NTRU vs (more) standard lattice problems

Alice Pellet-Mary

based on joint works with Joël Felderhoff and Damien Stehlé

London-ish Lattice Coding & Crypto Meetings

Context: NTRU

NTRU

(N-th degree truncated polynomial ring units)

- algorithmic problem based on lattices
- supposedly hard even with a quantum computer
- efficient
- used in post-quantum crypto: e.g., Falcon, NTRU and NTRUPrime
- old (for lattice-based crypto): introduced in 1996

[HPS98] Hoffstein, Pipher, and Silverman. NTRU: a ring based public key cryptosystem. ANTS.

Context: Ring / Module LWE

Ring LWE and Module LWE

(Ring / Module Learning With Errors)

- algorithmic problem based on lattices
- supposedly hard even with a quantum computer
- efficient
- used in post-quantum crypto: e.g., Dilithium, Saber and Kyber
- more recent: introduced in 2009

[SSTX09] Stehlé, Steinfeld, Tanaka, and Xagawa. Efficient public key encryption based on ideal lattices. Asiacrypt.
[LPR10] Lyubashevsky, Peikert, and Regev. On ideal lattices and learning with errors over rings. Eurocrypt.
[LS15] Langlois and Stehlé. Worst-case to average-case reductions for module lattices. Design Codes Cryptography.

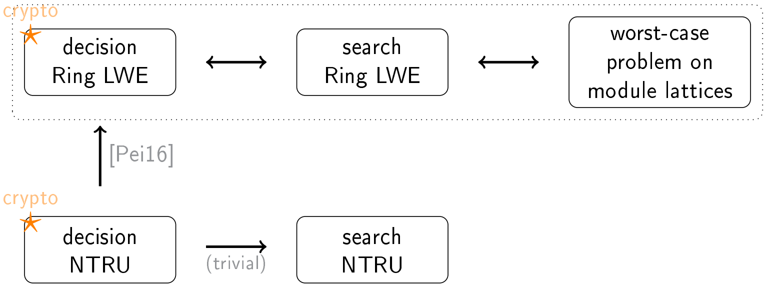
NTRU vs Ring LWE

- both are efficient
- both are versatile (but Ring LWE a bit more)
- NTRU is older

[Pei16] Peikert. A decade of lattice cryptography. Foundations and Trends in TCS.

NTRU vs Ring LWE

- both are efficient
- both are versatile (but Ring LWE a bit more)
- NTRU is older
- Ring LWE has stronger theoretical security guarantees (reductions)



[Pei16] Peikert. A decade of lattice cryptography. Foundations and Trends in TCS.

NTRU

Some definitions

If you like number fields

- $R = \mathbb{Z}[X]/(X^n + 1)$ $(n = 2^k)$
- $K = \mathbb{Q}[X]/(X^n + 1)$
- $q \in \mathbb{Z},\ q \geq 2$
  ($q \in R$, polynomial of degree 0)
- $R_q = (\mathbb{Z}/q\mathbb{Z})[X]/(X^n + 1)$
- $\|a\| = \sqrt{\sum_i a_i^2}$ $(a = \sum_{i=0}^{n-1} a_i X^i \in R)$

(K can be any other number field)

If you don't

- $R = \mathbb{Z}$
- $K = \mathbb{Q}$
- $q \in \mathbb{Z},\ q \geq 2$
- $R_q = \mathbb{Z}/q\mathbb{Z}$
- $\|a\| = |a|$ $(a \in R)$

Many NTRU variants

- search vs decision

Many NTRU variants

- search vs decision
- worst-case vs average case

Many NTRU variants

- search vs decision
- worst-case vs average case
- short vector vs dense sub-lattice

Many NTRU variants

- search vs decision
- worst-case vs average case
- short vector vs dense sub-lattice

In this talk: only worst-case variants (3 variants in total)

NTRU instances

NTRU instance

A γ-NTRU instance is $h \in R_q$ s.t.

- $h = f/g \bmod q$ (or $gh = f \bmod q$)
- $\|f\|, \|g\| \leq \frac{\sqrt{q}}{\gamma}$

The pair (f, g) is a trapdoor for h.

NTRU instances

NTRU instance

A γ-NTRU instance is $h \in R_q$ s.t.

- $h = f/g \bmod q$ (or $gh = f \bmod q$)
- $\|f\|, \|g\| \leq \frac{\sqrt{q}}{\gamma}$

The pair (f, g) is a trapdoor for h.

Claim: if (f, g) and (f', g') are two trapdoors for the same h,

$$\frac{f'}{g'} = \frac{f}{g} =: h_K \in K \quad \text{(division performed in K)}$$

NTRU instances

NTRU instance

A γ-NTRU instance is $h \in R_q$ s.t.

- $h = f/g \bmod q$ (or $gh = f \bmod q$)
- $\|f\|, \|g\| \leq \frac{\sqrt{q}}{\gamma}$

The pair (f, g) is a trapdoor for h.

Claim: if (f, g) and (f', g') are two trapdoors for the same h,

$$\frac{f'}{g'} = \frac{f}{g} =: h_K \in K \quad \text{(division performed in K)}$$

Proof: $\frac{f}{g} = \frac{f'}{g'} \bmod q \Rightarrow fg' = f'g \bmod q \Rightarrow fg' = f'g \Rightarrow \frac{f}{g} = \frac{f'}{g'}$

Decisional NTRU problem

(worst-case) decision NTRU

The γ-decisional NTRU problem asks, given $h \in R_q$, to decide whether

- h is a γ-NTRU instance (i.e., $h = f/g \bmod q$ with $\|f\|, \|g\| \leq \sqrt{q}/\gamma$)
- or not

Search NTRU problems

$\mathrm{NTRU}_{\mathrm{vec}}$

The γ-search NTRU vector problem (γ-$\mathrm{NTRU}_{\mathrm{vec}}$) asks, given a γ-NTRU instance h, to recover $(f, g) \in R^2$ s.t.

- $h = f/g \bmod q$
- $\|f\|, \|g\| \leq \sqrt{q}/\gamma$

Search NTRU problems

$\mathrm{NTRU}_{\mathrm{vec}}$

The γ-search NTRU vector problem (γ-$\mathrm{NTRU}_{\mathrm{vec}}$) asks, given a γ-NTRU instance h, to recover $(f, g) \in R^2$ s.t.

