Lattice cryptography and cryptanalysis

Wessel van Woerden (Université de Bordeaux, IMB, Inria).

université de BORDEAUX

Plan

Part I

Lattice theory

- Lattices
- Hard problems

Cryptography

- Trapdoor bases
- Encryption, Signature

Cryptanalysis

- Lattice Sieving
- Basis Reduction

Plan

Part I

Lattice theory

- Lattices
- Hard problems

Cryptography

- Trapdoor bases
- Encryption, Signature

Cryptanalysis

- Lattice Sieving
- Basis Reduction

Part II

Lattices used in cryptography

- SIS, LWE, decLWE
- Security proofs

Hardness Reductions

- search to decision
- WC to AC reductions

Algebraic Lattices

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

Plan

Part I

Lattice theory

- Lattices
- Hard problems

Cryptography

- Trapdoor bases
- Encryption, Signature

Cryptanalysis

- Lattice Sieving
- Basis Reduction

Part II

Lattices used in cryptography

- SIS, LWE, decLWE
- Security proofs

Hardness Reductions

- search to decision
- WC to AC reductions

Algebraic Lattices

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

acknowledgements: many slides adapted from Alice Pellet-Mary!

https://apelletm.pages.math.cnrs.fr/page-perso/research.html

Lattice theory

From codes to lattices

Similarities:

From codes to lattices

Similarities:

- Both are discrete additive groups

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

Differences:

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

Differences:

- Hamming distance in $\mathbb{F}_q^n$ $\longrightarrow$ Euclidean distance in $\mathbb{R}^n$

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

Differences:

- Hamming distance in $\mathbb{F}_q^n$ $\longrightarrow$ Euclidean distance in $\mathbb{R}^n$ (pictures!)

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

Differences:

- Hamming distance in $\mathbb{F}_q^n$ $\longrightarrow$ Euclidean distance in $\mathbb{R}^n$ (pictures!)
- Code with decoding algorithm $\longrightarrow$ Any lattice and a short basis
  (decoding for free!)

From codes to lattices

Similarities:

- Both are discrete additive groups
- Same problems: finding short or close lattice/code points

Differences:

- Hamming distance in $\mathbb{F}_q^n$ $\longrightarrow$ Euclidean distance in $\mathbb{R}^n$ (pictures!)
- Code with decoding algorithm $\longrightarrow$ Any lattice and a short basis

  (decoding for free!)

most important:

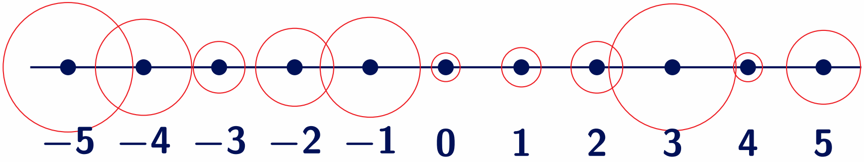
row vectors ($\boldsymbol{xG}$) $\longrightarrow$ column vectors ($\boldsymbol{Gx}$)

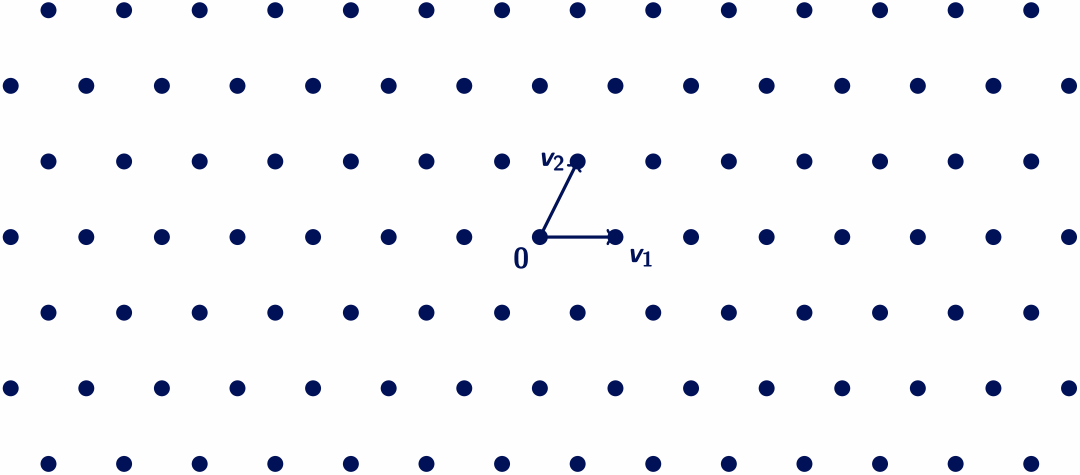
A lattice $\mathcal{L} \subset \mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$.

Discrete

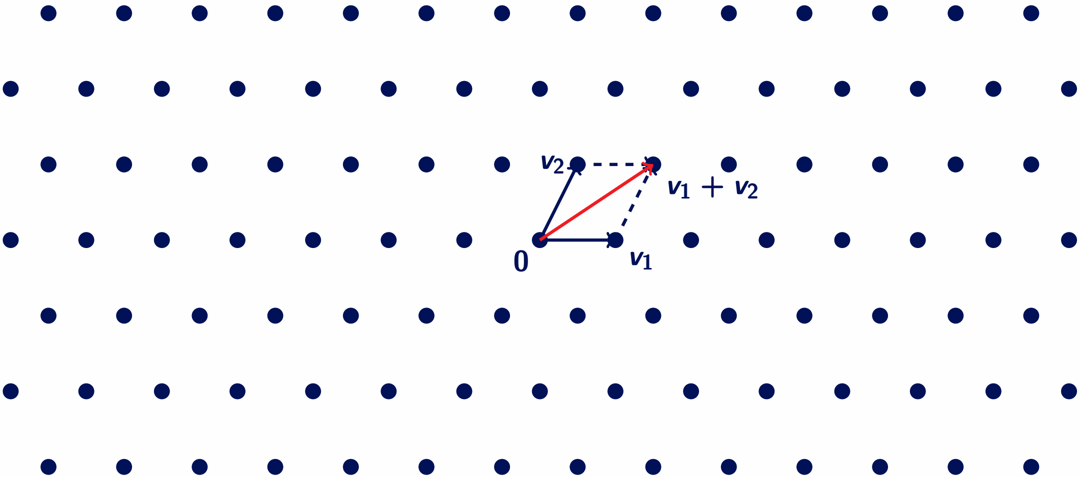
For every $v \in \mathcal{L}$ there exists an open ball around v that contains no other elements from $\mathcal{L}$.

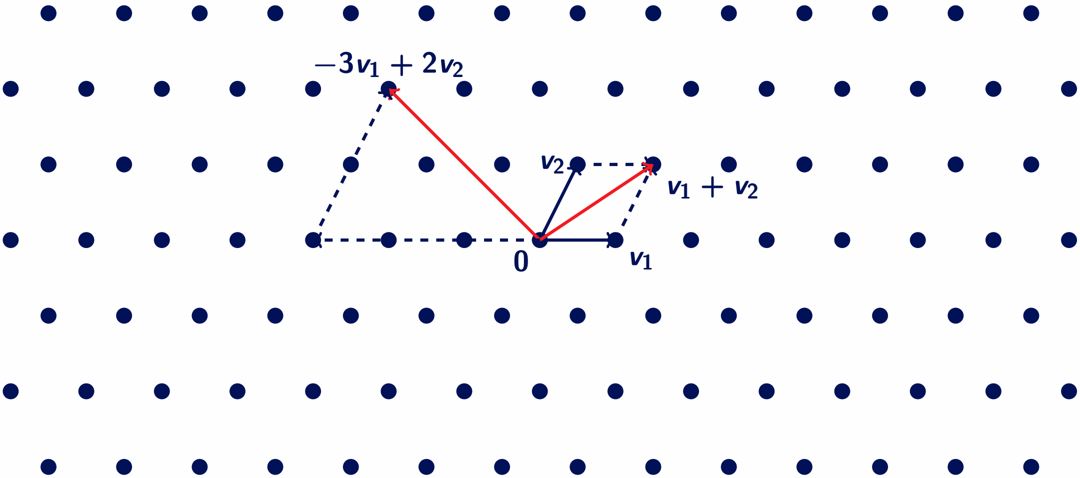
Example $\mathbb{Z} \subset \mathbb{R}$:

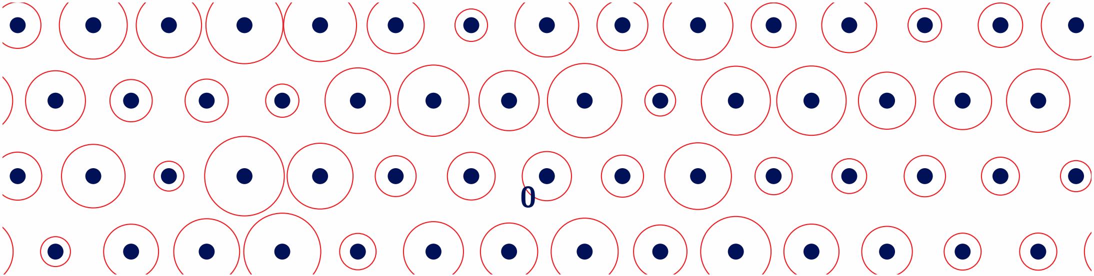



v_2
0
v_1

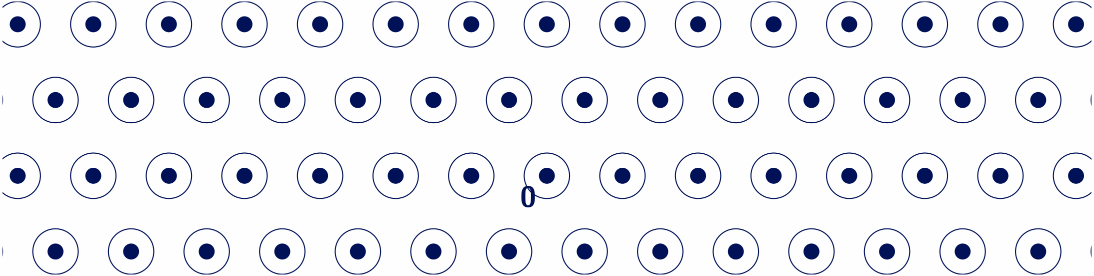
Additive




$-3v_1 + 2v_2$
v_2
$v_1 + v_2$
0
v_1

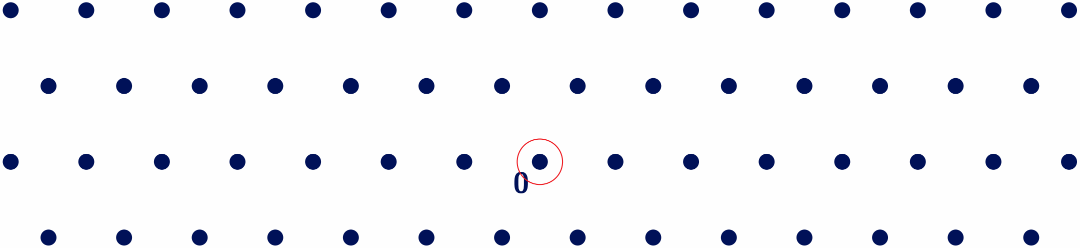

0

First minimum of a lattice



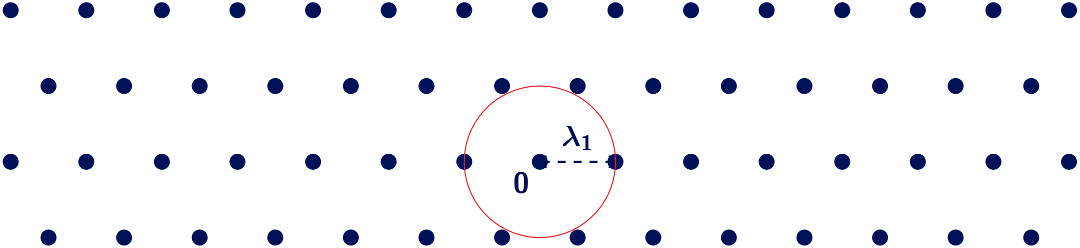
By the additivity the neighborhood of every lattice point looks the same.

First minimum of a lattice



By the additivity the neighborhood of every lattice point looks the same.

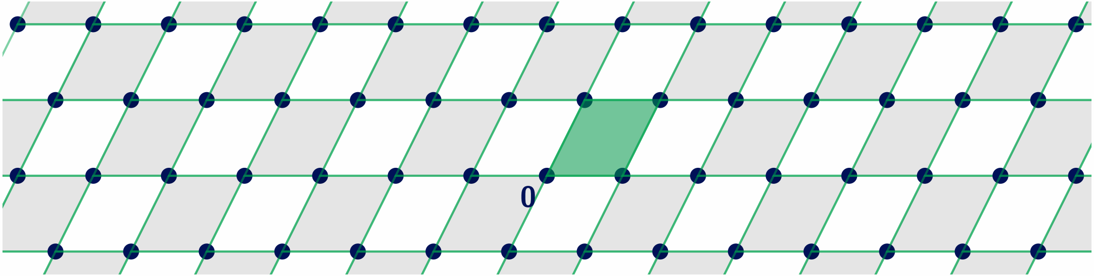
First minimum of a lattice



The first minimum $\lambda_1(\mathcal{L})$ of a lattice $\mathcal{L}$ is
the length of the shortest nonzero lattice vector:

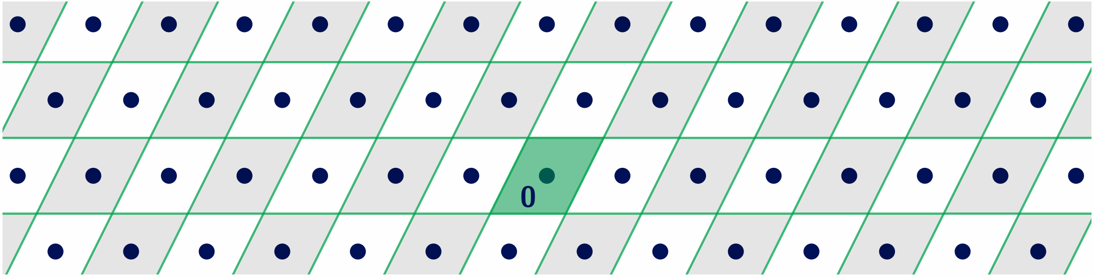
$$\lambda_1(\mathcal{L}) = \min_{x \in \mathcal{L} \setminus \{0\}} \{\|x\|\} > 0.$$

Volume of a lattice



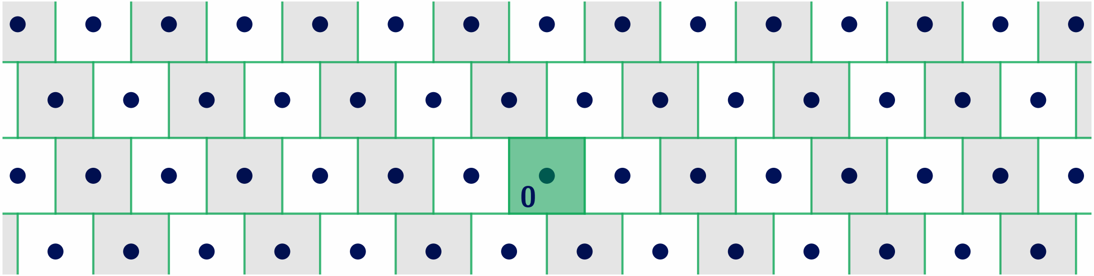
The volume $\mathrm{vol}(\mathcal{L})$ of a lattice $\mathcal{L}$ is the (co-)volume of any fundamental area w.r.t. translation of the lattice:

$$\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) \qquad (\mathrm{density}(\mathcal{L}) = 1/\,\mathrm{vol}(\mathcal{L}))$$



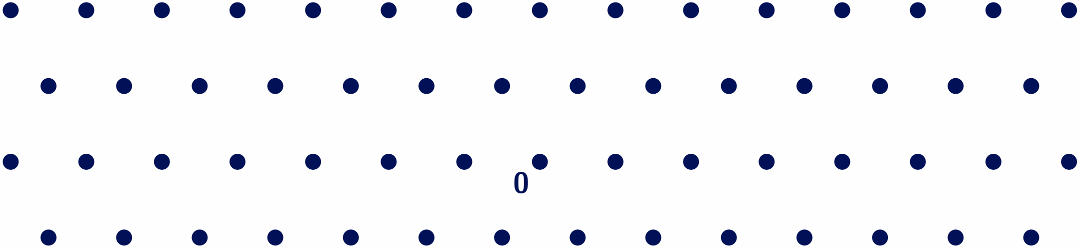
The volume $\mathrm{vol}(\mathcal{L})$ of a lattice $\mathcal{L}$ is the (co-)volume of any fundamental area w.r.t. translation of the lattice:

$$\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) \qquad (\mathrm{density}(\mathcal{L}) = 1/\,\mathrm{vol}(\mathcal{L}))$$



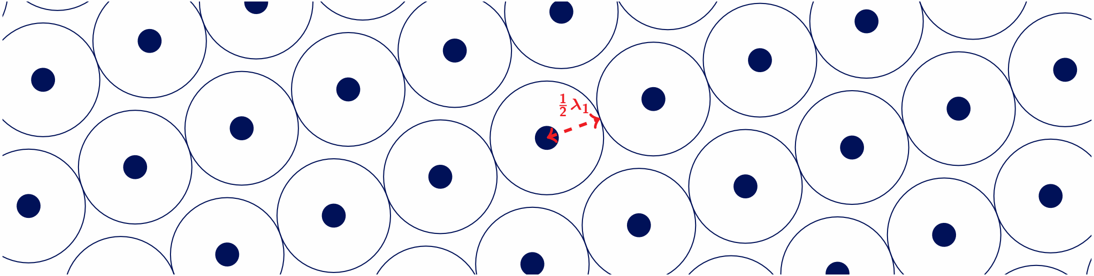
The volume $\mathbf{vol}(\mathcal{L})$ of a lattice $\mathcal{L}$ is the (co-)volume of any fundamental area w.r.t. translation of the lattice:

$$\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathbb{R}^n/\mathcal{L}) \qquad (\mathrm{density}(\mathcal{L}) = 1/\,\mathrm{vol}(\mathcal{L}))$$



