Introduction to lattice-based cryptography

Part II

Alice Pellet-Mary

PEPR PQ-TLS summer school, Anglet

Plan

Part I

Lattice problems

- Lattices
- Hard problems

Algorithms for lattice problems

- LLL, sieving, BKZ

Constructions

- Public key encryption
- Signatures

Part II

NTRU and signature

- NTRU, Falcon

LWE and public key encryption

- LWE
- Regev's encryption, Kyber

Algebraic Lattices

- Ideal and module lattices
- NTRU, RLWE, mod-LWE

Reductions between problems

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

- at least some lattices are hard
- but this does not mean that all lattices are hard

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

- at least some lattices are hard
- but this does not mean that all lattices are hard

Analogy with factoring: factoring is hard

- for some integers $\rightsquigarrow N = pq$ with p, q prime is hard
- but not for all integers $\rightsquigarrow N = p$ with p prime is easy

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

- at least some lattices are hard
- but this does not mean that all lattices are hard

Analogy with factoring: factoring is hard

- for some integers $\rightsquigarrow N = pq$ with p, q prime is hard
- but not for all integers $\rightsquigarrow N = p$ with p prime is easy

Sage demo

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

- at least some lattices are hard
- but this does not mean that all lattices are hard

Analogy with factoring: factoring is hard

- for some integers $\rightsquigarrow N = pq$ with p, q prime is hard
- but not for all integers $\rightsquigarrow N = p$ with p prime is easy

Sage demo

Episode 2: a random hard lattice $\mathcal{L}$ (+ a short basis) $\Rightarrow$ signature and PKE

Summary of previous episodes

Episode 1: no efficient algorithm that computes short bases for all lattices

- at least some lattices are hard
- but this does not mean that all lattices are hard

Analogy with factoring: factoring is hard

- for some integers $\rightsquigarrow N = pq$ with p, q prime is hard
- but not for all integers $\rightsquigarrow N = p$ with p prime is easy

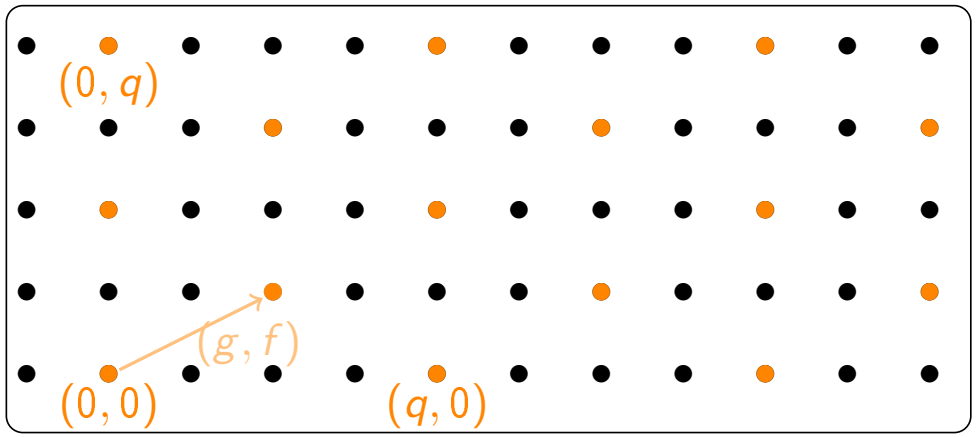
Sage demo

Episode 2: a random hard lattice $\mathcal{L}$ (+ a short basis) $\Rightarrow$ signature and PKE

How do we generate a hard lattice (+ a short basis)?

NTRU and Falcon's signature scheme

(one of the three 2022 NIST standard)



Digression: NIST standardization process

NIST competition for post-quantum key exchange and signatures:

https://csrc.nist.gov/projects/post-quantum-cryptography

Digression: NIST standardization process

NIST competition for post-quantum key exchange and signatures:

2016 NIST first call for proposals

https://csrc.nist.gov/projects/post-quantum-cryptography

Digression: NIST standardization process

NIST competition for post-quantum key exchange and signatures:

- 2016 NIST first call for proposals
- 2017 $\approx$ 70 submissions

https://csrc.nist.gov/projects/post-quantum-cryptography

Digression: NIST standardization process

NIST competition for post-quantum key exchange and signatures:

- 2016 NIST first call for proposals
- 2017 $\approx$ 70 submissions
- 2022 NIST announced the first 4 selected algorithms

  (some others are still being analyzed)
  - 1 key exchange: Kyber (lattices)
  - 3 signatures: Dilithium (lattices), Falcon (lattices), SPHINCS+ (hash)

https://csrc.nist.gov/projects/post-quantum-cryptography

Digression: NIST standardization process

NIST competition for post-quantum key exchange and signatures:

- 2016 NIST first call for proposals
- 2017 $\approx$ 70 submissions
- 2022 NIST announced the first 4 selected algorithms
  (some others are still being analyzed)
  - 1 key exchange: Kyber (lattices)
  - 3 signatures: Dilithium (lattices), Falcon (lattices), SPHINCS+ (hash)
- 2023 call for additional digital signatures ("not lattice-based")
  - 40 submissions, still under consideration

https://csrc.nist.gov/projects/post-quantum-cryptography

NTRU [HPS98]

Parameters:
- q prime and large (e.g., $q = 16411$)
- $B \ll \sqrt{q}$ integer (e.g., $B = 5$)

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

NTRU [HPS98]

Parameters:
- q prime and large (e.g., $q = 16411$)
- $B \ll \sqrt{q}$ integer (e.g., $B = 5$)

NTRU instance:
- sample f, g random integers in $\{-B, \dots, B\}$
  (e.g., $f = -2$, $g = 3$)
- return $h = f \cdot g^{-1} \bmod q$
  ($h = (-2) \times 3^{-1} \bmod 16411 = -5471$)

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

NTRU [HPS98]

Parameters:
- q prime and large (e.g., $q = 16411$)
- $B \ll \sqrt{q}$ integer (e.g., $B = 5$)

NTRU instance:
- sample f, g random integers in $\{-B, \dots, B\}$
  (e.g., $f = -2$, $g = 3$)
- return $h = f \cdot g^{-1} \bmod q$
  ($h = (-2) \times 3^{-1} \bmod 16411 = -5471$)

NTRU assumption: given h, there is no efficient algorithm that can find u and v such that $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

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

NTRU [HPS98]

Parameters:
- q prime and large (e.g., $q = 16411$)
- $B \ll \sqrt{q}$ integer (e.g., $B = 5$)

NTRU instance:
- sample f, g random integers in $\{-B, \dots, B\}$ (e.g., $f = -2$, $g = 3$)
- return $h = f \cdot g^{-1} \bmod q$ ($h = (-2) \times 3^{-1} \bmod 16411 = -5471$)

NTRU assumption: given h, there is no efficient algorithm that can find u and v such that $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

Wait... NTRU assumption obviously does not hold (we can test all the small (u, v))

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

NTRU [HPS98]

Parameters:
- q prime and large (e.g., $q = 16411$)
- $B \ll \sqrt{q}$ integer (e.g., $B = 5$)

NTRU instance:
- sample f, g random integers in $\{-B, \dots, B\}$ (e.g., $f = -2$, $g = 3$)
- return $h = f \cdot g^{-1} \bmod q$ ($h = (-2) \times 3^{-1} \bmod 16411 = -5471$)

NTRU assumption: given h, there is no efficient algorithm that can find u and v such that $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

Wait... NTRU assumption obviously does not hold (we can test all the small (u, v))

- we should replace integers by polynomials (degree 512 or 1024)
- for the moment, let's keep integers (and pretend it is hard)

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

Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Let $h = f \cdot g^{-1} \bmod q$ an NTRU instance.

$$\mathrm{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{and} \quad \mathcal{L}_h = \mathcal{L}(\mathrm{B}_h) \text{ (spanned by the columns of } \mathrm{B}_h)$$

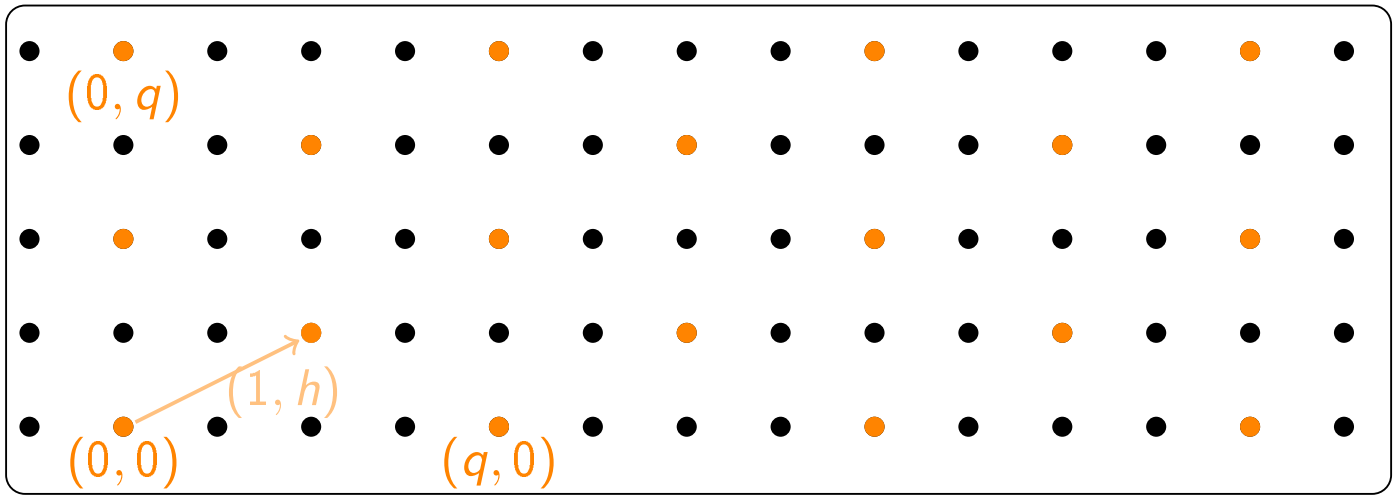
Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Let $h = f \cdot g^{-1} \bmod q$ an NTRU instance.

$$\mathrm{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{and} \quad \mathcal{L}_h = \mathcal{L}(\mathrm{B}_h) \quad \text{(spanned by the columns of } \mathrm{B}_h\text{)}$$



Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Let $h = f \cdot g^{-1} \bmod q$ an NTRU instance.

$$\mathrm{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{and} \quad \mathcal{L}_h = \mathcal{L}(\mathrm{B}_h) \quad \text{(spanned by the columns of } \mathrm{B}_h\text{)}$$

Property: $\begin{pmatrix} u \\ v \end{pmatrix} \in \mathcal{L}_h \iff h = v \cdot u^{-1} \bmod q$ (or $u = v = 0 \bmod q$)

Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Let $h = f \cdot g^{-1} \bmod q$ an NTRU instance.

$$\mathrm{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{and} \quad \mathcal{L}_h = \mathcal{L}(\mathrm{B}_h) \text{ (spanned by the columns of } \mathrm{B}_h)$$

Property: $\begin{pmatrix} u \\ v \end{pmatrix} \in \mathcal{L}_h \iff h = v \cdot u^{-1} \bmod q$ (or $u = v = 0 \bmod q$)

Finding a short vector in $\mathcal{L}_h$ $\implies$ solving NTRU

Hard lattice from NTRU assumption

NTRU assumption: given $h = f \cdot g^{-1} \bmod q$ ($|f|, |g| \leq B$), there is no efficient algorithm that can find (u, v) with $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

How do we get a hard lattice from this?

Let $h = f \cdot g^{-1} \bmod q$ an NTRU instance.

$$\mathrm{B}_h = \begin{pmatrix} 1 & 0 \\ h & q \end{pmatrix} \quad \text{and} \quad \mathcal{L}_h = \mathcal{L}(\mathrm{B}_h) \quad \text{(spanned by the columns of } \mathrm{B}_h\text{)}$$

Property: $\begin{pmatrix} u \\ v \end{pmatrix} \in \mathcal{L}_h \iff h = v \cdot u^{-1} \bmod q$ (or $u = v = 0 \bmod q$)

Finding a short vector in $\mathcal{L}_h$ $\implies$ solving NTRU

no adversary can compute a short basis of $\mathcal{L}_h$ $\impliedby$ NTRU assumption holds

NTRU: wrapping up

We have seen: under the NTRU assumption, the following algorithm produces random hard lattices.

- sample random f, g in $\{-B, \dots, B\}$
- compute $h = f \cdot g^{-1} \bmod q$
- return $\mathcal{L}_h$, spanned by $\begin{pmatrix} 1 \\ h \end{pmatrix}$ and $\begin{pmatrix} 0 \\ q \end{pmatrix}$

NTRU: wrapping up

We have seen: under the NTRU assumption, the following algorithm produces random hard lattices.

- sample random f, g in $\{-B, \dots, B\}$
- compute $h = f \cdot g^{-1} \bmod q$
- return $\mathcal{L}_h$, spanned by $\begin{pmatrix} 1 \\ h \end{pmatrix}$ and $\begin{pmatrix} 0 \\ q \end{pmatrix}$

Wait... $\mathcal{L}_h$ has dimension 2, how can it be hard?

NTRU: wrapping up

We have seen: under the NTRU assumption, the following algorithm produces random hard lattices.