- $h = f/g \bmod q$
- $\|f\|, \|g\| \leq \sqrt{q}/\gamma$

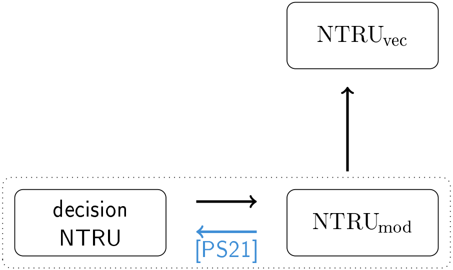
$\mathrm{NTRU}_{\mathrm{mod}}$

The γ-search NTRU module problem (γ-$\mathrm{NTRU}_{\mathrm{mod}}$) asks, given a γ-NTRU instance h, to recover h_K.
(Recall $h_K = f/g \in K$ for any trapdoor (f, g))
$\Leftrightarrow$ recover $(\alpha f, \alpha g)$ for any $\alpha \in K$

How do they compare?

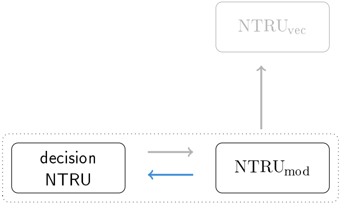
$$
\begin{tikzcd}
 & \boxed{\mathrm{NTRU}_{\mathrm{vec}}} \\
\boxed{\text{decision NTRU}} \arrow[r] & \boxed{\mathrm{NTRU}_{\mathrm{mod}}} \arrow[u]
\end{tikzcd}
$$

How do they compare?



[PS21] Pellet-Mary and Stehlé. On the hardness of the NTRU problem. Asiacrypt.

Proof: from $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU



Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (1/2)

Objective: given $h = f/g \bmod q$, recover $h_K = f/g \in K$ (division in K)

Can use an oracle for decision NTRU:

given $h' \in R_q$, the oracle outputs

- YES if $h' = f'/g' \bmod q$, with $\|f'\|, \|g'\| \leq B$ $(= \sqrt{q}/\gamma)$
- NO otherwise

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (1/2)

Objective: given $h = f/g \bmod q$, recover $h_K = f/g \in K$ (division in K)

Can use an oracle for decision NTRU:

given $h' \in R_q$, the oracle outputs

- YES if $h' = f'/g' \bmod q$, with $\|f'\|, \|g'\| \leq B$ $(= \sqrt{q}/\gamma)$
- NO otherwise

Idea:

- take $x, y \in R$
- create $h' = x \cdot h + y = \frac{xf+yg}{g} \bmod q$
- query the oracle on h'
- learn whether $\|xf + yg\| \leq B$ or not

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (1/2)

Objective: given $h = f/g \bmod q$, recover $h_K = f/g \in K$ (division in K)

Can use an oracle for decision NTRU:

given $h' \in R_q$, the oracle outputs

- YES if $h' = f'/g' \bmod q$, with $\|f'\|, \|g'\| \leq B$ $(= \sqrt{q}/\gamma)$
- NO otherwise

Idea:

- take $x, y \in R$
- create $h' = x \cdot h + y = \frac{xf+yg}{g} \bmod q$
- query the oracle on h'
- learn whether $\|xf + yg\| \leq B$ or not

$\Rightarrow$ we can choose x and y

$\Rightarrow$ we can modify the coordinates one by one

(uses canonical embedding here)

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (2/2)

Simplified problem

$f, g \in \mathbb{R}$ secret, $B' \geq 0$ unknown.
Given any $x, y \in \mathbb{R}$, we can learn whether $|xf + yg| \geq B'$ or not.
Objective: recover f/g

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (2/2)

Simplified problem

$f, g \in \mathbb{R}$ secret, $B' \geq 0$ unknown.
Given any $x, y \in \mathbb{R}$, we can learn whether $|xf + yg| \geq B'$ or not.
Objective: recover f/g

Algorithm:

- Find x_0, y_0 such that $x_0 f + y_0 g = B'$
  (Fix $x_0 \ll B'/|f|$ and increase y_0 until the oracle says NO)

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (2/2)

Simplified problem

$f, g \in \mathbb{R}$ secret, $B' \geq 0$ unknown.
Given any $x, y \in \mathbb{R}$, we can learn whether $|xf + yg| \geq B'$ or not.
Objective: recover f/g

Algorithm:

- Find x_0, y_0 such that $x_0 f + y_0 g = B'$
  (Fix $x_0 \ll B'/|f|$ and increase y_0 until the oracle says NO)
- Find x_1, y_1 such that $x_1 \neq x_0$ and $x_1 f + y_1 g = B'$

Reducing $\mathrm{NTRU}_{\mathrm{mod}}$ to decision NTRU (2/2)

Simplified problem

$f, g \in \mathbb{R}$ secret, $B' \geq 0$ unknown.
Given any $x, y \in \mathbb{R}$, we can learn whether $|xf + yg| \geq B'$ or not.
Objective: recover f/g

Algorithm:

- Find x_0, y_0 such that $x_0 f + y_0 g = B'$
  (Fix $x_0 \ll B'/|f|$ and increase y_0 until the oracle says NO)
- Find x_1, y_1 such that $x_1 \neq x_0$ and $x_1 f + y_1 g = B'$
- Solve for f/g

What about average case?

In the average case: the decision NTRU oracle is not perfect

- YES if $h \leftarrow \mathcal{D}$ for some distribution $\mathcal{D}$ over NTRU instances
- NO if $h \leftarrow \mathcal{U}(R_q)$ (uniform in R_q)

What about average case?

In the average case: the decision NTRU oracle is not perfect

- YES if $h \leftarrow \mathcal{D}$ for some distribution $\mathcal{D}$ over NTRU instances
- NO if $h \leftarrow \mathcal{U}(R_q)$ (uniform in R_q)

In this case:

We use the "oracle hidden center" framework [PRS17]

[PRS17] Peikert, Regev, and Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. STOC.

What about average case?

