On the hardness of the NTRU problem

Alice Pellet-Mary[1] and Damien Stehlé[2]

[1] CNRS and Université de Bordeaux, [2] ENS de Lyon

Asiacrypt 2021

https://eprint.iacr.org/2021/821.pdf

Context: NTRU

NTRU

(N-th degree truncated polynomial ring units)

- algorithmic problem based on lattices
- post-quantum
- efficient
- used in Falcon, NTRU and NTRUPrime
  (round 3 in the NIST post quantum standardization process)
- 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
- post-quantum
- efficient
- used in Dilithium, Saber and Kyber
  (round 3 in the NIST post quantum standardization process)
- 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.

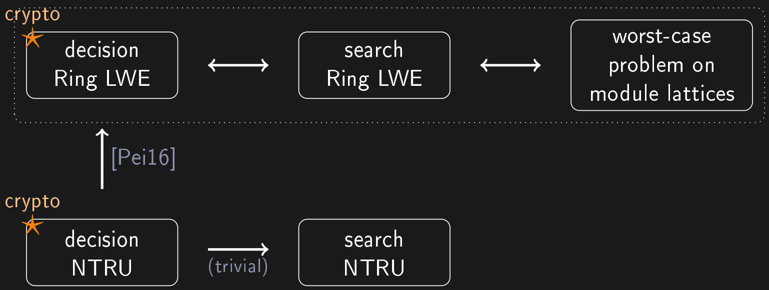
NTRU vs Ring LWE

- 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.

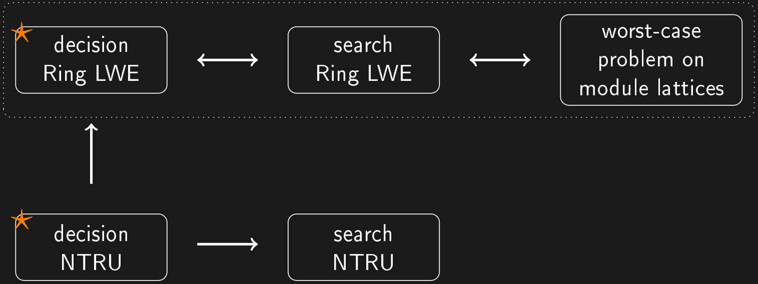
NTRU vs Ring LWE

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

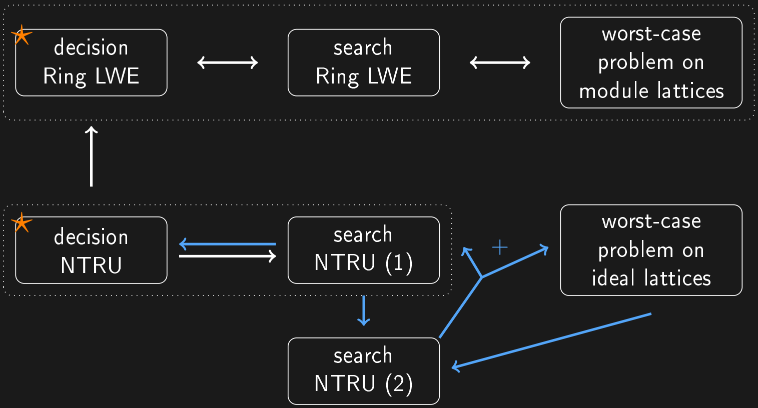


[Pei16] Peikert. A decade of lattice cryptography. Foundations and Trends in TCS.

Our result


decision
Ring LWE
search
Ring LWE
worst-case
problem on
module lattices
decision
NTRU
search
NTRU

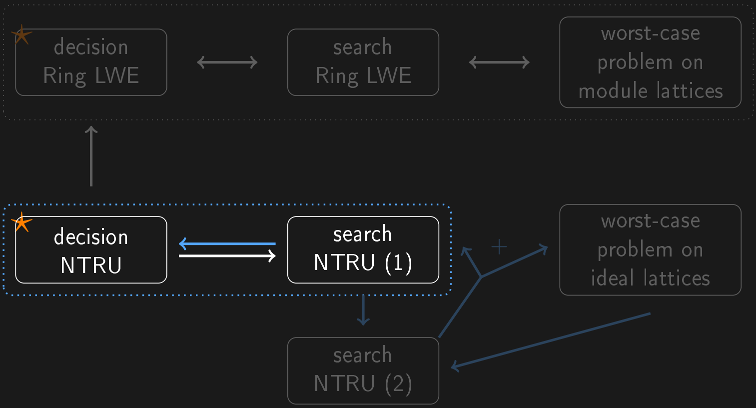
Our result



⚠ the reductions only work for certain distributions of NTRU instances ⚠

(the arrows may not compose)

Focus of this talk



Defining NTRU problems

$$\boxed{\text{decision NTRU}} \leftrightarrows \boxed{\text{search NTRU (1)}}$$

Notations

If you like polynomial rings

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

(K can be any other number field)

If you don't

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

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,

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

Decisional NTRU problem

decision NTRU

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

- $h \leftarrow \mathcal{D}$ where $\mathcal{D}$ is a distribution over γ-NTRU instances
- $h \leftarrow \mathcal{U}(R_q)$

Search NTRU problem (1)

Search NTRU (1)

The γ-search NTRU (1) problem asks, given a γ-NTRU instance h, to recover h_K.

