On the Lattice Isomorphism Problem, Cryptography and the Signature Scheme HAWK

Léo Ducas, Eamonn Postlethwaite, Ludo Pulles (CWI, Cryptology Group), Wessel van Woerden (Université de Bordeaux, IMB).

Centrum Wiskunde & Informatica

Motivation

- LWE, SIS, NTRU lattices: versatile, but poor decoding.

Motivation

- LWE, SIS, NTRU lattices: versatile, but poor decoding.
- Many wonderful lattices exist with great geometric properties.

Motivation

- LWE, SIS, NTRU lattices: versatile, but poor decoding.
- Many wonderful lattices exist with great geometric properties.
- Can we use these in cryptography?

Motivation

- LWE, SIS, NTRU lattices: versatile, but poor decoding.
- Many wonderful lattices exist with great geometric properties.
- Can we use these in cryptography?

Contributions

- General identification, encryption and signature scheme based on the Lattice Isomorphism Problem.

Motivation

- LWE, SIS, NTRU lattices: versatile, but poor decoding.
- Many wonderful lattices exist with great geometric properties.
- Can we use these in cryptography?

Contributions

- General identification, encryption and signature scheme based on the Lattice Isomorphism Problem.
- Better lattice $\implies$ better efficiency and security.

Motivation

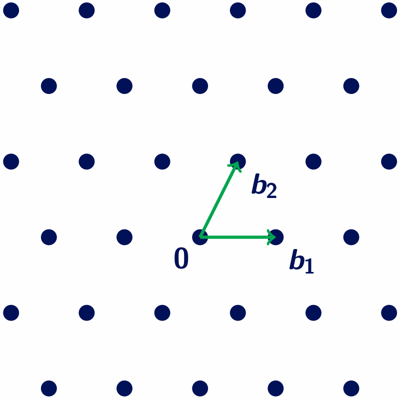
- LWE, SIS, NTRU lattices: versatile, but poor decoding.
- Many wonderful lattices exist with great geometric properties.
- Can we use these in cryptography?

Contributions

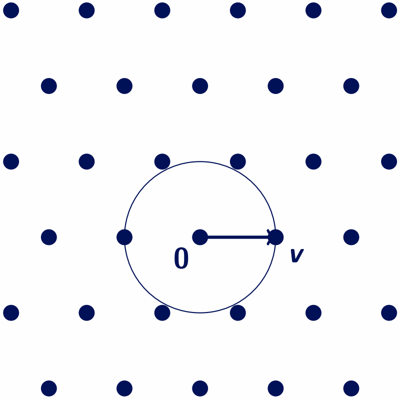
- General identification, encryption and signature scheme based on the Lattice Isomorphism Problem.
- Better lattice $\Longrightarrow$ better efficiency and security.
- HAWK: a simple and efficient signature scheme from $\mathbb{Z}^n$.

Lattices

Lattice $\mathcal{L}(\boldsymbol{B}) := \{\sum_i x_i \boldsymbol{b}_i : \boldsymbol{x} \in \mathbb{Z}^n\} \subset \mathbb{R}^n$



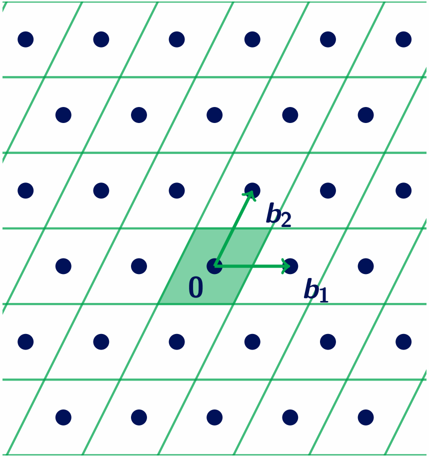
Lattice $\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n$



First minimum

$$\lambda_1(\mathcal{L}) := \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2$$

Lattice $\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n$



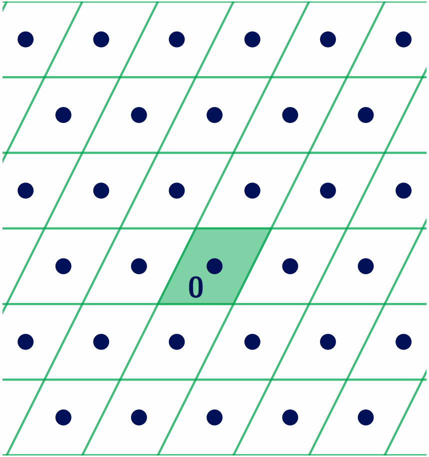
<u>First minimum</u>

$$\lambda_1(\mathcal{L}) := \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2$$

<u>Determinant</u>

$$\det(\mathcal{L}) := \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) = |\det(B)|$$

Lattice $\mathcal{L}(B) := \{\sum_i x_i b_i : x \in \mathbb{Z}^n\} \subset \mathbb{R}^n$



First minimum

$$\lambda_1(\mathcal{L}) := \min_{x \in \mathcal{L} \setminus \{0\}} \|x\|_2$$

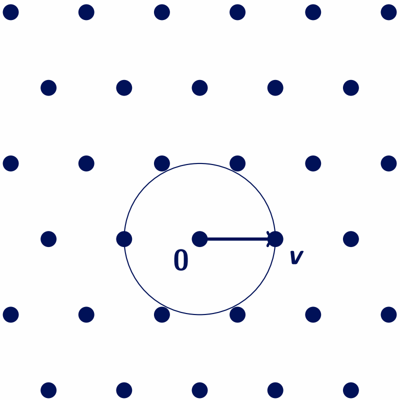
Determinant

$$\det(\mathcal{L}) := \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) = |\det(B)|$$

Minkowski's Theorem

$$\lambda_1(\mathcal{L}) \leq \underbrace{2\frac{\det(\mathcal{L})^{1/n}}{\mathrm{vol}(\mathcal{B}^n)^{1/n}}}_{\mathrm{Mk}(\mathcal{L})} \leq \sqrt{n}\det(\mathcal{L})^{1/n}$$

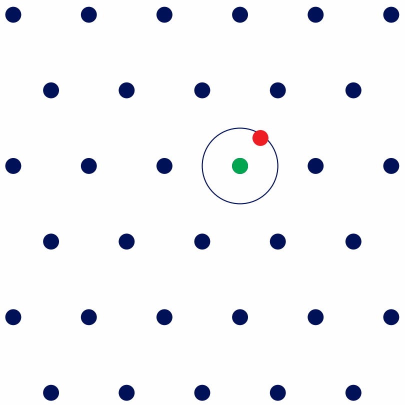
Lattice $\mathcal{L} \subset \mathbb{R}^n$



SVP

Find a *shortest* nonzero vector $\boldsymbol{v} \in \mathcal{L}$ of length $\lambda_1(\mathcal{L}) \leq \mathsf{Mk}(\mathcal{L})$.