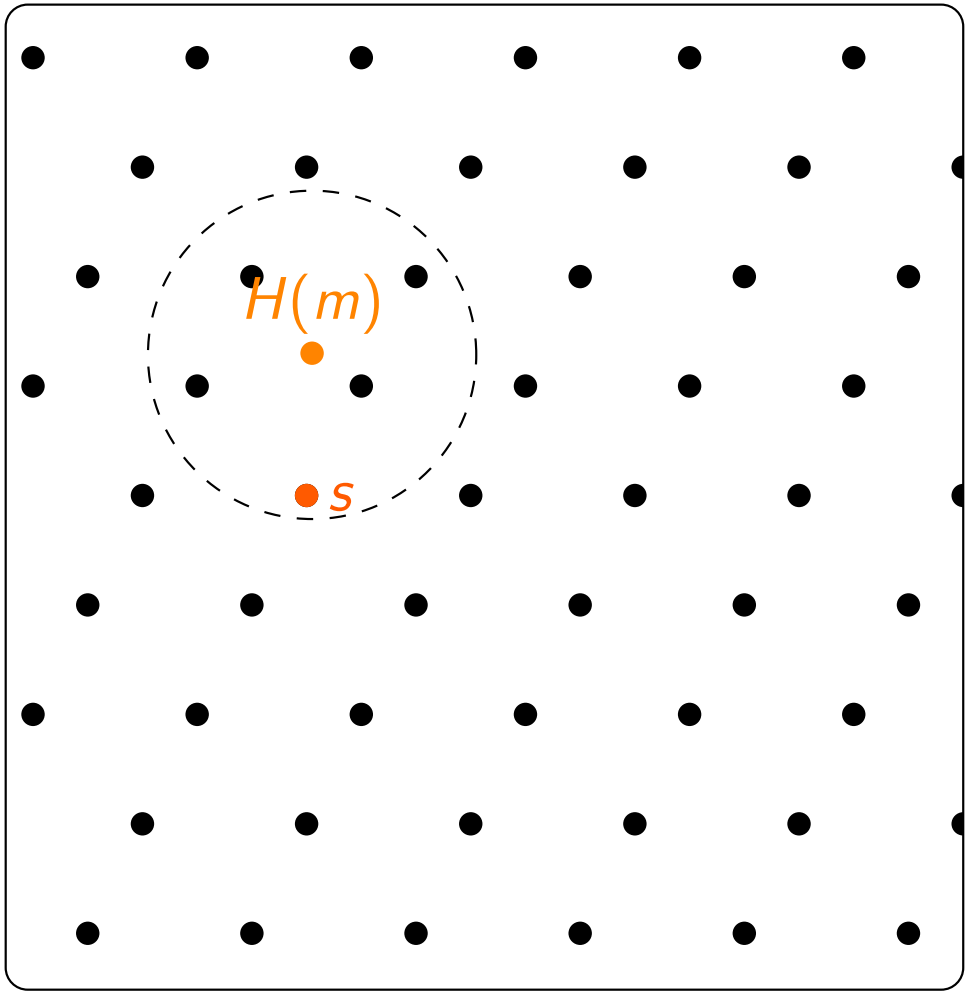
- sample random f, g in $\{-B, \dots, B\}$
- compute $h = f \cdot g^{-1} \bmod q$
- return $\mathcal{L}_h$, spanned by $\begin{pmatrix} 1 \\ h \end{pmatrix}$ and $\begin{pmatrix} 0 \\ q \end{pmatrix}$

Wait... $\mathcal{L}_h$ has dimension 2, how can it be hard?

This is because we work with integers instead of polynomials (for simplicity)

$\Rightarrow$ if f, g and h are polynomials of degree d, $\mathcal{L}_h$ has dimension $2d$ instead of 2

Remember: hash-and-sign signatures



KeyGen:

- pk = bad basis of $\mathcal{L}$
- sk = short basis of $\mathcal{L}$

Sign(m, sk):

- $x = H(m)$ (hash the message)
- sample $s \in \mathcal{L} \cap \mathcal{B}_r(x)$ (small radius r)

Verify(m, s, pk):

- check that $s \in \mathcal{L}$
- check that $H(m) - s$ is small

Lemma: if an adversary can forge signatures, then she can recover a short basis of $\mathcal{L}$ using only pk (in the ROM)

Falcon signature scheme

Falcon signature:

Hash-and-sign signature

+

hard lattice from NTRU assumption

with polynomials of degree $d = 512$ or 1024 instead of integers

Falcon is one of the three post-quantum signature scheme standardized by the NIST in 2022.

[Falcon] Fouque, Hoffstein, Kirchner, Lyubashevsky, Pornin, Prest, Ricosset, Seiler, Whyte, Zhang. NIST standard

Falcon's performances

			Post-quantum standards (NIST 2022)		
	RSA	EdDSA	**Falcon**	Dilithium	SPHINCS+
Public key size (bytes)	256	32	**897**	1312	32
Signature size (bytes)	256	64	**666**	2420	17088
Signature time (µs)	665	51	**241**	208	35584
Verification time (µs)	19	142	**52**	74	2091

- Comparison for $\approx$ 128-bits security
- Timings for a processor of 1.6 GHz
  - RSA and EdDSA are obtained experimentally with openssl on my laptop
  - Falcon, Dilithium and SPHINCS+ are obtained from the number of cycles provided on their websites (assuming a 1.6 GHz processor)
- for SPHINCS+, this is the variant using SHAKE hash function

Pros and cons of Falcon

Falcon signature:

- post-quantum security
- fast (comparable to RSA/EdDSA)
- relatively compact ($\approx$ 3 times RSA, $\approx$ 15 times EdDSA)

Pros and cons of Falcon

Falcon signature:

- post-quantum security
- fast (comparable to RSA/EdDSA)
- relatively compact ($\approx$ 3 times RSA, $\approx$ 15 times EdDSA)

But there are still important open questions for actual deployment

Pros and cons of Falcon

Falcon signature:

- post-quantum security
- fast (comparable to RSA/EdDSA)
- relatively compact ($\approx$ 3 times RSA, $\approx$ 15 times EdDSA)

But there are still important open questions for actual deployment

- requires floating point
  - not all devices have floating point units

Pros and cons of Falcon

Falcon signature:

- post-quantum security
- fast (comparable to RSA/EdDSA)
- relatively compact ($\approx$ 3 times RSA, $\approx$ 15 times EdDSA)

But there are still important open questions for actual deployment

- requires floating point
  - not all devices have floating point units
- difficult to mask
  - can be subject to certain side-channel attacks
  - see Raccoon if you want a masking-friendly lattice-based signature

[Raccoon] del Pino, Espitau, Katsumata, Maller, Mouhartem, Prest, Rossi, Saarinen. Technical report, NIST.

Pros and cons of Falcon

Falcon signature:

- post-quantum security
- fast (comparable to RSA/EdDSA)
- relatively compact ($\approx$ 3 times RSA, $\approx$ 15 times EdDSA)

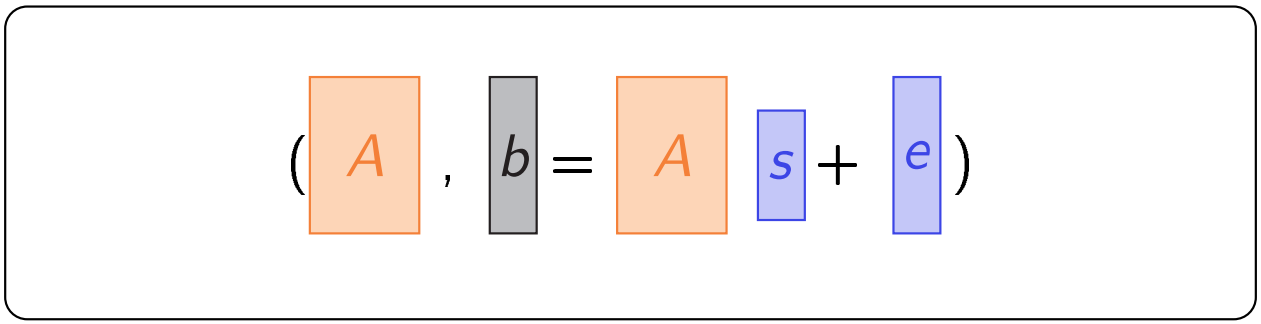
But there are still important open questions for actual deployment

- requires floating point
  - not all devices have floating point units
- difficult to mask
  - can be subject to certain side-channel attacks
  - see Raccoon if you want a masking-friendly lattice-based signature
- KeyGen and Sign are complex to implement
  - can lead to bugs in implementation

[Raccoon] del Pino, Espitau, Katsumata, Maller, Mouhartem, Prest, Rossi, Saarinen. Technical report, NIST.

LWE and Regev's encryption scheme

(and a little bit of Kyber, another 2022 NIST standard)



The LWE problem [Reg05]

Notations: q, B integers, $1 \leq B \ll q$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

LWE (Learning With Errors) [Reg05]

Sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{m \times n})$ and $s, e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e

[Reg05] Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC.

The LWE problem [Reg05]

Notations: q, B integers, $1 \leq B \ll q$, $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$

LWE (Learning With Errors) [Reg05]

Sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{m \times n})$ and $s, e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e

LWE assumption: there is no efficient algorithm solving LWE

[Reg05] Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC.

LWE is a lattice problem

LWE (Learning With Errors)

Sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e

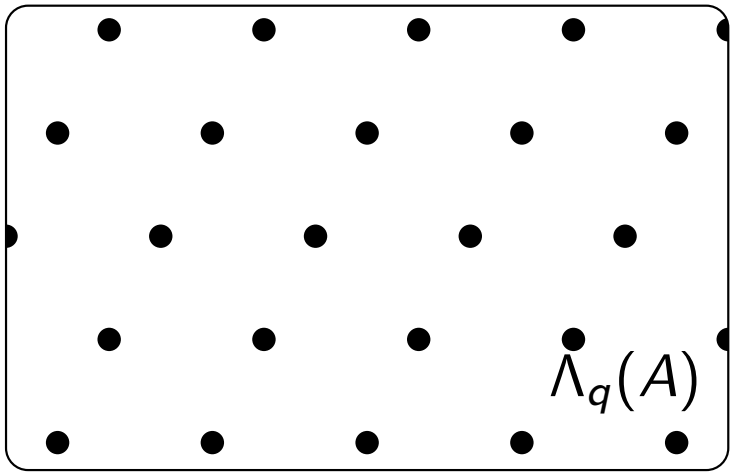
LWE is a lattice problem

LWE (Learning With Errors)

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e



$$\Lambda_q(A) = \{x \in \mathbb{Z}^m \mid \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

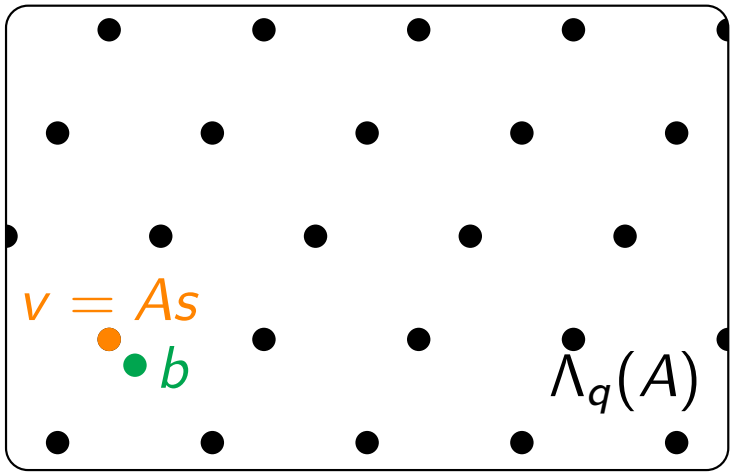
LWE is a lattice problem

LWE (Learning With Errors)

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e



$$\Lambda_q(A) = \{x \in \mathbb{Z}^m \mid \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

$$b = v + e,$$
where $v \in \Lambda_q(A)$ and e small

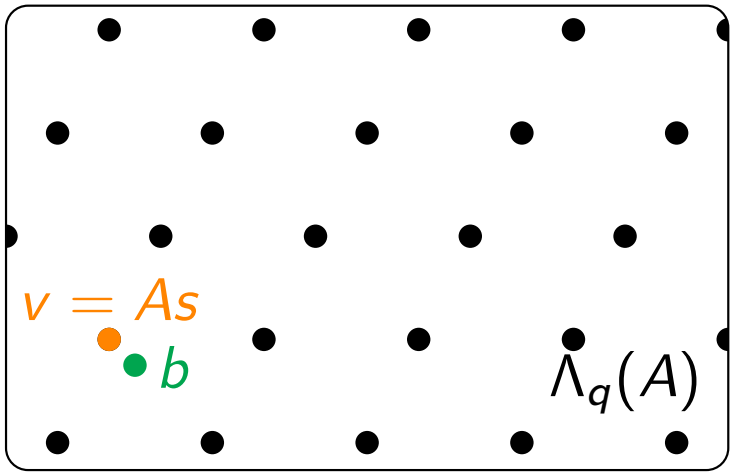
LWE is a lattice problem

LWE (Learning With Errors)

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where $b := As + e \bmod q$

Recover s or e



$$\Lambda_q(A) = \{x \in \mathbb{Z}^m \mid \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

$$b = v + e,$$
where $v \in \Lambda_q(A)$ and e small

LWE $\approx$ CVP in $\Lambda_q(A)$

Decision variant of LWE

decision-LWE

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times m})$ and $s, e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where

$$b := A\,s + e \bmod q \quad \text{or} \quad b \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^n)$$

Guess whether b is uniform or not.