In the average case: the decision NTRU oracle is not perfect

- YES if $h \leftarrow \mathcal{D}$ for some distribution $\mathcal{D}$ over NTRU instances
- NO if $h \leftarrow \mathcal{U}(R_q)$ (uniform in R_q)

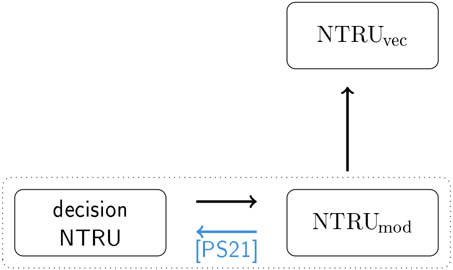
In this case:

We use the "oracle hidden center" framework [PRS17]

- we continuously transform $\mathcal{D}$ into $\mathcal{U}(R_q)$
- need to prove that the continuous transformation behaves nicely (lipschitz,...)
- then call [PRS17]

[PRS17] Peikert, Regev, and Stephens-Davidowitz. Pseudorandomness of ring-LWE for any ring and modulus. STOC.

Where are we?



[PS21] Pellet-Mary and Stehlé. On the hardness of the NTRU problem. Asiacrypt.

NTRU vs ideal lattice problems

SVP in ideal lattices

Recall: $R = \mathbb{Z}[X]/(X^n + 1)$ (or $R = \mathbb{Z}$)

SVP in ideal lattices

Recall: $R = \mathbb{Z}[X]/(X^n + 1)$ (or $R = \mathbb{Z}$)

For this talk: pretend all ideals are principal and computing generators is easy

(Principal) Ideals: $I = \langle z \rangle = \{zr \,|\, r \in R\}$

(e.g., $\langle 2 \rangle = \{2x \,|\, x \in \mathbb{Z}\}$)

SVP in ideal lattices

Recall: $R = \mathbb{Z}[X]/(X^n + 1)$ (or $R = \mathbb{Z}$)

For this talk: pretend all ideals are principal and computing generators is easy

(Principal) Ideals: $I = \langle z \rangle = \{zr \mid r \in R\}$
(e.g., $\langle 2 \rangle = \{2x \mid x \in \mathbb{Z}\}$)

ideal-SVP: Given $\langle z \rangle$, find $a \in \langle z \rangle$ such that $\|a\|$ is small
(recall: $\|a\| = \sqrt{\sum_i |a_i|^2}$ if $a = \sum_i a_i X^i$)

SVP in ideal lattices

Recall: $R = \mathbb{Z}[X]/(X^n + 1)$ (or $R = \mathbb{Z}$)

For this talk: pretend all ideals are principal and computing generators is easy

(Principal) Ideals: $I = \langle z \rangle = \{zr \mid r \in R\}$

(e.g., $\langle 2 \rangle = \{2x \mid x \in \mathbb{Z}\}$)

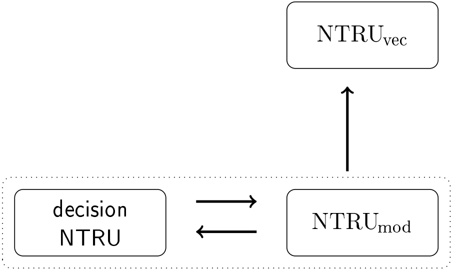
ideal-SVP: Given $\langle z \rangle$, find $a \in \langle z \rangle$ such that $\|a\|$ is small

(recall: $\|a\| = \sqrt{\sum_i |a_i|^2}$ if $a = \sum_i a_i X^i$)

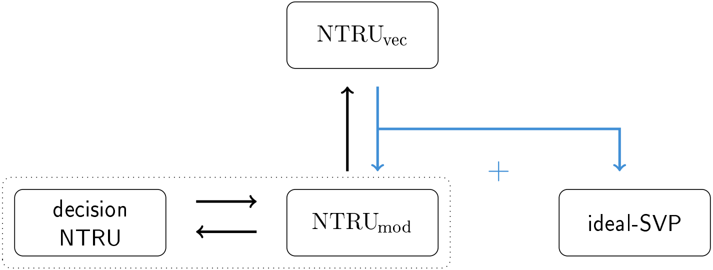
Remark: $x|y \not\Rightarrow \|x\| \leq \|y\|$

smallness for divisibility is different from smallness for Euclidean norm

From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$



From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$



From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$: proof

Objective: given $h_k \in K$ with the promise that $\exists (f, g) \in R^2$ with $h_K = f/g$ and $\|f\|, \|g\| \leq B$, recover one such pair (f, g).

From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$: proof

Objective: given $h_k \in K$ with the promise that $\exists (f, g) \in R^2$ with $h_K = f/g$ and $\|f\|, \|g\| \leq B$, recover one such pair (f, g).

Algorithm:

- Recover large $(f', g') \in R^2$ with $h_K = f'/g'$

From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$: proof

Objective: given $h_k \in K$ with the promise that $\exists (f, g) \in R^2$ with $h_K = f/g$ and $\|f\|, \|g\| \leq B$, recover one such pair (f, g).

Algorithm:

- Recover large $(f', g') \in R^2$ with $h_K = f'/g'$
- Compute $\delta = \gcd(f', g')$ and $f'' = f'/\delta$, $g'' = g'/\delta$
  (f'' and g'' are small for divisibility, but not for Euclidean norm)

From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$: proof

Objective: given $h_k \in K$ with the promise that $\exists (f, g) \in R^2$ with $h_K = f/g$ and $\|f\|, \|g\| \leq B$, recover one such pair (f, g).

Algorithm:

- Recover large $(f', g') \in R^2$ with $h_K = f'/g'$
- Compute $\delta = \gcd(f', g')$ and $f'' = f'/\delta$, $g'' = g'/\delta$
  (f'' and g'' are small for divisibility, but not for Euclidean norm)
- Define $z = |f''| + |g''|$ and solve SVP in $\langle z \rangle$
  $\Rightarrow$ find $r \in R$ s.t. $\|r \cdot z\|$ small

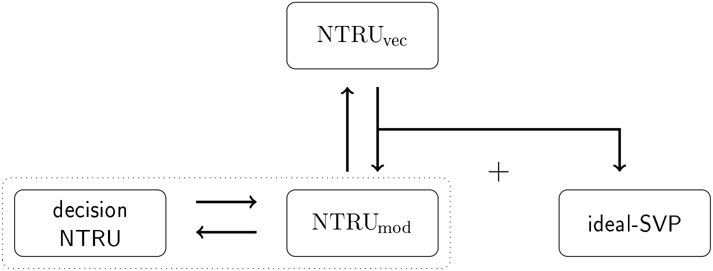
From $\mathrm{NTRU}_{\mathrm{vec}}$ to $\mathrm{NTRU}_{\mathrm{mod}}$: proof

Objective: given $h_k \in K$ with the promise that $\exists (f, g) \in R^2$ with $h_K = f/g$ and $\|f\|, \|g\| \leq B$, recover one such pair (f, g).