The volume $\mathbf{vol}(\mathcal{L})$ of a lattice $\mathcal{L}$ is the (co-)volume of any fundamental area w.r.t. translation of the lattice:

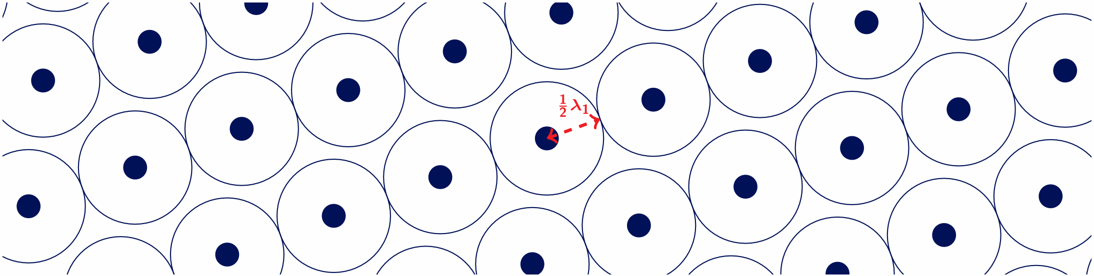
$$\mathbf{vol}(\mathcal{L}) = \mathbf{vol}(\mathrm{Span}_{\mathbb{R}}(\mathcal{L})/\mathcal{L}), \qquad (\mathbf{density}(\mathcal{L}) = 1/\mathbf{vol}(\mathcal{L}))$$



Minkowski's Theorem

For a full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$ we have

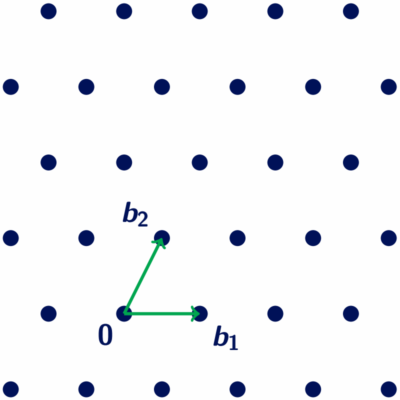
$$\mathrm{vol}\left(\frac{1}{2}\lambda_1(\mathcal{L}) \cdot \mathcal{B}^n\right) \leq \mathbf{vol}(\mathcal{L})$$



Minkowski's Theorem

For a full-rank lattice $\mathcal{L} \subset \mathbb{R}^n$ we have

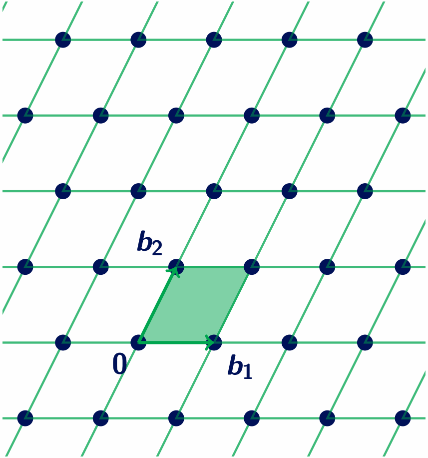
$$\lambda_1(\mathcal{L}) \leq \underbrace{2\frac{\mathrm{vol}(\mathcal{L})^{1/n}}{\mathrm{vol}(\mathcal{B}^n)^{1/n}}}_{\mathrm{Mk}(\mathcal{L})} \approx 2 \cdot \sqrt{n/2\pi e} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$



Lattice basis

$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n$

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



Lattice basis

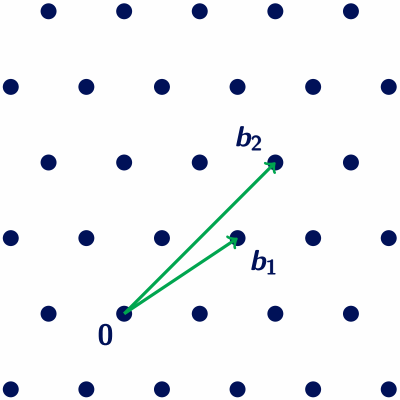
$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n$

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

Fundamental Parallelepiped

$\mathcal{P}(B) = B \cdot [0,1)^n$

$\mathbf{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{P}(B)) = |\det(B)|$



Lattice basis

$\mathbb{R}$-linearly independent $\mathbf{b}_1, \ldots, \mathbf{b}_n$

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

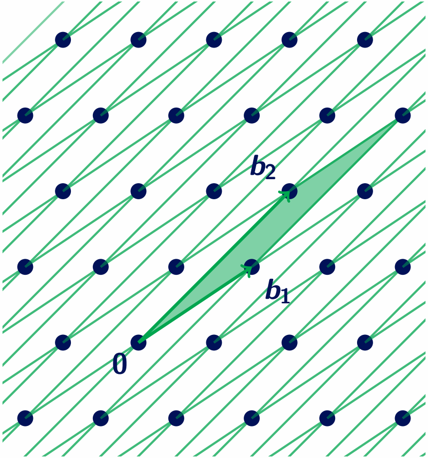
Fundamental Parallelepiped

$$\mathcal{P}(B) = B \cdot [0,1)^n$$

$$\mathbf{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{P}(B)) = |\det(B)|$$

Infinitely many distinct bases

$$B' = B \cdot U \text{ for } U \in \mathcal{GL}_n(\mathbb{Z}).$$



Lattice basis

$\mathbb{R}$-linearly independent $\mathbf{b_1, \ldots, b_n}$

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

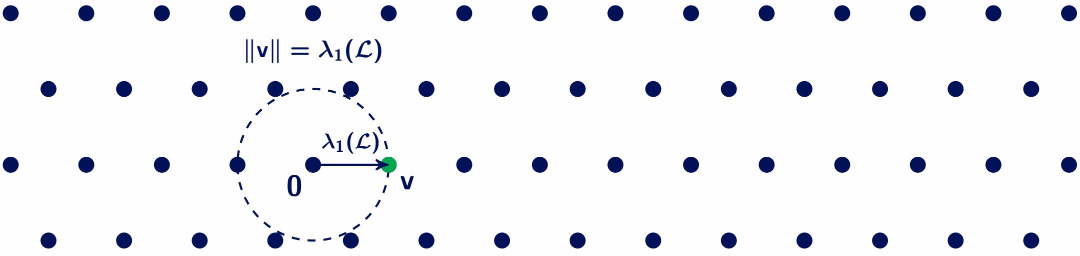
Fundamental Parallelepiped

$$\mathcal{P}(B) = B \cdot [0,1)^n$$

$$\mathbf{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{P}(B)) = |\,\mathbf{det}(B)|$$

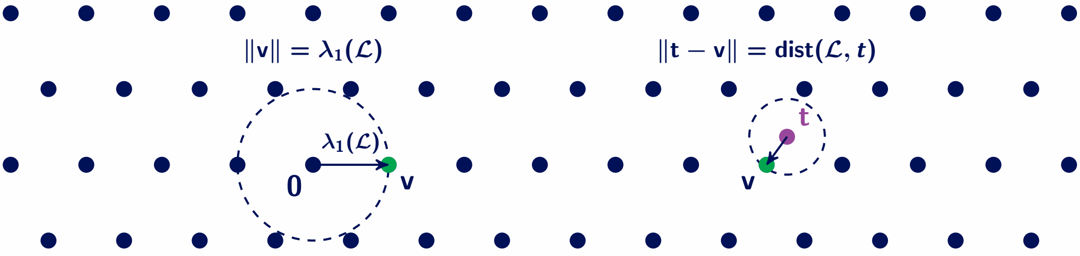
Infinitely many distinct bases

$$B' = B \cdot U \text{ for } U \in \mathcal{GL}_n(\mathbb{Z}).$$



Shortest Vector Problem (SVP)

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

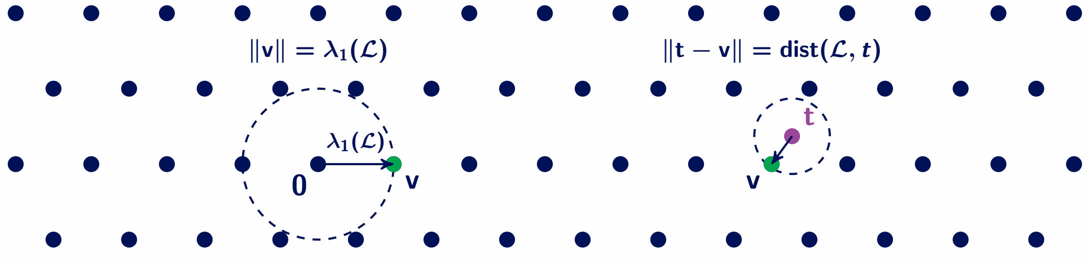


Shortest Vector Problem (SVP)

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

Closest Vector Problem (CVP)

Given a target $\mathbf{t} \in \mathbb{R}^n$, find a closest vector $\boldsymbol{v} \in \mathcal{L}$ to $\mathbf{t}$.



Shortest Vector Problem (SVP)

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

Closest Vector Problem (CVP)

Given a target $\mathbf{t} \in \mathbb{R}^n$, find a closest vector $\boldsymbol{v} \in \mathcal{L}$ to $\mathbf{t}$.

Supposedly hard to solve when n is large

(even with a quantum computer)

How hard is SVP/CVP?

In theory: best algorithm has asymptotic complexity $2^{c \cdot n + o(n)}$

classical: $c \approx 0.292$, or quantum: $c \approx 0.265$)

$\Rightarrow$ not polynomial

How hard is SVP/CVP?

In theory: best algorithm has asymptotic complexity $2^{c \cdot n + o(n)}$

classical: $c \approx 0.292$, or quantum: $c \approx 0.265$)

$\Rightarrow$ not polynomial

In practice:

- $n = 2 \rightsquigarrow$ easy, very efficient in practice

How hard is SVP/CVP?

In theory: best algorithm has asymptotic complexity $2^{c \cdot n + o(n)}$

classical: $c \approx 0.292$, or quantum: $c \approx 0.265$)

$\Rightarrow$ not polynomial

In practice:

- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 60$ or $n = 80 \rightsquigarrow$ a few minutes on a personal laptop

How hard is SVP/CVP?

In theory: best algorithm has asymptotic complexity $2^{c \cdot n + o(n)}$

classical: $c \approx 0.292$, or quantum: $c \approx 0.265$)

$\Rightarrow$ not polynomial

In practice:

- $n = 2$ $\rightsquigarrow$ easy, very efficient in practice
- up to $n = 60$ or $n = 80$ $\rightsquigarrow$ a few minutes on a personal laptop
- up to $n = 180$ $\rightsquigarrow$ few weeks on a big computer with good code

How hard is SVP/CVP?

In theory: best algorithm has asymptotic complexity $2^{c \cdot n + o(n)}$

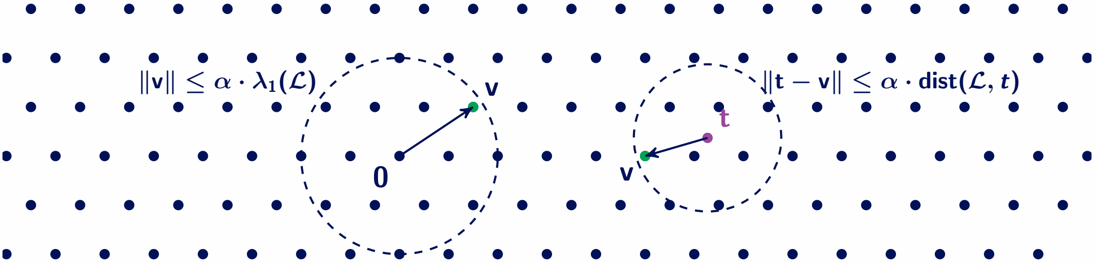
classical: $c \approx 0.292$, or quantum: $c \approx 0.265$)

$\Rightarrow$ not polynomial

In practice:

- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 60$ or $n = 80 \rightsquigarrow$ a few minutes on a personal laptop
- up to $n = 180 \rightsquigarrow$ few weeks on a big computer with good code
- from $n = 400$ to $n = 1000 \rightsquigarrow$ cryptography

Approximate versions

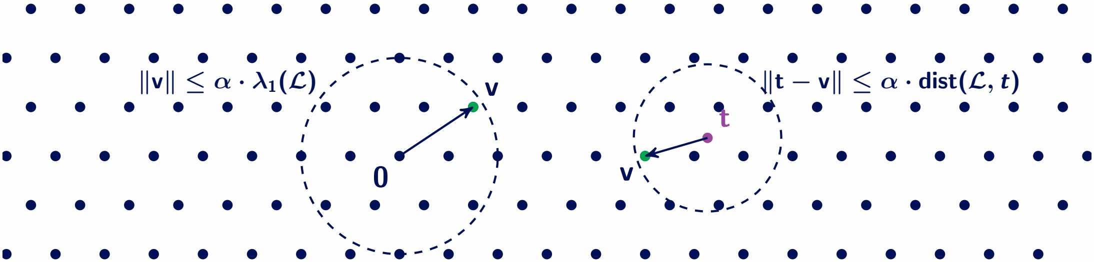


α-approx-SVP

Find a short nonzero vector $\boldsymbol{v} \in \mathcal{L}$ of length $\leq \alpha \cdot \lambda_1(\mathcal{L})$.

α-approx-CVP

Given a target $\mathbf{t} \in \mathbb{R}^n$, find a close vector $\boldsymbol{v} \in \mathcal{L}$ to $\mathbf{t}$.



α-approx-SVP

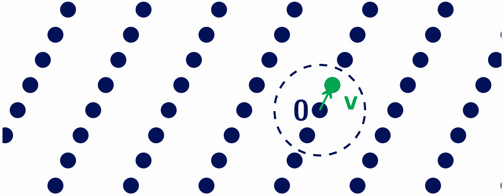
Find a short nonzero vector $\boldsymbol{v} \in \mathcal{L}$ of length $\leq \alpha \cdot \lambda_1(\mathcal{L})$.

α-approx-CVP

Given a target $\mathbf{t} \in \mathbb{R}^n$, find a close vector $\boldsymbol{v} \in \mathcal{L}$ to $\mathbf{t}$.

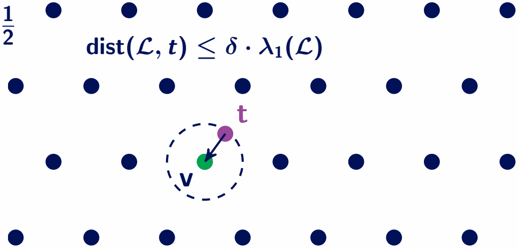
Supposedly hard to solve when n is large and the approximation factor α is small (poly(n))

$\lambda_1(\mathcal{L}) \leq \delta \cdot \mathsf{Mk}(\mathcal{L})$



$\delta \leq \frac{1}{2}$

$\mathsf{dist}(\mathcal{L}, t) \leq \delta \cdot \lambda_1(\mathcal{L})$

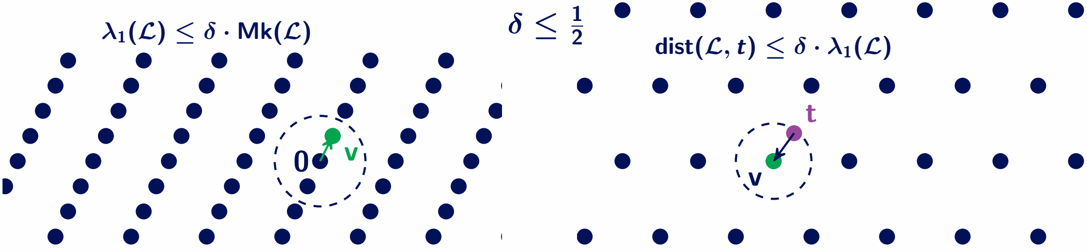


δ-uSVP

Find unusually short vector $\boldsymbol{v} \in \mathcal{L}$.

Bounded Distance Decoding (δ-BDD)

CVP with a target unusually close to the lattice.



δ-uSVP

Find unusually short vector $v \in \mathcal{L}$.

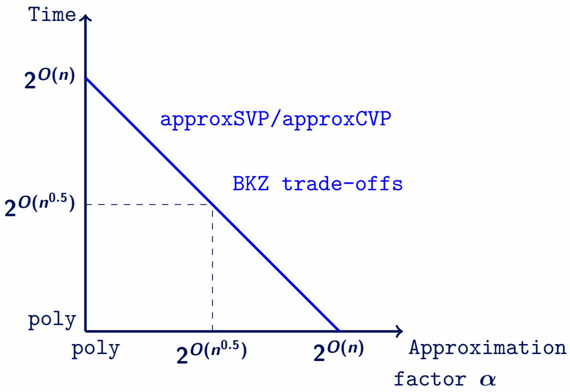
Bounded Distance Decoding (δ-BDD)

CVP with a target unusually close to the lattice.