Decision variant of LWE

decision-LWE

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times m})$ and s, $e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where

$$b := A\,s + e \bmod q \quad \text{or} \quad b \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^n)$$

Guess whether b is uniform or not.

$$\text{decision LWE} \overset{\sim}{\Longleftrightarrow} \text{(search) LWE}$$

Decision variant of LWE

decision-LWE

Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times m})$ and s, $e \leftarrow \mathrm{Uniform}(\{-B, \cdots, B\}^n)$

Given A and b, where

$$b := As + e \bmod q \quad \text{or} \quad b \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^n)$$

Guess whether b is uniform or not.

decision LWE $\overset{\sim}{\Longleftrightarrow}$ (search) LWE

$\Rightarrow$ decision problems can be easier to use for crypto

Regev-like encryption scheme [LPS10,LP11]

Parameters: $n, q, B \in \mathbb{Z}_{>0}$ (notation: U_B uniform distribution over $\{-B, \cdots, B\}$)

[LPS10] Lyubashevsky, Palacio, Segev. Public-key cryptographic primitives provably as secure as subset sum. TCC.

[LP11] Lindner and Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA.

Regev-like encryption scheme [LPS10,LP11]

Parameters: $n, q, B \in \mathbb{Z}_{>0}$ (notation: U_B uniform distribution over $\{-B, \cdots, B\}$)

KeyGen: Sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow U_B^n$

Return $pk = (A, b = As + e \bmod q)$ and $sk = s$

[LPS10] Lyubashevsky, Palacio, Segev. Public-key cryptographic primitives provably as secure as subset sum. TCC.

[LP11] Lindner and Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA.

Regev-like encryption scheme [LPS10,LP11]

Parameters: $n, q, B \in \mathbb{Z}_{>0}$ (notation: U_B uniform distribution over $\{-B, \cdots, B\}$)

KeyGen: Sample $A \leftarrow \mathrm{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow U_B^n$

Return $pk = (A, b = As + e \bmod q)$ and $sk = s$

Encrypt: message $m \in \{0, 1\}$

sample $\tilde{s}^T, \tilde{e}^T \leftarrow U_B^n$ and $e' \leftarrow U_B$

return $c = (\tilde{s}^T A + \tilde{e}^T, \tilde{s}^T b + e' + m \cdot \lfloor q/2 \rceil)$

(all $\bmod q$)

[LPS10] Lyubashevsky, Palacio, Segev. Public-key cryptographic primitives provably as secure as subset sum. TCC.

[LP11] Lindner and Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA.

Regev-like encryption scheme [LPS10,LP11]

Parameters: $n, q, B \in \mathbb{Z}_{>0}$ (notation: U_B uniform distribution over $\{-B, \cdots, B\}$)

KeyGen: Sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n \times n})$ and $s, e \leftarrow U_B^n$

Return $pk = (A, b = As + e \bmod q)$ and $sk = s$

Encrypt: message $m \in \{0, 1\}$

sample $\tilde{s}^T, \tilde{e}^T \leftarrow U_B^n$ and $e' \leftarrow U_B$

return $c = (\tilde{s}^T A + \tilde{e}^T, \tilde{s}^T b + e' + m \cdot \lfloor q/2 \rceil)$

(all mod q)

Decrypt: $c = (c_1{}^T, c_2)$

compute $r = c_1{}^T s - c_2 \bmod q$ ($r \in [0, q]$)

return 1 if r is in $[q/4, 3q/4]$ and 0 otherwise

[LPS10] Lyubashevsky, Palacio, Segev. Public-key cryptographic primitives provably as secure as subset sum. TCC.

[LP11] Lindner and Peikert. Better key sizes (and attacks) for LWE-based encryption. CT-RSA.

Correctness

Theorem

If $q \geq 8 \cdot n \cdot B^2 + 4 \cdot B$, then the scheme is correct.

Correctness: for any message m and any $(pk, sk) \leftarrow \mathsf{KeyGen}$, it holds that

$$\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m.$$

Correctness

Theorem

If $q \geq 8 \cdot n \cdot B^2 + 4 \cdot B$, then the scheme is correct.

Correctness: for any message m and any $(pk, sk) \leftarrow \mathsf{KeyGen}$, it holds that

$$\mathsf{Dec}(sk, \mathsf{Enc}(pk, m)) = m.$$

Proof: on the board

Correctness

Theorem

If $q \geq 8 \cdot n \cdot B^2 + 4 \cdot B$, then the scheme is correct.

Correctness: for any message m and any $(pk, sk) \leftarrow \mathrm{KeyGen}$, it holds that

$$\mathrm{Dec}(sk, \mathrm{Enc}(pk, m)) = m.$$

Proof: on the board

Decryption failures: if we replace U_B by a Gaussian distribution, the scheme might fail with very small probability (s and e might have coefficients $\geq B$)

Security: high level

Public information:

A

$b = As + e$

$c_1^T = \tilde{s}^T A + \tilde{e}^T$

$c_2 = \tilde{s}^T b + e' + m \cdot \lfloor q/2 \rceil$

Security: high level

Public information:

A

$b = As + e$

$c_1^T = \tilde{s}^T A + \tilde{e}^T$

$c_2 = \tilde{s}^T b + e' + m \cdot \lfloor q/2 \rceil$

Decision-LWE: $b \approx b$

Security: high level

Public information:

A

$b = A\,s + e$

$c_1^T = \tilde{s}^T\,A + \tilde{e}^T$

$c_2 = \tilde{s}^T\,b + e' + m \cdot \lfloor q/2 \rceil$

Decision-LWE: $b \approx b$

Security: high level

Public information:

$$A$$

$$b = A\,s + e$$

$$c_1^T = \tilde{s}^T\,A + \tilde{e}^T$$

$$c_2 = \tilde{s}^T\,b + e' + m \cdot \lfloor q/2 \rceil$$

Decision-LWE: $b \approx b$

Decision-LWE:

$$\tilde{s}^T \,|\, A \,|\, b + \tilde{e}^T \,|\, e' \approx c_1^T \,|\, c$$

Security: high level

Public information:

A

$b = A\,s + e$

$c_1^T = \tilde{s}^T\,A + \tilde{e}^T$

$c_2 = c + m \cdot \lfloor q/2 \rceil$

Decision-LWE: $b \approx b$

Decision-LWE:

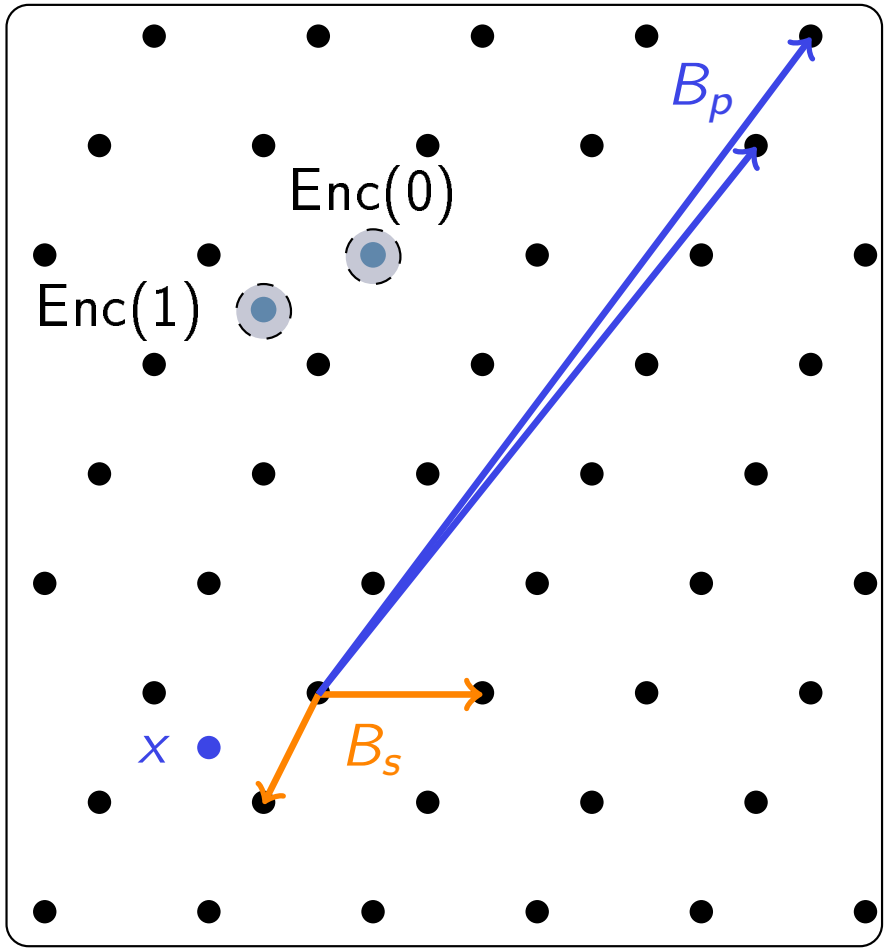
$\tilde{s}^T\,|\,A\,|\,b + \tilde{e}^T\,|\,e' \approx c_1^T\,|\,c$

What is the relation between

KeyGen:
- $sk = s$
- $pk = (A, b = As + e)$

Encrypt: $c = (c_1^T, c_2)$ with
- $c_1^T = \tilde{s}^T A + \tilde{e}^T$
- $c_2 = \tilde{s}^T b + e' + m \cdot \lfloor q/2 \rceil$

Decrypt:
- compute $r = c_1^T s - c_2 \bmod q$
- return $\begin{cases} 1 & \text{if } r \in [q/4, 3q/4] \\ 0 & \text{otherwise} \end{cases}$



Kyber

Kyber key exchange:

Regev-like public key encryption

(in its KEM version)

+

module variant of LWE

(see next part)

Kyber is the only post-quantum key encapsulation mechanism standardized by the NIST in 2022.

[Kyber] Avanzi, Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, Schwabe, Seiler, Stehlé. NIST standard

Summary on NTRU and LWE

NTRU and LWE are average-case problems

Summary on NTRU and LWE

NTRU and LWE are average-case problems

$\Rightarrow$ Good for crypto

(negligible probability to sample a weak key)

Summary on NTRU and LWE

NTRU and LWE are average-case problems
$\Rightarrow$ Good for crypto
(negligible probability to sample a weak key)

NTRU: random hard instance of the short basis problem

LWE: random hard instance of the close vector problem

Summary on NTRU and LWE

NTRU and LWE are average-case problems
$\Rightarrow$ Good for crypto
(negligible probability to sample a weak key)

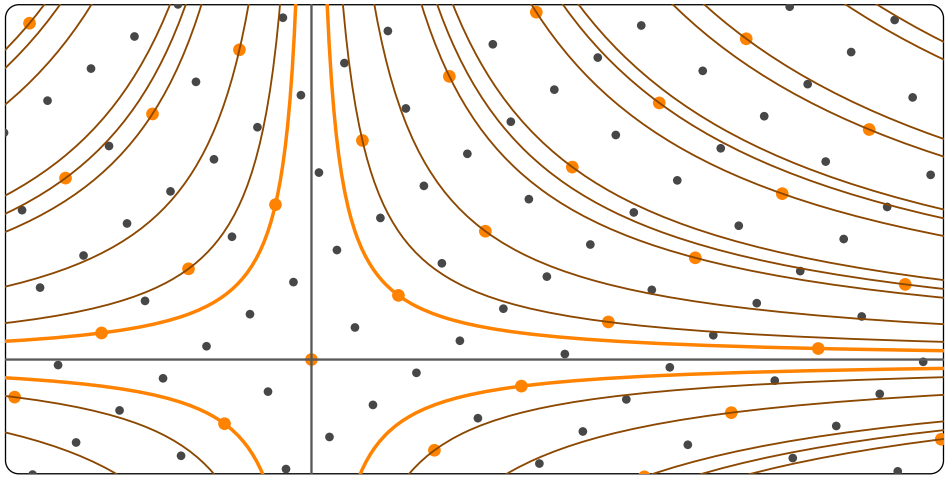
NTRU: random hard instance of the short basis problem

LWE: random hard instance of the close vector problem

We have seen:
- signature based on NTRU
- encryption based on LWE

We can also do:
- signature based on LWE
- encryption based on NTRU

Algebraic lattices



Number fields

Number field: $K = \mathbb{Q}[X]/P(X)$ (P irreducible, $\deg(P) = d$)

Number fields

Number field: $K = \mathbb{Q}[X]/P(X)$ (P irreducible, $\deg(P) = d$)

- $K = \mathbb{Q}$
- $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^\ell \rightsquigarrow$ power-of-two cyclotomic field
- $K = \mathbb{Q}[X]/(X^d - X - 1)$ with d prime $\rightsquigarrow$ NTRUPrime field

Number fields

Number field: $K = \mathbb{Q}[X]/P(X)$ (P irreducible, $\deg(P) = d$)

- $K = \mathbb{Q}$
- $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^\ell \rightsquigarrow$ power-of-two cyclotomic field
- $K = \mathbb{Q}[X]/(X^d - X - 1)$ with d prime $\rightsquigarrow$ NTRUPrime field

Ring of integers: $\mathcal{O}_K \subset K$, for this talk $\mathcal{O}_K = \mathbb{Z}[X]/P(X)$

(more generally $\mathbb{Z}[X]/P(X) \subseteq \mathcal{O}_K$ but $\mathcal{O}_K$ can be larger)

Number fields

Number field: $K = \mathbb{Q}[X]/P(X)$ (P irreducible, $\deg(P) = d$)

- $K = \mathbb{Q}$
- $K = \mathbb{Q}[X]/(X^d + 1)$ with $d = 2^\ell \rightsquigarrow$ power-of-two cyclotomic field
- $K = \mathbb{Q}[X]/(X^d - X - 1)$ with d prime $\rightsquigarrow$ NTRUPrime field