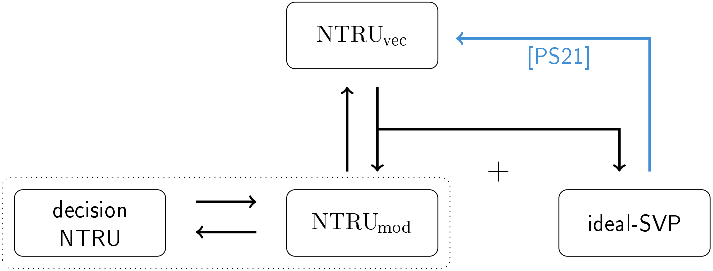
Algorithm:

- Recover large $(f', g') \in R^2$ with $h_K = f'/g'$
- Compute $\delta = \gcd(f', g')$ and $f'' = f'/\delta$, $g'' = g'/\delta$
  (f'' and g'' are small for divisibility, but not for Euclidean norm)
- Define $z = |f''| + |g''|$ and solve SVP in $\langle z \rangle$
  $\Rightarrow$ find $r \in R$ s.t. $\|r \cdot z\|$ small
- Output $(r \cdot f'', r \cdot g'')$

NTRU vs ideal-SVP

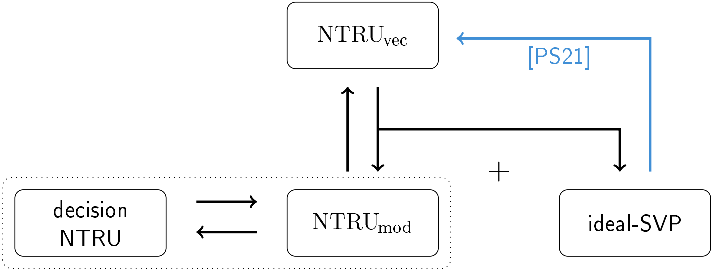


NTRU vs ideal-SVP



[PS21] Pellet-Mary and Stehlé. On the hardness of the NTRU problem. Asiacrypt.

NTRU vs ideal-SVP



$$\boxed{\mathrm{NTRU}_{\mathrm{vec}} \approx \mathrm{NTRU}_{\mathrm{mod}} + \text{ideal-SVP}}$$

$\mathrm{NTRU}_{\mathrm{mod}}$ and ideal-SVP seem incomparable

[PS21] Pellet-Mary and Stehlé. On the hardness of the NTRU problem. Asiacrypt.

NTRU vs module lattice problems

Module lattices

(free) Module: $M = \{\sum_{i=1}^{k} x_i \cdot \boldsymbol{b}_i \mid x_i \in R\}$,
where $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k \in K^k$ are linearly independent

Module lattices

(free) Module: $M = \{\sum_{i=1}^{k} x_i \cdot \boldsymbol{b}_i \mid x_i \in R\}$,
where $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k \in K^k$ are linearly independent

Properties:

- k is the rank of M

Module lattices

(free) Module: $M = \{\sum_{i=1}^{k} x_i \cdot \boldsymbol{b}_i \mid x_i \in R\}$,
where $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k \in K^k$ are linearly independent

Properties:

- k is the rank of M
- rank-1 module = ideal

Module lattices

(free) Module: $M = \{\sum_{i=1}^{k} x_i \cdot \boldsymbol{b}_i \,|\, x_i \in R\}$,
where $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_k \in K^k$ are linearly independent

Properties:

- k is the rank of M
- rank-1 module = ideal
- $\sigma(M)$ is a lattice of rank kn, where

$$\sigma : K = \mathbb{Q}[X]/(X^n + 1) \to \mathbb{Q}^n$$
$$\sum_{i=0}^{n-1} a_i X^i \mapsto (a_0, \cdots, a_{n-1})$$

$\sigma(M)$ is a module lattice

Modules with exceptionally short vectors

unique-SVP (uSVP): input is a rank-N lattice L with

$$\lambda_1(L) \ll \det(L)^{1/N}$$

Modules with exceptionally short vectors

unique-SVP (uSVP): input is a rank-N lattice L with

$$\lambda_1(L) \ll \det(L)^{1/N}$$

Special case of module: if $L = \sigma(M)$ is a module-lattice $(N = nk)$

- 1 short vector in $L \Rightarrow n$ short vectors in L

Modules with exceptionally short vectors

unique-SVP (uSVP): input is a rank-N lattice L with

$$\lambda_1(L) \ll \det(L)^{1/N}$$

Special case of module: if $L = \sigma(M)$ is a module-lattice $(N = nk)$

- 1 short vector in $L \Rightarrow n$ short vectors in L
  If $\boldsymbol{s} \in M$ is small, then $\boldsymbol{b}_i = X^i \cdot \boldsymbol{s} \in M$ satisfies
  - $\|\boldsymbol{b}_i\| = \|\boldsymbol{s}\|$
  - $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{n-1}$ are $\mathbb{Z}$-linearly independent

Modules with exceptionally short vectors

unique-SVP (uSVP): input is a rank-N lattice L with

$$\lambda_1(L) \ll \det(L)^{1/N}$$

Special case of module: if $L = \sigma(M)$ is a module-lattice ($N = nk$)

- 1 short vector in $L \Rightarrow n$ short vectors in L
  If $\boldsymbol{s} \in M$ is small, then $\boldsymbol{b}_i = X^i \cdot \boldsymbol{s} \in M$ satisfies
  - $\|\boldsymbol{b}_i\| = \|\boldsymbol{s}\|$
  - $\boldsymbol{b}_0, \ldots, \boldsymbol{b}_{n-1}$ are $\mathbb{Z}$-linearly independent
- 1 exceptionally short vector in L
  $\Rightarrow$ an exceptionally dense rank-n sublattice (rank-1 submodule)

mod-uSVP instances (in rank 2)