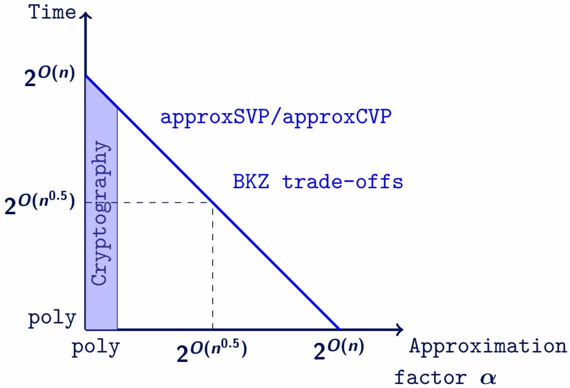
Supposedly hard to solve when n is large and the promise gap $1/\delta$ is small (poly(n))

Best Time/Approximation trade-off for SVP, CVP (even quantumly):

BKZ algorithm

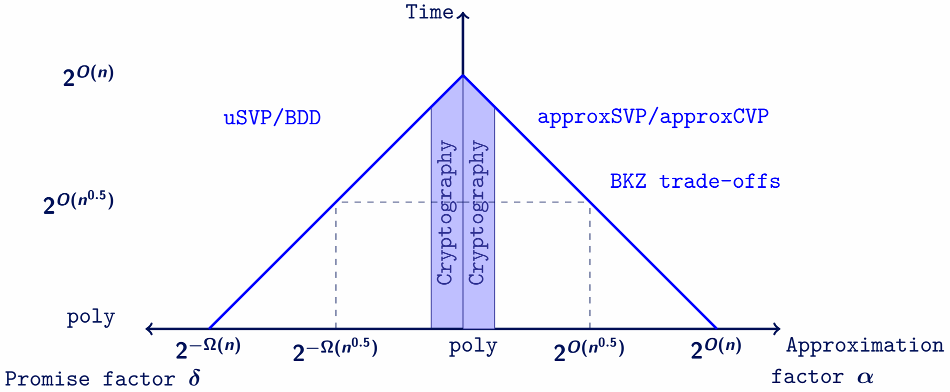


Best Time/Approximation trade-off for SVP, CVP (even quantumly):
BKZ algorithm



Asymptotic hardness of approx-SVP/CVP

Best Time/Approximation trade-off for SVP, CVP (even quantumly):
BKZ algorithm



Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$

Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$
- Lattices can be efficiently represented by a basis

Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$
- Lattices can be efficiently represented by a basis

For large dimension n and small approximation factors the following problems are supposedly hard:

- SVP, approxSVP, uSVP

Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$
- Lattices can be efficiently represented by a basis

For large dimension n and small approximation factors the following problems are supposedly hard:

- SVP, approxSVP, uSVP
- CVP, approxCVP, BDD

Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$
- Lattices can be efficiently represented by a basis

For large dimension n and small approximation factors the following problems are supposedly hard:

- SVP, approxSVP, uSVP
- CVP, approxCVP, BDD

Many more variants possible: search vs decisional, one vs more solutions, ...)

Recap

We have seen:

- Lattices are discrete subgroups of $\mathbb{R}^n$
- Lattices can be efficiently represented by a basis

For large dimension n and small approximation factors the following problems are supposedly hard:

- SVP, approxSVP, uSVP
- CVP, approxCVP, BDD

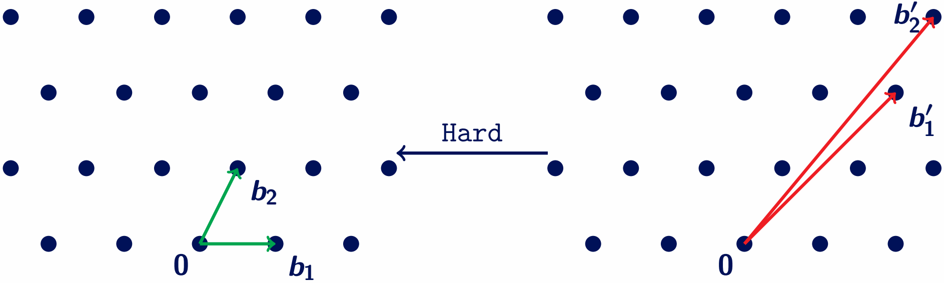
Many more variants possible: search vs decisional, one vs more solutions, ...)

$\boxed{\text{How to build cryptography from this?}}$

Lattice-based cryptography

Good vs bad basis

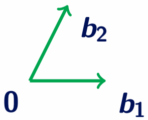
Good basis (Secret key) | Bad basis (Public key)

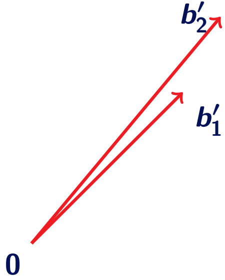


Short and close to orthogonal

Good vs bad basis

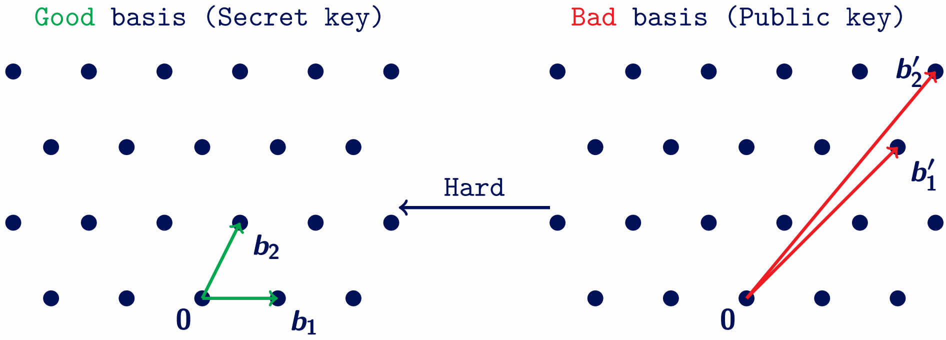
Good basis (Secret key) Bad basis (Public key)





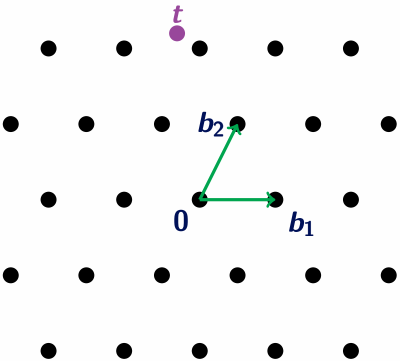
Short and close to orthogonal

Good vs bad basis



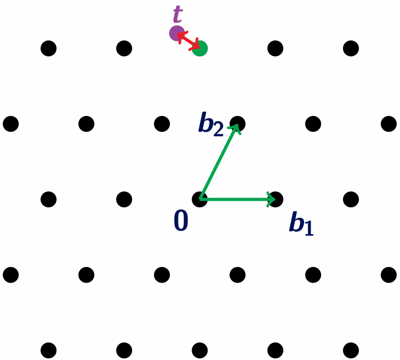
Short and close to orthogonal

Keygen: Generate a random lattice along with a good basis (NTRU, LWE, SIS, ...)



Input: $t = -1.4 \cdot b_1 + 2.2 \cdot b_2$

Solving CVP with a short basis

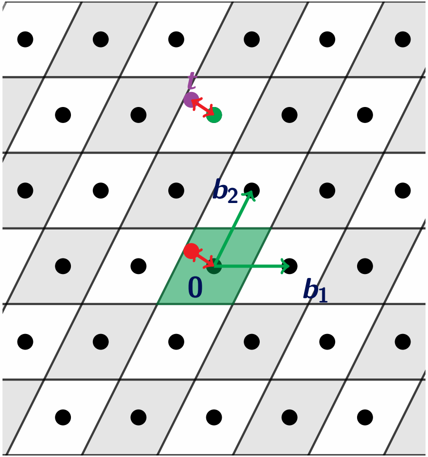


Input: $t = -1.4 \cdot b_1 + 2.2 \cdot b_2$

$\big\downarrow$ round coordinates

Output: $v = -1 \cdot b_1 + 2 \cdot b_2$

Solving CVP with a short basis



Input: $t = -1.4 \cdot b_1 + 2.2 \cdot b_2$

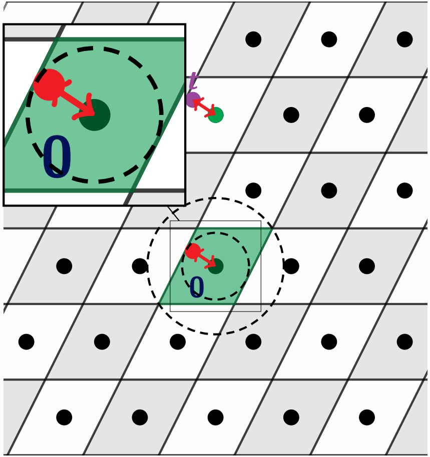
$\downarrow$ round coordinates

Output: $v = -1 \cdot b_1 + 2 \cdot b_2$

$e = t - v = -.4 \cdot b_1 + 0.2 \cdot b_2$

$e \in B \cdot \left[-\frac{1}{2}, \frac{1}{2}\right)^n$

Solving CVP with a short basis



Input: $t = -1.4 \cdot b_1 + 2.2 \cdot b_2$

$\downarrow$ round coordinates

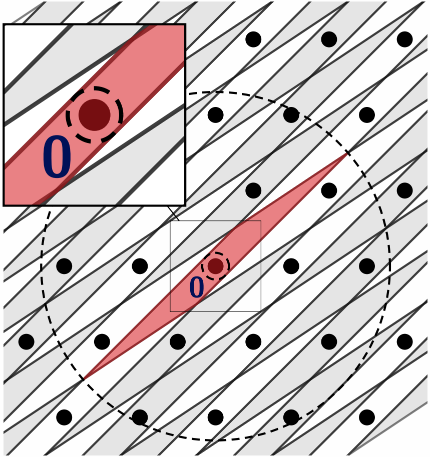
Output: $v = -1 \cdot b_1 + 2 \cdot b_2$

$$e = t - v = -.4 \cdot b_1 + 0.2 \cdot b_2$$

$$e \in B \cdot \left[-\tfrac{1}{2}, \tfrac{1}{2}\right)^n$$

BDD: inner-radius approxCVP: outer-radius

Solving CVP with a short basis



Input: $t = -1.4 \cdot b_1 + 2.2 \cdot b_2$

$\downarrow$ round coordinates

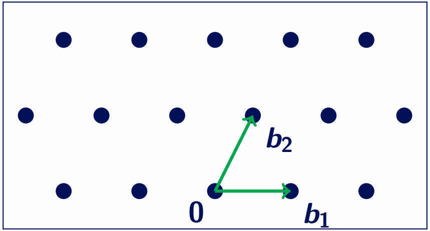
Output: $v = -1 \cdot b_1 + 2 \cdot b_2$

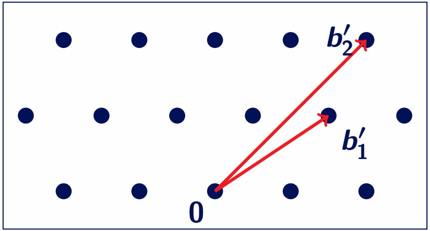
$e = t - v = -.4 \cdot b_1 + 0.2 \cdot b_2$

$e \in B \cdot \left[-\frac{1}{2}, \frac{1}{2}\right)^n$

The better the basis,
the closer the solution

BDD: inner-radius approxCVP: outer-radius

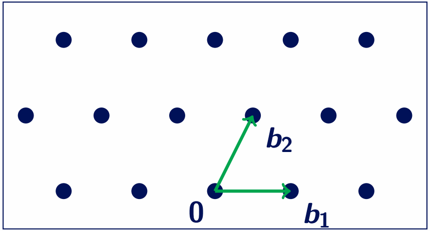


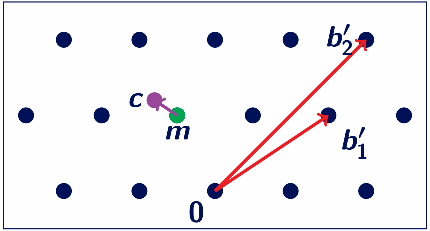


KeyGen:

sk = *good* basis of $\mathcal{L}$.

pk = *bad* basis of $\mathcal{L}$.





KeyGen:

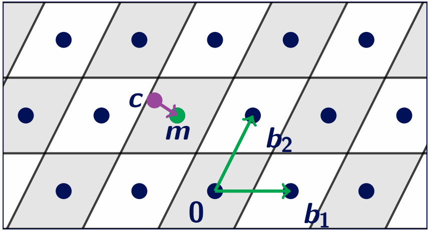
sk = *good* basis of $\mathcal{L}$.

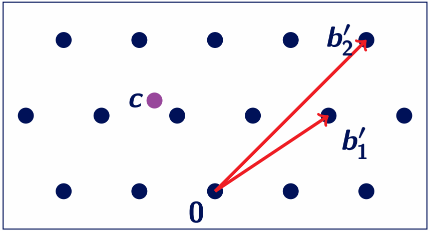
pk = *bad* basis of $\mathcal{L}$.

Encrypt(m, pk) :

Input: encode message $m \in \mathcal{L}$ using **pk**.

Output: noisy message $c = m + e$.





KeyGen:

sk = ***good*** basis of $\mathcal{L}$.

pk = ***bad*** basis of $\mathcal{L}$.

Encrypt(m, pk) :

Input: encode message $m \in \mathcal{L}$ using **pk**.

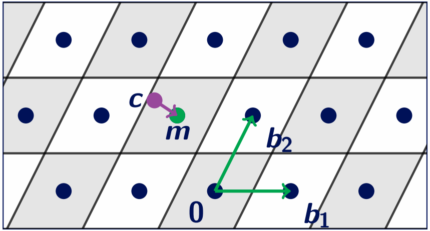
Output: noisy message $c = m + e$.

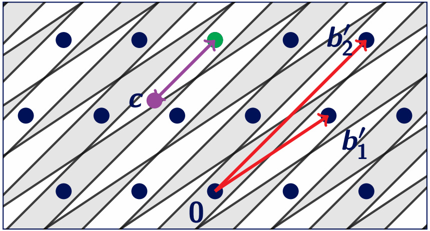
Decrypt(c, sk):

Input: $c = m + e$.

Output: recover m using **sk**.

Encryption via BDD





KeyGen:

sk = ***good*** basis of $\mathcal{L}$.

pk = ***bad*** basis of $\mathcal{L}$.

Encrypt(m, pk) :

Input: encode message $m \in \mathcal{L}$ using **pk**.

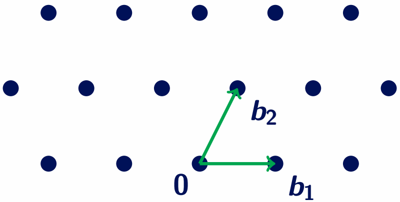
Output: noisy message $c = m + e$.

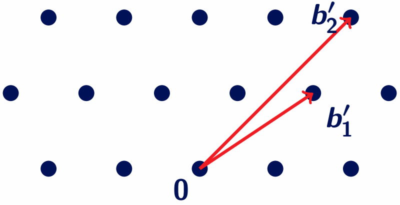
Decrypt(c, sk):

Input: $c = m + e$.

Output: recover m using **sk**.

Assumption: Hard to solve BDD in $\mathcal{L}$ with bad basis.

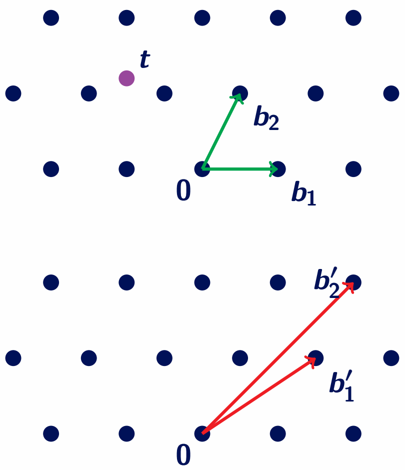




KeyGen:

sk = ***good*** basis of $\mathcal{L}$.

pk = ***bad*** basis of $\mathcal{L}$.



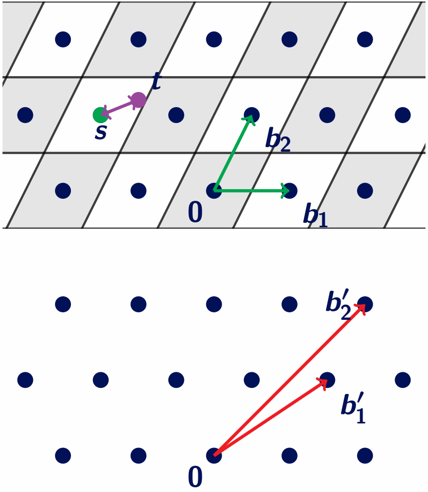
KeyGen:

sk = *good* basis of $\mathcal{L}$.

pk = *bad* basis of $\mathcal{L}$.

Sign(m, sk) :

Hash $\boldsymbol{m}$ to a target $\boldsymbol{t} = \boldsymbol{H}(\boldsymbol{m}) \in \mathbb{R}^n$.



KeyGen:

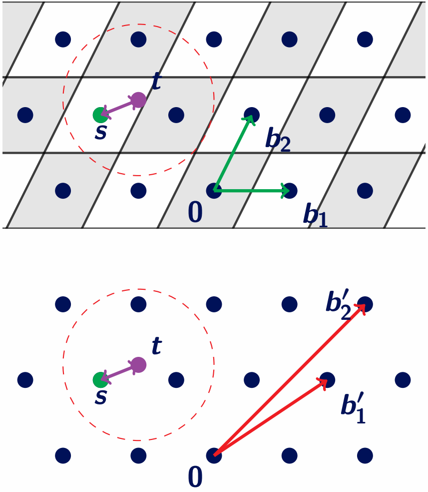
sk = *good* basis of $\mathcal{L}$.

pk = *bad* basis of $\mathcal{L}$.