Ring of integers: $\mathcal{O}_K \subset K$, for this talk $\mathcal{O}_K = \mathbb{Z}[X]/P(X)$

(more generally $\mathbb{Z}[X]/P(X) \subseteq \mathcal{O}_K$ but $\mathcal{O}_K$ can be larger)

- $\mathcal{O}_K = \mathbb{Z}$
- $\mathcal{O}_K = \mathbb{Z}[X]/(X^d + 1)$ with $d = 2^\ell \rightsquigarrow$ power-of-two cyclotomic ring
- $\mathcal{O}_K = \mathbb{Z}[X]/(X^d - X - 1)$ with d prime $\rightsquigarrow$ NTRUPrime ring of integers

Embeddings

$(K = \mathbb{Q}[X]/P(X), \quad \alpha_1, \cdots, \alpha_d$ complex roots of $P(X))$

Coefficient embedding: $\Sigma : \begin{array}{rcl} K & \to & \mathbb{R}^d \\ \sum_{i=0}^{d-1} y_i X^i & \mapsto & (y_0, \cdots, y_{d-1}) \end{array}$

Canonical embedding: $\sigma : \begin{array}{rcl} K & \to & \mathbb{C}^d \\ y(X) & \mapsto & (y(\alpha_1), \cdots, y(\alpha_d)) \end{array}$

Embeddings

$(K = \mathbb{Q}[X]/P(X), \quad \alpha_1, \cdots, \alpha_d$ complex roots of $P(X))$

Coefficient embedding: $\Sigma : \begin{array}{rcl} K & \to & \mathbb{R}^d \\ \sum_{i=0}^{d-1} y_i X^i & \mapsto & (y_0, \cdots, y_{d-1}) \end{array}$

Canonical embedding: $\sigma : \begin{array}{rcl} K & \to & \mathbb{C}^d \\ y(X) & \mapsto & (y(\alpha_1), \cdots, y(\alpha_d)) \end{array}$

- both embeddings induce a (different) geometry on K

Embeddings

$(K = \mathbb{Q}[X]/P(X), \quad \alpha_1, \cdots, \alpha_d$ complex roots of $P(X))$

Coefficient embedding: $\Sigma :$

$$\begin{array}{rcl} K & \to & \mathbb{R}^d \\ \sum_{i=0}^{d-1} y_i X^i & \mapsto & (y_0, \cdots, y_{d-1}) \end{array}$$

Canonical embedding: $\sigma :$

$$\begin{array}{rcl} K & \to & \mathbb{C}^d \\ y(X) & \mapsto & (y(\alpha_1), \cdots, y(\alpha_d)) \end{array}$$

- both embeddings induce a (different) geometry on K

Which embedding should we choose?

- coefficient embedding is used for constructions (efficient implementation)
- canonical embedding is used in cryptanalysis / reductions (nice mathematical properties)

Embeddings

$(K = \mathbb{Q}[X]/P(X), \quad \alpha_1, \cdots, \alpha_d$ complex roots of $P(X))$

Coefficient embedding: $\Sigma : \begin{array}{rcl} K & \to & \mathbb{R}^d \\ \sum_{i=0}^{d-1} y_i X^i & \mapsto & (y_0, \cdots, y_{d-1}) \end{array}$

Canonical embedding: $\sigma : \begin{array}{rcl} K & \to & \mathbb{C}^d \\ y(X) & \mapsto & (y(\alpha_1), \cdots, y(\alpha_d)) \end{array}$

- both embeddings induce a (different) geometry on K

Which embedding should we choose?

- coefficient embedding is used for constructions (efficient implementation)
- canonical embedding is used in cryptanalysis / reductions (nice mathematical properties)
- for fields used in crypto, both geometries are $\approx$ the same

Ideals

Ideal: $I \subseteq \mathcal{O}_K$ is an ideal if

- $x + y \in I$ for all $x, y \in I$
- $a \cdot x \in I$ for all $a \in \mathcal{O}_K$ and $x \in I$

Ideals

Ideal: $I \subseteq \mathcal{O}_K$ is an ideal if

- $x + y \in I$ for all $x, y \in I$
- $a \cdot x \in I$ for all $a \in \mathcal{O}_K$ and $x \in I$

- $I_1 = \{2a \,|\, a \in \mathbb{Z}\}$ and $J_1 = \{6a \,|\, a \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}$
- $I_2 = \{a + b \cdot X \,|\, a + b = 0 \bmod 2,\ a, b \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}[X]/(X^2 + 1)$

Ideals

Ideal: $I \subseteq \mathcal{O}_K$ is an ideal if

- $x + y \in I$ for all $x, y \in I$
- $a \cdot x \in I$ for all $a \in \mathcal{O}_K$ and $x \in I$

- $I_1 = \{2a \,|\, a \in \mathbb{Z}\}$ and $J_1 = \{6a \,|\, a \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}$
- $I_2 = \{a + b \cdot X \,|\, a + b = 0 \bmod 2,\ a, b \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}[X]/(X^2 + 1)$

Principal ideals: $\langle g \rangle := \{g \cdot a \mid a \in O_K\}$

Ideals

Ideal: $I \subseteq \mathcal{O}_K$ is an ideal if

- $x + y \in I$ for all $x, y \in I$
- $a \cdot x \in I$ for all $a \in \mathcal{O}_K$ and $x \in I$

- $I_1 = \{2a \mid a \in \mathbb{Z}\}$ and $J_1 = \{6a \mid a \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}$
- $I_2 = \{a + b \cdot X \mid a + b = 0 \bmod 2,\ a, b \in \mathbb{Z}\}$ in $\mathcal{O}_K = \mathbb{Z}[X]/(X^2 + 1)$

Principal ideals: $\langle g \rangle := \{g \cdot a \mid a \in O_K\}$

- $I_1 = \{2a \mid a \in \mathbb{Z}\} = \langle 2 \rangle$
- $I_2 = \{a + b \cdot X \mid a + b = 0 \bmod 2,\ a, b \in \mathbb{Z}\} = \langle 1 + X \rangle$

Ideal lattices

$\mathcal{O}_K$ is a lattice:

- $\mathcal{O}_K = 1 \cdot \mathbb{Z} + X \cdot \mathbb{Z} + \cdots + X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\mathcal{O}_K) = \Sigma(1) \cdot \mathbb{Z} + \cdots + \Sigma(X^{d-1}) \cdot \mathbb{Z}$

Ideal lattices

$\mathcal{O}_K$ is a lattice:

- $\mathcal{O}_K = 1 \cdot \mathbb{Z} + X \cdot \mathbb{Z} + \cdots + X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\mathcal{O}_K) = \Sigma(1) \cdot \mathbb{Z} + \cdots + \Sigma(X^{d-1}) \cdot \mathbb{Z}$

$\Sigma(\mathcal{O}_K)$ is a lattice of rank d in $\mathbb{Z}^d$ with basis $(\Sigma(X^i))_{0 \leq i < d}$

Ideal lattices

$\mathcal{O}_K$ is a lattice:

- $\mathcal{O}_K = 1 \cdot \mathbb{Z} + X \cdot \mathbb{Z} + \cdots + X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\mathcal{O}_K) = \Sigma(1) \cdot \mathbb{Z} + \cdots + \Sigma(X^{d-1}) \cdot \mathbb{Z}$

$\Sigma(\mathcal{O}_K)$ is a lattice of rank d in $\mathbb{Z}^d$ with basis $(\Sigma(X^i))_{0 \leq i < d}$

$\langle g \rangle$ is a lattice:

- $\langle g \rangle = g \cdot \mathcal{O}_K = g \cdot 1 \cdot \mathbb{Z} + g \cdot X \cdot \mathbb{Z} + \cdots + g \cdot X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\langle g \rangle) = \Sigma(g) \cdot \mathbb{Z} + \cdots + \Sigma(g \cdot X^{d-1}) \cdot \mathbb{Z}$

Ideal lattices

$\mathcal{O}_K$ is a lattice:

- $\mathcal{O}_K = 1 \cdot \mathbb{Z} + X \cdot \mathbb{Z} + \cdots + X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\mathcal{O}_K) = \Sigma(1) \cdot \mathbb{Z} + \cdots + \Sigma(X^{d-1}) \cdot \mathbb{Z}$

$\Sigma(\mathcal{O}_K)$ is a lattice of rank d in $\mathbb{Z}^d$ with basis $(\Sigma(X^i))_{0 \leq i < d}$

$\langle g \rangle$ is a lattice:

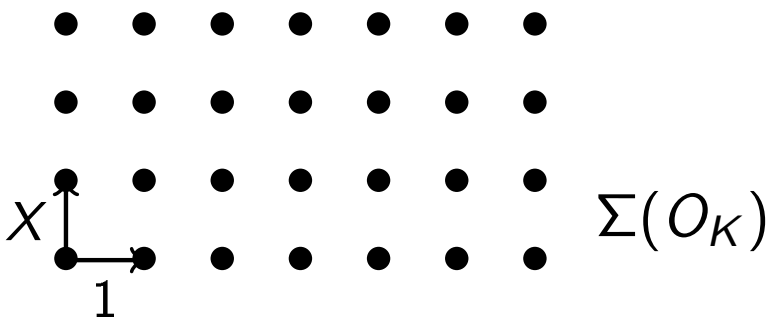
- $\langle g \rangle = g \cdot \mathcal{O}_K = g \cdot 1 \cdot \mathbb{Z} + g \cdot X \cdot \mathbb{Z} + \cdots + g \cdot X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\langle g \rangle) = \Sigma(g) \cdot \mathbb{Z} + \cdots + \Sigma(g \cdot X^{d-1}) \cdot \mathbb{Z}$

$\Sigma(\langle g \rangle)$ is a lattice of rank d in $\mathbb{Z}^d$ with basis $(\Sigma(g \cdot X^i))_{0 \leq i < d}$

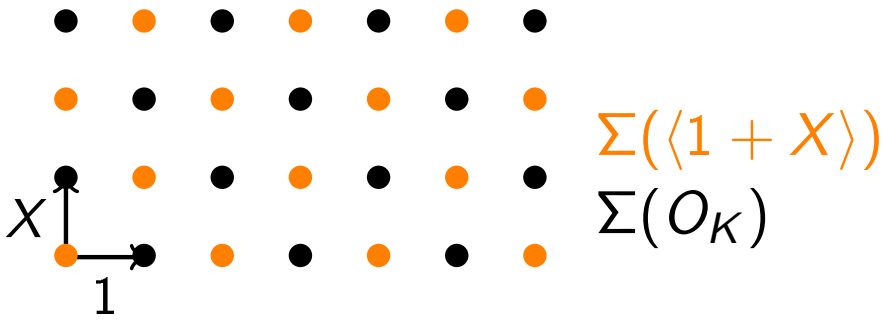
(this is also true for non principal ideals)

(we can replace Σ by σ and $\mathbb{Z}^d$ by $\mathbb{C}^d$)

Ideal lattices (2)

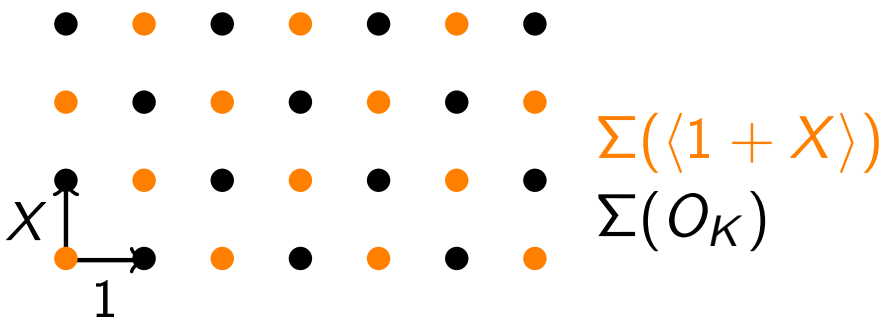


Ideal lattices (2)

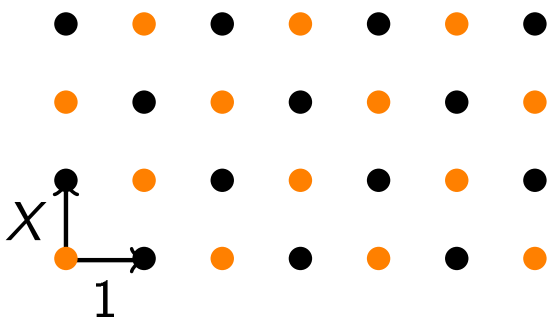


Ideal lattices (2)

Basis of $\langle g \rangle$: $g, g \cdot X, \cdots, g \cdot X^{d-1}$



Ideal lattices (2)



$\Sigma(\langle 1 + X \rangle)$

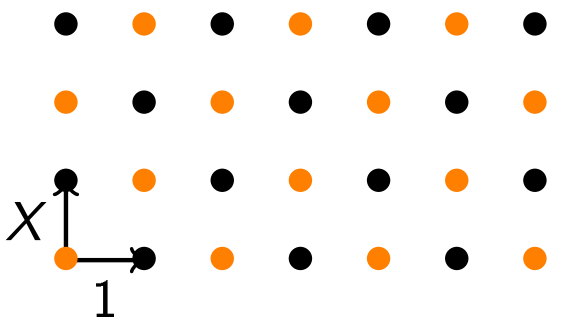
$\Sigma(O_K)$

Basis of $\langle g \rangle$: $g, g \cdot X, \cdots, g \cdot X^{d-1}$

$$\begin{pmatrix} g_0 & & \\ g_1 & & \\ \vdots & & \\ g_{d-1} & & \end{pmatrix}$$

(in $K = \mathbb{Q}[X]/X^d + 1$)

Ideal lattices (2)