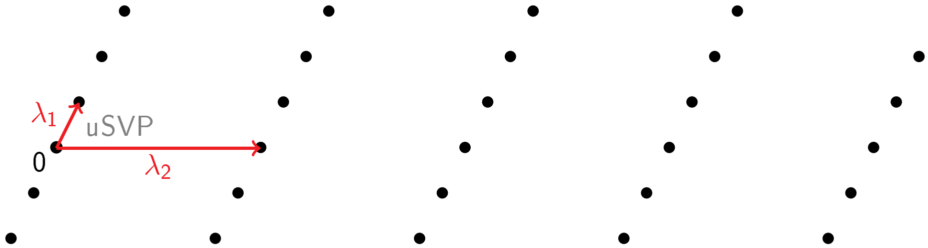
From now on: all modules have rank 2

mod-uSVP instance

A module unique SVP instance (γ-mod-uSVP) is $\boldsymbol{B} \in K^{2\times 2}$, basis of a rank-2 module M, s.t.

$$\|\lambda_1(M)\| \leq 1/\gamma \cdot \det(M)^{1/(2n)}.$$

(when $K = \mathbb{Q}$)



NTRU is a mod-uSVP

NTRU lattice: For $h \in R$, define

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{(in columns)}$$

$$\boxed{h \text{ is an NTRU instance} \Leftrightarrow \boldsymbol{B}_h \text{ is a mod-uSVP instance}}$$

NTRU is a mod-uSVP

NTRU lattice: For $h \in R$, define

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{(in columns)}$$

h is an NTRU instance $\Leftrightarrow$ $\boldsymbol{B}_h$ is a mod-uSVP instance

Proof of $\Rightarrow$: assume $h = f/g \bmod q$ with $\|f\|, \|g\| \leq \sqrt{q}/\gamma$
Define M_h rank-2 module spanned by $\boldsymbol{B}_h$

- $(g, f)^T \in M_h \Rightarrow \lambda_1(M_h) \leq \sqrt{2q}/\gamma$
- $\det(M_h) = q^n \Rightarrow \det(M_h)^{1/(2n)} = \sqrt{q}$

$\Rightarrow$ $\boldsymbol{B}_h$ is a $(\gamma/\sqrt{2})$-mod-uSVP instance

NTRU is a mod-uSVP

NTRU lattice: For $h \in R$, define

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{(in columns)}$$

h is an NTRU instance $\Leftrightarrow$ $\boldsymbol{B}_h$ is a mod-uSVP instance

Proof of $\Rightarrow$: assume $h = f/g \bmod q$ with $\|f\|, \|g\| \leq \sqrt{q}/\gamma$
Define M_h rank-2 module spanned by $\boldsymbol{B}_h$

- $(g, f)^T \in M_h \Rightarrow \lambda_1(M_h) \leq \sqrt{2q}/\gamma$
- $\det(M_h) = q^n \Rightarrow \det(M_h)^{1/(2n)} = \sqrt{q}$

$\Rightarrow$ $\boldsymbol{B}_h$ is a $(\gamma/\sqrt{2})$-mod-uSVP instance

Proof of $\Leftarrow$: similar, but requires a slightly more general definition of NTRU
($gh = f \bmod q$ instead of $h = f/g \bmod q$)

mod-uSVP problems

mod-uSVP$_{\mathrm{vec}}$

The γ-mod-uSVP vector problem (γ-mod-uSVP$_{\mathrm{vec}}$) asks, given a γ-mod-uSVP instance $\boldsymbol{B}$ spanning a module M, to recover $\boldsymbol{s} \in M$ s.t.

$$\|\boldsymbol{s}\| \leq 1/\gamma \cdot \det(M)^{1/(2n)}.$$

mod-uSVP problems

$\text{mod-uSVP}_{\text{vec}}$

The γ-mod-uSVP vector problem ($\gamma\text{-mod-uSVP}_{\text{vec}}$) asks, given a γ-mod-uSVP instance $\boldsymbol{B}$ spanning a module M, to recover $\boldsymbol{s} \in M$ s.t.

$$\|\boldsymbol{s}\| \leq 1/\gamma \cdot \det(M)^{1/(2n)}.$$

$\text{mod-uSVP}_{\text{mod}}$

The γ-mod-uSVP module problem ($\gamma\text{-mod-uSVP}_{\text{mod}}$) asks, given a γ-mod-uSVP instance $\boldsymbol{B}$ spanning a module M, to recover $\boldsymbol{v} \in M$ s.t.

$$\det(R \cdot \boldsymbol{v})^{1/n} \leq 1/\gamma \cdot \det(M)^{1/(2n)}.$$

($R \cdot \boldsymbol{v}$ is a dense rank-1 submodule of M)

NTRU is a mod-uSVP (2)

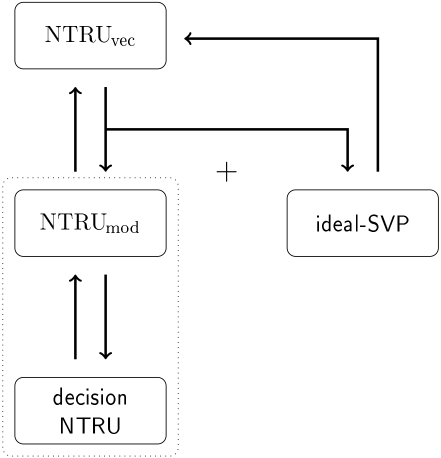
$\boxed{\mathrm{NTRU}_{\mathrm{vec}} = \text{mod-uSVP}_{\mathrm{vec}} \text{ restricted to NTRU instances}}$

NTRU is a mod-uSVP (2)