Lattice $\mathcal{L} \subset \mathbb{R}^n$



SVP

Find a *shortest* nonzero vector $v \in \mathcal{L}$ of length $\lambda_1(\mathcal{L}) \leq \mathsf{Mk}(\mathcal{L})$.

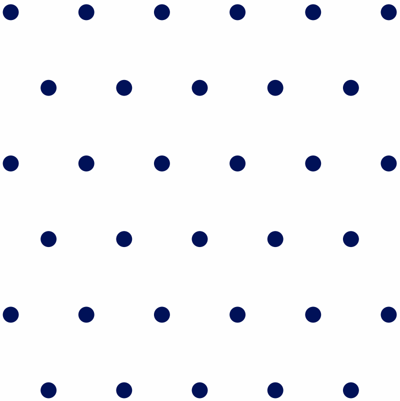
BDD

Given a target $t = v + e \in \mathbb{R}^n$ with

$$v \in \mathcal{L} \text{ and } \|e\| < \rho \leq \frac{1}{2}\lambda_1(\mathcal{L}) \leq \frac{1}{2}\mathsf{Mk}(\mathcal{L}),$$

recover $v \in \mathcal{L}$.

Lattice $\mathcal{L} \subset \mathbb{R}^n$



SVP

Find a *shortest* nonzero vector $\boldsymbol{v} \in \mathcal{L}$ of length $\underbrace{\lambda_1(\mathcal{L}) \leq \mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L})}$.

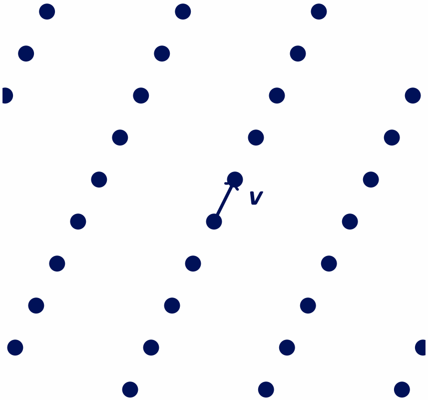
BDD

Given a target $\boldsymbol{t} = \boldsymbol{v} + \boldsymbol{e} \in \mathbb{R}^n$ with $\boldsymbol{v} \in \mathcal{L}$ and $\|\boldsymbol{e}\| < \underbrace{\rho \leq \frac{1}{2}\lambda_1(\mathcal{L}) \leq \frac{1}{2}\mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L},\rho)}$, recover $\boldsymbol{v} \in \mathcal{L}$.

Hardness depends on the *gaps*!

Hard Problems

Lattice $\mathcal{L} \subset \mathbb{R}^n$



SVP

Find a *shortest* nonzero vector $\boldsymbol{v} \in \mathcal{L}$ of length $\underbrace{\lambda_1(\mathcal{L}) \leq \mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L})}$.

BDD

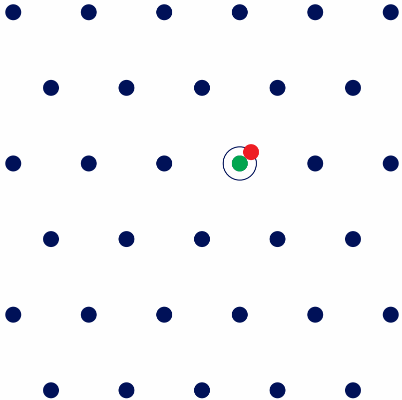
Given a target $\boldsymbol{t} = \boldsymbol{v} + \boldsymbol{e} \in \mathbb{R}^n$ with $\boldsymbol{v} \in \mathcal{L}$ and $\|\boldsymbol{e}\| < \underbrace{\rho \leq \frac{1}{2}\lambda_1(\mathcal{L}) \leq \frac{1}{2}\mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L},\rho)}$,

recover $\boldsymbol{v} \in \mathcal{L}$.

Hardness depends on the *gaps*!

Hard Problems

Lattice $\mathcal{L} \subset \mathbb{R}^n$



<u>SVP</u>

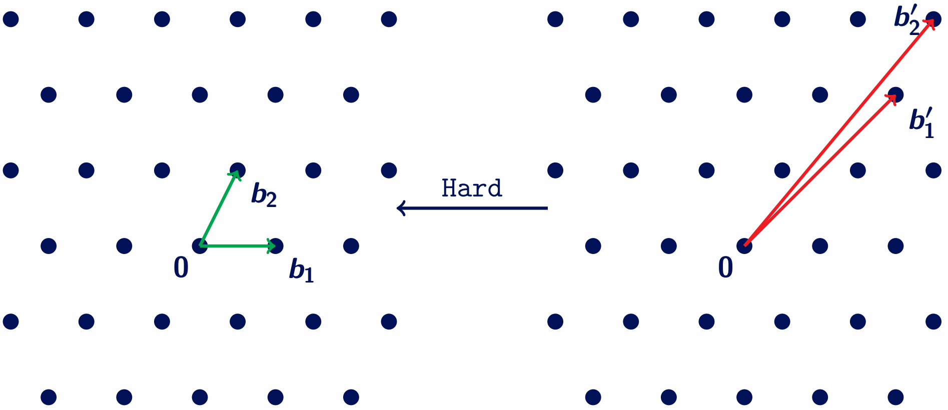
Find a *shortest* <u>nonzero</u> vector $\boldsymbol{v} \in \mathcal{L}$ of length $\underbrace{\lambda_1(\mathcal{L}) \leq \mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L})}$.

