = q \cdot \mathbb{Z}_K$, computes $g \in K$ such that $\mathfrak{a} = g \cdot \mathbb{Z}_K$ and $g\overline{g} = q$ (if it exists).

- **Fundamental tool** for the cryptanalysis of (rank-2) search module-LIP.
- A generalization to **quaternionic** ideals would break HAWK.

# Focus on decision module-LIP (8/10)

**Remark:** Testing $\mathfrak{a} = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1$ is basically module-LIP between $\mathbb{Z}_K$ and $\mathfrak{a}$.

Testing if an ideal $\mathfrak{a}$ is principal is a **hard problem** for (classical) computers. Luckily:

**Gentry-Szydlo's algorithm**

There is a (classical) **poly-time** algorithm that given an ideal $\mathfrak{a}$ and $q \in K$ s.t. $\mathfrak{a} \cdot \overline{\mathfrak{a}} = q \cdot \mathbb{Z}_K$, computes $g \in K$ such that $\mathfrak{a} = g \cdot \mathbb{Z}_K$ and $g\overline{g} = q$ (if it exists).

- **Fundamental tool** for the cryptanalysis of (rank-2) search module-LIP.
- A generalization to **quaternionic** ideals would break HAWK.

$\longrightarrow$ We obtain a (classical) poly-time algo to test if $\mathcal{M} \in \mathrm{sgen}(\mathcal{L})$.

# Focus on decision module-LIP (9/10)

The algorithm extends to arbitrary $\mathcal{L} \subset K^\ell$ and $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$.

# Focus on decision module-LIP (9/10)

The algorithm extends to arbitrary $\mathcal{L} \subset K^\ell$ and $\mathcal{M} \in \text{gen}(\mathcal{L})$. In general:

- Need to check if $[\mathcal{L} : \mathcal{M}] = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1 \longrightarrow$ OK! (Gentry-Szydlo).

# Focus on decision module-LIP (9/10)

The algorithm extends to arbitrary $\mathcal{L} \subset K^\ell$ and $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$. In general:

- Need to check if $[\mathcal{L} : \mathcal{M}] = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1 \longrightarrow$ OK! (Gentry-Szydlo).
- Need to compute $\det(\mathbf{\Theta}_\mathfrak{p})$ with good enough precision, for a small number of prime ideals (which depends on $\mathcal{L}$).

# Focus on decision module-LIP (9/10)

The algorithm extends to arbitrary $\mathcal{L} \subset K^\ell$ and $\mathcal{M} \in \text{gen}(\mathcal{L})$. In general:

- Need to check if $[\mathcal{L} : \mathcal{M}] = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1 \longrightarrow$ OK! (Gentry-Szydlo).
- Need to compute $\det(\mathbf{\Theta}_\mathfrak{p})$ with good enough precision, for a small number of prime ideals (which depends on $\mathcal{L}$).

Second item is hard to analyze. No complexity result in the general case.

# Focus on decision module-LIP (9/10)

The algorithm extends to arbitrary $\mathcal{L} \subset K^\ell$ and $\mathcal{M} \in \mathrm{gen}(\mathcal{L})$. In general:

- Need to check if $[\mathcal{L} : \mathcal{M}] = g \cdot \mathbb{Z}_K$ with $g\overline{g} = 1 \longrightarrow$ OK! (Gentry-Szydlo).
- Need to compute $\det(\mathbf{\Theta}_\mathfrak{p})$ with good enough precision, for a small number of prime ideals (which depends on $\mathcal{L}$).

Second item is hard to analyze. No complexity result in the general case.

**Remark:** For several module lattices (including HAWK), **no** local conditions to check. $\longrightarrow$ provable poly-time complexity!

# Focus on decision module-LIP (10/10)

Fix $K = \mathbb{Q}[X]/(X^d + 1)$ and $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$ as in HAWK.

We have an effective distinguisher for $\mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

# Focus on decision module-LIP (10/10)

Fix $K = \mathbb{Q}[X]/(X^d + 1)$ and $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ as in HAWK.

We have an effective distinguisher for $\mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

**Question 1:** Recall that HAWK reduces to several decision module-LIP instances. Are these module-lattices in the same special genus?