(Recall $h_K = f/g \in K$ for any trapdoor (f, g))

Search NTRU problem (1)

Search NTRU (1)

The γ-search NTRU (1) problem 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 $\alpha \in K$

Search NTRU problem (1)

Search NTRU (1)

The γ-search NTRU (1) problem 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 $\alpha \in K$

Note: for the rest of the talk, "search NTRU" = " search NTRU (1)"

Techniques of the reduction

$$\text{decision NTRU} \rightleftarrows \text{search NTRU (1)}$$

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 small ($\leq B$)
- NO otherwise

(we assume for now that the oracle is perfect)

Reducing search NTRU 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 small $(\leq B)$
- NO otherwise

(we assume for now that the oracle is perfect)

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 $xf + yg$ is small or not

Reducing search NTRU 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 small ($\leq B$)
- NO otherwise

(we assume for now that the oracle is perfect)

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 $xf + yg$ is small or not

$\Rightarrow$ we can choose x and y

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

Reducing search NTRU 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 search NTRU 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

Remark: we can only learn f/g (not f and g)
(multiply f, g, B by the same $\alpha \rightsquigarrow$ oracle has the same behavior)

Reducing search NTRU 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

Remark: we can only learn f/g (not f and g)
(multiply f, g, B by the same $\alpha \rightsquigarrow$ oracle has the same behavior)

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 search NTRU 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

Remark: we can only learn f/g (not f and g)
(multiply f, g, B by the same $\alpha \rightsquigarrow$ oracle has the same behavior)

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 search NTRU 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

Remark: we can only learn f/g (not f and g)
(multiply f, g, B by the same $\alpha \rightsquigarrow$ oracle has the same behavior)

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

Handling imperfect oracles

If the oracle is not perfect:

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

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

If the oracle is not perfect:

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

- we continuously transform $\mathcal{D}$ into $\mathcal{U}(R_q)$
  (recall that $\mathcal{D}$ is a distribution over NTRU instances)
- 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.

Conclusion

More related works

Security guarantees:

[SS11, WW18] If $f, g \leftarrow D_{R,\sigma}$ with $\sigma \geq \mathrm{poly}(n) \cdot \sqrt{q}$
then $f/g \approx \mathcal{U}(R_q)$ (cyclotomic fields)

- decision NTRU is statistically hard when $\gamma \leq \frac{1}{\mathrm{poly}(n)}$

[SS11] Stehlé and Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. Eurocrypt.
[WW18] Wang and Wang. Provably secure NTRUEncrypt over any cyclotomic field. SAC.

More related works

Security guarantees:

[SS11, WW18] If $f, g \leftarrow D_{R,\sigma}$ with $\sigma \geq \mathrm{poly}(n) \cdot \sqrt{q}$
then $f/g \approx \mathcal{U}(R_q)$ (cyclotomic fields)

- decision NTRU is statistically hard when $\gamma \leq \frac{1}{\mathrm{poly}(n)}$

Attacks: (polynomial time)

[LLL82] decision/search(1) NTRU are broken if $\gamma \geq 2^n$

[LLL82] Lenstra, Lenstra, Lovász. Factoring polynomials with rational coefficients. Mathematische Annalen.

More related works

Security guarantees:

[SS11, WW18] If $f, g \leftarrow D_{R,\sigma}$ with $\sigma \geq \mathrm{poly}(n) \cdot \sqrt{q}$
then $f/g \approx \mathcal{U}(R_q)$ (cyclotomic fields)

- decision NTRU is statistically hard when $\gamma \leq \frac{1}{\mathrm{poly}(n)}$

Attacks: (polynomial time)

[LLL82] decision/search(1) NTRU are broken if $\gamma \geq 2^n$

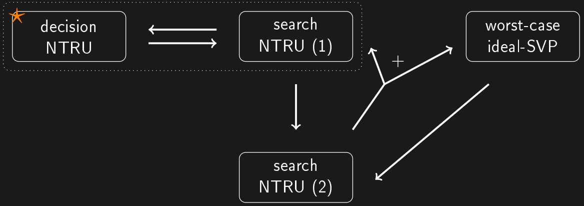
[ABD16, CJL16] [KF17] decision/search(1) NTRU are broken if $(\log q)^2 \geq n \cdot \log \frac{\sqrt{q}}{\gamma}$
(e.g., $q \approx 2^{\sqrt{n}}$ and $\gamma = \sqrt{q}/\mathrm{poly}(n)$)

[ABD16] Albrecht, Bai, and Ducas. A subfield lattice attack on overstretched NTRU assumptions. Crypto.

[CJL16] Cheon, Jeong, and Lee. An algorithm for NTRU problems. LMS J Comput Math.

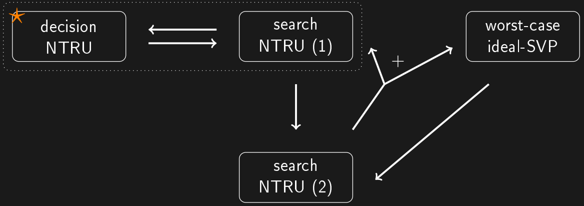
[KF17] Kirchner and Fouque. Revisiting lattice attacks on overstretched NTRU parameters. Eurocrypt

Conclusion and open problems



- Can we make the distributions of the reductions match?
- Can we prove hardness of decision NTRU from worst case lattice problems?
- Can we prove a reduction from module problems with rank ≥ 2?

Conclusion and open problems



- Can we make the distributions of the reductions match?
- Can we prove hardness of decision NTRU from worst case lattice problems?
- Can we prove a reduction from module problems with rank ≥ 2?

Thank you