$\Sigma(\langle 1 + X \rangle)$

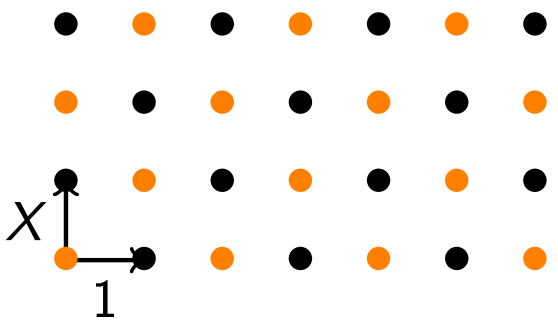
$\Sigma(O_K)$

Basis of $\langle g \rangle$: $g, g \cdot X, \cdots, g \cdot X^{d-1}$

$$\begin{pmatrix} g_0 & -g_{d-1} \\ g_1 & g_0 \\ \vdots & \vdots \\ g_{d-1} & g_{d-2} \end{pmatrix}$$

(in $K = \mathbb{Q}[X]/X^d + 1$)

Ideal lattices (2)



$\Sigma(\langle 1 + X \rangle)$

$\Sigma(O_K)$

Basis of $\langle g \rangle$: $g, g \cdot X, \cdots, g \cdot X^{d-1}$

$$\begin{pmatrix} g_0 & -g_{d-1} & \cdots & -g_1 \\ g_1 & g_0 & \cdots & -g_2 \\ \vdots & \vdots & \ddots & \vdots \\ g_{d-1} & g_{d-2} & \cdots & g_0 \end{pmatrix}$$

(in $K = \mathbb{Q}[X]/X^d + 1$)

Module lattices

(Free) module:

$$M = \{B \cdot x \,|\, x \in {\mathcal{O}_K}^k\} \text{ for some matrix } B \in {\mathcal{O}_K}^{k \times k} \text{ with } \det_K(B) \neq 0$$

Module lattices

(Free) module:

$$M = \{B \cdot x \mid x \in {\mathcal{O}_K}^k\} \text{ for some matrix } B \in {\mathcal{O}_K}^{k \times k} \text{ with } \det_K(B) \neq 0$$

- k is the module rank
- B is a module basis of M

  (if the module is not free, it has a "pseudo-basis" instead)

$\Sigma(M)$ is a lattice:

- of $\mathbb{Z}$-rank $n := d \cdot k$, included in $\mathbb{Z}^n$

Module lattices

(Free) module:

$$M = \{B \cdot x \mid x \in {\mathcal{O}_K}^k\} \text{ for some matrix } B \in {\mathcal{O}_K}^{k \times k} \text{ with } \det_K(B) \neq 0$$

- k is the module rank
- B is a module basis of M

  (if the module is not free, it has a "pseudo-basis" instead)

$\Sigma(M)$ is a lattice:

- of $\mathbb{Z}$-rank $n := d \cdot k$, included in $\mathbb{Z}^n$
- with basis $(\Sigma(b_i X^j))_{\substack{1 \leq i \leq k \\ 0 \leq j < d}}$ (b_i columns of B)

Modules vs ideals

Ideal	$=$	Module of rank 1
(principal ideal	$=$	free module of rank 1)

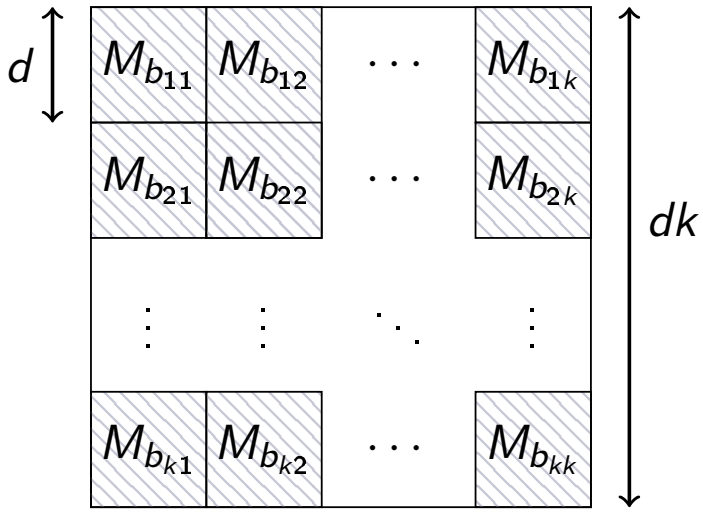
Modules vs ideals

Ideal	=	Module of rank 1
(principal ideal	=	free module of rank 1)

In $K = \mathbb{Q}[X]/(X^d + 1)$:

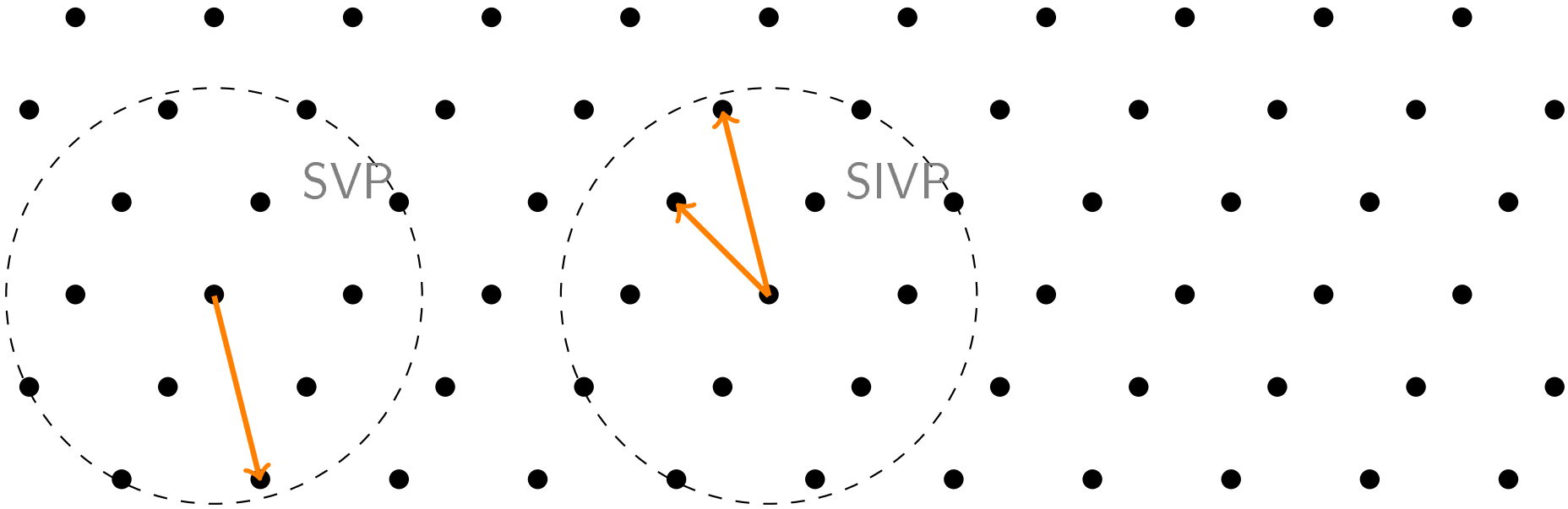
$$M_a = \begin{pmatrix} a_1 & -a_d & \cdots & -a_2 \\ a_2 & a_1 & \cdots & -a_3 \\ \vdots & \ddots & \ddots & \vdots \\ a_d & a_{d-1} & \cdots & a_1 \end{pmatrix}$$

basis of a
principal ideal lattice



basis of a free module lattice
of rank k

Algorithmic problems



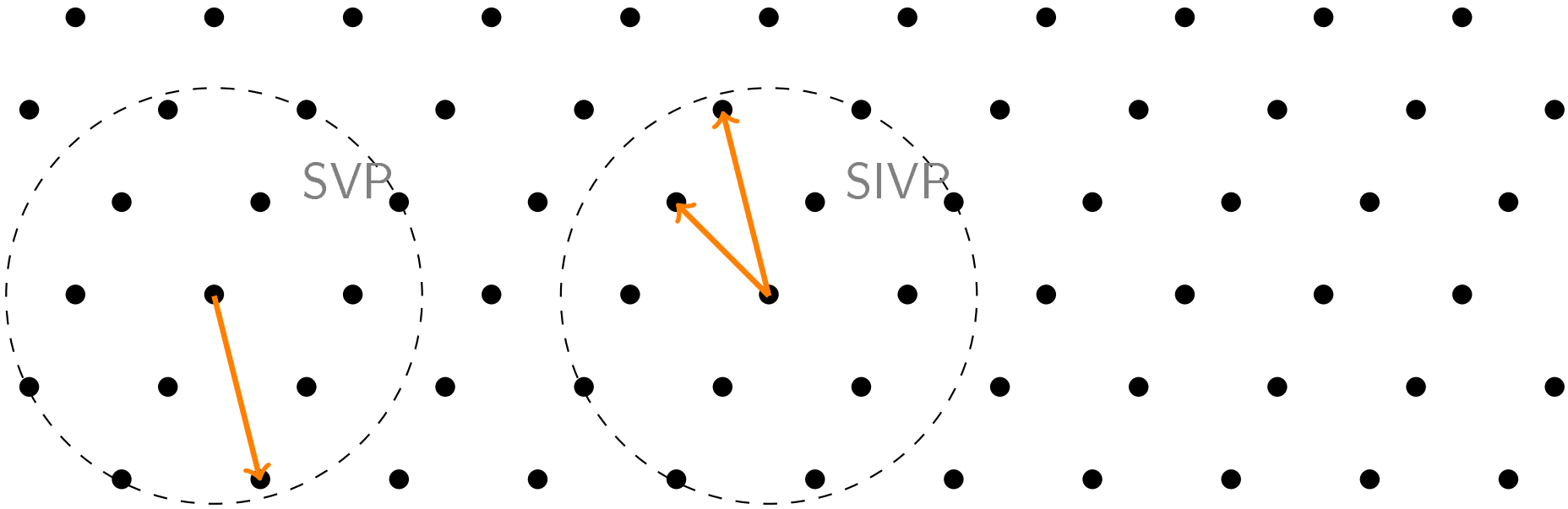
γ-SVP

shortest vector problem

γ-short basis

shortest basis problem

Algorithmic problems



γ-SVP
shortest vector problem

γ-short basis
shortest basis problem

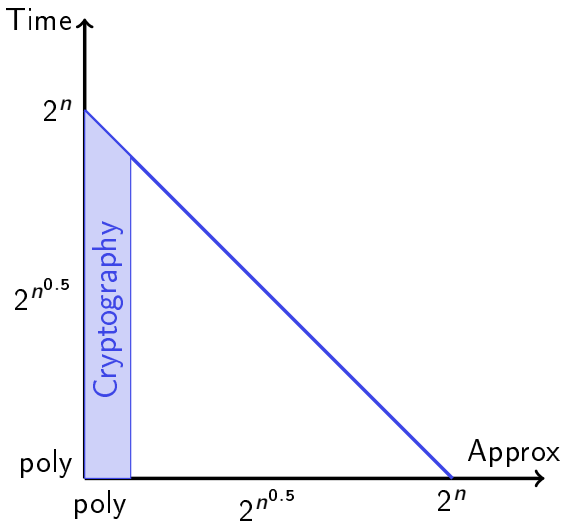
Notations:

- id-X = problem X restricted to ideal lattices
- mod-X_k = problem X restricted to module lattices of rank k

(worst-case: we want algorithms for all ideal/module lattices)

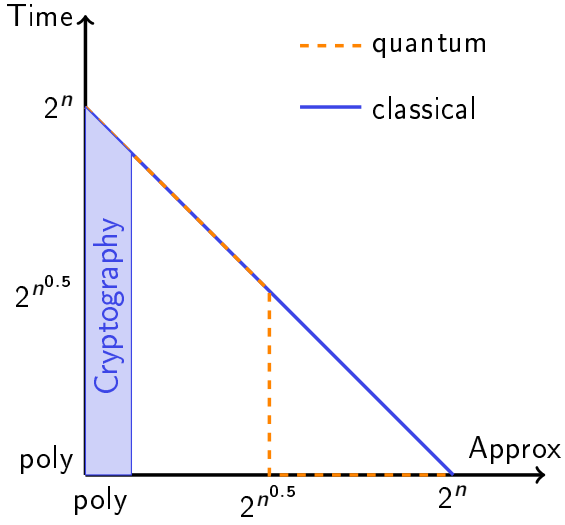
Hardness of SVP

Asymptotics:



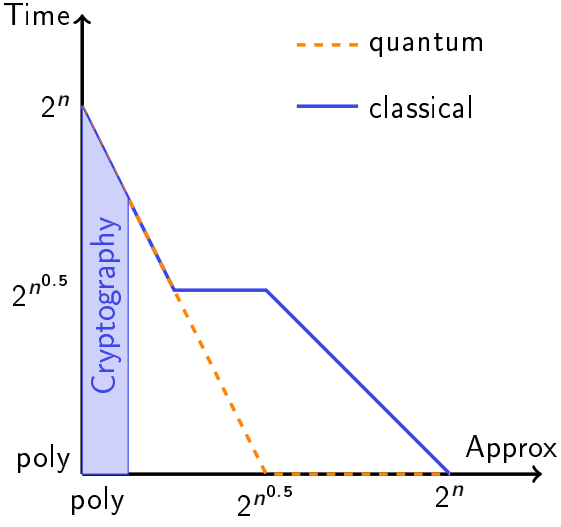
SVP and mod-SVP_k

($k \geq 2$)



id-SVP [CDW17]

(in cyclotomic fields)



id-SVP [PHS19,BR20]

(with $2^{O(n)}$ pre-processing)

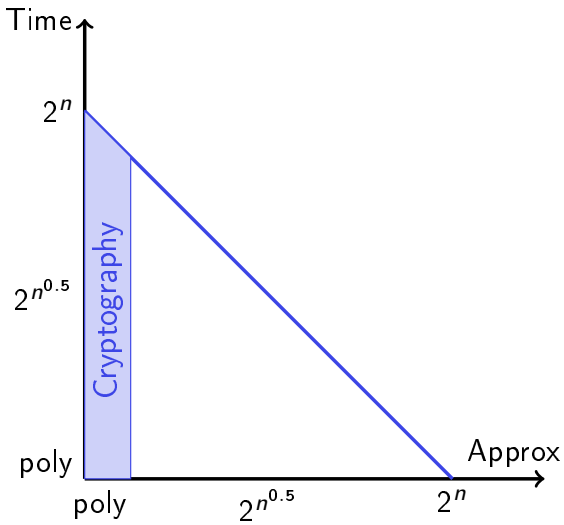
[CDW17] Cramer, Ducas, Wesolowski. Short stickelberger class relations and application to ideal-SVP. Eurocrypt.

