

# On the hardness of the NTRU problem

Alice Pellet-Mary<sup>1</sup> and Damien Stehlé<sup>2</sup>

<sup>1</sup> CNRS and Université de Bordeaux, <sup>2</sup> ENS de Lyon

Asiacrypt 2021

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



# NTRU

Algorithmic problem based on lattices

- ▶ post-quantum
- ▶ efficient
- ▶ used in Falcon and NTRU / NTRUPrime (NIST finalists)
- ▶ old (for lattice-based crypto): introduced in 1996

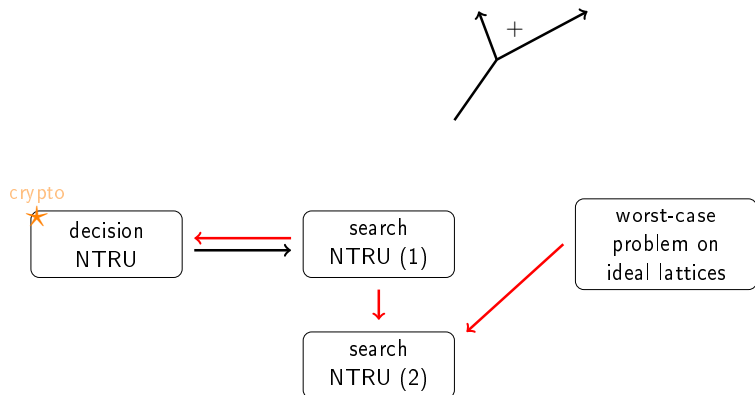
## Our result

We prove reductions between NTRU variants and other lattice problems



# Our result

We prove reductions between NTRU variants and other lattice problems



# The different NTRU problems

## NTRU instances

$$R = \mathbb{Z}[X]/(X^n + 1), \quad K = \mathbb{Q}[X]/(X^n + 1), \quad n = 2^k, \quad R_q = R/(qR)$$

## NTRU instances

$$R = \mathbb{Z}[X]/(X^n + 1), \quad K = \mathbb{Q}[X]/(X^n + 1), \quad n = 2^k, \quad R_q = R/(qR)$$

### NTRU instance

A  $(\gamma, q)$ -NTRU instance is  $h \in R_q$  s.t.

- ▶  $h = f/g \bmod q$  (or  $gh = f \bmod q$ )
- ▶  $\|f\|, \|g\| \leq \frac{\sqrt{q}}{\gamma}$  (if  $y = \sum_{i=0}^{n-1} y_i X^i \in R$ , then  $\|y\| = \sqrt{\sum_i y_i^2}$ )

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

## NTRU instances

$$R = \mathbb{Z}[X]/(X^n + 1), \quad K = \mathbb{Q}[X]/(X^n + 1), \quad n = 2^k, \quad R_q = R/(qR)$$

### NTRU instance

A  $(\gamma, q)$ -NTRU instance is  $h \in R_q$  s.t.

- ▶  $h = f/g \bmod q$  (or  $gh = f \bmod q$ )
- ▶  $\|f\|, \|g\| \leq \frac{\sqrt{q}}{\gamma}$  (if  $y = \sum_{i=0}^{n-1} y_i X^i \in R$ , then  $\|y\| = \sqrt{\sum_i y_i^2}$ )

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

## dNTRU

The  $(\gamma, q)$ -decisional NTRU problem ( $(\gamma, q)$ -dNTRU) asks, given  $h \in R_q$ , to decide whether

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

# Search NTRU problems

## NTRU<sub>vec</sub>

The  $(\gamma, q)$ -search NTRU vector problem ( $(\gamma, q)$ -NTRU<sub>vec</sub>) asks, given a  $(\gamma, q)$ -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

## NTRU<sub>vec</sub>

The  $(\gamma, q)$ -search NTRU vector problem ( $(\gamma, q)$ -NTRU<sub>vec</sub>) asks, given a  $(\gamma, q)$ -NTRU instance  $h$ , to recover  $(f, g) \in R^2$  s.t.