<u>BDD</u>

Given a target $\boldsymbol{t} = \boldsymbol{v} + \boldsymbol{e} \in \mathbb{R}^n$ with $\boldsymbol{v} \in \mathcal{L}$ and $\|\boldsymbol{e}\| < \underbrace{\rho \leq \frac{1}{2}\lambda_1(\mathcal{L}) \leq \frac{1}{2}\mathsf{Mk}(\mathcal{L})}_{\mathsf{gap}(\mathcal{L},\rho)}$, recover $\boldsymbol{v} \in \mathcal{L}$.

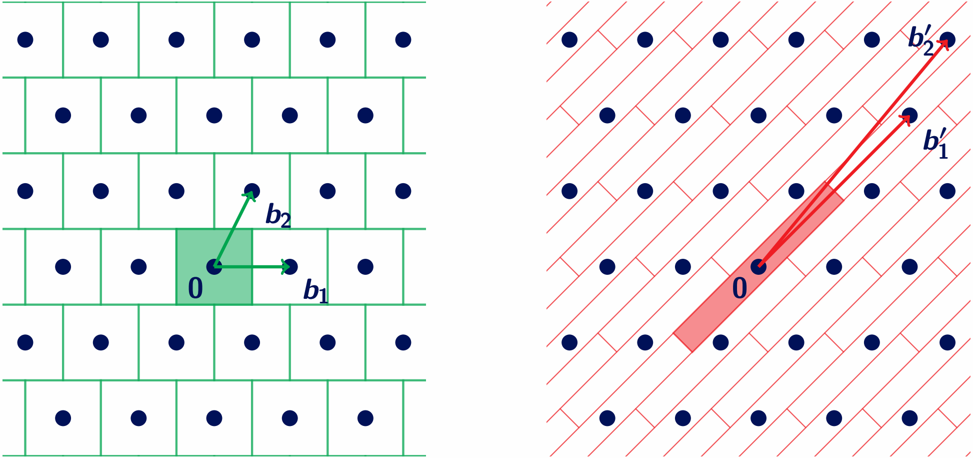
Hardness depends on the *gaps*!

Good basis (Secret key)   Bad basis (Public key)



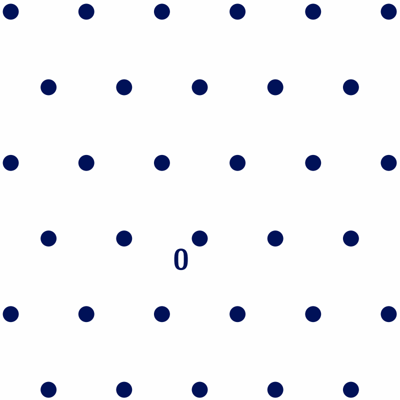
$\boldsymbol{B}$ $\boldsymbol{B}' = \boldsymbol{B} \cdot \boldsymbol{U}, \;\; \boldsymbol{U} \in \mathrm{GL}_d(\mathbb{Z})$

Good basis (Secret key) Bad basis (Public key)

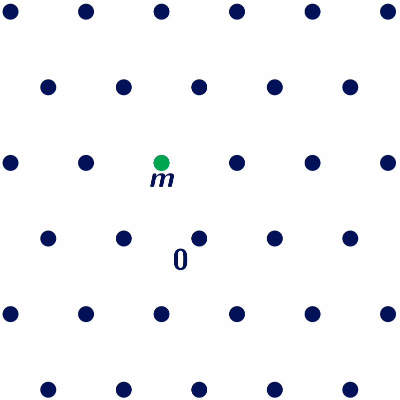


Babai's nearest plane algorithm

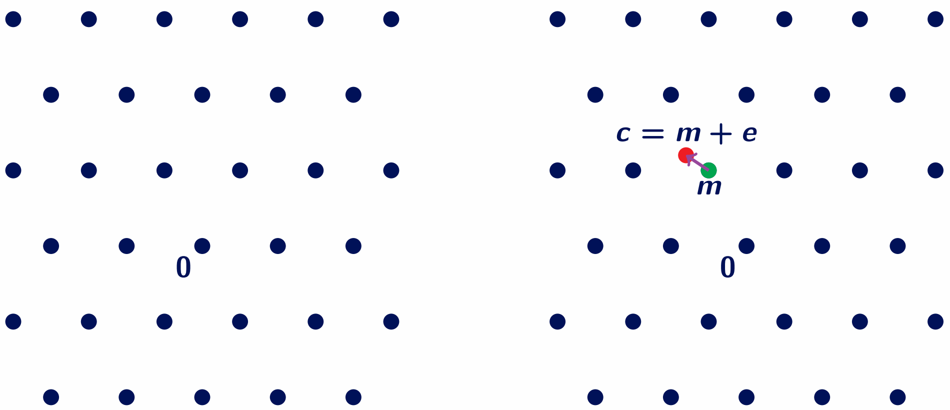
Good basis (Secret key)



Bad basis (Public key)



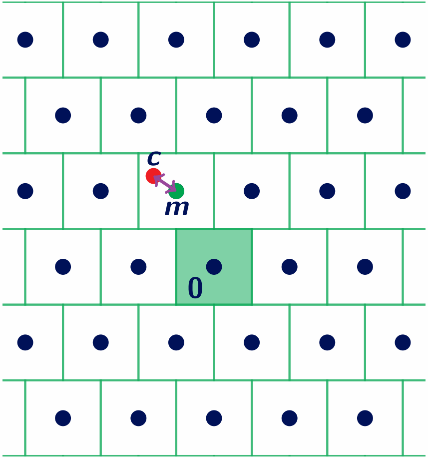
Good basis (Secret key) Bad basis (Public key)



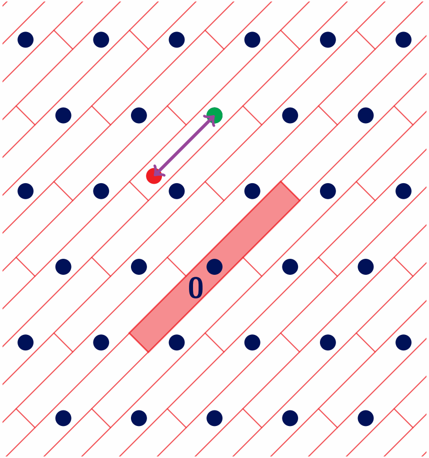
Encrypt by adding a small error

Encryption, legacy approach

Good basis (Secret key)



Bad basis (Public key)



Decrypt using the good basis

Large gap

Current lattice based crypto relies on hardness of decoding with

$$\mathrm{gap}(\mathcal{L}, \rho) \geq \Omega(\sqrt{n}).$$

Broken by SVP in dimension $\beta \leq n/2 + o(n)$, e.g. $n = 1024 \implies \beta \approx 450$.