[PHS19] Pellet-Mary, Hanrot, Stehlé. Approx-SVP in ideal lattices with pre-processing. Eurocrypt.

[BR20] Bernard, Roux-Langlois. Twisted-PHS: using the product formula to solve approx-SVP in ideal lattices. AC.

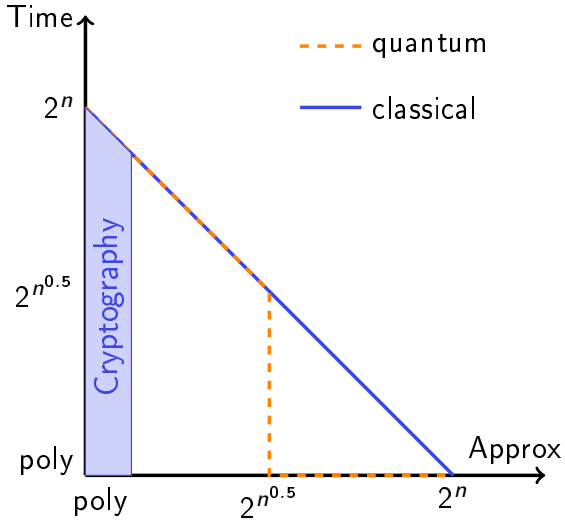
Hardness of SVP

Asymptotics:



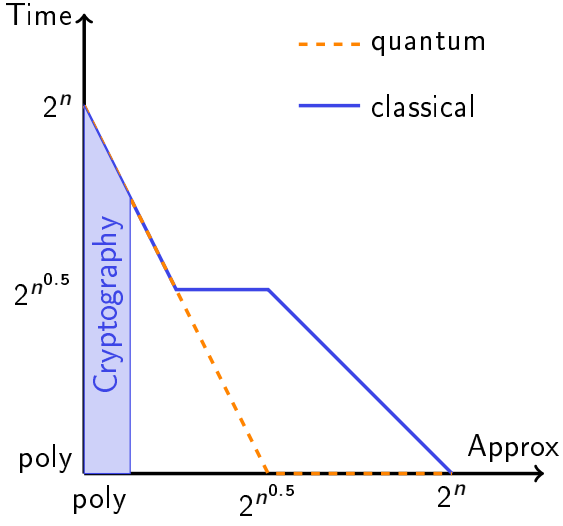
SVP and mod-SVP$_k$

($k \geq 2$)



id-SVP [CDW17]

(in cyclotomic fields)



id-SVP [PHS19,BR20]

(with $2^{O(n)}$ pre-processing)

Practice: Darmstadt challenge[1]

$\rightsquigarrow$ max dim for SVP: 186

$\rightsquigarrow$ max dim for id-SVP: 176

[1] https://www.latticechallenge.org/

Ring and Module-LWE

(search) mod-LWE$_k$

Parameters: q and B

Problem: Sample

- $A \leftarrow \text{Uniform}((\mathcal{O}_K/(q\mathcal{O}_K))^{m\times k})$
- $s, e \in {\mathcal{O}_K}^k \times {\mathcal{O}_K}^m$ with coefficients in $\{-B, \cdots, B\}$

Given A and $b = A \cdot s + e \bmod q$, recover s

(size of s and e can be defined using Σ or σ)

Ring and Module-LWE

(search) mod-LWE$_k$

Parameters: q and B

Problem: Sample

- $A \leftarrow \text{Uniform}((\mathcal{O}_K/(q\mathcal{O}_K))^{m\times k})$
- $s, e \in {\mathcal{O}_K}^k \times {\mathcal{O}_K}^m$ with coefficients in $\{-B, \cdots, B\}$

Given A and $b = A \cdot s + e \bmod q$, recover s

(size of s and e can be defined using Σ or σ)

RLWE = mod-LWE$_1$

Ring and Module-LWE

(search) mod-LWE$_k$

Parameters: q and B

Problem: Sample

- $A \leftarrow \text{Uniform}((\mathcal{O}_K/(q\mathcal{O}_K))^{m\times k})$
- $s, e \in {\mathcal{O}_K}^k \times {\mathcal{O}_K}^m$ with coefficients in $\{-B, \cdots, B\}$

Given A and $b = A \cdot s + e \bmod q$, recover s

(size of s and e can be defined using Σ or σ)

RLWE = mod-LWE$_1$

Module-LWE is a lattice problem:

mod-LWE$_k \approx$ CVP in a module lattice of rank m $(\Lambda_q^{\text{mod}}(A))$

NTRU [HPS98]

Parameters:
- q prime and large
- $B \ll \sqrt{q}$ integer

NTRU instance:
- sample f, g randomly in $\mathcal{O}_K$ with coefficients $\leq B$
- return $h = f \cdot g^{-1} \bmod q$

NTRU assumption: given h, there is no efficient algorithm that can find u and v such that $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

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

NTRU [HPS98]

Parameters:
- q prime and large
- $B \ll \sqrt{q}$ integer

NTRU instance:
- sample f, g randomly in $\mathcal{O}_K$ with coefficients $\leq B$
- return $h = f \cdot g^{-1} \bmod q$

NTRU assumption: given h, there is no efficient algorithm that can find u and v such that $|u|, |v| \leq B$ and $h = u \cdot v^{-1} \bmod q$

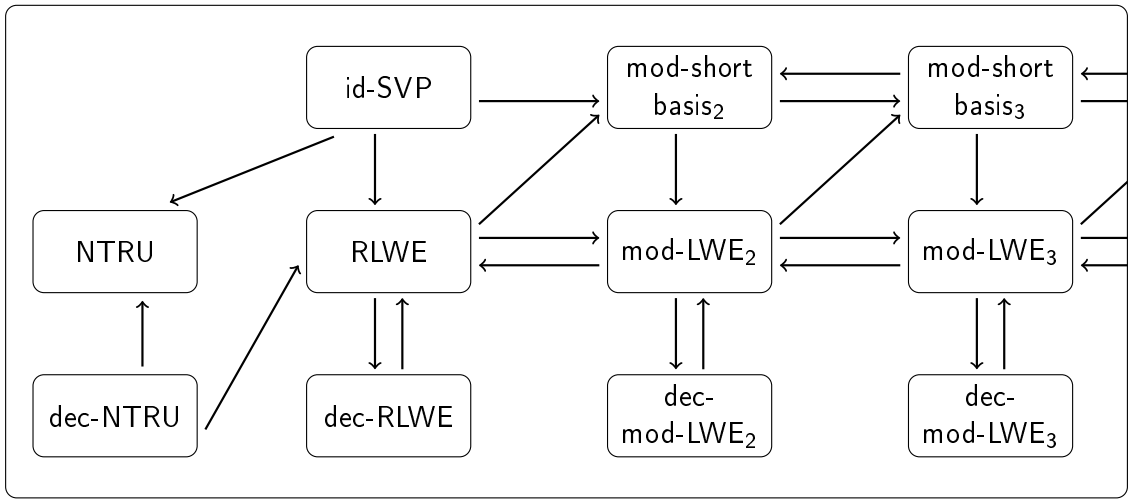
Now the NTRU assumption should hold

(or if it does not, then you have a paper!)

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

Reductions between problems

(or how to compare hardness of different problems)



LWE is as hard as worst-case lattice problems

Theorem [Reg05]

For any $m = \mathrm{poly}(n)$, modulus $q \leq 2^{\mathrm{poly}(n)}$ and $B \geq 2\sqrt{n}$, solving LWE is at least as hard as quantumly solving γ-SIVP on arbitrary n-dimensional lattice, for some approximation factor $\gamma = \tilde{O}(n \cdot q/B)$.

⚠ the reduction is for a variant of LWE where s and e are sampled from a discrete Gaussian distribution of parameter B ⚠

[Reg05] Regev. On lattices, learning with errors, random linear codes, and cryptography. STOC.

LWE is as hard as worst-case lattice problems

Theorem [Reg05]

For any $m = \mathrm{poly}(n)$, modulus $q \leq 2^{\mathrm{poly}(n)}$ and $B \geq 2\sqrt{n}$, solving LWE is at least as hard as quantumly solving γ-SIVP on arbitrary n-dimensional lattice, for some approximation factor $\gamma = \tilde{O}(n \cdot q/B)$.

⚠ the reduction is for a variant of LWE where s and e are sampled from a discrete Gaussian distribution of parameter B ⚠

Remark: the reduction can be made fully classical [Pei09, BLPRS13]
(with some extra loss on parameters)

[Pei09] Peikert. Public-key cryptosystems from the worst-case shortest vector problem. STOC.

[BLPRS13] Brakerski, Langlois, Peikert, Regev, and Stehlé. Classical hardness of learning with errors. STOC

LWE is as hard as worst-case lattice problems

Theorem [Reg05]

For any $m = \mathrm{poly}(n)$, modulus $q \leq 2^{\mathrm{poly}(n)}$ and $B \geq 2\sqrt{n}$, solving LWE is at least as hard as quantumly solving γ-SIVP on arbitrary n-dimensional lattice, for some approximation factor $\gamma = \tilde{O}(n \cdot q/B)$.

⚠ the reduction is for a variant of LWE where s and e are sampled from a discrete Gaussian distribution of parameter B ⚠

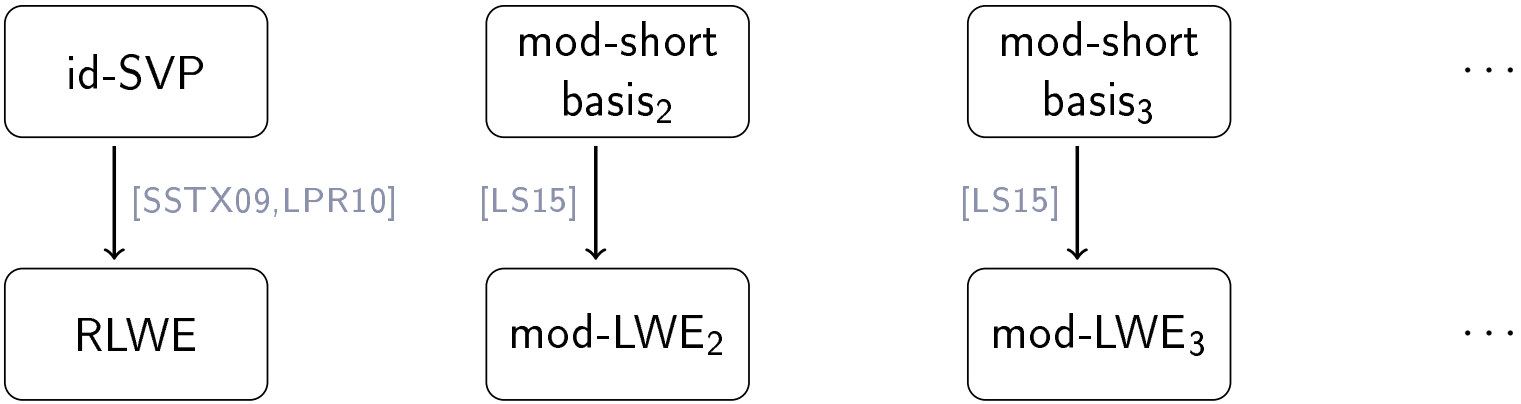
Remark: the reduction can be made fully classical [Pei09, BLPRS13]
(with some extra loss on parameters)

Solving LWE on average $\gtrsim$ (quantumly!) Computing short basis in any lattice of rank n

[Pei09] Peikert. Public-key cryptosystems from the worst-case shortest vector problem. STOC.