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

## NTRU<sub>mod</sub>

The  $(\gamma, q)$ -search NTRU module problem ( $(\gamma, q)$ -NTRU<sub>mod</sub>) asks, given a  $(\gamma, q)$ -NTRU instance  $h$ , to recover  $h_K$ .

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

# Search NTRU problems

## NTRU<sub>vec</sub>

The  $(\gamma, q)$ -search NTRU vector problem ( $(\gamma, q)$ -NTRU<sub>vec</sub>) asks, given a  $(\gamma, q)$ -NTRU instance  $h$ , to recover  $(f, g) \in R^2$  s.t.

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

## NTRU<sub>mod</sub>

The  $(\gamma, q)$ -search NTRU module problem ( $(\gamma, q)$ -NTRU<sub>mod</sub>) asks, given a  $(\gamma, q)$ -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  $\alpha \in K$

(Both problems exist in worst-case and average-case variants)

## NTRU is a (module) lattice problem

**NTRU lattice:** For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

## NTRU is a (module) lattice problem

NTRU lattice: For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

### Properties

- ▶ if  $h \leftarrow \mathcal{U}(R_q)$ :  $\lambda_1(\Lambda_h) \approx \sqrt{q} \cdot \sqrt{n}$

# NTRU is a (module) lattice problem

**NTRU lattice:** For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

## Properties

- ▶ if  $h \leftarrow \mathcal{U}(R_q)$ :  $\lambda_1(\Lambda_h) \approx \sqrt{q} \cdot \sqrt{n}$
- ▶ if  $h$  is  $(\gamma, q)$ -NTRU:  $\lambda_1(\Lambda_h) \leq \sqrt{q}/\gamma$  (this is a unique-SVP instance)

## NTRU is a (module) lattice problem

**NTRU lattice:** For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

### Properties

- ▶ if  $h \leftarrow \mathcal{U}(R_q)$ :  $\lambda_1(\Lambda_h) \approx \sqrt{q} \cdot \sqrt{n}$
- ▶ if  $h$  is  $(\gamma, q)$ -NTRU:  $\lambda_1(\Lambda_h) \leq \sqrt{q}/\gamma$  (this is a unique-SVP instance)

**dNTRU:** decide if  $\lambda_1(\Lambda_h)$  is small or not



## NTRU is a (module) lattice problem

**NTRU lattice:** For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

### Properties

- ▶ if  $h \leftarrow \mathcal{U}(R_q)$ :  $\lambda_1(\Lambda_h) \approx \sqrt{q} \cdot \sqrt{n}$
- ▶ if  $h$  is  $(\gamma, q)$ -NTRU:  $\lambda_1(\Lambda_h) \leq \sqrt{q}/\gamma$  (this is a unique-SVP instance)

**dNTRU:** decide if  $\lambda_1(\Lambda_h)$  is small or not

**NTRU<sub>vec</sub>:** recover  $(f, g) \leftrightarrow$  find a shortest vector in  $\Lambda_h$

# NTRU is a (module) lattice problem

**NTRU lattice:** For  $h \in R$ , define  $\Lambda_h$  the (module) lattice with basis

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

$$\Lambda_h = \{(g, f)^T \in R^2 \mid g \cdot h = f \pmod{q}\}$$

## Properties

- ▶ if  $h \leftarrow \mathcal{U}(R_q)$ :  $\lambda_1(\Lambda_h) \approx \sqrt{q} \cdot \sqrt{n}$
- ▶ if  $h$  is  $(\gamma, q)$ -NTRU:  $\lambda_1(\Lambda_h) \leq \sqrt{q}/\gamma$  (this is a unique-SVP instance)

**dNTRU:** decide if  $\lambda_1(\Lambda_h)$  is small or not

**NTRU<sub>vec</sub>:** recover  $(f, g) \leftrightarrow$  find a shortest vector in  $\Lambda_h$

**NTRU<sub>mod</sub>:** recover  $\alpha \cdot (f, g) \leftrightarrow$  find the direction where  $\Lambda_h$  is dense

# What we know about NTRU

## Previous works

### Reductions:

- [SS11, WW18] If  $f, g \leftarrow D_{R, \sigma}$  with  $\sigma \geq \text{poly}(n) \cdot \sqrt{q}$   
then  $f/g \approx \mathcal{U}(R_q)$  (cyclotomic fields)
- ▶ dNTRU is provably hard when  $\gamma \leq \frac{1}{\text{poly}(n)}$

---

[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.