Sign(m, sk) :

Hash $\boldsymbol{m}$ to a target $\boldsymbol{t} = H(\boldsymbol{m}) \in \mathbb{R}^n$.

Output: $\boldsymbol{s} \in \mathcal{L}$ close to $\boldsymbol{t}$ using **sk**.



KeyGen:

sk = ***good*** basis of $\mathcal{L}$.

pk = ***bad*** basis of $\mathcal{L}$.

Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

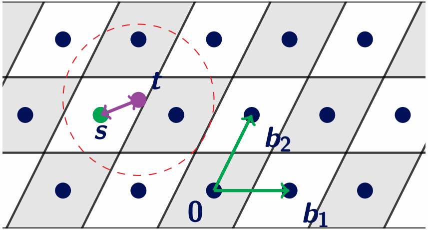
Output: $s \in \mathcal{L}$ close to t using **sk**.

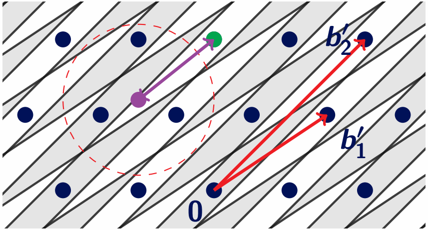
Verify(s, pk):

Check that $s \in \mathcal{L}$ using **pk**.

Check that s is close to $H(m)$.

Hash-and-sign signature scheme via approxCVP





KeyGen:

sk = *good* basis of $\mathcal{L}$.

pk = *bad* basis of $\mathcal{L}$.

Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

Output: $s \in \mathcal{L}$ close to t using sk.

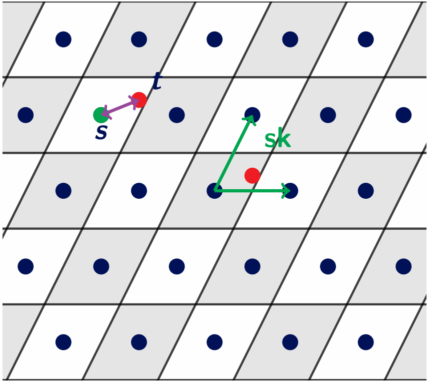
Verify(s, pk):

Check that $s \in \mathcal{L}$ using pk.

Check that s is close to $H(m)$.

Assumption: Hard to solve approxCVP in $\mathcal{L}$ with bad basis.

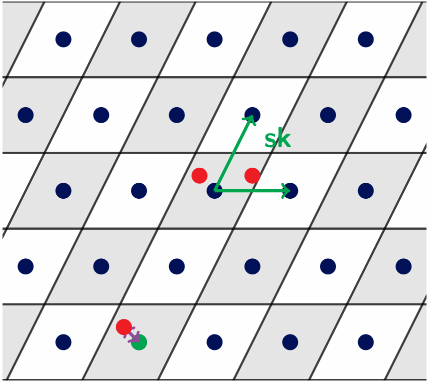
Learning attack on the signature scheme



Parallelepiped attack:

- ask for a signature s on m
- plot $H(m) - s$

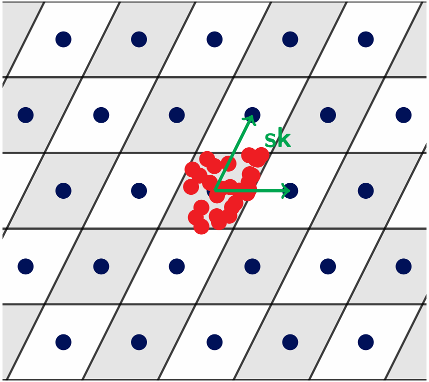
[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology



Parallelepiped attack:

- ask for a signature $\boldsymbol{s}$ on $\boldsymbol{m}$
- plot $\boldsymbol{H}(\boldsymbol{m}) - \boldsymbol{s}$
- repeat

[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology



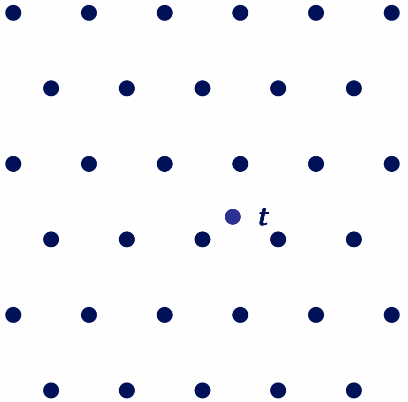
Parallelepiped attack:

- ask for a signature $\boldsymbol{s}$ on $\boldsymbol{m}$
- plot $\boldsymbol{H}(\boldsymbol{m}) - \boldsymbol{s}$
- repeat

From the shape of the parallelepiped, one can recover the short basis

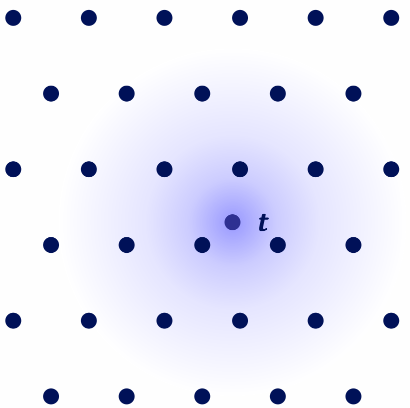
[NR06] Nguyen and Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. J. Cryptology

Preventing the attack



Idea: solve approxCVP randomly

Preventing the attack

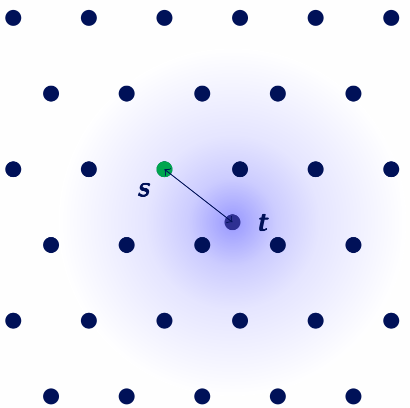


Idea: solve approxCVP randomly

Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

Output: (discrete Gaussian) sample $s \in \mathcal{L}$ close to t using sk.

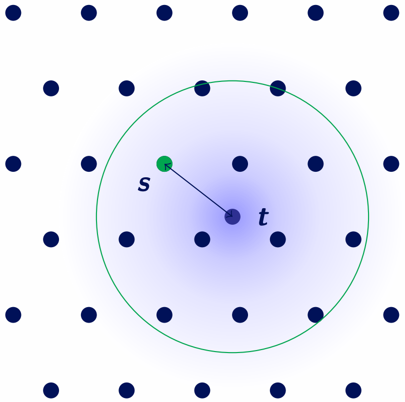


Idea: solve approxCVP randomly

Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

Output: (discrete Gaussian) sample $s \in \mathcal{L}$ close to t using sk.

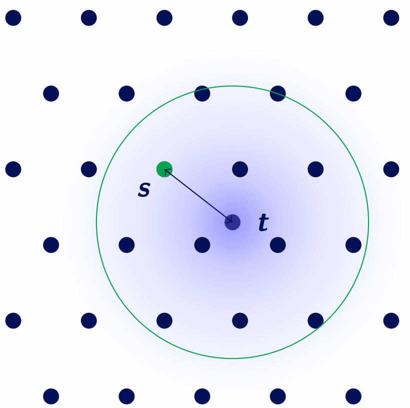


Idea: solve approxCVP randomly

Sign(m, sk) :

Hash $\boldsymbol{m}$ to a target $\boldsymbol{t} = \boldsymbol{H}(\boldsymbol{m}) \in \mathbb{R}^n$.

Output: (discrete Gaussian) sample $\boldsymbol{s} \in \mathcal{L}$ close to $\boldsymbol{t}$ using **sk**.



Idea: solve approxCVP randomly

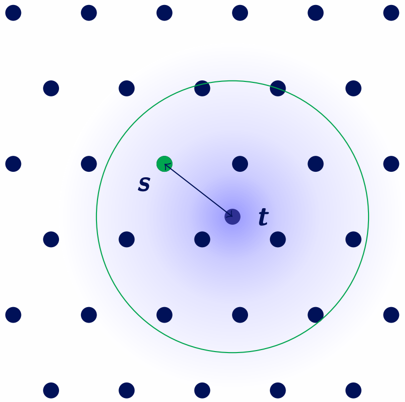
Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

Output: (discrete Gaussian) sample $s \in \mathcal{L}$ close to t using **sk**.

Signature does not depend on secret basis $\Rightarrow$ no leakage!

Preventing the attack



Idea: solve approxCVP randomly

Sign(m, sk) :

Hash m to a target $t = H(m) \in \mathbb{R}^n$.

Output: (discrete Gaussian) sample $s \in \mathcal{L}$ close to t using **sk**.

Signature does not depend on secret basis $\Rightarrow$ no leakage!

FALCON = the above + NTRU lattices.

We have seen:

- BDD is hard (in a family of random lattices) $\Rightarrow$ encryption scheme.

We have seen:

- BDD is hard (in a family of random lattices) $\Rightarrow$ encryption scheme.
- approxCVP is hard (...) $\Rightarrow$ signature scheme.

We have seen:

- BDD is hard (in a family of random lattices) $\Rightarrow$ encryption scheme.
- approxCVP is hard (...) $\Rightarrow$ signature scheme.

More on these families of lattices in part II!

Recap and advanced constructions

We have seen:

- BDD is hard (in a family of random lattices) $\Rightarrow$ encryption scheme.
- approxCVP is hard (...) $\Rightarrow$ signature scheme.

More on these families of lattices in part II!

One can construct many advanced primitives from lattices:

- (fully) homomorphic encryption
- identity based encryption
- functional encryption for linear functions
- ...

Recap and advanced constructions

We have seen:

- BDD is hard (in a family of random lattices) $\Rightarrow$ encryption scheme.
- approxCVP is hard (...) $\Rightarrow$ signature scheme.

How hard?

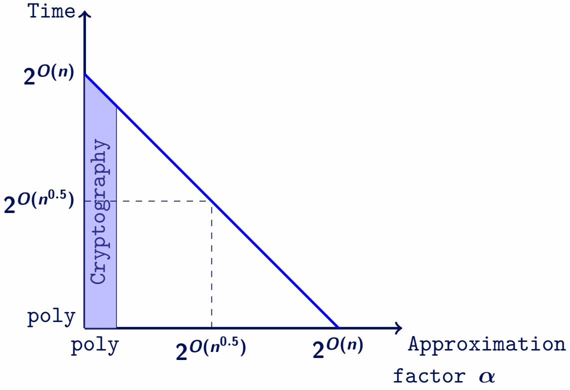
More on these families of lattices in part II!

One can construct many advanced primitives from lattices:

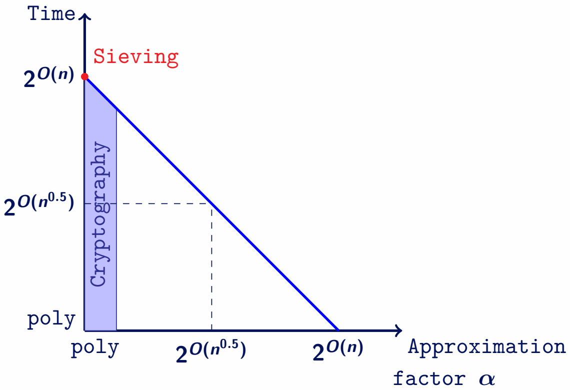
- (fully) homomorphic encryption
- identity based encryption
- functional encryption for linear functions
- ...

Cryptanalysis - Algorithms to solve (approx)SVP

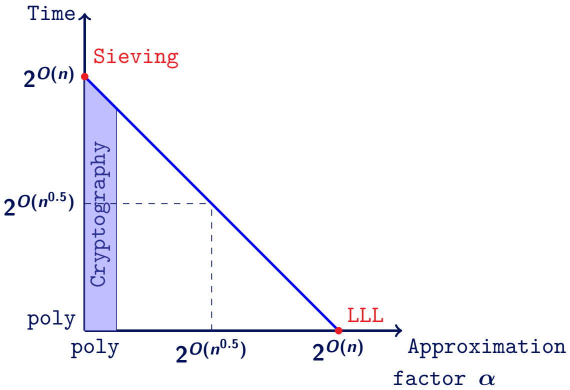
Best Time/Approximation trade-off for SVP, CVP (even quantumly):



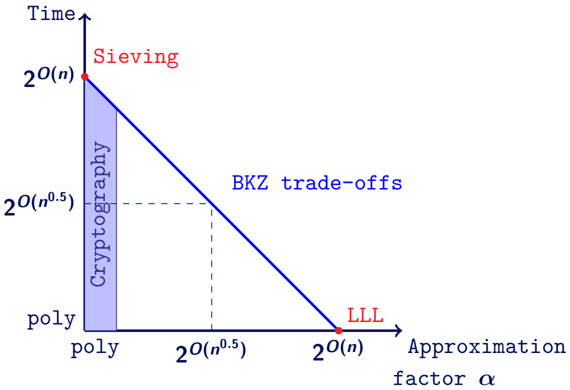
Best Time/Approximation trade-off for SVP, CVP (even quantumly):



Best Time/Approximation trade-off for SVP, CVP (even quantumly):



Best Time/Approximation trade-off for SVP, CVP (even quantumly):



Heuristically solving SVP with lattice sieving

Heuristic assumptions allow to..

Heuristics in lattice-based cryptanalysis

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms
- derive asymptotic and concrete hardness estimates

Heuristics in lattice-based cryptanalysis

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms
- derive asymptotic and concrete hardness estimates

Provable:	worst-case analysis
Heuristic:	simplified average-case analysis

Heuristics in lattice-based cryptanalysis

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms
- derive asymptotic and concrete hardness estimates

Provable:	worst-case analysis
Heuristic:	simplified average-case analysis

Why is this ok for lattice problems?

Heuristics in lattice-based cryptanalysis

Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms
- derive asymptotic and concrete hardness estimates

Provable:	worst-case analysis
Heuristic:	simplified average-case analysis

Why is this ok for lattice problems?

- average-case is often the worst case (see part II!)

Heuristics in lattice-based cryptanalysis

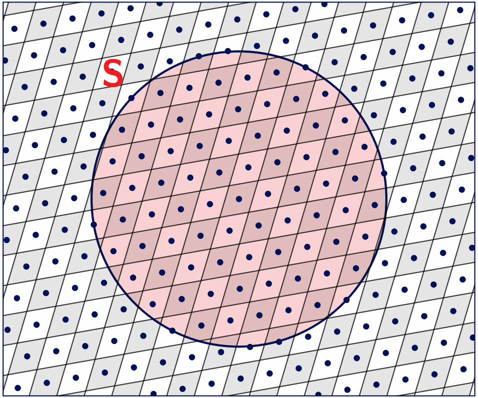
Heuristic assumptions allow to..

- bridge the gap between provable and practical algorithms
- reason about the practical behavior of algorithms
- derive asymptotic and concrete hardness estimates

Provable:	worst-case analysis
Heuristic:	simplified average-case analysis

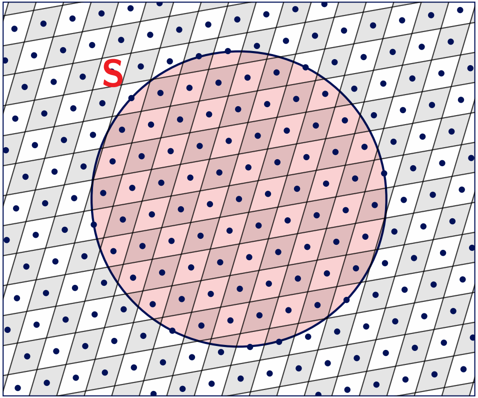
Why is this ok for lattice problems?

- average-case is often the worst case (see part II!)
- matches with practical experiments



For a 'nice' volume $S \subset \mathbb{R}^n$:

$$|S \cap \mathcal{L}| \approx \frac{\mathrm{vol}(S)}{\mathrm{vol}(\mathcal{L})} = \mathrm{vol}(S) \cdot \mathrm{density}(\mathcal{L})$$

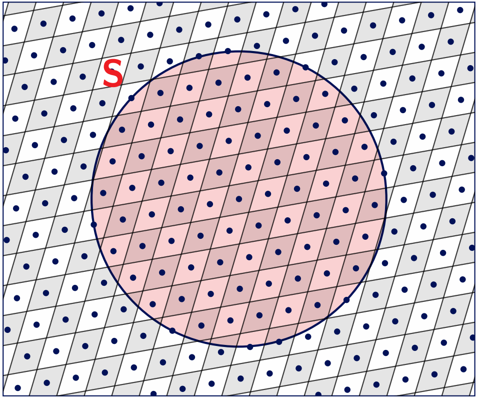


For a 'nice' volume $S \subset \mathbb{R}^n$:

$$|S \cap \mathcal{L}| \approx \frac{\mathrm{vol}(S)}{\mathrm{vol}(\mathcal{L})} = \mathrm{vol}(S) \cdot \mathrm{density}(\mathcal{L})$$

lattice points are uniformly distributed with a certain density.

Gaussian Heuristic

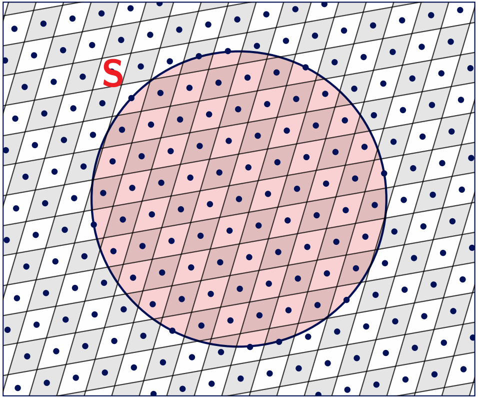


For a 'nice' volume $S \subset \mathbb{R}^n$:

$$|S \cap \mathcal{L}| \approx \frac{\mathrm{vol}(S)}{\mathrm{vol}(\mathcal{L})} = \mathrm{vol}(S) \cdot \mathrm{density}(\mathcal{L})$$

lattice points are uniformly distributed with a certain density.