Large gap

Current lattice based crypto relies on hardness of decoding with

$$\mathrm{gap}(\mathcal{L}, \rho) \geq \Omega(\sqrt{n}).$$

Broken by SVP in dimension $\beta \leq n/2 + o(n)$, e.g. $n = 1024 \implies \beta \approx 450$.

An example: Prime Lattice [CR88]

Let $p_1, \ldots, p_n$ be distinct small primes not dividing m, we define:

$$\mathcal{L}_{\mathtt{prime}} := \{x = (x_1, \ldots, x_n) \in \mathbb{Z}^n : \prod_i p_i^{x_i} = 1 \bmod m\}.$$

Large gap

Current lattice based crypto relies on hardness of decoding with

$$\mathrm{gap}(\mathcal{L}, \rho) \geq \Omega(\sqrt{n}).$$

Broken by SVP in dimension $\beta \leq n/2 + o(n)$, e.g. $n = 1024 \implies \beta \approx 450$.

An example: Prime Lattice [CR88]

Let $p_1, \ldots, p_n$ be distinct small primes not dividing m, we define:

$$\mathcal{L}_{\mathrm{prime}} := \{x = (x_1, \ldots, x_n) \in \mathbb{Z}^n : \prod_i p_i^{x_i} = 1 \bmod m\}.$$

- Efficiently decode up to large radius ρ by trial division.

Large gap

Current lattice based crypto relies on hardness of decoding with

$$\mathsf{gap}(\mathcal{L}, \rho) \geq \Omega(\sqrt{n}).$$

Broken by SVP in dimension $\beta \leq n/2 + o(n)$, e.g. $n = 1024 \implies \beta \approx 450$.

An example: Prime Lattice [CR88]

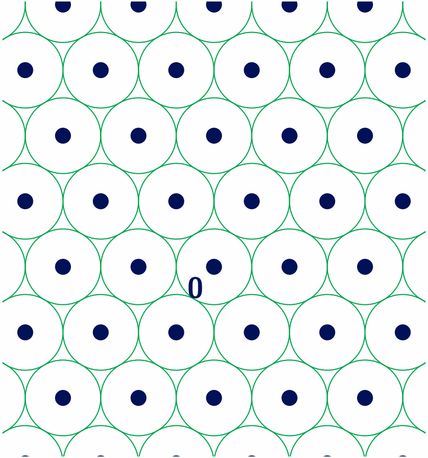
Let $p_1, \ldots, p_n$ be distinct small primes not dividing m, we define:

$$\mathcal{L}_{\texttt{prime}} := \{x = (x_1, \ldots, x_n) \in \mathbb{Z}^n : \prod_i p_i^{x_i} = 1 \bmod m\}.$$

- Efficiently decode up to large radius ρ by trial division.
- With the right parameters $\mathsf{gap}(\mathcal{L}_{\texttt{prime}}, \rho) = \Theta(\log(n))$ [DP19].

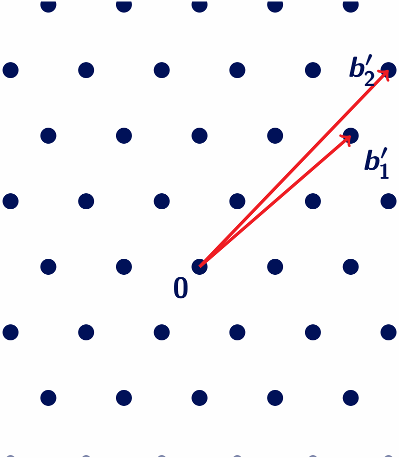
How to hide the remarkable lattice?

Good lattice (~~Secret key~~)



$\mathcal{L}$

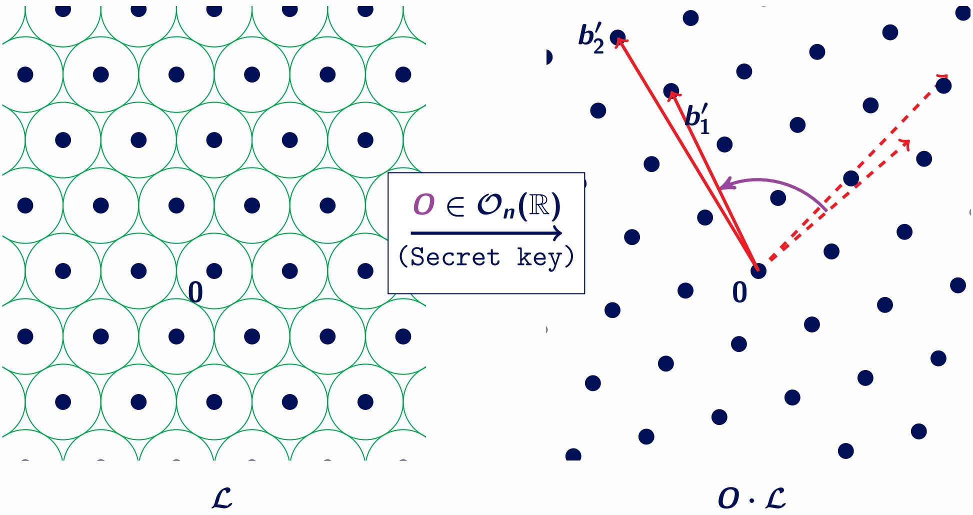
Bad basis (Public key)