[BLPRS13] Brakerski, Langlois, Peikert, Regev, and Stehlé. Classical hardness of learning with errors. STOC

Reductions



⚠ Arrows may not all compose (different parameters) ⚠

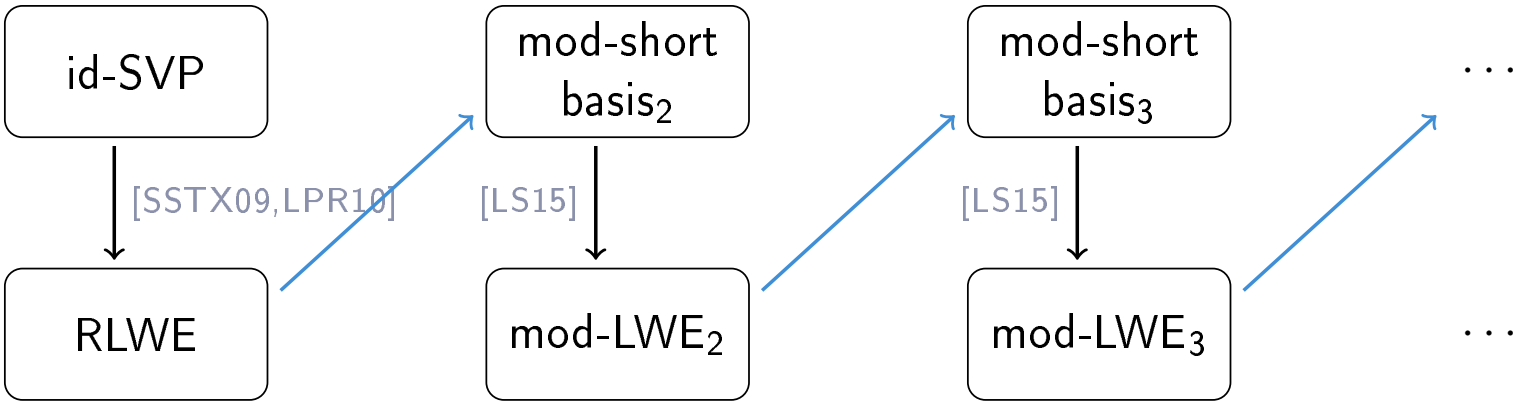
- reductions may be quantum
- reductions hold for σ and Gaussian noise

[SSTX09] Stehlé, Steinfeld, Tanaka, Xagawa. Efficient public key encryption based on ideal lattices. Asiacrypt.

[LPR10] Lyubashevsky, Peikert, Regev. On ideal lattices and learning with errors over rings. Eurocrypt.

[LS15] Langlois, Stehlé. Worst-case to average-case reductions for module lattices. DCC.

Reductions



⚠ Arrows may not all compose (different parameters) ⚠

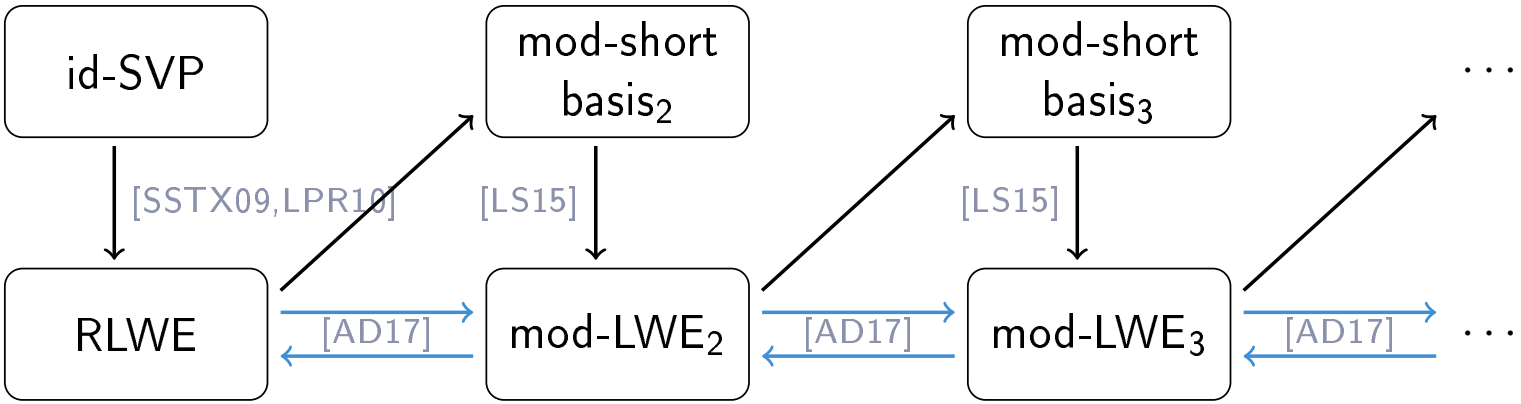
- reductions may be quantum
- reductions hold for σ and Gaussian noise

[SSTX09] Stehlé, Steinfeld, Tanaka, Xagawa. Efficient public key encryption based on ideal lattices. Asiacrypt.

[LPR10] Lyubashevsky, Peikert, Regev. On ideal lattices and learning with errors over rings. Eurocrypt.

[LS15] Langlois, Stehlé. Worst-case to average-case reductions for module lattices. DCC.

Reductions

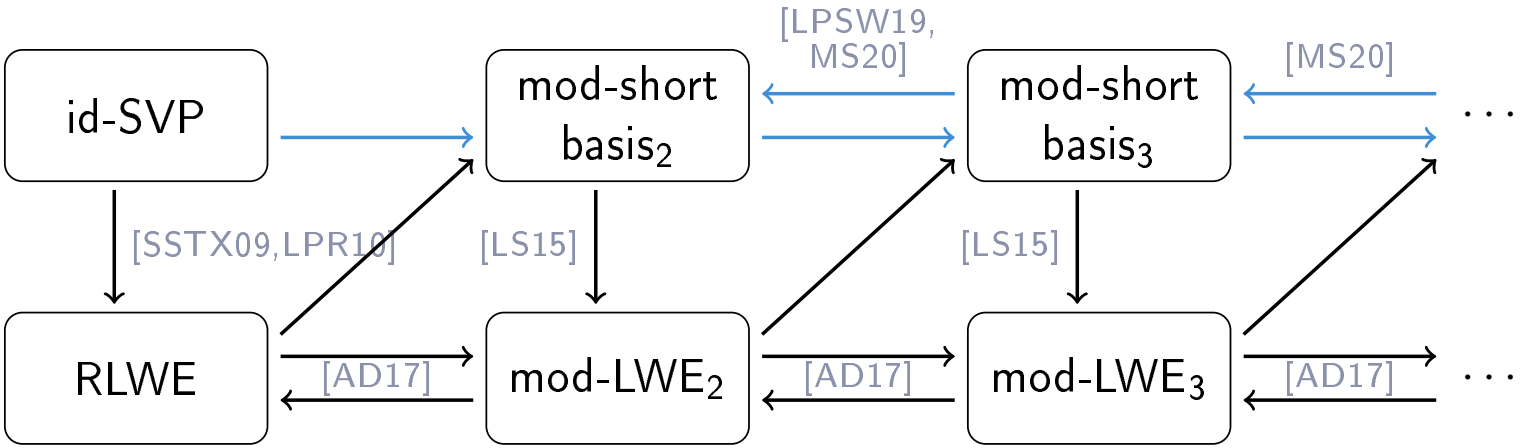


⚠ Arrows may not all compose (different parameters) ⚠

- reductions may be quantum
- reductions hold for σ and Gaussian noise

[AD17] Albrecht, Deo. Large modulus ring-LWE $\geq$ module-LWE. Asiacrypt.

Reductions



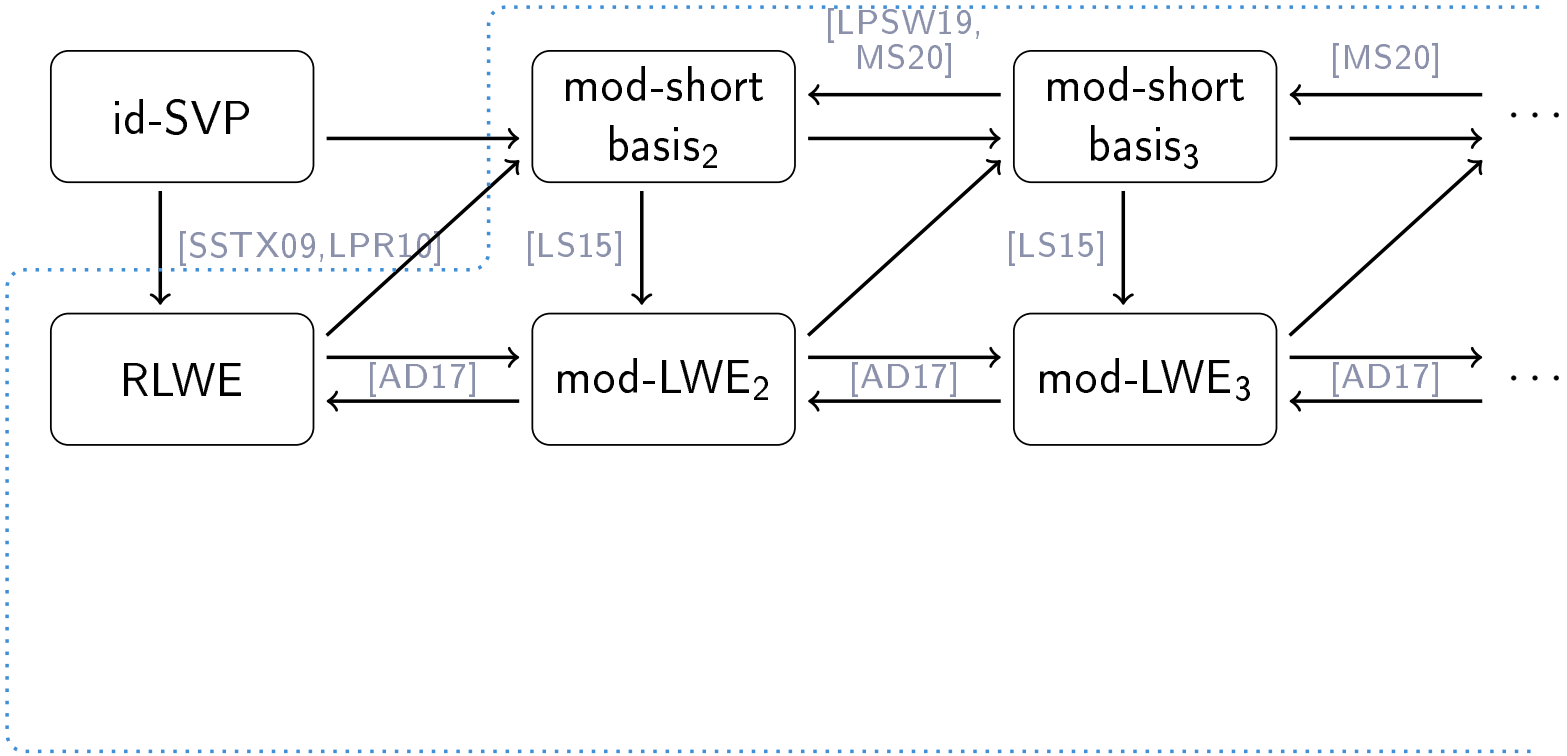
⚠ Arrows may not all compose (different parameters) ⚠

- reductions may be quantum
- reductions hold for σ and Gaussian noise

[LPSW19] Lee, Pellet-Mary, Stehlé, and Wallet. An LLL algorithm for module lattices. Asiacrypt.

[MS20] Mukherjee and Stephens-Davidowitz. Lattice reduction for modules, or how to reduce moduleSVP to moduleSVP. Crypto.

Reductions



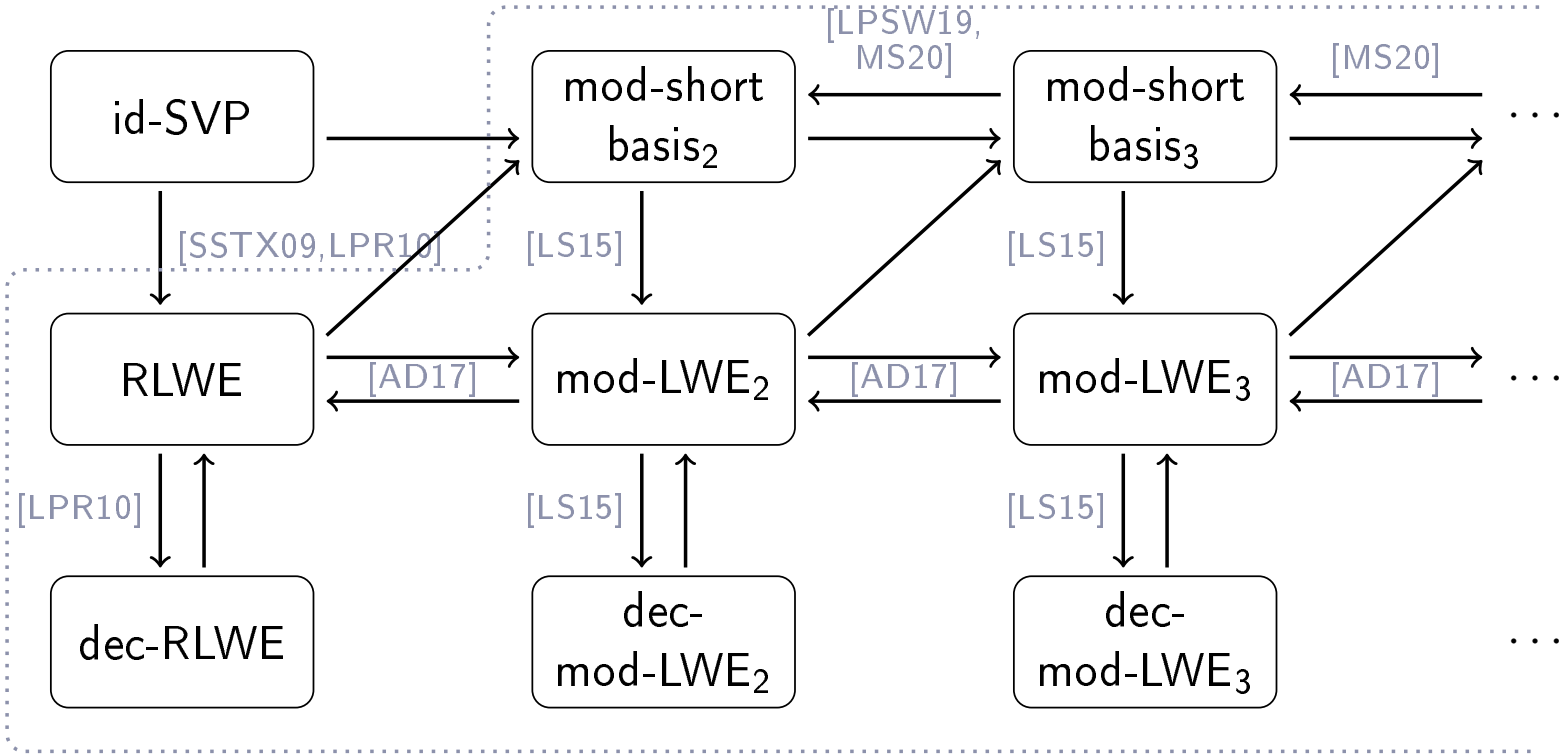
⚠ Arrows may not all compose (different parameters) ⚠

- reductions may be quantum
- reductions hold for σ and Gaussian noise

[LPSW19] Lee, Pellet-Mary, Stehlé, and Wallet. An LLL algorithm for module lattices. Asiacrypt.