# Focus on decision module-LIP (10/10)

Fix $K = \mathbb{Q}[X]/(X^d + 1)$ and $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ as in HAWK.

We have an effective distinguisher for $\mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

**Question 1:** Recall that HAWK reduces to several decision module-LIP instances. Are these module-lattices in the same special genus? **YES** $\longrightarrow$ No impact on HAWK...

# Focus on decision module-LIP (10/10)

Fix $K = \mathbb{Q}[X]/(X^d + 1)$ and $\mathcal{L} = \mathbb{Z}_K\binom{1}{0} + \mathbb{Z}_K\binom{0}{1}$ as in HAWK.

We have an effective distinguisher for $\mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

**Question 1:** Recall that HAWK reduces to several decision module-LIP instances.
Are these module-lattices in the same special genus? **YES** $\longrightarrow$ No impact on HAWK...

**Question 2:** Is $\mathrm{sgen}(\mathcal{L})$ really finer than $\mathrm{gen}(\mathcal{L})$?
How many iso. classes in $\mathrm{gen}(\mathcal{L})$? in $\mathrm{sgen}(\mathcal{L})$?

# Focus on decision module-LIP (10/10)

Fix $K = \mathbb{Q}[X]/(X^d + 1)$ and $\mathcal{L} = \mathbb{Z}_K\left(\begin{smallmatrix}1\\0\end{smallmatrix}\right) + \mathbb{Z}_K\left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$ as in HAWK.

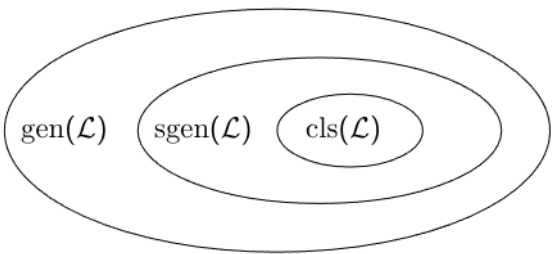
We have an effective distinguisher for $\mathrm{sgen}(\mathcal{L}) \subseteq \mathrm{gen}(\mathcal{L})$.

**Question 1:** Recall that HAWK reduces to several decision module-LIP instances.
Are these module-lattices in the same special genus? **YES** $\longrightarrow$ No impact on HAWK...

**Question 2:** Is $\mathrm{sgen}(\mathcal{L})$ really finer than $\mathrm{gen}(\mathcal{L})$?
How many iso. classes in $\mathrm{gen}(\mathcal{L})$? in $\mathrm{sgen}(\mathcal{L})$? For $d = 512$, we (heuristically) have:

$\#\{\text{iso. classes in } \mathrm{gen}(\mathcal{L})\} \approx 2^{1000}$

$\#\{\text{iso. classes in } \mathrm{sgen}(\mathcal{L})\} \approx 2^{800}$



# Takeaway and perspectives

- Search module-LIP is broken for almost all choices of $K$, but HAWK still flies!

# Takeaway and perspectives

- Search module-LIP is broken for almost all choices of $K$, but HAWK still flies!
- Search module-LIP and decision module-LIP are connected problems.
  This suggests a new direction to target HAWK.

# Takeaway and perspectives

- Search module-LIP is broken for almost all choices of $K$, but HAWK still flies!
- Search module-LIP and decision module-LIP are connected problems. This suggests a new direction to target HAWK.
- The special genus is a finer invariant than the genus. Improves the state-of-the-art on decision module-LIP. To make a hard instance of decision module-LIP, lattices must be chosen inside the same special genus.

# Takeaway and perspectives

- Search module-LIP is broken for almost all choices of $K$, but HAWK still flies!
- Search module-LIP and decision module-LIP are connected problems. This suggests a new direction to target HAWK.
- The special genus is a finer invariant than the genus. Improves the state-of-the-art on decision module-LIP. To make a hard instance of decision module-LIP, lattices must be chosen inside the same special genus.
- Despite the exponential gain, there is still an exponential number of classes in $\mathrm{sgen}(\mathcal{L})$. No practical impact on the security of HAWK.