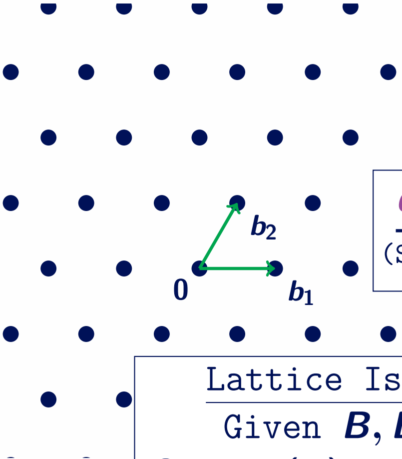
$\mathcal{L}$

Good lattice (~~Secret key~~)

Bad basis (Public key)

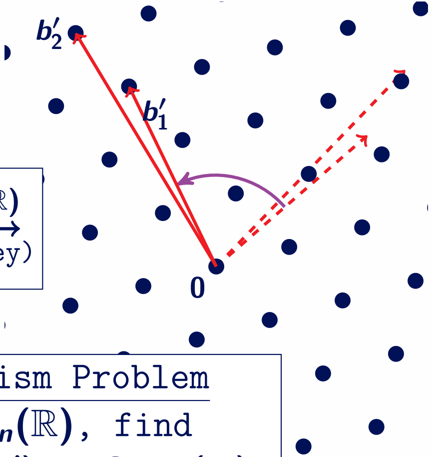


Good lattice (~~Secret key~~)



$\xrightarrow[\text{(Secret key)}]{O \in \mathcal{O}_n(\mathbb{R})}$

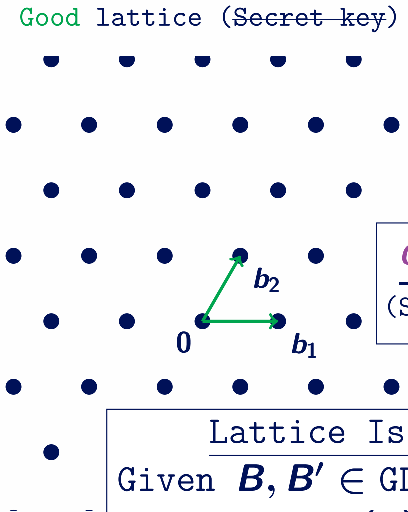
Bad basis (Public key)

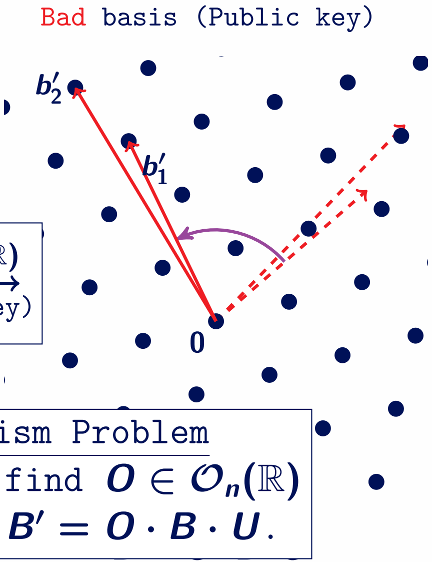


Lattice Isomorphism Problem

Given $B, B' \in \mathrm{GL}_n(\mathbb{R})$, find $O \in \mathcal{O}_n(\mathbb{R})$ s.t. $\mathcal{L}(B') = O \cdot \mathcal{L}(B)$.

How to hide the remarkable lattice?





Lattice Isomorphism Problem

Given $B, B' \in \mathrm{GL}_n(\mathbb{R})$, find $O \in \mathcal{O}_n(\mathbb{R})$ and $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $B' = O \cdot B \cdot U$.

Lattice Isomorphism Problem

LIP

Given $\mathbf{B}, \mathbf{B}' \in \mathrm{GL}_n(\mathbb{R})$ of isomorphic lattices, find $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ and $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $\mathbf{B}' = \mathbf{O} \cdot \mathbf{B} \cdot \mathbf{U}$.

Lattice Isomorphism Problem

LIP

Given $\mathbf{B}, \mathbf{B}' \in \mathrm{GL}_n(\mathbb{R})$ of isomorphic lattices, find $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ and $\mathbf{U} \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $\mathbf{B}' = \mathbf{O} \cdot \mathbf{B} \cdot \mathbf{U}$.

- The lattice analogue of 'vintage' McEliece $\mathbf{G}' = \mathbf{P} \cdot \mathbf{G} \cdot \mathbf{S}$,
- and Oil and Vinegar $\mathcal{P} = \mathcal{Q} \circ \mathcal{S}$.

LIP

Given $B, B' \in \mathrm{GL}_n(\mathbb{R})$ of isomorphic lattices, find $O \in \mathcal{O}_n(\mathbb{R})$ and $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $B' = O \cdot B \cdot U$.

- The lattice analogue of 'vintage' McEliece $G' = P \cdot G \cdot S$,
- and Oil and Vinegar $\mathcal{P} = \mathcal{Q} \circ \mathcal{S}$.
- Best known attacks require to solve SVP.

Algorithms

- $\mathrm{Min}(\mathcal{L}(B')) = O \cdot \mathrm{Min}(\mathcal{L}(B))$.
- Best practical algorithm: backtrack search all isometries between the sets of short vectors.
- Best proven algorithm uses short primal and dual vectors ($n^{O(n)}$ time and space).

$$\boldsymbol{B}' = \boldsymbol{O} \cdot \boldsymbol{B} \cdot \boldsymbol{U}.$$

$$\boldsymbol{B}' = \boldsymbol{O} \cdot \boldsymbol{B} \cdot \boldsymbol{U}.$$

Sidestep real values!