In theory: true in expectation over all translations of S or for a random lattice $\mathcal{L}$.



For a ‘nice’ volume $S \subset \mathbb{R}^n$:

$$|S \cap \mathcal{L}| \approx \frac{\mathrm{vol}(S)}{\mathrm{vol}(\mathcal{L})} = \mathrm{vol}(S) \cdot \mathrm{density}(\mathcal{L})$$

lattice points are uniformly distributed with a certain density.

In theory: true in expectation over all translations of S or for a random lattice $\mathcal{L}$.

In practice: true for random lattices.
(for a very weak heuristic notion of randomness)

Intermezzo on high dimensional geometry (1)

High dimensional volumes can behave unintuitively

High dimensional volumes can behave unintuitively

$$\mathrm{vol}([-1,1]^n) = 2^n, \quad \mathrm{vol}(\mathcal{B}^n) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} = \left(\frac{2\pi e}{n}\right)^{n/2+o(n)} \to 0$$

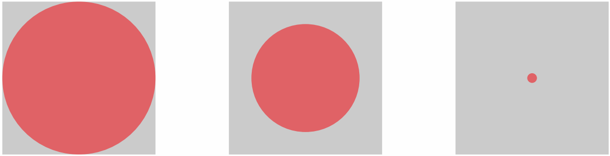
High dimensional volumes can behave unintuitively

$$\mathrm{vol}([-1,1]^n) = 2^n, \quad \mathrm{vol}(\mathcal{B}^n) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} = \left(\frac{2\pi e}{n}\right)^{n/2+o(n)} \to 0$$

$n = 2$ 78.5%

$n = 4$ 31%

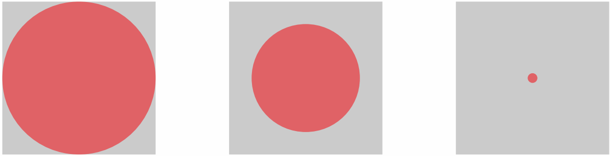
$n = 10$ 0.25%



High dimensional volumes can behave unintuitively

$$\mathrm{vol}([-1,1]^n) = 2^n, \quad \mathrm{vol}(\mathcal{B}^n) = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)} = \left(\frac{2\pi e}{n}\right)^{n/2+o(n)} \to 0$$

$n = 2$ 78.5% | $n = 4$ 31% | $n = 10$ 0.25%



n-dimensional balls with a fixed radius ‘disappear’ for large n.

Intermezzo on high dimensional geometry (2)

Scaling by R changes volume by factor R^n.

Intermezzo on high dimensional geometry (2)

Scaling by $\boldsymbol{R}$ changes volume by factor $\boldsymbol{R^n}$.

Example: suppose we have a ball $\gamma \cdot \mathcal{B}^{500}$ with the same volume as a $\mathbf{500}$-dimensional lattice $\mathcal{L} \subset \mathbb{R}^{500}$.

Intermezzo on high dimensional geometry (2)

Scaling by $\boldsymbol{R}$ changes volume by factor $\boldsymbol{R^n}$.

Example: suppose we have a ball $\gamma \cdot \mathcal{B}^{500}$ with the same volume as a $\mathbf{500}$-dimensional lattice $\mathcal{L} \subset \mathbb{R}^{500}$.

Gaussian Heuristic $\Rightarrow$

$$\left|\left(\gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1$$

$$\left|\left(1.05 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1.05^{500} = 3.9 \cdot 10^{10}$$

$$\left|\left(0.95 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| = 7.3 \cdot 10^{-12} \approx 0$$

Intermezzo on high dimensional geometry (2)

Scaling by R changes volume by factor R^n.

Example: suppose we have a ball $\gamma \cdot \mathcal{B}^{500}$ with the same volume as a 500-dimensional lattice $\mathcal{L} \subset \mathbb{R}^{500}$.

Gaussian Heuristic $\Rightarrow$

$$\left|\left(\gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1$$

$$\left|\left(1.05 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1.05^{500} = 3.9 \cdot 10^{10}$$

$$\left|\left(0.95 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| = 7.3 \cdot 10^{-12} \approx 0$$

$$\boxed{\lambda_1(\mathcal{L}) \approx \gamma.}$$

Scaling by R changes volume by factor R^n.

Example: suppose we have a ball $\gamma \cdot \mathcal{B}^{500}$ with the same volume as a 500-dimensional lattice $\mathcal{L} \subset \mathbb{R}^{500}$.

Gaussian Heuristic $\Rightarrow$

$$\left|\left(\gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1$$

$$\left|\left(1.05 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1.05^{500} = 3.9 \cdot 10^{10}$$

$$\left|\left(0.95 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| = 7.3 \cdot 10^{-12} \approx 0$$

$$\boxed{\lambda_1(\mathcal{L}) \approx \gamma.}$$

$$\boxed{\lambda_1 \approx \mathrm{gh}(\mathcal{L}) := \frac{\mathrm{vol}(\mathcal{L})^{1/n}}{\mathrm{vol}(\mathcal{B}^n)^{1/n}} \sim \sqrt{n/2\pi e} \cdot \mathrm{vol}(\mathcal{L})^{1/n}.}$$

Scaling by R changes volume by factor R^n.

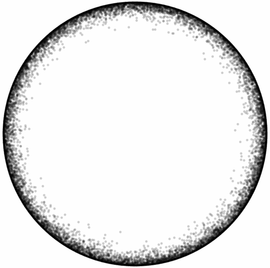
Example: suppose we have a ball $\gamma \cdot \mathcal{B}^{500}$ with the same volume as a 500-dimensional lattice $\mathcal{L} \subset \mathbb{R}^{500}$.

Gaussian Heuristic $\Rightarrow$

$$\left|\left(\gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1$$

$$\left|\left(1.05 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| \approx 1.05^{500} = 3.9 \cdot 10^{10}$$

$$\left|\left(0.95 \cdot \gamma \cdot \mathcal{B}^{500} \setminus \{0\}\right) \cap \mathcal{L}\right| = 7.3 \cdot 10^{-12} \approx 0$$



$$\lambda_1(\mathcal{L}) \approx \gamma.$$

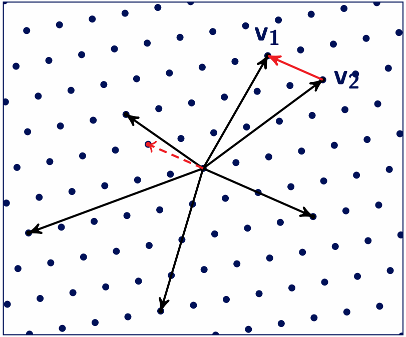
$$\lambda_1 \approx \mathrm{gh}(\mathcal{L}) := \frac{\mathrm{vol}(\mathcal{L})^{1/n}}{\mathrm{vol}(\mathcal{B}^n)^{1/n}} \sim \sqrt{n/2\pi e} \cdot \mathrm{vol}(\mathcal{L})^{1/n}.$$

SVP via Lattice Sieving

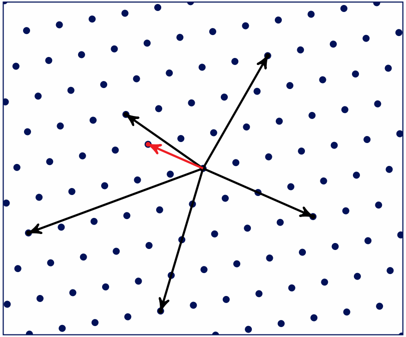
1. Sample a list $\boldsymbol{L} \subset \mathcal{L}$ of (long) lattice vectors.

SVP via Lattice Sieving

1. Sample a list $\mathbf{L} \subset \mathcal{L}$ of (long) lattice vectors.
2. Repeat:



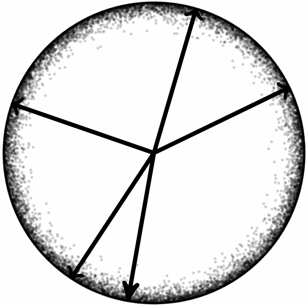
Find close vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbf{L}$.



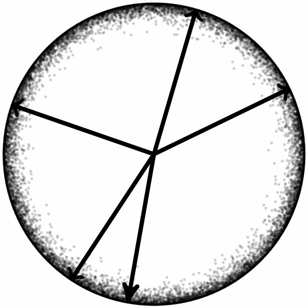
Replace $\mathbf{v}_2 \leftarrow \mathbf{v}_1 - \mathbf{v}_2$.

Heuristic complexity analysis

Start with a list $\boldsymbol{L}$ of $\boldsymbol{N}$ vectors of length $\leq \gamma$.



Start with a list L of
N vectors of length $\leq \gamma$.



Heuristic assumption

vectors in list L have uniform directions.

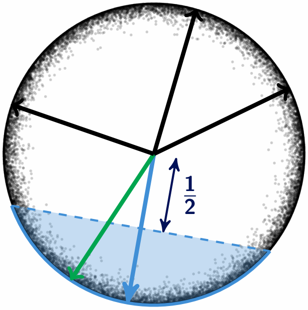
Start with a list L of
N vectors of length $\leq \gamma$.

Heuristic assumption

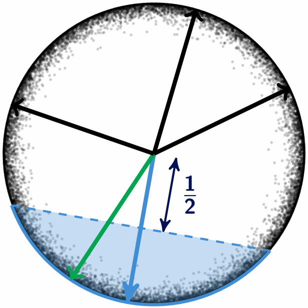
vectors in list L have uniform directions.

Probability $\|v_1 - v_2\| \leq 0.999 \cdot \gamma$ equals

relative volume spherical cap $\approx (3/4 + \epsilon)^{n/2+o(n)}$



Start with a list $\boldsymbol{L}$ of
$\boldsymbol{N}$ vectors of length $\leq \gamma$.



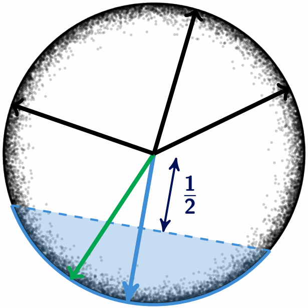
Heuristic assumption

vectors in list $\boldsymbol{L}$ have uniform directions.

Probability $\|\boldsymbol{v}_1 - \boldsymbol{v}_2\| \leq \mathbf{0.999} \cdot \gamma$ equals
relative volume spherical cap $\approx (3/4 + \epsilon)^{n/2+o(n)}$

$\boldsymbol{N}^2$ pairs, new list size $\boldsymbol{N}$, so need $\boldsymbol{N}^2 \cdot (3/4)^{n/2} \geq \boldsymbol{N}$.

Start with a list L of
N vectors of length $\leq \gamma$.



<u>Heuristic assumption</u>

vectors in list L have uniform directions.

Probability $\|v_1 - v_2\| \leq 0.999 \cdot \gamma$ equals

relative volume spherical cap $\approx (3/4 + \epsilon)^{n/2+o(n)}$

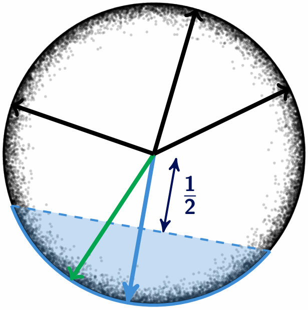
N^2 pairs, new list size N, so need $N^2 \cdot (3/4)^{n/2} \geq N$.

Space: $N \cdot \mathrm{poly}(n) = (4/3)^{n/2+o(n)} = 2^{0.2075+o(n)}$

Time: $N^2 \cdot \mathrm{poly}(n) = (4/3)^{n+o(n)} = 2^{0.415n+o(n)}$.

Heuristic complexity analysis

Start with a list L of
N vectors of length $\leq \gamma$.



Heuristic assumption

vectors in list L have uniform directions.

Probability $\|v_1 - v_2\| \leq 0.999 \cdot \gamma$ equals
relative volume spherical cap $\approx (3/4 + \epsilon)^{n/2+o(n)}$

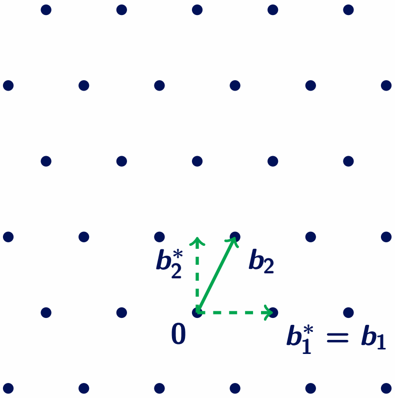
N^2 pairs, new list size N, so need $N^2 \cdot (3/4)^{n/2} \geq N$.

Space: $N \cdot \mathrm{poly}(n) = (4/3)^{n/2+o(n)} = 2^{0.2075+o(n)}$
Time: $N^2 \cdot \mathrm{poly}(n) = (4/3)^{n+o(n)} = 2^{0.415n+o(n)}$.

Can be improved to $2^{0.292n+o(n)}$!.

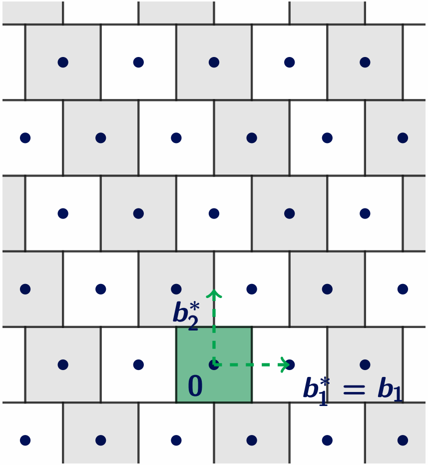
Solving approxSVP/CVP via basis reduction

Gram-Schmidt Orthogonalisation



GSO: $b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$

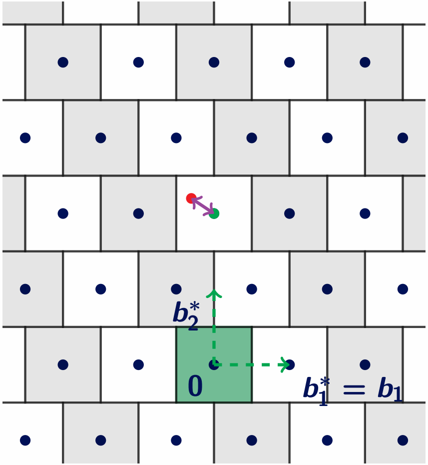
Gram-Schmidt Orthogonalisation



GSO: $b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$

Fundamental Area: $\mathcal{F}_{B^*} := \prod_{i=1}^{k} \left[-\frac{1}{2}b_i^*, \frac{1}{2}b_i^*\right)$

Gram-Schmidt Orthogonalisation



GSO: $b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$

Fundamental Area: $\mathcal{F}_{B^*} := \prod_{i=1}^{k} \left[-\frac{1}{2}b_i^*, \frac{1}{2}b_i^*\right)$

Nearest plane algorithm

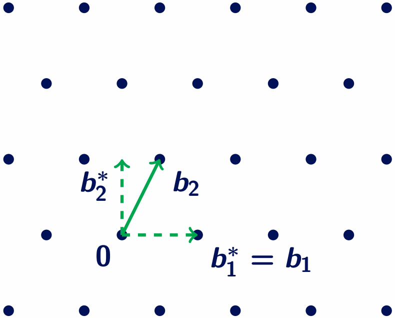
Input: target $t = e$

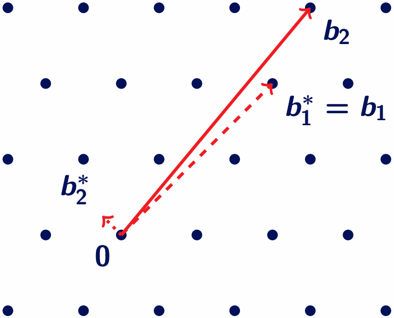
For $j = n, \ldots, 1$:

$$e \leftarrow e - \left\lfloor \frac{\langle e, b_i^* \rangle}{\langle b_i^*, b_i^* \rangle} \right\rceil b_i \, .$$

Output: $e \in \mathcal{F}_{B^*}$

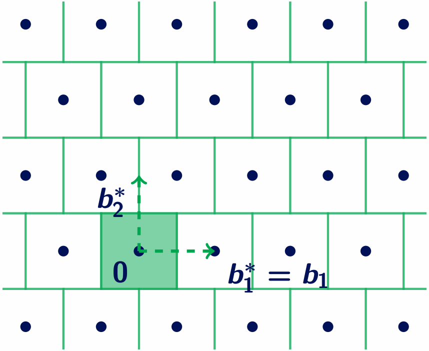
$$b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$$

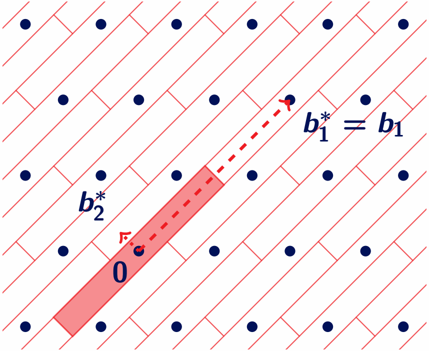




$$b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$$

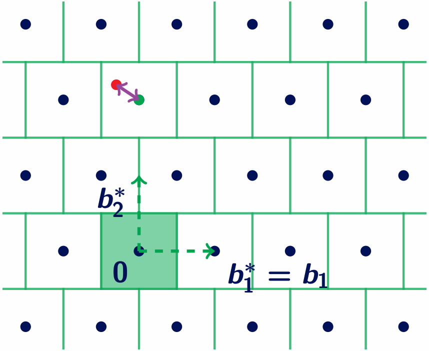
$$\boxed{\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{F}_{B^*}) = \prod_{i=1}^{k} \|b_i^*\|}$$

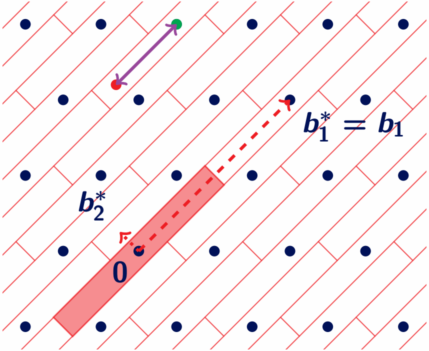




$$b_i^* := \underbrace{\pi_{(b_1,\dots,b_{i-1})^\perp}}_{\pi_i}(b_i)$$

$$\boxed{\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{F}_{B^*}) = \prod_{i=1}^{k} \|b_i^*\|}$$





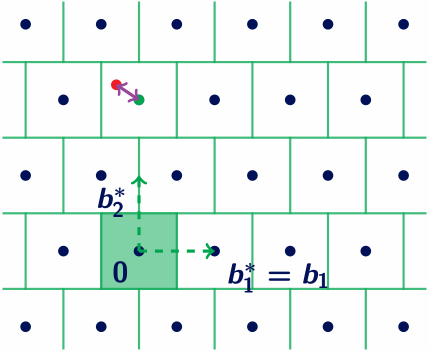
BDD: $\|e\| < \frac{1}{2} \min_i \|b_i^*\|$,

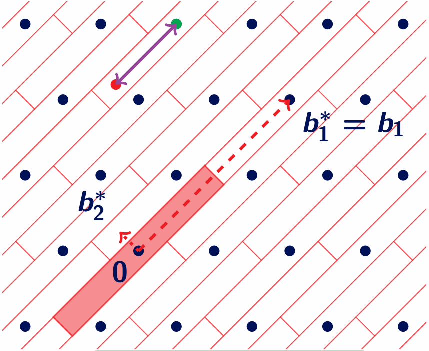
approxCVP: $\|e\|^2 \leq \frac{1}{4} \sum_i \|b_i^*\|^2$.