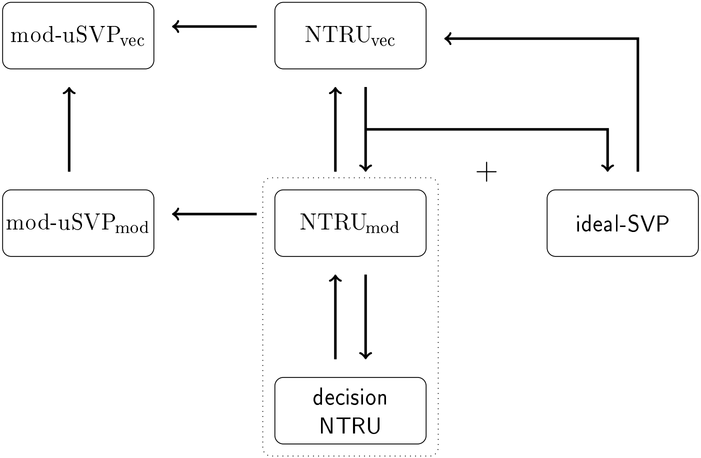
$\boxed{\mathrm{NTRU}_{\mathrm{vec}} = \text{mod-uSVP}_{\mathrm{vec}} \text{ restricted to NTRU instances}}$

$\boxed{\mathrm{NTRU}_{\mathrm{mod}} = \text{mod-uSVP}_{\mathrm{mod}} \text{ restricted to NTRU instances}}$

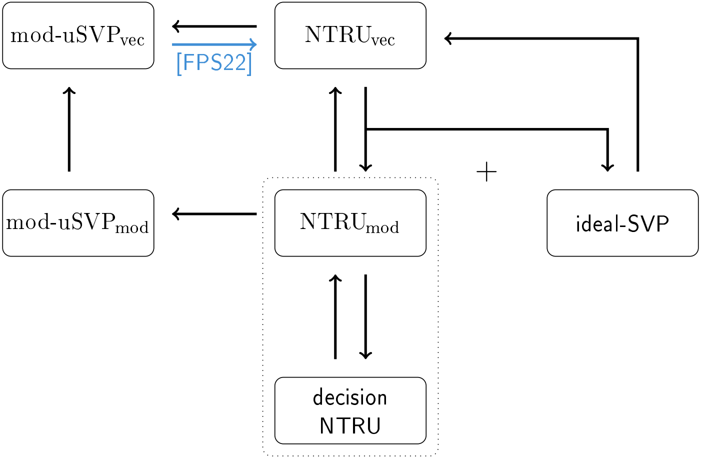
NTRU vs mod-uSVP



NTRU vs mod-uSVP

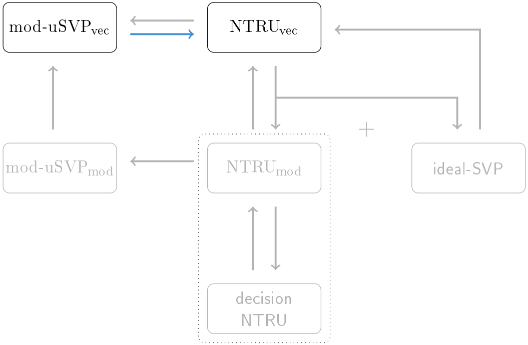


NTRU vs mod-uSVP



[FPS22] Felderhoff, Pellet-Mary, and Stehlé. On Module Unique-SVP and NTRU. Asiacrypt.

Proof: from mod-uSVP$_{\mathrm{vec}}$ to NTRU$_{\mathrm{vec}}$



Reminder and objective

$\text{mod-uSVP}_{\text{vec}}$

find a short vector in rank 2 module generated by

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

with $b_{ij} \in R$.

NTRU_{vec}

find a short vector in rank 2 module generated by

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix}$$

with $h \in R$ (and $q \in \mathbb{Z}$).

In both cases, promise that there exists an exceptionally short vector

Reminder and objective

mod-uSVP$_{\mathrm{vec}}$

find a short vector in rank 2 module generated by

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

with $b_{ij} \in R$.

NTRU$_{\mathrm{vec}}$

find a short vector in rank 2 module generated by

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix}$$

with $h \in R$ (and $q \in \mathbb{Z}$).

In both cases, promise that there exists an exceptionally short vector

Strategy: transform input $\boldsymbol{B}$ into some $\boldsymbol{B}_h$ with $\approx$ the same geometry

Reminder and objective

$\text{mod-uSVP}_{\text{vec}}$

find a short vector in rank 2 module generated by

$$\boldsymbol{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

with $b_{ij} \in R$.

NTRU_{vec}

find a short vector in rank 2 module generated by

$$\boldsymbol{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix}$$

with $h \in R$ (and $q \in \mathbb{Z}$).

In both cases, promise that there exists an exceptionally short vector

Strategy: transform input $\boldsymbol{B}$ into some $\boldsymbol{B}_h$ with $\approx$ the same geometry

Limitation: we will use an ideal-SVP oracle

(ok because we have a reduction ideal-SVP $\to \text{NTRU}_{\text{vec}}$)

Step 1: HNF

Input: M_0	$\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$	Compute the (module) HNF
	$\downarrow$	$\downarrow$
$M_1 = M_0$	$\begin{pmatrix} 1 & 0 \\ b'_{21} & b'_{22} \end{pmatrix}$	(with good probability $\gcd(b_{11}, b_{12}) = 1$)

Module unchanged $\Rightarrow$ geometry unchanged

Step 2: ideal-SVP

M_1 $\quad \left| \begin{pmatrix} 1 & 0 \\ b'_{21} & b'_{22} \end{pmatrix} \right|$

Step 2: ideal-SVP

$$M_1 \quad \left| \begin{pmatrix} 1 & 0 \\ b'_{21} & b'_{22} \end{pmatrix} \right| \quad \begin{array}{c} \text{compute } s = r \cdot b'_{22} \text{ with } r \in R \\ \text{s.t. } s = q + \varepsilon \ (\varepsilon \in R \text{ and } \|\varepsilon\| < q/n) \end{array}$$

- requires $q \geq \det(M_1)^{1/(2n)} \cdot \mathrm{poly}(n)$
- uses an ideal-SVP solver

Step 2: ideal-SVP

$$\begin{array}{l|c|c} M_1 & \begin{pmatrix} 1 & 0 \\ b'_{21} & b'_{22} \end{pmatrix} & \begin{array}{c} \text{compute } s = r \cdot b'_{22} \text{ with } r \in R \\ \text{s.t. } s = q + \varepsilon \ (\varepsilon \in R \text{ and } \|\varepsilon\| < q/n) \end{array} \\ & \downarrow & \downarrow \\ M_2 \subseteq M_1 & \begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix} & (s \in \langle b'_{22} \rangle \Rightarrow M_2 \subseteq M_1) \end{array}$$