$\boldsymbol{O} \in \mathcal{O}_n(\mathbb{R})$

Sample $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{n}}(\mathbb{Z})$ s.t.
$\boldsymbol{B}'$ is independent of $\boldsymbol{B}$.

$$\boldsymbol{B}' = \boldsymbol{O} \cdot \boldsymbol{B} \cdot \boldsymbol{U}.$$

Sidestep real values!
$\boldsymbol{O} \in \mathcal{O}_{\boldsymbol{n}}(\mathbb{R})$

Orthonormal $\boldsymbol{O} \in \mathcal{O}_n(\mathbb{R})$

Orthonormal $O \in \mathcal{O}_n(\mathbb{R})$

$$(B')^t B' = U^t B^t O^t O B U = U^t B^t B U.$$

Orthonormal $O \in \mathcal{O}_n(\mathbb{R})$

$$(B')^t B' = U^t B^t O^t O B U = U^t B^t B U.$$

$$Q := B^t B \in \mathcal{S}_n^{>0}$$

Lattices $\implies$ Quadratic Forms

$$(\mathcal{L} \subset \mathbb{R}^n, \langle x, y \rangle) \implies (\mathbb{Z}^n, \langle x, y \rangle_Q := x^t Q y)$$

Keep the geometry, forget the embedding.

Orthonormal $O \in \mathcal{O}_n(\mathbb{R})$

$$(B')^t B' = U^t B^t O^t O B U = U^t B^t B U .$$

$$Q := B^t B \in \mathcal{S}_n^{>0}$$

Lattices $\implies$ Quadratic Forms

$$(\mathcal{L} \subset \mathbb{R}^n, \langle x, y \rangle) \implies (\mathbb{Z}^n, \langle x, y \rangle_Q := x^t Q y)$$

Keep the geometry, forget the embedding.

LIP restated:

Find $U \in \mathrm{GL}_n(\mathbb{Z})$ s.t. $Q' = U^t Q U$.

An average-case distribution

Unimodular $\boldsymbol{U} \in \mathrm{GL}_{\boldsymbol{n}}(\mathbb{Z})$

An average-case distribution

Unimodular $U \in \mathrm{GL}_n(\mathbb{Z})$

Equivalence class $[Q] := \{U^t Q U : U \in \mathrm{GL}_n(\mathbb{Z})\}$.

Def: Distribution $\mathcal{D}_\sigma([Q])$ over $[Q]$,

An average-case distribution

Unimodular $U \in \mathrm{GL}_n(\mathbb{Z})$

Equivalence class $[Q] := \{U^t Q U : U \in \mathrm{GL}_n(\mathbb{Z})\}$.

Def: Distribution $\mathcal{D}_\sigma([Q])$ over $[Q]$,

+ Efficient sampler $(Q', U) \leftarrow \mathrm{Sample}_\sigma(Q)$ s.t. $Q' \sim \mathcal{D}_\sigma([Q])$ and $Q' = U^t Q U$.

Q' only depends on the class $[Q]$ and not on Q itself.

Unimodular $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$

Equivalence class $[\boldsymbol{Q}] := \{\boldsymbol{U}^t \boldsymbol{Q} \boldsymbol{U} : \boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})\}$.

Def: Distribution $\mathcal{D}_\sigma([\boldsymbol{Q}])$ over $[\boldsymbol{Q}]$,

+ Efficient sampler $(\boldsymbol{Q}', \boldsymbol{U}) \leftarrow \mathrm{Sample}_\sigma(\boldsymbol{Q})$ s.t. $\boldsymbol{Q}' \sim \mathcal{D}_\sigma([\boldsymbol{Q}])$ and $\boldsymbol{Q}' = \boldsymbol{U}^t \boldsymbol{Q} \boldsymbol{U}$.

$\boldsymbol{Q}'$ *only depends on the class* $[\boldsymbol{Q}]$ *and not on* $\boldsymbol{Q}$ *itself.*

$\Longrightarrow$ average-case LIP, ZKPoK, identification scheme.

An average-case distribution

Unimodular $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$

Equivalence class $[\boldsymbol{Q}] := \{\boldsymbol{U}^t \boldsymbol{Q} \boldsymbol{U} : \boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})\}$.

Def: Distribution $\mathcal{D}_\sigma([\boldsymbol{Q}])$ over $[\boldsymbol{Q}]$,

$+$ Efficient sampler $(\boldsymbol{Q}', \boldsymbol{U}) \leftarrow \mathrm{Sample}_\sigma(\boldsymbol{Q})$
s.t. $\boldsymbol{Q}' \sim \mathcal{D}_\sigma([\boldsymbol{Q}])$ and $\boldsymbol{Q}' = \boldsymbol{U}^t \boldsymbol{Q} \boldsymbol{U}$.

$\boldsymbol{Q}'$ *only depends on the class* $[\boldsymbol{Q}]$ *and not on* $\boldsymbol{Q}$ *itself.*

$\implies$ average-case LIP, ZKPoK, identification scheme.

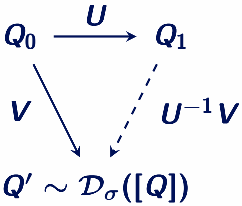
$\implies$ Worst-case to average-case reduction over $[\boldsymbol{Q}]$.

An average-case distribution

- ac-LIP$^{\boldsymbol{Q}}_{\sigma}$: given $\boldsymbol{Q}$ and $\boldsymbol{Q}' \leftarrow \mathcal{D}_\sigma([\boldsymbol{Q}])$, recover $\boldsymbol{U} \in \mathrm{GL}_n(\mathbb{Z})$.

An average-case distribution

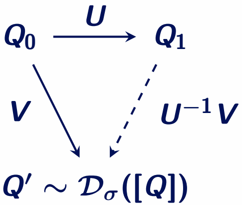
- ac-LIP$_\sigma^Q$: given Q and $Q' \leftarrow \mathcal{D}_\sigma([Q])$, recover $U \in \mathrm{GL}_n(\mathbb{Z})$.
- ZKPoK: Given public $Q_0, Q_1 \in [Q]$, prove knowledge of a U s.t. $Q_1 = U^t Q_0 U$, without revealing U.