# Takeaway and perspectives

- Search module-LIP is broken for almost all choices of $K$, but HAWK still flies!
- Search module-LIP and decision module-LIP are connected problems. This suggests a new direction to target HAWK.
- The special genus is a finer invariant than the genus. Improves the state-of-the-art on decision module-LIP. To make a hard instance of decision module-LIP, lattices must be chosen inside the same special genus.
- Despite the exponential gain, there is still an exponential number of classes in $\mathrm{sgen}(\mathcal{L})$. No practical impact on the security of HAWK.
- **Open question:** Other computable invariants?

# Bonus: Gentry-Szydlo's algorithm (1/2)

Let $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^r$. Fix $g \in \mathbb{Z}_K$.

# Bonus: Gentry-Szydlo's algorithm (1/2)

Let $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^r$. Fix $g \in \mathbb{Z}_K$.

`GentrySzydlo`

**Input:** A basis of $g \cdot \mathbb{Z}_K$ and $g\overline{g}$.
**Output:** $g$ (up to a root of unity of $K$).

# Bonus: Gentry-Szydlo's algorithm (1/2)

Let $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^r$. Fix $g \in \mathbb{Z}_K$.

`GentrySzydlo`

**Input:** A basis of $g \cdot \mathbb{Z}_K$ and $g\overline{g}$.
**Output:** $g$ (up to a root of unity of $K$).

1. Find a prime number $p = 1 \pmod{2d}$, so that $\mathbb{Z}_K/(p) \simeq (\mathbb{F}_p)^d$.

2. Compute a LLL-reduced basis of $(g \cdot \mathbb{Z}_K)^{p-1} = g^{p-1} \cdot \mathbb{Z}_K$.
   At each step, divide the basis by $g\overline{g}$ to avoid coefficient blow-up.

# Bonus: Gentry-Szydlo's algorithm (1/2)

Let $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^r$. Fix $g \in \mathbb{Z}_K$.

`GentrySzydlo`

**Input:** A basis of $g \cdot \mathbb{Z}_K$ and $g\overline{g}$.
**Output:** $g$ (up to a root of unity of $K$).

1. Find a prime number $p = 1 \pmod{2d}$, so that $\mathbb{Z}_K/(p) \simeq (\mathbb{F}_p)^d$.
2. Compute a LLL-reduced basis of $(g \cdot \mathbb{Z}_K)^{p-1} = g^{p-1} \cdot \mathbb{Z}_K$.
   At each step, divide the basis by $g\overline{g}$ to avoid coefficient blow-up.
3. The first basis vector is $g^{p-1} \cdot v$ with $v$ **short**. Reduce it modulo $p$.

# Bonus: Gentry-Szydlo's algorithm (2/2)

4. By Fermat's theorem, $g^{p-1} = 1 \pmod{p}$, so we have $v \pmod{p}$ .

# Bonus: Gentry-Szydlo's algorithm (2/2)

4. By Fermat's theorem, $g^{p-1} = 1 \pmod{p}$, so we have $v \pmod{p}$ .
5. If $p$ is **big enough**, obtain $v$ exactly and deduce $g^{p-1}$.

# Bonus: Gentry-Szydlo's algorithm (2/2)

4. By Fermat's theorem, $g^{p-1} = 1 \pmod{p}$, so we have $v \pmod{p}$ .
5. If $p$ is **big enough**, obtain $v$ exactly and deduce $g^{p-1}$.
6. With some trick, reduce the exponent and obtain $g^d$.

# Bonus: Gentry-Szydlo's algorithm (2/2)

4. By Fermat's theorem, $g^{p-1} = 1 \pmod{p}$, so we have $v \pmod{p}$ .
5. If $p$ is **big enough**, obtain $v$ exactly and deduce $g^{p-1}$.
6. With some trick, reduce the exponent and obtain $g^d$.
7. Compute a $d$-th root and get $g$, up to a root of unity.

# Bonus: Gentry-Szydlo's algorithm (2/2)

4. By Fermat's theorem, $g^{p-1} = 1 \pmod{p}$, so we have $v \pmod{p}$ .
5. If $p$ is **big enough**, obtain $v$ exactly and deduce $g^{p-1}$.
6. With some trick, reduce the exponent and obtain $g^d$.
7. Compute a $d$-th root and get $g$, up to a root of unity.

$\rightarrow$ We have an implementation in `SageMath`!