- requires $q \geq \det(M_1)^{1/(2n)} \cdot \mathrm{poly}(n)$
- uses an ideal-SVP solver

Step 2: ideal-SVP

M_1	$\begin{pmatrix} 1 & 0 \\ b'_{21} & b'_{22} \end{pmatrix}$	compute $s = r \cdot b'_{22}$ with $r \in R$ s.t. $s = q + \varepsilon$ ($\varepsilon \in R$ and $\\|\varepsilon\\| < q/n$)
	$\downarrow$	$\downarrow$
$M_2 \subseteq M_1$	$\begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix}$	$(s \in \langle b'_{22} \rangle \Rightarrow M_2 \subseteq M_1)$

- requires $q \geq \det(M_1)^{1/(2n)} \cdot \mathrm{poly}(n)$
- uses an ideal-SVP solver

$$\boxed{\lambda_1(M_2) \leq \lambda_1(M_1) \cdot \mathrm{poly}(n) \quad \text{and} \quad \det(M_2)^{1/(2n)} \geq \det(M_1)^{1/(2n)}}$$

(provided $q \approx \det(M_1)^{1/(2n)}$)

Step 3: distortion

M_2	$\begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix}$	$s = q + \varepsilon \quad (\|\varepsilon\| \leq q/n)$

Step 3: distortion

$$
\begin{array}{l|c|c}
M_2 & \begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix} & s = q + \varepsilon \quad (\|\varepsilon\| \leq q/n) \\
 & & \text{distort (second coordinate } \times q/s \approx 1 + \tfrac{1}{n}) \\
 & \downarrow & \downarrow \\
M_3 \approx M_2 & \begin{pmatrix} 1 & 0 \\ b'_{21} \cdot q/s & q \end{pmatrix} &
\end{array}
$$

Step 3: distortion

$$
\begin{array}{l|c|c}
M_2 & \begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix} & s = q + \varepsilon \quad (\|\varepsilon\| \leq q/n) \\
 & & \text{distort (second coordinate } \times q/s \approx 1 + \frac{1}{n}) \\
 & \downarrow & \downarrow \\
M_3 \approx M_2 & \begin{pmatrix} 1 & 0 \\ b'_{21} \cdot q/s & q \end{pmatrix} & \text{round} \\
 & \downarrow & \downarrow \\
M_4 \approx M_3 & \begin{pmatrix} 1 & 0 \\ \lfloor b'_{21} \cdot q/s \rceil & q \end{pmatrix} & h = \lfloor b'_{21} \cdot q/s \rceil \in R
\end{array}
$$

Step 3: distortion

$$M_2 \qquad \begin{pmatrix} 1 & 0 \\ b'_{21} & s \end{pmatrix} \qquad s = q + \varepsilon \quad (\|\varepsilon\| \leq q/n)$$

distort (second coordinate $\times q/s \approx 1 + \frac{1}{n}$)

$$\downarrow$$

$$M_3 \approx M_2 \qquad \begin{pmatrix} 1 & 0 \\ b'_{21} \cdot q/s & q \end{pmatrix}$$

round

$$\downarrow$$

$$M_4 \approx M_3 \qquad \begin{pmatrix} 1 & 0 \\ \lfloor b'_{21} \cdot q/s \rceil & q \end{pmatrix} \qquad h = \lfloor b'_{21} \cdot q/s \rceil \in R$$

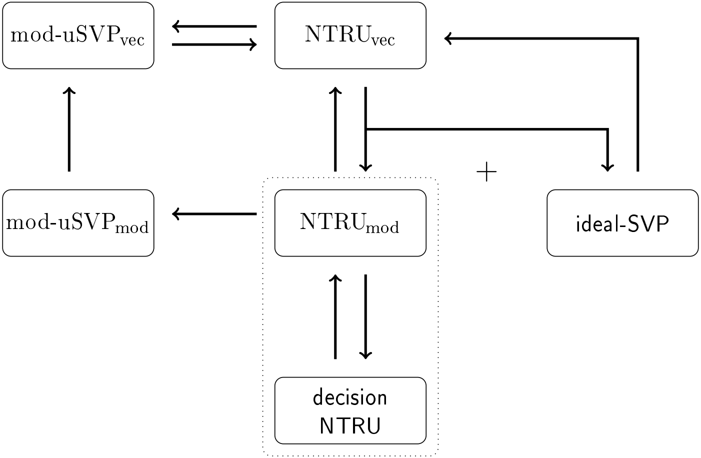
$M_4 \approx M_2$ is still a mod-uSVP instance

$+$

$\boldsymbol{B}_4$ has NTRU shape

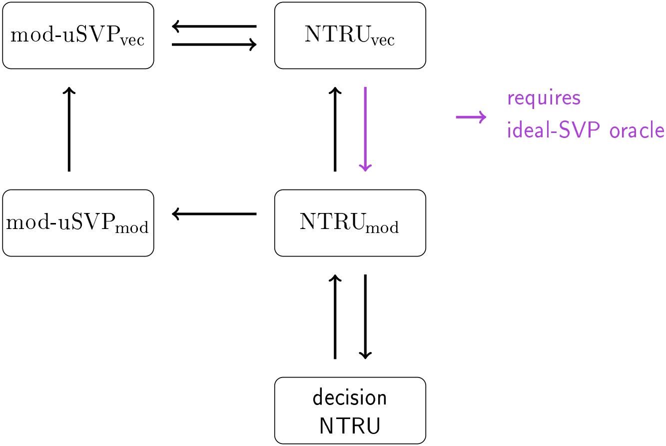
Conclusion

Summary



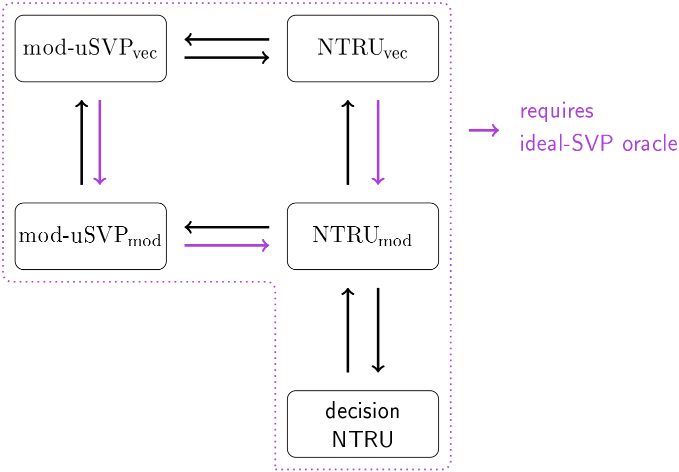
mod-uSVP$_{\text{vec}}$
NTRU$_{\text{vec}}$
+
mod-uSVP$_{\text{mod}}$
NTRU$_{\text{mod}}$
ideal-SVP
decision NTRU