$(Q', V) \leftarrow \mathrm{Sample}_\sigma(Q_0)$

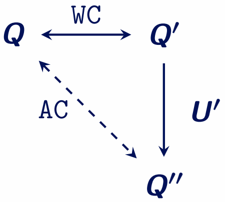
An average-case distribution

- ac-LIP$_\sigma^Q$: given Q and $Q' \leftarrow \mathcal{D}_\sigma([Q])$, recover $U \in GL_n(\mathbb{Z})$.
- ZKPoK: Given public $Q_0, Q_1 \in [Q]$, prove knowledge of a U s.t. $Q_1 = U^t Q_0 U$, without revealing U.



$$(Q', V) \leftarrow \mathrm{Sample}_\sigma(Q_0)$$

- Worst-case to average-case reduction:



$$(Q'', U') \leftarrow \mathrm{Sample}_\sigma(Q')$$

Decodable lattice $\mathcal{L} \implies$ Encryption scheme.

Best attack: generic lattice reduction.

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Best attack: generic lattice reduction.

SVP attack: $\mathbf{gap}(\mathcal{L})$.

Decodable lattice $\mathcal{L}$ $\Longrightarrow$ Encryption scheme.

Best attack: generic lattice reduction.

SVP attack: $\mathbf{gap}(\mathcal{L})$.

Dual SVP attack: $\mathbf{gap}(\mathcal{L}^*)$.

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Best attack: generic lattice reduction.

SVP attack: $\mathbf{gap}(\mathcal{L})$.

Dual SVP attack: $\mathbf{gap}(\mathcal{L}^*)$.

Decoding attack (BDD): $\mathbf{gap}(\mathcal{L}, \rho)$.

Decodable Lattices

	Primal	Dual	Decoding
Decodable Lattice	$\mathrm{gap}(\mathcal{L})$	$\mathrm{gap}(\mathcal{L}^*)$	$\mathrm{gap}(\mathcal{L}, \rho)$
Random Lattice	$\Theta(1)$	$\Theta(1)$	$2^{\Theta(n)}$
$\mathbb{Z}^n$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$	$\Theta(\sqrt{n})$
NTRU [HPS98]	$\Omega(\alpha)$	$\Omega(\alpha)$	$\Omega(n/\alpha)$
LWE [Ajt99, AP11, MP12]	$\Omega(1)$	$\Omega(\alpha)$	$\Omega(n/\alpha)$
Prime Lattice [CR88, DP19]	$\Theta(\log n)$	$\Omega(\sqrt{n})$	$\Theta(\log n)$
Barnes-Sloane [MP21]	$\Theta(\sqrt{\log n})$	$\Omega(\sqrt{n})$	$\Theta(\sqrt{\log n})$
Reed-Solomon [BP22]	$\Theta(\sqrt{\log n})$	$\Omega(\sqrt{n})$	$\Theta(\sqrt{\log n})$
Barnes-Wall [MN08]	$\Theta(\sqrt[4]{n})$	$\Theta(\sqrt[4]{n})$	$\Theta(\sqrt[4]{n})$

Interesting cases

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Decodable lattice $\mathcal{L} \implies$ Encryption scheme.

$\mathbb{Z}^n$: similar geometry to NTRU, LWE, but extremely simple and efficient.

$$n = 1024 \implies \beta \approx 440$$

Interesting cases

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

$\mathbb{Z}^n$: similar geometry to NTRU, LWE,
but extremely simple and efficient.

$$n = 1024 \implies \beta \approx 440$$

BW^n: better geometry and decoding $O(\sqrt[4]{n})$,

$$n = 1024 \implies \beta \approx 710.$$

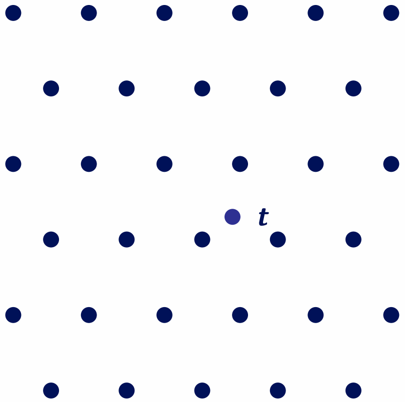
Decodable lattice $\mathcal{L} \implies$ Encryption scheme.

$\mathbb{Z}^n$: similar geometry to NTRU, LWE,
but extremely simple and efficient.
$$n = 1024 \implies \beta \approx 440$$

BW^n: better geometry and decoding $O(\sqrt[4]{n})$,
$$n = 1024 \implies \beta \approx 710.$$

?: gaps $\leq$ poly-log(n),
$$\beta \approx n.$$

FALCON (and MITAKA) use the hash-and-sign design with NTRU lattices.



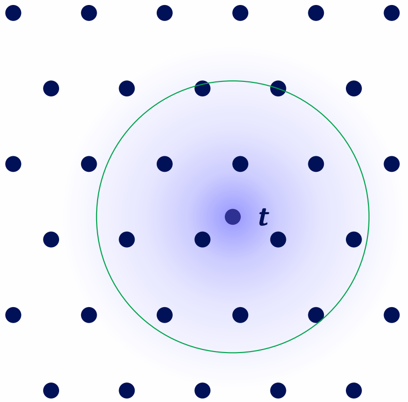
Sign(m):

- Hash m to a target t.
- (Gaussian) sample a nearby lattice point s using a good basis.

Verify(m, s):

- Hash m to a target t.
- Check $s \in \mathcal{L}$ and $\|s - t\|$ small.

FALCON (and MITAKA) use the hash-and-sign design with NTRU lattices.



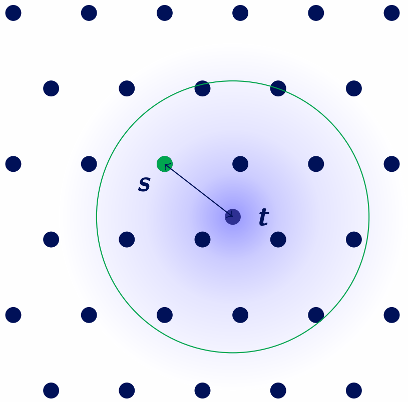
Sign(m):

- Hash m to a target t.
- (Gaussian) sample a nearby lattice point s using a good basis.