Good vs Bad basis

$$b_i^* := \underbrace{\pi_{(b_1,\ldots,b_{i-1})^\perp}}_{\pi_i}(b_i)$$

$$\mathrm{vol}(\mathcal{L}) = \mathrm{vol}(\mathcal{F}_{B^*}) = \prod_{i=1}^{k} \|b_i^*\|$$



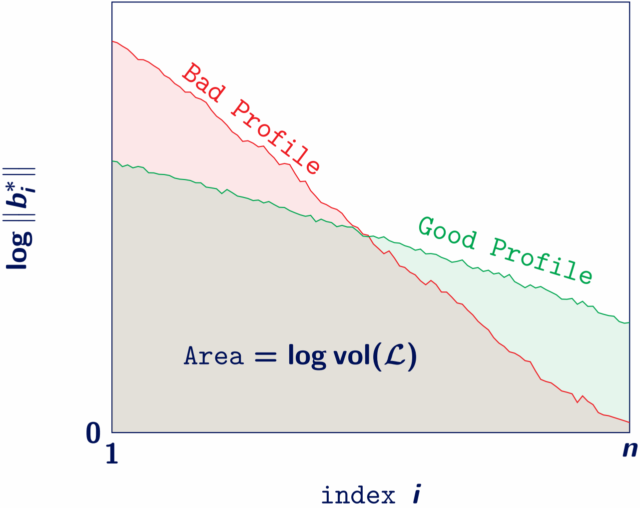


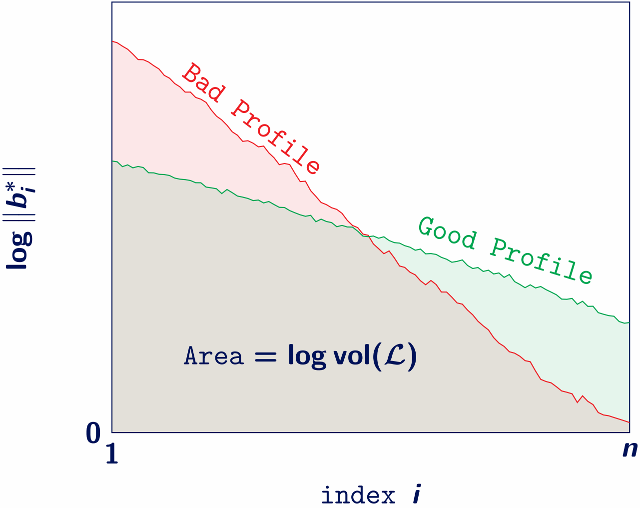
BDD: $\|e\| < \frac{1}{2}\min_i \|b_i^*\|$,

approxCVP: $\|e\|^2 \leq \frac{1}{4}\sum_i \|b_i^*\|^2$.

Good Basis:
$\|b_1^*\| \approx \ldots \approx \|b_k^*\|$

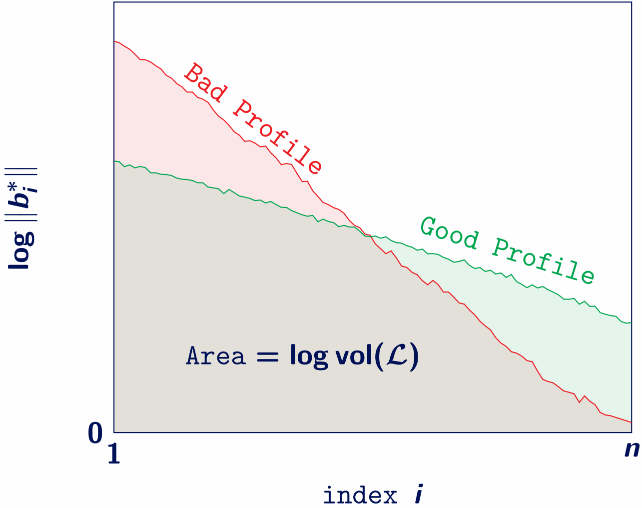
Basis Profile





Basis profile

Measures the length and orthogonality of a basis



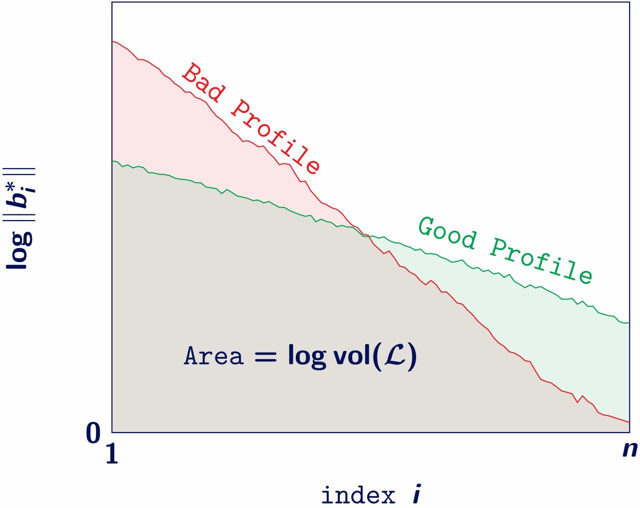
Basis profile

Measures the length and orthogonality of a basis

Basis reduction

Input: Bad basis

Output: good basis



Basis profile

Measures the length and orthogonality of a basis

Basis reduction

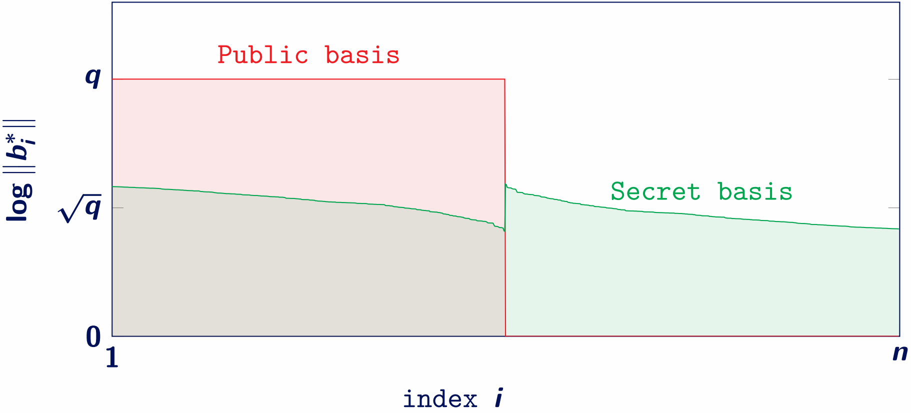
Input: Bad basis

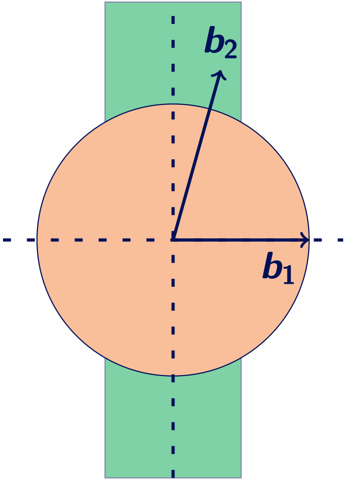
Output: good basis

Flatten the profile!

Example: NTRU public vs secret basis

public and secret bases generated from the NTRU problem





Wristwatch Lemma

For any lattice $\mathcal{L}$ of rank 2
there exists a basis (b_1, b_2) s.t.

$$\|b_1\| \leq \|b_2\|$$

$$|\langle b_1, b_2 \rangle| \leq \frac{1}{2}\|b_1\|$$

$$\Downarrow$$

$$\boxed{\|b_1^*\| \leq \sqrt{\frac{4}{3}} \cdot \|b_2^*\|}$$

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange Reduced for all $i < n$.

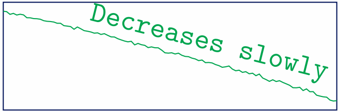
Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange Reduced for all $i < n$.

$$\Downarrow$$

$$\forall i < n,\ \left\| b_i^* \right\| \leq \sqrt{4/3} \cdot \left\| b_{i+1}^* \right\|$$

$\log \left\| b_i^* \right\|$



index i

<u>Definition</u>

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange Reduced for all $i < n$.

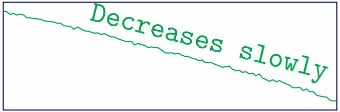
$$\Downarrow$$

$$\forall i < n,\ \|b_i^*\| \leq \sqrt{4/3} \cdot \left\|b_{i+1}^*\right\|$$

$$\Downarrow$$

$$\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$

$\log \|b_i^*\|$



index i

LLL Reduction

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange Reduced for all $i < n$.

Algorithm

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not Lagrange Reduced, Langrange Reduce it.

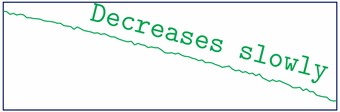
$$\Downarrow$$

$$\forall i < n,\ \|b_i^*\| \leq \sqrt{4/3} \cdot \|b_{i+1}^*\|$$

$$\Downarrow$$

$$\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$

$\log \|b_i^*\|$



index i

LLL Reduction

Definition

A basis B of $\mathcal{L}$ is LLL-reduced if $(\pi_i(b_i), \pi_i(b_{i+1}))$ is Lagrange Reduced for all $i < n$.

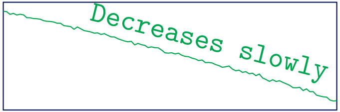
$$\Downarrow$$

$$\forall i < n, \ \left\|b_i^*\right\| \leq \sqrt{4/3} \cdot \left\|b_{i+1}^*\right\|$$

$$\Downarrow$$

$$\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$

$\log \left\|b_i^*\right\|$



index i

Algorithm

While $\exists i$ s.t. $(\pi_i(b_i), \pi_i(b_{i+1}))$ is not Lagrange Reduced, Langrange Reduce it.

Termination in poly-time:

Requires a slight relaxation. (ϵ-Lagrange Reduced)

Proof argument:

$$P = \sum_{i \leq n} (n+1-i) \cdot \log \left\|b_i^*\right\|$$

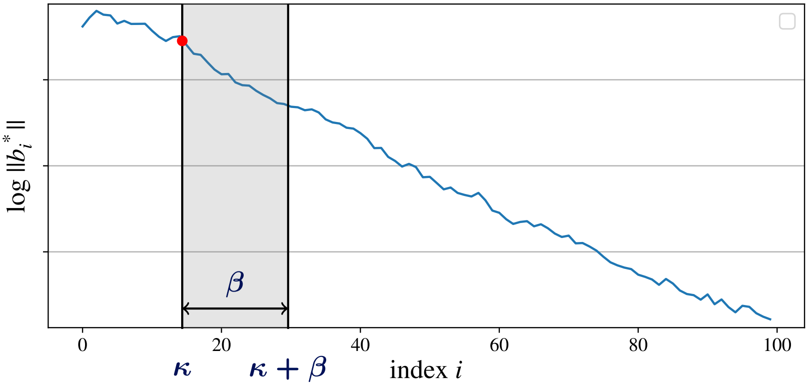
Decreases by ϵ at each step and is lower-bounded.

BKZ algorithm

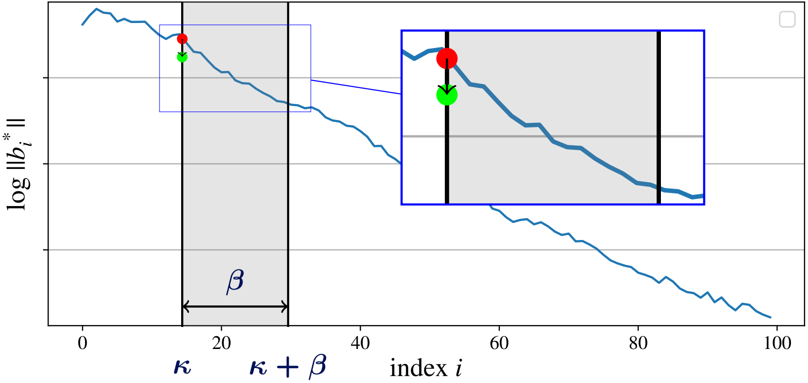
- Define the projected sublattice basis $\boldsymbol{B}_{l:r} := (\pi_l(\boldsymbol{b}_l), \ldots, \pi_l(\boldsymbol{b}_{r-1}))$.

BKZ algorithm

- Define the projected sublattice basis $B_{l:r} := (\pi_l(b_l), \ldots, \pi_l(b_{r-1}))$.
- For $\kappa = 1, \ldots, n$ solve SVP in $\mathcal{L}(B_{\kappa:\min\{n+1,\kappa+\beta\}})$ and replace b_κ.

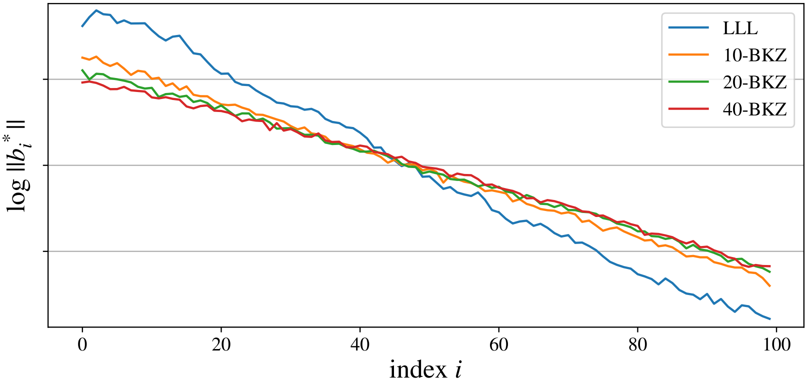


- Define the projected sublattice basis $B_{l:r} := (\pi_l(b_l), \ldots, \pi_l(b_{r-1}))$.
- For $\kappa = 1, \ldots, n$ solve SVP in $\mathcal{L}(B_{\kappa:\min\{n+1,\kappa+\beta\}})$ and replace b_κ.

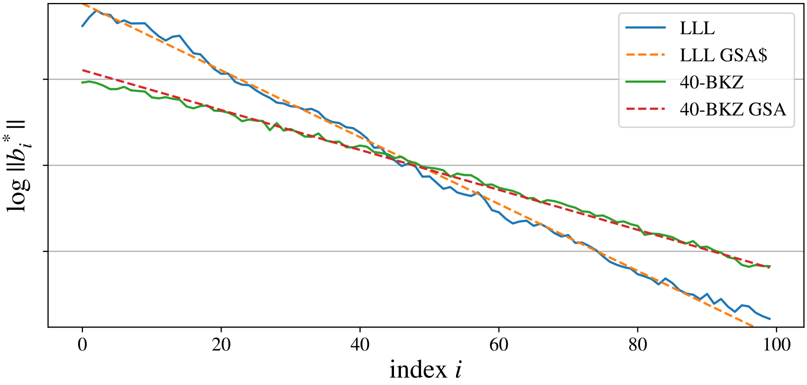


BKZ algorithm

- Define the projected sublattice basis $B_{l:r} := (\pi_l(b_l), \ldots, \pi_l(b_{r-1}))$.
- For $\kappa = 1, \ldots, n$ solve SVP in $\mathcal{L}(B_{\kappa:\min\{n+1,\kappa+\beta\}})$ and replace b_κ.
- Reduction better for larger blocksize β, but cost $2^{0.292\beta+o(n)}$.