## Previous works

### Reductions:

- [SS11, WW18] If  $f, g \leftarrow D_{R, \sigma}$  with  $\sigma \geq \text{poly}(n) \cdot \sqrt{q}$   
then  $f/g \approx \mathcal{U}(R_q)$  (cyclotomic fields)  
▶ dNTRU is provably hard when  $\gamma \leq \frac{1}{\text{poly}(n)}$
- [Pei16] dNTRU  $\leq$  RLWE

---

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

## Previous works

### Reductions:

- [SS11, WW18] If  $f, g \leftarrow D_{R, \sigma}$  with  $\sigma \geq \text{poly}(n) \cdot \sqrt{q}$   
then  $f/g \approx \mathcal{U}(R_q)$  (cyclotomic fields)  
▶ dNTRU is provably hard when  $\gamma \leq \frac{1}{\text{poly}(n)}$
- [Pei16] dNTRU  $\leq$  RLWE

### Attacks: (polynomial time)

- [LLL82] dNTRU, NTRU<sub>mod</sub> broken if  $\gamma \geq 2^n$

---

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

## Previous works

### Reductions:

- [SS11, WW18] If  $f, g \leftarrow D_{R, \sigma}$  with  $\sigma \geq \text{poly}(n) \cdot \sqrt{q}$   
then  $f/g \approx \mathcal{U}(R_q)$  (cyclotomic fields)  
▶ dNTRU is provably hard when  $\gamma \leq \frac{1}{\text{poly}(n)}$
- [Pei16] dNTRU  $\leq$  RLWE

### Attacks: (polynomial time)

- [LLL82] dNTRU, NTRU<sub>mod</sub> broken if  $\gamma \geq 2^n$
- [ABD16, CJL16] dNTRU, NTRU<sub>mod</sub> broken if  $(\log q)^2 \geq n \cdot \log \frac{\sqrt{q}}{\gamma}$   
[KF17] (e.g.,  $q \approx 2^{\sqrt{n}}$  and  $\gamma = \sqrt{q}/\text{poly}(n)$ )

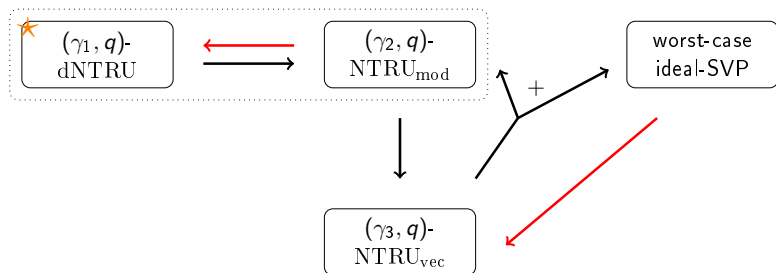
---

[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*

## Our results (with more details)

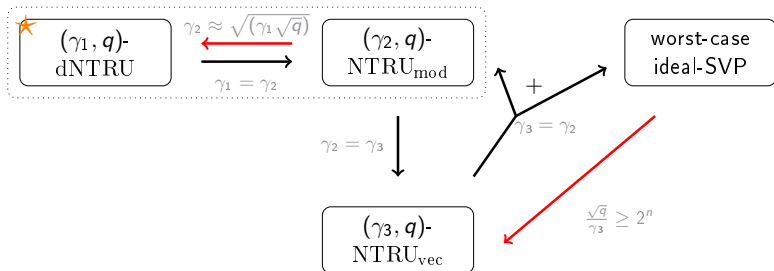


⚠ the reductions only work for certain distributions of NTRU instances ⚠  
(the arrows may not compose)

**Worst-case ideal-SVP:** given any ideal lattice  $I \subset R$ , find  $v \in I \setminus \{0\}$  such that  $\|v\| \leq \min_{w \in I \setminus \{0\}} \|w\|$ .