[MS20] Mukherjee and Stephens-Davidowitz. Lattice reduction for modules, or how to reduce moduleSVP to moduleSVP. Crypto.

Reductions



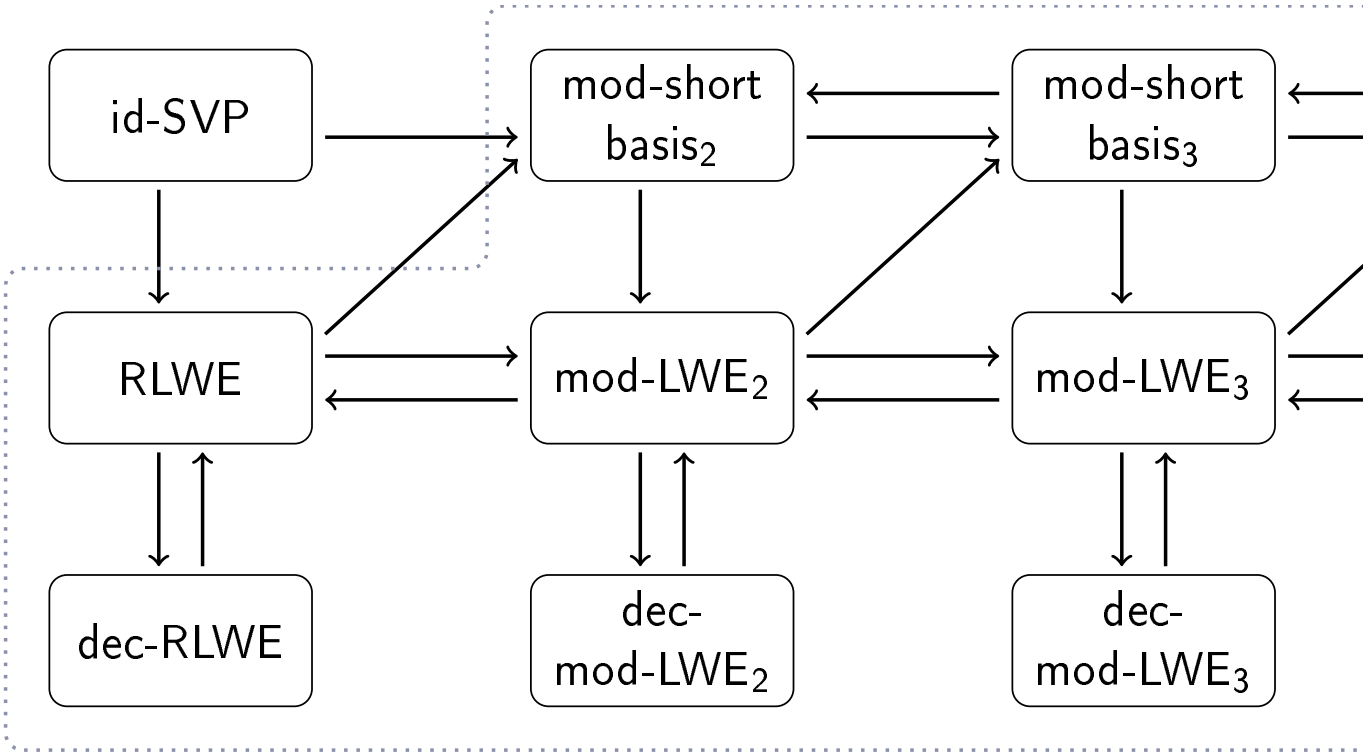
⚠ Arrows may not all compose (different parameters) ⚠

- reductions may be quantum
- reductions hold for σ and Gaussian noise

[LPSW19] Lee, Pellet-Mary, Stehlé, and Wallet. An LLL algorithm for module lattices. Asiacrypt.

[MS20] Mukherjee and Stephens-Davidowitz. Lattice reduction for modules, or how to reduce moduleSVP to moduleSVP. Crypto.

Reductions

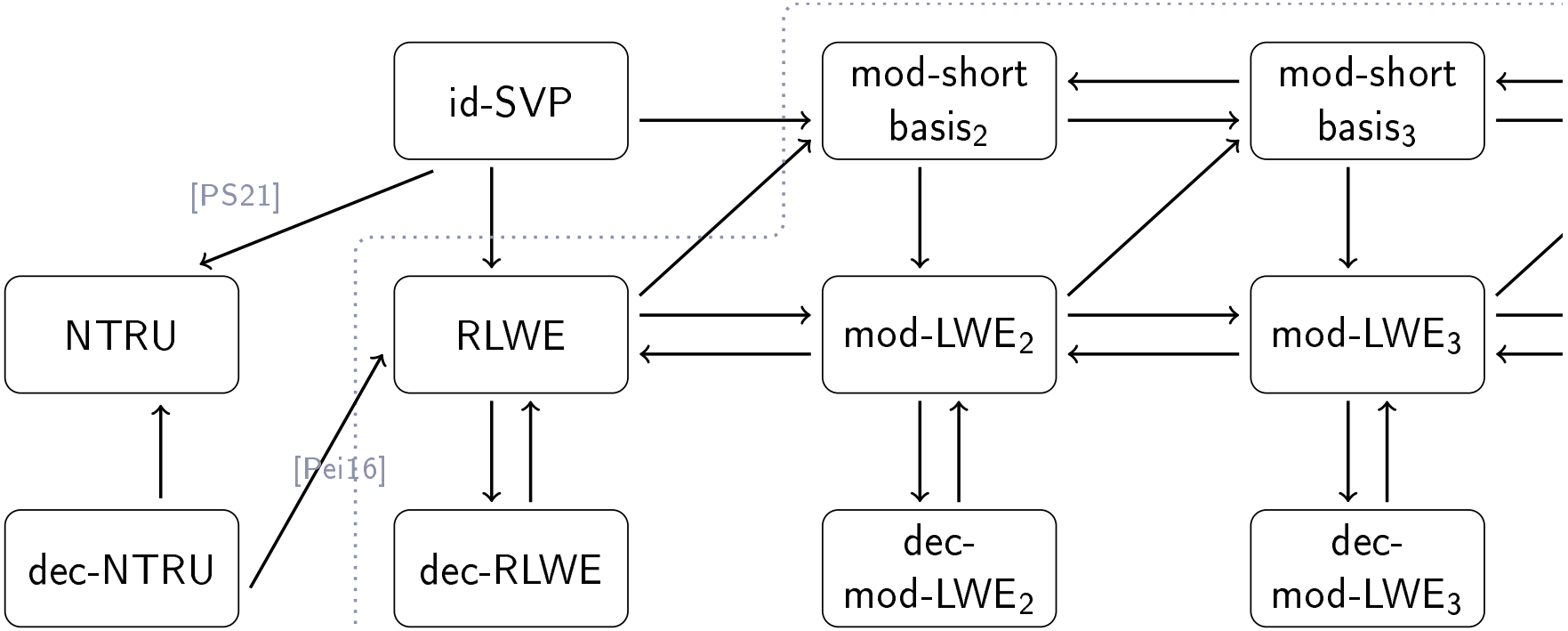


⚠ Arrows may not all compose (different parameters) ⚠

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

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

Reductions

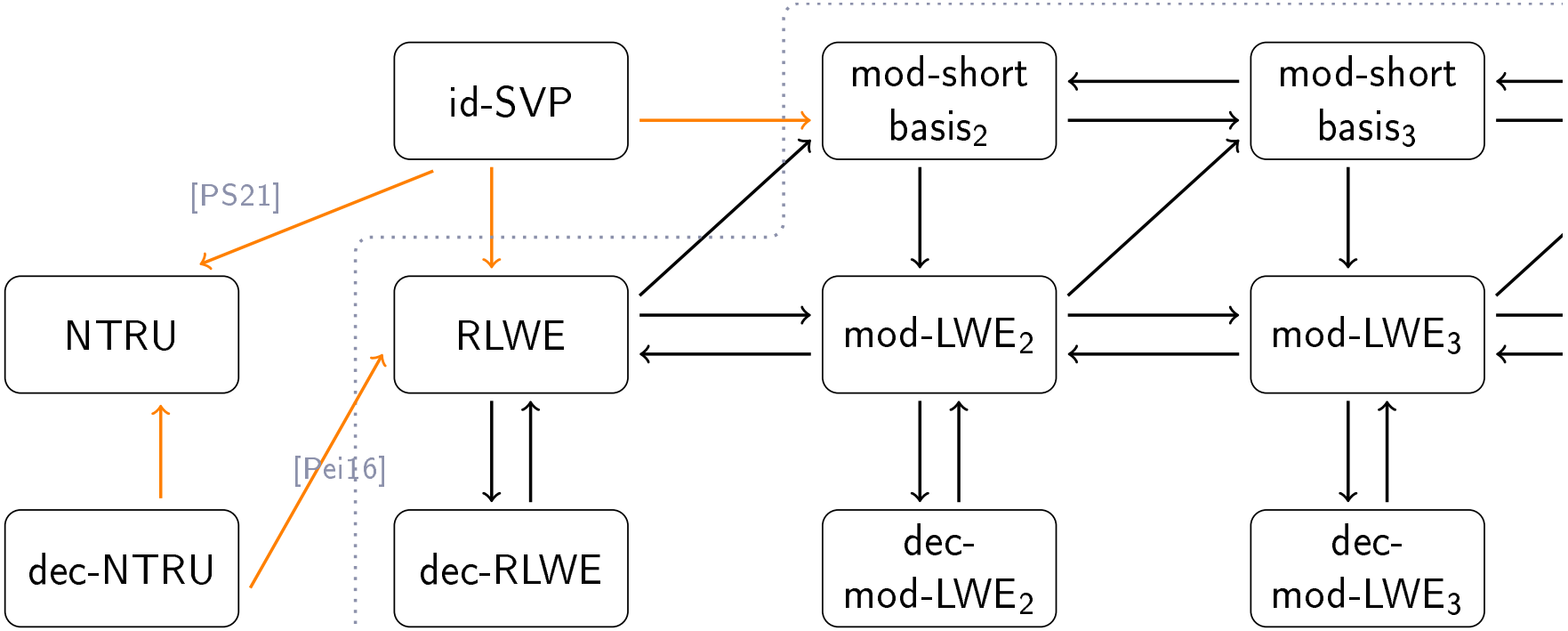


⚠ Arrows may not all compose (different parameters) ⚠

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

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

Reductions



possibly crude reductions

⚠ Arrows may not all compose (different parameters) ⚠

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

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

Take-away from this section

id-SVP is a lower bound
on the hardness of RLWE, mod-LWE, NTRU

Take-away from this section

id-SVP is a lower bound
on the hardness of RLWE, mod-LWE, NTRU

Breaking id-SVP does not break:

- RLWE, mod-LWE, NTRU
- most lattice-based crypto using algebraic lattices

Take-away from this section

id-SVP is a lower bound
on the hardness of RLWE, mod-LWE, NTRU

Breaking id-SVP does not break:

- RLWE, mod-LWE, NTRU
- most lattice-based crypto using algebraic lattices

Breaking id-SVP do break:

- some early FHE schemes
- the PV-Knap problem [HPS+14,BSS22]

[HPS+14] Hoffstein, Pipher, Schanck, Silverman, and Whyte. Practical signatures from the partial Fourier recovery problem. ACNS.

[BSS22] Boudgoust, Sakzad, and Steinfeld. Vandermonde meets Regev: Public Key Encryption Schemes Based on Partial Vandermonde Problems. DCC.

Conclusion

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`
- quite efficient if using structured lattices

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`
- quite efficient if using structured lattices
- can be used in many constructions

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`
- quite efficient if using structured lattices
- can be used in many constructions

Drawbacks:

- big keysizes and ciphertexts/signatures

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`
- quite efficient if using structured lattices
- can be used in many constructions

Drawbacks:

- big keysizes and ciphertexts/signatures
- structured lattice problems are still young
  - more cryptanalysis is needed

Conclusion

Advantages:

- many reductions (worst-case to average-case, search to decision, ...)
  - some parameters might still be broken
  - but gives confidence that there are no major flaws in the problems
- complexity of the best algorithms is quite well understood
  - LWE estimator: `https://github.com/malb/lattice-estimator`
- quite efficient if using structured lattices
- can be used in many constructions

Drawbacks:

- big keysizes and ciphertexts/signatures
- structured lattice problems are still young
  - more cryptanalysis is needed

Thank you