- Define the projected sublattice basis $B_{l:r} := (\pi_l(b_l), \dots, \pi_l(b_{r-1}))$.
- For $\kappa = 1, \dots, n$ solve SVP in $\mathcal{L}(B_{\kappa:\min\{n+1,\kappa+\beta\}})$ and replace b_κ.
- Reduction better for larger blocksize β, but cost $2^{0.292\beta+o(n)}$.
- Behaviour well understood for 'random' lattices. [GSA]



Recap

We have seen:

- SVP can be solved in time $2^{0.292n+o(n)}$ via lattice sieving

Recap

We have seen:

- SVP can be solved in time $2^{0.292n+o(n)}$ via lattice sieving
- Lattice reduction: flattening the basis profile

We have seen:

- SVP can be solved in time $2^{0.292n+o(n)}$ via lattice sieving
- Lattice reduction: flattening the basis profile

LLL algorithm:

$$\boxed{\begin{array}{c}\text{SVP for rank } 2 \\ \text{(Lagrange reduction)}\end{array}} \longrightarrow \boxed{\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}}$$

Recap

We have seen:

- SVP can be solved in time $2^{0.292n+o(n)}$ via lattice sieving
- Lattice reduction: flattening the basis profile

LLL algorithm:

SVP for rank 2 (Lagrange reduction)	$\longrightarrow$	$\|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$

BKZ algorithm:

SVP for rank β (sieving)	$\longrightarrow$	$\|b_1\| \leq O(\beta)^{\frac{n-1}{2(\beta-1)}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$

Recap

We have seen:

- SVP can be solved in time $2^{0.292n+o(n)}$ via lattice sieving
- Lattice reduction: flattening the basis profile

LLL algorithm:

SVP for rank 2
(Lagrange reduction)

$$\longrightarrow \quad \|b_1\| \leq \sqrt{4/3}^{\frac{n-1}{2}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$

BKZ algorithm:

SVP for rank β
(sieving)

$$\longrightarrow \quad \|b_1\| \leq O(\beta)^{\frac{n-1}{2(\beta-1)}} \cdot \mathrm{vol}(\mathcal{L})^{1/n}$$

- Same algorithms also solve promise variants uSVP and BDD

Conclusion

We have seen:

- Basics of lattice theory and hard problems

Conclusion

We have seen:

- Basics of lattice theory and hard problems
- How these hard problems can be used for cryptography

Conclusion

We have seen:

- Basics of lattice theory and hard problems
- How these hard problems can be used for cryptography
- The best (known) algorithms to solve these problems

Conclusion

We have seen:

- Basics of lattice theory and hard problems
- How these hard problems can be used for cryptography
- The best (known) algorithms to solve these problems

What's next?

- Keygen: what families of lattices to use? (SIS, LWE, NTRU, ...)

Conclusion

We have seen:

- Basics of lattice theory and hard problems
- How these hard problems can be used for cryptography
- The best (known) algorithms to solve these problems

What's next?

- Keygen: what families of lattices to use? (SIS, LWE, NTRU, ...)
- Why do we trust these lattices? (hardness reductions)

Conclusion

We have seen:

- Basics of lattice theory and hard problems
- How these hard problems can be used for cryptography
- The best (known) algorithms to solve these problems

What's next?

- Keygen: what families of lattices to use? (SIS, LWE, NTRU, ...)
- Why do we trust these lattices? (hardness reductions)
- More efficiency: algebraic lattices (ideal and module lattices)

Part II

Plan

Part I

Lattice theory

- Lattices
- Hard problems

Cryptography

- Trapdoor bases
- Encryption, Signature

Cryptanalysis

- Lattice Sieving
- Basis Reduction

Part II

Lattices used in cryptography

- SIS, LWE, decLWE
- Security proofs

Hardness Reductions

- search to decision
- WC to AC reductions

Algebraic Lattices

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

Limitations of SVP (and CVP)

SVP and CVP are hard in the worst case

Limitations of SVP (and CVP)

SVP and CVP are hard in the worst case

- no efficient algorithm that works for any lattice

SVP and CVP are hard in the worst case

- no efficient algorithm that works for any lattice
- but for some lattice it might be easier

Limitations of SVP (and CVP)

SVP and CVP are hard in the worst case

- no efficient algorithm that works for any lattice
- but for some lattice it might be easier

For crypto, we need problems that are hard on average

(i.e., for a random instance, the problem is hard with overwhelming probability)

random q-ary lattices

q-ary lattices

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

- A lattice $\mathcal{L} \subset \mathbb{R}^m$ of dimension m is called q-ary if

$$q\mathbb{Z}^m \subset \mathcal{L} \subset \mathbb{Z}^m.$$

- Let $A \in \mathbb{Z}_q^{m \times n}$, then we define the row-generated q-ary lattice

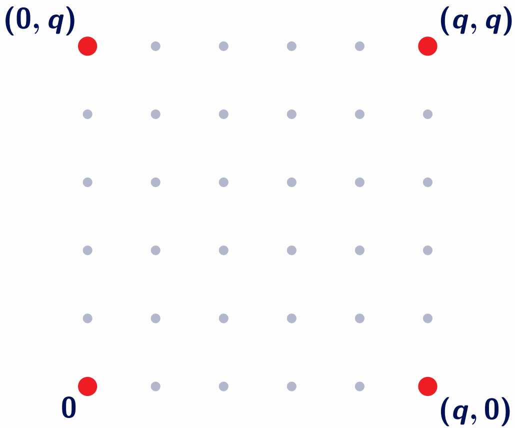
$$\Lambda_q(A) := \{y \in \mathbb{Z}^m : y \equiv Ax \bmod q \text{ for some } x \in \mathbb{Z}_q^n\} = A\mathbb{Z}^n + q\mathbb{Z}^m$$

- and the parity-check q-ary lattice

$$\Lambda_q^{\perp}(A) := \{x \in \mathbb{Z}^m : x^\top A \equiv 0 \bmod q\} = \ker(A^\top : \mathbb{Z}^m \to \mathbb{Z}_q^n)$$

- Exercise: if q prime and A has full column-rank, then

$$\mathrm{vol}(\Lambda_q(A)) = q^{m-n}, \quad \mathrm{vol}(\Lambda_q^{\perp}(A)) = q^n$$

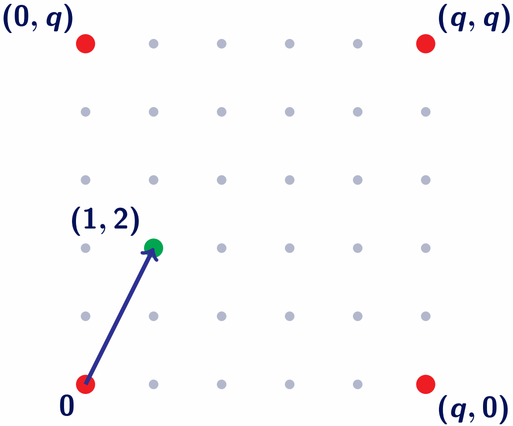


Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

Example

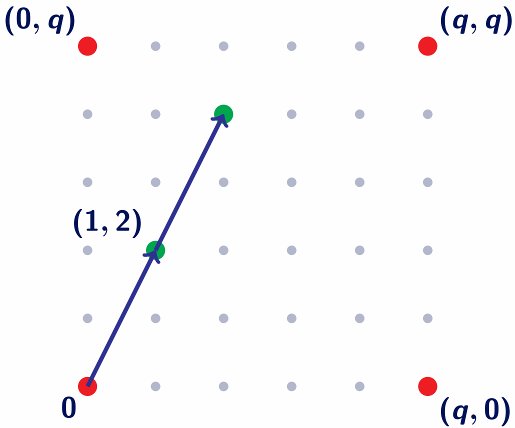


Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

Example

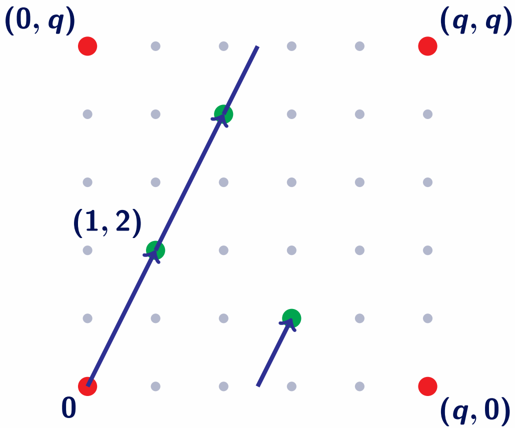


Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

Example

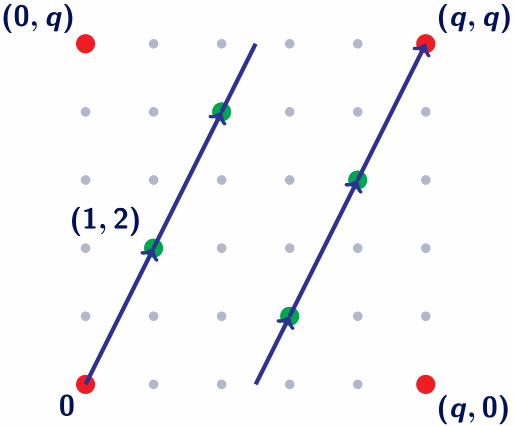


Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

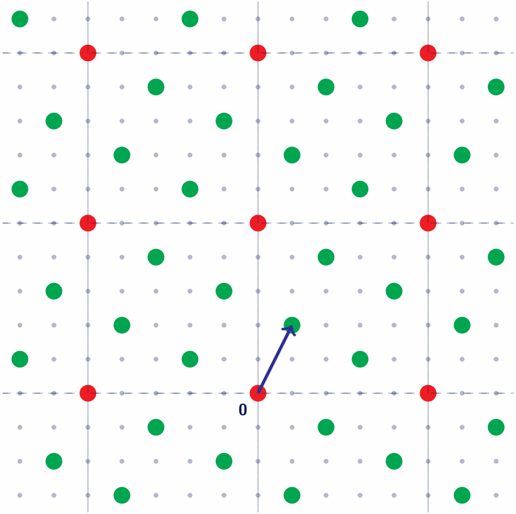
Example



Suppose $q = 5, n = 1, m = 2,$

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

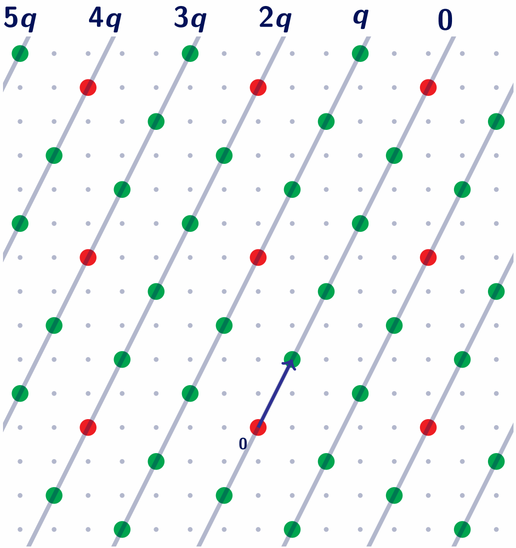


Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

Example



Suppose $q = 5, n = 1, m = 2$,

$$A = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$\Lambda_q(A) = A\mathbb{Z}^n + q\mathbb{Z}^m = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \cdot \mathbb{Z} + 5\mathbb{Z}^2$$

Parity-check representation:

$$\Lambda_q\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix}\right) = \Lambda_q^{\perp}\left(\begin{pmatrix} -2 \\ 1 \end{pmatrix}\right)$$

$$= \{(x, y) \in \mathbb{Z}^2 : -2x + y \equiv 0 \bmod q\}$$

Family of random q-ary lattices

- Random $\boldsymbol{q}$-ary lattice: sample $\boldsymbol{A} \in \mathcal{U}\left(\mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{m}\times\boldsymbol{n}}\right)$, and consider $\boldsymbol{\Lambda}_{\boldsymbol{q}}(\boldsymbol{A})$

Family of random q-ary lattices

- Random $\boldsymbol{q}$-ary lattice: sample $\boldsymbol{A} \in \mathcal{U}\left(\mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{m} \times \boldsymbol{n}}\right)$, and consider $\boldsymbol{\Lambda}_{\boldsymbol{q}}(\boldsymbol{A})$
- equivalently: sample $\boldsymbol{A} \in \mathcal{U}\left(\mathbb{Z}_{\boldsymbol{q}}^{\boldsymbol{m} \times (\boldsymbol{m}-\boldsymbol{n})}\right)$, and consider $\boldsymbol{\Lambda}_{\boldsymbol{q}}^{\perp}(\boldsymbol{A})$

- Random q-ary lattice: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times n}\right)$, and consider $\Lambda_q(A)$
- equivalently: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times(m-n)}\right)$, and consider $\Lambda_q^{\perp}(A)$
- Defines average-case problems!

Family of random q-ary lattices

- Random q-ary lattice: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m \times n}\right)$, and consider $\Lambda_q(A)$
- equivalently: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m \times (m-n)}\right)$, and consider $\Lambda_q^{\perp}(A)$
- Defines average-case problems!
- For $X \in \{$approxSVP, approxCVP, uSVP, BDD$\}$ and $m = \mathrm{poly}(n)$ we have

Solving X in any lattice of rank m	$\geq$	Solving X with non-negligible prob. in a random q-ary lattice

- These average-case problems are also known as (I)SIS and LWE.

Family of random q-ary lattices

- Random q-ary lattice: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m \times n}\right)$, and consider $\Lambda_q(A)$
- equivalently: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m \times (m-n)}\right)$, and consider $\Lambda_q^{\perp}(A)$
- Defines average-case problems!
- For $X \in \{$approxSVP, approxCVP, uSVP, BDD$\}$ and $m = \mathrm{poly}(n)$ we have

Solving X in any lattice of rank m	$\geq$	Solving X with non-negligible prob. in a random q-ary lattice	$\gtrsim$	Solving approx-SVP in any lattice of rank $\min(n, m-n)$

- These average-case problems are also known as (I)SIS and LWE.

Family of random q-ary lattices

- Random q-ary lattice: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times n}\right)$, and consider $\Lambda_q(A)$
- equivalently: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times(m-n)}\right)$, and consider $\Lambda_q^{\perp}(A)$
- Defines average-case problems!
- For $X \in \{\text{approxSVP}, \text{approxCVP}, \text{uSVP}, \text{BDD}\}$ and $m = \mathrm{poly}(n)$ we have

Solving X in any lattice of rank m	$\geq$	Solving X with non-negligible prob. in a random q-ary lattice	$\gtrsim$	Solving approx-SVP in any lattice of rank $\min(n, m-n)$

Worst-case to average-case reduction

Family of random q-ary lattices

- Random q-ary lattice: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times n}\right)$, and consider $\Lambda_q(A)$
- equivalently: sample $A \in \mathcal{U}\left(\mathbb{Z}_q^{m\times (m-n)}\right)$, and consider $\Lambda_q^{\perp}(A)$
- Defines average-case problems!
- For $X \in \{$approxSVP, approxCVP, uSVP, BDD$\}$ and $m = \mathbf{poly}(n)$ we have

Solving X in any lattice of rank m	$\geq$	Solving X with non-negligible prob. in a random q-ary lattice	$\gtrsim$	Solving approx-SVP in any lattice of rank $\min(n, m-n)$

Worst-case to average-case reduction

- These average-case problems are also known as (I)SIS and LWE.

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

SIS (Short Integer Solution) [Ajt96]

Parameters: B and q

Problem: Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ s.t. $\boxed{x^T}\,\boxed{A} = 0 \bmod q$ with $\|x\| \leq B$ and $x \neq 0$.

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

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

SIS (Short Integer Solution) [Ajt96]

Parameters: B and q

Problem: Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ s.t. $\boxed{x^T}\,\boxed{A} = 0 \bmod q$ with $\|x\| \leq B$ and $x \neq 0$.

Solving SIS with non-negligible probability	$\gtrsim$	Solving approx-SVP in any lattice of rank n

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

The SIS problem

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

SIS (Short Integer Solution) [Ajt96]

Parameters: B and q

Problem: Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ s.t. $\boxed{x^T}\,\boxed{A} = 0 \bmod q$ with $\|x\| \leq B$ and $x \neq 0$.

Solving approx-SVP in any lattice lattice of rank m	$\geq$	Solving SIS with non-negligible probability	$\gtrsim$	Solving approx-SVP in any lattice of rank n

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

The SIS problem

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

ISIS (Inhomogeneous Short Integer Solution) [Ajt96]

Parameters: B and q

Problem: Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{y} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$

Find $x \in \mathbb{Z}^m$ s.t. $\boxed{x^T}\,\boxed{A} = \boxed{y^T} \bmod q$ with $\|x\| \leq B$.

Solving approx-CVP in any lattice lattice of rank m	$\geq$	Solving ISIS with non-negligible probability	$\gtrsim$	Solving approx-SVP in any lattice of rank n

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

(I)SIS is as hard as worst-case lattice problems

Theorem [Ajt96]

For any $m = \text{poly}(n)$ and $B > 0$ and sufficiently large $q \geq B \cdot \text{poly}(n)$, it holds that solving SIS is at least as hard as solving γ-SIVP on arbitrary n-dimensional lattice, for some approximation factor $\gamma = B \cdot \text{poly}(n)$.

(SIVP = shortest independent vectors problems.

Objective: find n short linearly independent vectors in the lattice)

[Ajt96] Ajtai. Generating hard instances of lattice problems. STOC.

(I)SIS is as hard as worst-case lattice problems

Theorem [Ajt96]

For any $m = \mathsf{poly}(n)$ and $B > 0$ and sufficiently large $q \geq B \cdot \mathsf{poly}(n)$, it holds that solving SIS is at least as hard as solving γ-SIVP on arbitrary n-dimensional lattice, for some approximation factor $\gamma = B \cdot \mathsf{poly}(n)$.