Summary



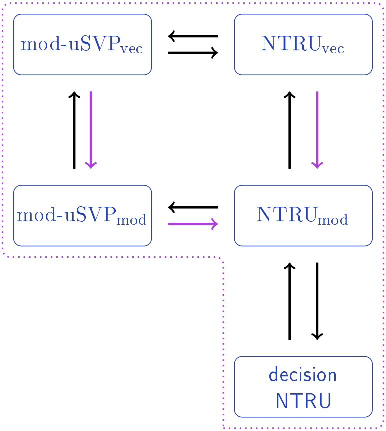
mod-uSVP$_{\text{vec}}$
NTRU$_{\text{vec}}$
mod-uSVP$_{\text{mod}}$
NTRU$_{\text{mod}}$
decision NTRU
requires ideal-SVP oracle

Summary



mod-uSVP$_{\text{vec}}$
NTRU$_{\text{vec}}$
mod-uSVP$_{\text{mod}}$
NTRU$_{\text{mod}}$
decision NTRU
requires ideal-SVP oracle

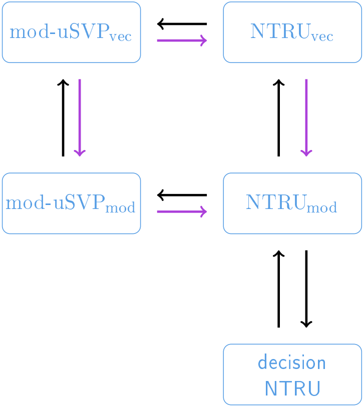
What about average case?



$\longrightarrow$ requires ideal-SVP oracle

Worst-case

What about average case?



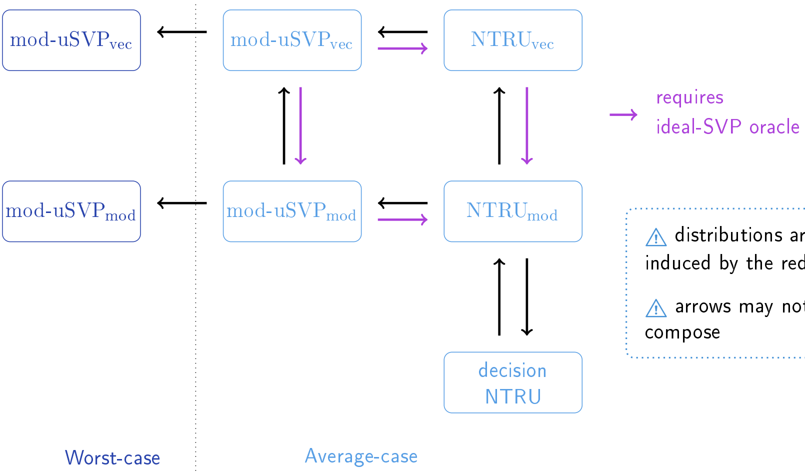
Average-case

$\longrightarrow$ requires ideal-SVP oracle

⚠ distributions are induced by the reductions

⚠ arrows may not compose

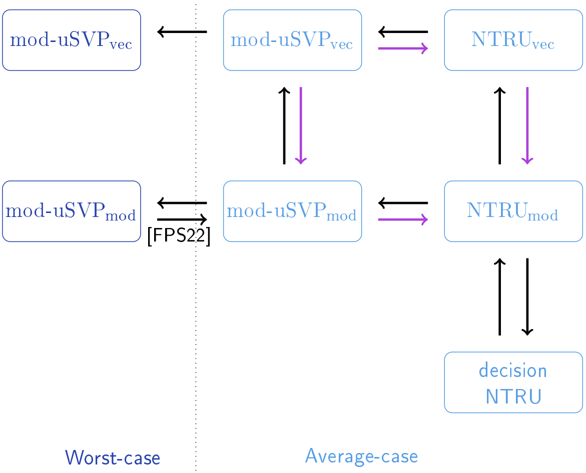
What about average case?



⚠ distributions are induced by the reductions

⚠ arrows may not compose

What about average case?



→ requires ideal-SVP oracle

⚠ distributions are induced by the reductions

⚠ arrows may not compose

[FPS22] Felderhoff, Pellet-Mary, and Stehlé. On Module Unique-SVP and NTRU. Asiacrypt.

Open problems

$\boxed{\text{NTRU} \approx \text{worst case uSVP in rank 2 modules}}$

Open problems

NTRU $\approx$ worst case uSVP in rank 2 modules

Limitations:

- Modulus q needs to be exponential for some reductions

Open problems

NTRU $\approx$ worst case uSVP in rank 2 modules

Limitations:

- Modulus q needs to be exponential for some reductions
- We do not always know how to sample from the average-case distribution

Open problems

$\boxed{\text{NTRU} \approx \text{worst case uSVP in rank 2 modules}}$

Limitations:

- Modulus q needs to be exponential for some reductions
- We do not always know how to sample from the average-case distribution

Open problems:

- How hard is worst-case uSVP in modules of rank 2?
  (e.g., compared to SVP / gap-SVP / SIVP in modules of rank 2, or to ideal-SVP?)
- Worst-case to average-case reduction directly for NTRU?

Open problems

NTRU $\approx$ worst case uSVP in rank 2 modules

Limitations:

- Modulus q needs to be exponential for some reductions
- We do not always know how to sample from the average-case distribution

Open problems:

- How hard is worst-case uSVP in modules of rank 2?
  (e.g., compared to SVP / gap-SVP / SIVP in modules of rank 2, or to ideal-SVP?)
- Worst-case to average-case reduction directly for NTRU?

Thank you