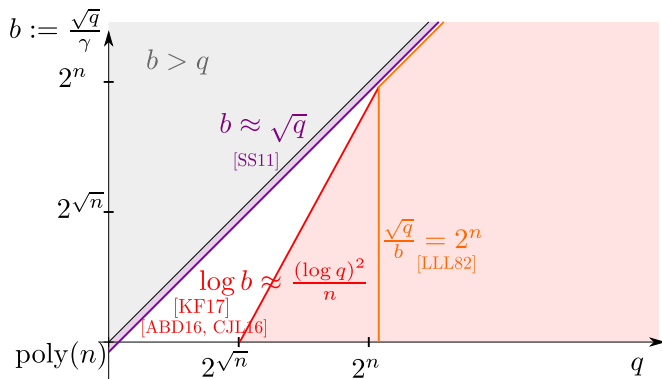
## Our results (with more details)



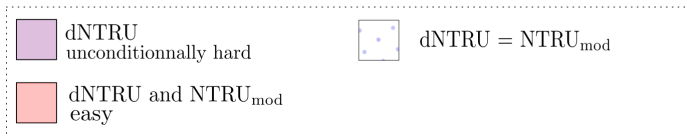
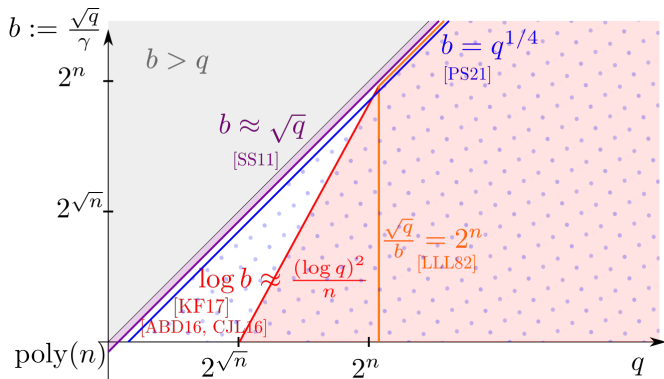
⚠ the reductions only work for certain distributions of NTRU instances ⚠  
(the arrows may not compose)

**Worst-case ideal-SVP:** given any ideal lattice  $I \subset R$ , find  $v \in I \setminus \{0\}$  such that  $\|v\| \leq \min_{w \in I \setminus \{0\}} \|w\|$ .

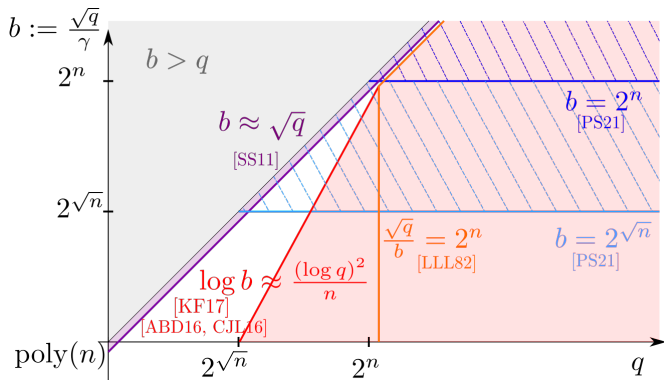
# One big picture: poly time attacks and reductions (cyclotomics)



# One big picture: poly time attacks and reductions (cyclotomics)

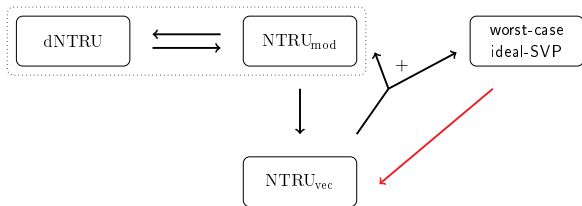


# One big picture: poly time attacks and reductions (cyclotomics)



	dNTRU unconditionnally hard		w.c. id-SVP $\leq$ NTRU <sub>vec</sub>
	dNTRU and NTRU <sub>mod</sub> easy		w.c. id-SVP $\leq$ NTRU <sub>vec</sub> quantumly, for cyclotomic fields