Verify(m, s):

- Hash m to a target t.
- Check $s \in \mathcal{L}$ and $\|s - t\|$ small.

Falcon (and Mitaka) use the hash-and-sign design with NTRU lattices.



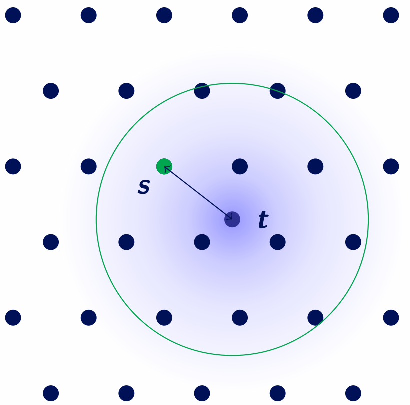
Sign(m):

- Hash m to a target t.
- (Gaussian) sample a nearby lattice point s using a good basis.

Verify(m, s):

- Hash m to a target t.
- Check $s \in \mathcal{L}$ and $\|s - t\|$ small.

FALCON (and MITAKA) use the hash-and-sign design with NTRU lattices.



Sign(m):

- Hash m to a target t.
- (Gaussian) sample a nearby lattice point s using a good basis.

Complicated and slow!

Verify(m, s):

- Hash m to a target t.
- Check $s \in \mathcal{L}$ and $\|s - t\|$ small.

Idea: Gaussian sampling in (cosets of) $\mathbb{Z}^n$ is (almost) trivial.

Idea: Gaussian sampling in (cosets of) $\mathbb{Z}^n$ is (almost) trivial.

How to make it competitive?

Idea: Gaussian sampling in (cosets of) $\mathbb{Z}^n$ is (almost) trivial.

How to make it competitive?

1. We add structure: module-LIP.

Idea: Gaussian sampling in (cosets of) $\mathbb{Z}^n$ is (almost) trivial.

How to make it competitive?

1. We add structure: module-LIP.
2. We compress keys and signatures.

Idea: Gaussian sampling in (cosets of) $\mathbb{Z}^n$ is (almost) trivial.

How to make it competitive?

1. We add structure: module-LIP.
2. We compress keys and signatures.
3. Only hash to targets in $\frac{1}{2}\mathbb{Z}^n$.

Performance of Hawk

- Hawk has an *isochronous* implementation in C.

	Falcon-512	Hawk-512		Falcon-1024	Hawk-1024	
KeyGen *	**7.95** ms	**4.25** ms	$\downarrow$ /**1.9**	**23.60** ms	**17.88** ms	$\downarrow$ /**1.3**
Sign *	**193** µs	**50** µs	$\downarrow$ /**3.9**	**382** µs	**99** µs	$\downarrow$ /**3.9**
Verify *	**50** µs	**19** µs	$\downarrow$ /**2.6**	**99** µs	**46** µs	$\downarrow$ /**2.2**
$\lvert$**sk**$\rvert$	**1281**	**1153**	$\downarrow$ /**1.1**	**2305**	**2561**	$\uparrow$ $\times$**1.1**
$\lvert$**pk**$\rvert$	**897**	**1006** $\pm$ **6**	$\uparrow$ $\times$**1.2**	**1793**	**2329** $\pm$ **11**	$\uparrow$ $\times$**1.29**
$\lvert$**sig**$\rvert$	**652** $\pm$ **3**	**542** $\pm$ **4**	$\downarrow$ /**1.20**	**1261** $\pm$ **4**	**1195** $\pm$ **6**	$\downarrow$ /**1.06**

Table: Performance on an i5-4590 @**3.30**GHz CPU.
*: AVX2 implementation using floats.

Performance of Hawk

- HAWK has an *isochronous* implementation in C.
- HAWK remains fast when floating points are unavailable.

	FALCON-512	HAWK-512		FALCON-1024	HAWK-1024	
KeyGen *	**7.95** ms	**4.25** ms	↓ /1.9	**23.60** ms	**17.88** ms	↓ /1.3
KeyGen	**19.32** ms	**13.14** ms	↓ /1.5	**54.65** ms	**41.39** ms	↓ /1.3
Sign *	**193** µs	**50** µs	↓ /3.9	**382** µs	**99** µs	↓ /3.9
Sign	**2449** µs	**168**µs	↓ /15	**5273** µs	**343**µs	↓ /15
Verify *	**50** µs	**19** µs	↓ /2.6	**99** µs	**46** µs	↓ /2.2
Verify	**53** µs	**178** µs	↑ ×3.4	**105** µs	**392** µs	↑ ×3.7
$\|sk\|$	**1281**	**1153**	↓ /1.1	**2305**	**2561**	↑ ×1.1
$\|pk\|$	**897**	**1006** ± **6**	↑ ×1.2	**1793**	**2329** ± **11**	↑ ×1.29
$\|sig\|$	**652** ± **3**	**542** ± **4**	↓ /1.20	**1261** ± **4**	**1195** ± **6**	↓ /1.06

Table: Performance on an i5-4590 @3.30GHz CPU.
*: AVX2 implementation using floats.

Conclusion

Any lattice $\Longrightarrow$ Identification scheme.

Decodable lattice $\mathcal{L}$ $\Longrightarrow$ Encryption scheme.

Gaussian sampleable lattice $\mathcal{L}$ $\Longrightarrow$ Signature scheme.

Conclusion

Any lattice $\implies$ Identification scheme.

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Gaussian sampleable lattice $\mathcal{L}$ $\implies$ Signature scheme.

$\mathbb{Z}^n$ is enough to
match LWE and NTRU.

Conclusion

Any lattice $\implies$ Identification scheme.

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Gaussian sampleable lattice $\mathcal{L}$ $\implies$ Signature scheme.

$\mathbb{Z}^n$ is enough to
match LWE and NTRU.

End goal: do even better.

Conclusion

Any lattice $\implies$ Identification scheme.

Decodable lattice $\mathcal{L}$ $\implies$ Encryption scheme.

Gaussian sampleable lattice $\mathcal{L}$ $\implies$ Signature scheme.

$\mathbb{Z}^n$ is enough to
match LWE and NTRU.

End goal: do even better.

Thanks! :)