# Techniques



## From ideal-SVP to $\text{NTRU}_{\text{vec}}$

**Objective:** Transform an ideal  $I$  into an NTRU instance  $h$

- $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  (assume for simplicity that  $I$  is principal)
- $g$  short vector of  $I$

## From ideal-SVP to $\text{NTRU}_{\text{vec}}$

**Objective:** Transform an ideal  $I$  into an NTRU instance  $h$

- $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  (assume for simplicity that  $I$  is principal)
- $g$  short vector of  $I$

$$\begin{aligned}g &= z \cdot r && (r \in R) \\ \Leftrightarrow g \cdot \frac{q}{z} &= qr \\ \Leftrightarrow g \cdot h &= f \pmod{q}\end{aligned}$$

- ▶  $h = q/z, f = 0$
- ▶  $\|f\|, \|g\|$  small

## From ideal-SVP to $\text{NTRU}_{\text{vec}}$

**Objective:** Transform an ideal  $I$  into an NTRU instance  $h$

- $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  (assume for simplicity that  $I$  is principal)
- $g$  short vector of  $I$

$$g = z \cdot r \quad (r \in R)$$

$$\Leftrightarrow g \cdot \frac{q}{z} = qr$$

$$\Leftrightarrow g \cdot h = f \pmod{q}$$

- ▶  $h = q/z, f = 0$
- ▶  $\|f\|, \|g\|$  small

⚠ Not an NTRU instance ( $h \in K$  is not in  $R_q$ )



## From ideal-SVP to $\text{NTRU}_{\text{vec}}$

**Objective:** Transform an ideal  $I$  into an NTRU instance  $h$

- $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  (assume for simplicity that  $I$  is principal)
- $g$  short vector of  $I$

$$\begin{aligned} g &= z \cdot r && (r \in R) \\ \Leftrightarrow g \cdot \frac{q}{z} &= qr \\ \Leftrightarrow g \cdot \left\lfloor \frac{q}{z} \right\rfloor &= -g \cdot \left\{ \frac{q}{z} \right\} \pmod{q} && \{x\} = x - \lfloor x \rfloor \\ \Leftrightarrow g \cdot h &= f \pmod{q} \end{aligned}$$

- ▶  $h = \lfloor q/z \rfloor$ ,  $f = -g\{q/z\}$
- ▶  $\|f\| \approx \|g\|$  small

## From ideal-SVP to $\text{NTRU}_{\text{vec}}$

**Objective:** Transform an ideal  $I$  into an NTRU instance  $h$

- $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  (assume for simplicity that  $I$  is principal)
- $g$  short vector of  $I$

$$\begin{aligned} g &= z \cdot r && (r \in R) \\ \Leftrightarrow g \cdot \frac{q}{z} &= qr \\ \Leftrightarrow g \cdot \left\lfloor \frac{q}{z} \right\rfloor &= -g \cdot \left\{ \frac{q}{z} \right\} \pmod{q} && \{x\} = x - \lfloor x \rfloor \\ \Leftrightarrow g \cdot h &= f \pmod{q} \end{aligned}$$

- ▶  $h = \lfloor q/z \rfloor$ ,  $f = -g\{q/z\}$
- ▶  $\|f\| \approx \|g\|$  small

This is an NTRU instance

## From ideal-SVP to $\text{NTRU}_{\text{vec}}$ (2)

Summing up: If  $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  and  $z$  known

- we can construct an NTRU instance  $h$  from  $I$ 
  - ▶ any short  $g \in I$  provides a trapdoor  $(f, g)$  for  $h$

## From ideal-SVP to $\text{NTRU}_{\text{vec}}$ (2)

Summing up: If  $I = \langle z \rangle = \{z \cdot r \mid r \in R\}$  and  $z$  known

- we can construct an NTRU instance  $h$  from  $I$ 
  - ▶ any short  $g \in I$  provides a trapdoor  $(f, g)$  for  $h$

What we need to conclude the reduction:

- any trapdoor  $(f', g')$  for  $h$  is such that  $g' \in I$ 
  - ▶  $g'$  solution to ideal-SVP in  $I$

# Extending the reduction

## Non principal ideals:

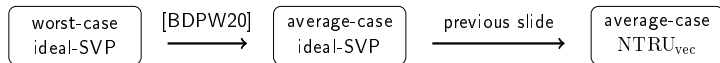
- $I = R \cap \langle z \rangle$  with  $z$  easily computed
  - ▶ everything still works with this  $z$

# Extending the reduction

## Non principal ideals:

- $I = R \cap \langle z \rangle$  with  $z$  easily computed
  - ▶ everything still works with this  $z$

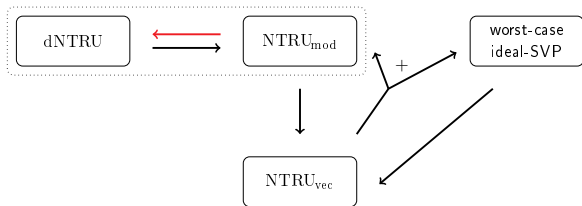
## Worst-case to average-case reduction:



---

[BDPW20] de Boer, Ducas, Pellet-Mary, and Wesolowski. Random Self-reducibility of Ideal-SVP via Arakelov Random Walks. Crypto.

# Techniques



## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$

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

**Can use an oracle:** given  $h \in R_q$ , the oracle outputs

- ▶ YES if  $h = f/g \bmod q$ , with  $f, g$  small ( $\leq B$ )
- ▶ NO otherwise



## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$

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

**Can use an oracle:** given  $h \in R_q$ , the oracle outputs

- ▶ YES if  $h = f/g \bmod q$ , with  $f, g$  small ( $\leq B$ )
- ▶ 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  $xf + yg$  is small or not

# From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$

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

**Can use an oracle:** given  $h \in R_q$ , the oracle outputs

- ▶ YES if  $h = f/g \bmod q$ , with  $f, g$  small ( $\leq B$ )
- ▶ 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  $xf + yg$  is small or not

$\Rightarrow$  we can choose  $x$  and  $y$

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

## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$ (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  $f/g$

## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$ (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)

## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$ (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  $x_1 f + y_1 g = B$

## From $\text{NTRU}_{\text{mod}}$ to $\text{dNTRU}$ (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  $f/g$

# More technical details

If the oracle is not perfect:

- ▶ need to handle distributions
- ▶ use the “oracle hidden center” framework [PRS17]

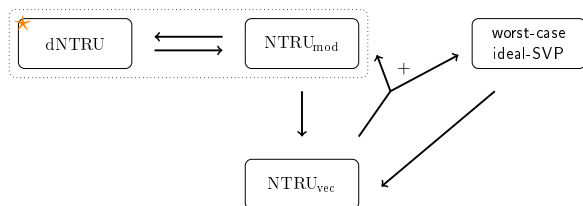
---

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

# Conclusion

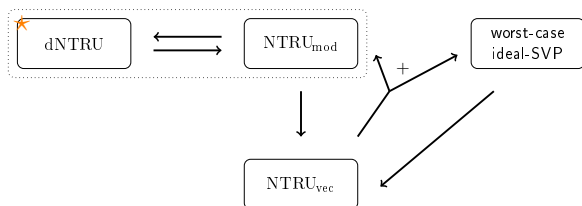


# Conclusion and open problems



- Can we make the distributions of the reductions match?
- Can we relate  $\text{NTRU}_{\text{mod}}$  and ideal-SVP?
  - ▶ maybe not since any “natural reduction” would provide new attacks
- Can we prove reduction from module problems with  $\text{rank} \geq 2$ ?

# Conclusion and open problems



- Can we make the distributions of the reductions match?
- Can we relate  $\text{NTRU}_{\text{mod}}$  and ideal-SVP?
  - ▶ maybe not since any “natural reduction” would provide new attacks
- Can we prove reduction from module problems with  $\text{rank} \geq 2$ ?

Thank you