(SIVP = shortest independent vectors problems.
Objective: find n short linearly independent vectors in the lattice)

- the **poly** quantities have been improved in more recent works
- for typical parameters: SIS $\cong$ ISIS
- see [Pei16] for a survey

[Pei16] Peikert. A decade of lattice cryptography. Foundations and trends in theoretical computer science

SIS is a lattice problem

SIS (Short Integer Solution)

Given $\boxed{\boldsymbol{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ (with $n \log q < m$)

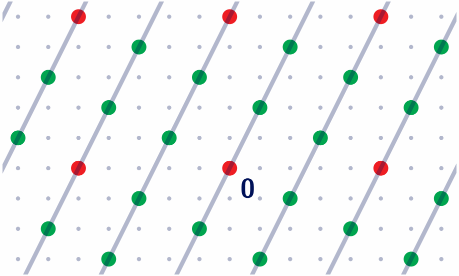
Find $\boldsymbol{x} \in \mathbb{Z}^m$ with $\|\boldsymbol{x}\| \leq B$ and $\boldsymbol{x} \neq \boldsymbol{0}$ s.t. $\boxed{\boldsymbol{x}^T}\ \boxed{\boldsymbol{A}} = \boldsymbol{0} \mod \boldsymbol{q}$.

SIS is a lattice problem

SIS (Short Integer Solution)

Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ with $\|x\| \leq B$ and $x \neq 0$ s.t. $\boxed{x^T}\ \boxed{A} = 0 \mod q$.



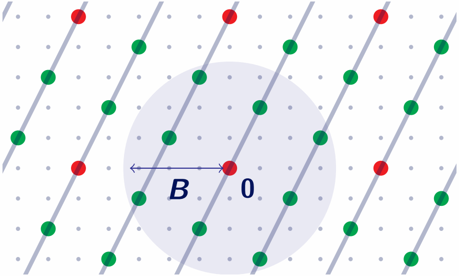
$$\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m \mid x^T A = 0 \bmod q\}$$

SIS is a lattice problem

SIS (Short Integer Solution)

Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ with $\|x\| \leq B$ and $x \neq 0$ s.t. $\boxed{x^T}\,\boxed{A} = \quad \bmod q$.



$$\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m \mid x^T A = 0 \bmod q\}$$

SIS $\approx$ approx-SVP in random $\Lambda_q^{\perp}(A)$

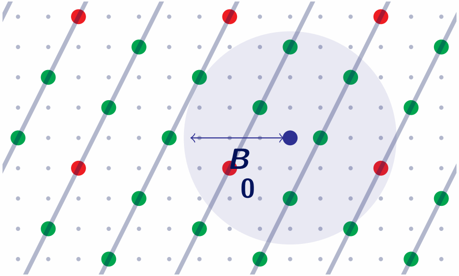
Average-case approx-SVP problem

SIS is a lattice problem

SIS (Short Integer Solution)

Given $\boxed{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{y} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ (with $n \log q < m$)

Find $x \in \mathbb{Z}^m$ with $\|x\| \leq B$ s.t. $\boxed{x^T}\,\boxed{A} = \boxed{y^T} \bmod q$.



$$\Lambda_q^{\perp}(A) = \{x \in \mathbb{Z}^m \mid x^T A = 0 \bmod q\}$$

ISIS $\approx$ approx-CVP in random $\Lambda_q^{\perp}(A)$

Average-case approx-CVP problem

Trapdoor basis

Lemma [Ajt99]

One can efficiently create a uniform SIS lattice $\boldsymbol{\Lambda_q^\perp(A)}$ together with a short basis of it.

[Ajt99] Ajtai. Generating hard instances of the short basis problem. ICALP.

Trapdoor basis

Lemma [Ajt99]

One can efficiently create a uniform SIS lattice $\Lambda_q^\perp(A)$ together with a short basis of it.

Idea: start with a short basis, then perturb and randomize it

[Ajt99] Ajtai. Generating hard instances of the short basis problem. ICALP.

Trapdoor basis

Lemma [Ajt99]

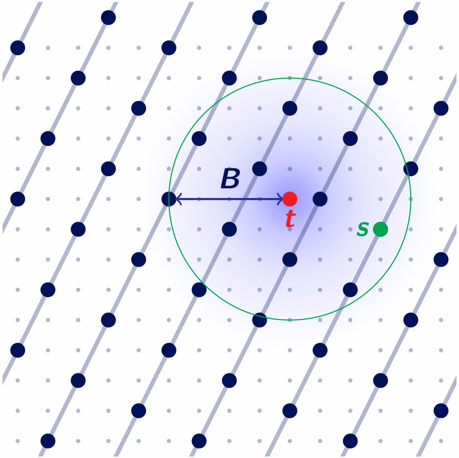
One can efficiently create a uniform SIS lattice $\Lambda_q^{\perp}(A)$ together with a short basis of it.

Idea: start with a short basis, then perturb and randomize it

$$\begin{array}{ccc} & \nearrow & \textbf{pk} = \text{uniform random } A \\ \text{SIS } \textbf{KeyGen} & & \Big\downarrow \geq \text{ SIS problem} \\ & \searrow & \textbf{sk} = \text{good basis } S \text{ of } \Lambda_q^{\perp}(A) \end{array}$$

[Ajt99] Ajtai. Generating hard instances of the short basis problem. ICALP.

Hash-and-sign signature scheme from SIS

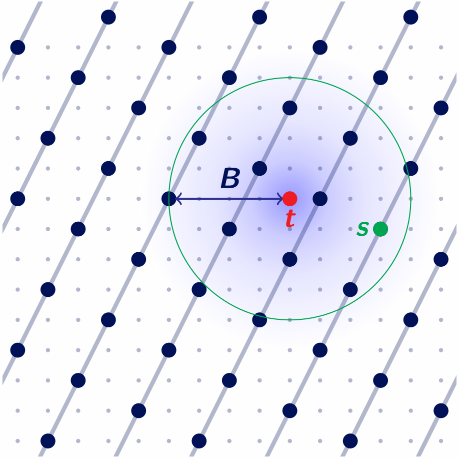


Sign: hash message to $\boldsymbol{t} \in \mathbb{Z}_q^m$,

sample nearby $\boldsymbol{s} \in \Lambda_q^{\perp}(\boldsymbol{A})$ with sk

Verify: $\boldsymbol{s} \in \Lambda_q^{\perp}(\boldsymbol{A}) \wedge \|\boldsymbol{t} - \boldsymbol{s}\| \leq B$

Hash-and-sign signature scheme from SIS



Sign: hash message to $\boldsymbol{t} \in \mathbb{Z}_q^m$,

sample nearby $\boldsymbol{s} \in \Lambda_q^{\perp}(\boldsymbol{A})$ with sk

Verify: $\boldsymbol{s} \in \Lambda_q^{\perp}(\boldsymbol{A}) \wedge \|\boldsymbol{t} - \boldsymbol{s}\| \leq \boldsymbol{B}$

Security proof

key-recovery $\geq$ SIS problem

signature forgery $\geq$ ISIS problem

(assuming no leakage from sampling

can be proven in Random Oracle Model) .

Signature scheme based on hard average-case lattice problem

The LWE problem

LWE (Learning With Errors) [Reg05]

Sample $\boxed{\boldsymbol{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{\boldsymbol{s}} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $\boxed{\boldsymbol{e}} \leftarrow \mathcal{U}(\{-B, \cdots, B\}^m)$

Given $\boxed{\boldsymbol{A}}$ and $\boxed{\boldsymbol{b}}$, where $\boxed{\boldsymbol{b}} := \boxed{\boldsymbol{A}}\,\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q$

Recover $\boldsymbol{s}$ or $\boldsymbol{e}$

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

The LWE problem

LWE (Learning With Errors) [Reg05]

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

Given $\boxed{A}$ and $\boxed{b}$, where $\boxed{b} := \boxed{A}\,\boxed{s} + \boxed{e} \bmod q$

Recover s or e

Remark. Sometimes $\boxed{s}$ is small in $\mathbb{Z}_q$ (not uniform)

- this is (almost) equivalent
- prove it (*hint:* you are allowed to change m)

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

LWE (Learning With Errors) [Reg05]

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

Given $\boxed{A}$ and $\boxed{b}$, where $\boxed{b} := \boxed{A}\boxed{s} + \boxed{e} \bmod q$

Recover s or e

Solving LWE with non-negligible probability	$\gtrsim$ quantumly!	Solving approx-SVP in any lattice of rank n

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

LWE (Learning With Errors) [Reg05]

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

Given $\boxed{A}$ and $\boxed{b}$, where $\boxed{b} := \boxed{A}\,\boxed{s} + \boxed{e} \bmod q$

Recover s or e

Solving BDD in any lattice of rank m	$\gtrsim$	Solving LWE with non-negligible probability	$\gtrsim$ quantumly!	Solving approx-SVP in any lattice of rank n

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

LWE is quantumly 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 quantumly 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]

[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 a lattice problem

LWE instance $\left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} = \boxed{\boldsymbol{A}}\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q \right)$, $\boldsymbol{e}$ small

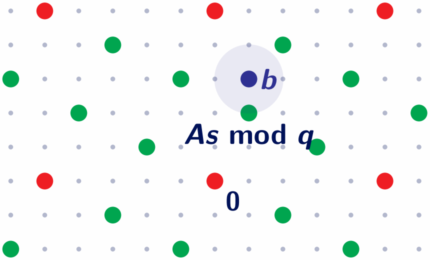
LWE is a lattice problem

LWE instance $\left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} = \boxed{\boldsymbol{A}}\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q\right)$, $\boldsymbol{e}$ small

BDD

target $\boldsymbol{b} = \boldsymbol{v} + \boldsymbol{e}$

lattice $\Lambda_q(\boldsymbol{A})$



LWE is a lattice problem

LWE instance $\left(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} \bmod q\right)$, $\boldsymbol{e}$ small

BDD

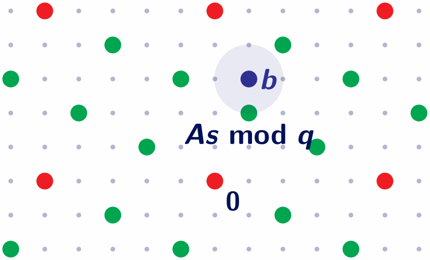
target $\boldsymbol{b} = \boldsymbol{v} + \boldsymbol{e}$

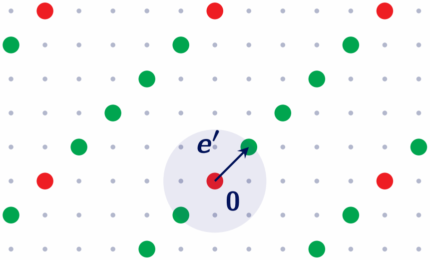
lattice $\boldsymbol{\Lambda}_q(\boldsymbol{A})$

uSVP

lattice $\boldsymbol{\Lambda}_q\left(\begin{pmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{0}_n & \boldsymbol{1} \end{pmatrix}\right)$

contains short $\boldsymbol{e}' := (\boldsymbol{e}^{\perp}, \boldsymbol{1})^{\perp}$





decision-LWE (1)

Sample $\boxed{\boldsymbol{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{\boldsymbol{s}} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $\boxed{\boldsymbol{e}} \leftarrow \mathcal{U}(\{-B, \cdots, B\}^m)$

Given $\boxed{\boldsymbol{A}}$ and $\boxed{\boldsymbol{b}}$, where $\boxed{\boldsymbol{b}} := \boxed{\boldsymbol{A}}\,\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q$ or $\boxed{\boldsymbol{b}} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$

Guess whether $\boxed{\boldsymbol{b}}$ is uniform or not.

Decision variant of LWE (1)

decision-LWE (1)

Sample $\boxed{\mathbf{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{\mathbf{s}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n})$ and $\boxed{\mathbf{e}} \leftarrow \mathcal{U}(\{-B,\cdots,B\}^m)$

Given $\boxed{\mathbf{A}}$ and $\boxed{\mathbf{b}}$, where $\boxed{\mathbf{b}} := \boxed{\mathbf{A}}\,\boxed{\mathbf{s}} + \boxed{\mathbf{e}} \bmod q$ or $\boxed{\mathbf{b}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m})$

Guess whether $\boxed{\mathbf{b}}$ is uniform or not.

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

Decision variant of LWE (1)

decision-LWE (1)

Sample $\boxed{\boldsymbol{A}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m\times n})$, $\boxed{\boldsymbol{s}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{n})$ and $\boxed{\boldsymbol{e}} \leftarrow \mathcal{U}(\{-B,\cdots,B\}^m)$

Given $\boxed{\boldsymbol{A}}$ and $\boxed{\boldsymbol{b}}$, where $\boxed{\boldsymbol{b}} := \boxed{\boldsymbol{A}}\,\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q$ or $\boxed{\boldsymbol{b}} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m})$

Guess whether $\boxed{\boldsymbol{b}}$ is uniform or not.

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

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

Decision variant of LWE (2)

if dec-LWE is hard: $\left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} = \boxed{\boldsymbol{A}}\,\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod q\right) \approx \left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}}\right)$

Decision variant of LWE (2)

if dec-LWE is hard: $\left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} = \boxed{\boldsymbol{A}}\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \ \textbf{mod}\ \boldsymbol{q} \right) \approx \left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} \right)$

BDD:

For a random $\boldsymbol{q}$-ary lattice:

BDD target $\boldsymbol{b} \approx$ uniform random target

Decision variant of LWE (2)

if dec-LWE is hard: $\left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} = \boxed{\boldsymbol{A}}\boxed{\boldsymbol{s}} + \boxed{\boldsymbol{e}} \bmod \boldsymbol{q} \right) \approx \left(\boxed{\boldsymbol{A}}, \boxed{\boldsymbol{b}} \right)$

BDD: For a random $\boldsymbol{q}$-ary lattice:

BDD target $\boldsymbol{b} \approx$ uniform random target

uSVP: random $\boldsymbol{q}$-ary lattice with planted short vector

$\approx$

random $\boldsymbol{q}$-ary lattice

Decision variant of LWE (2)

if dec-LWE is hard: $\left(\boxed{A}, \boxed{b} = \boxed{A}\,\boxed{s} + \boxed{e} \bmod q\right) \approx \left(\boxed{A}, \boxed{b}\right)$

BDD: For a random q-ary lattice:

BDD target $b \approx$ uniform random target

uSVP: random q-ary lattice with planted short vector

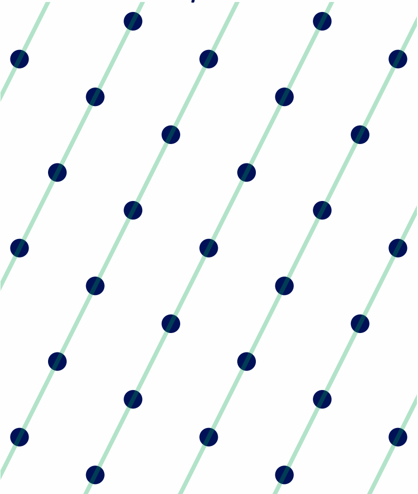
$\approx$

random q-ary lattice

useful in security proofs!

1 bit encryption scheme from LWE

$\Lambda_q^{\perp}(P)$

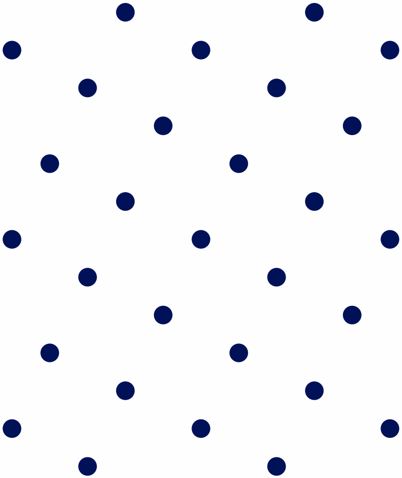


KeyGen:

$\mathbf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}.$

$\mathbf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

$\Lambda_q^{\perp}(P)$



KeyGen:

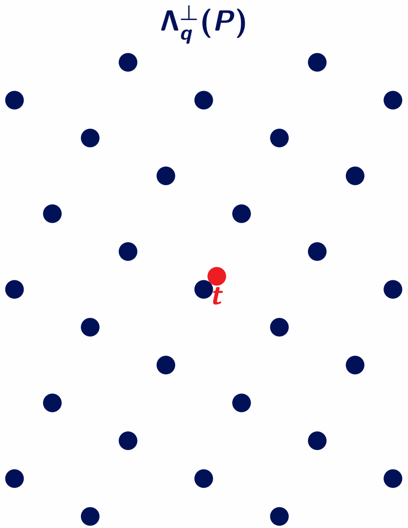
$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^{\perp}(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^{\top}$.



KeyGen:

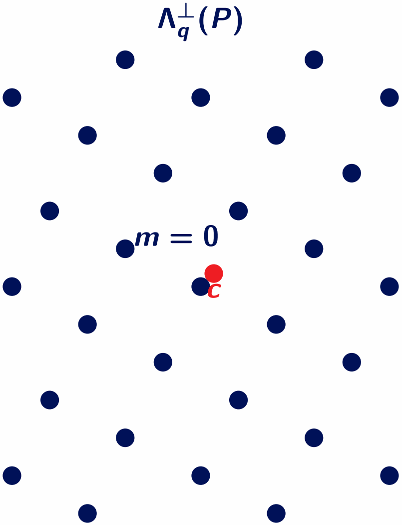
$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^\perp(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^\top$.



KeyGen:

$\textcolor{red}{\mathsf{pk}} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}.$

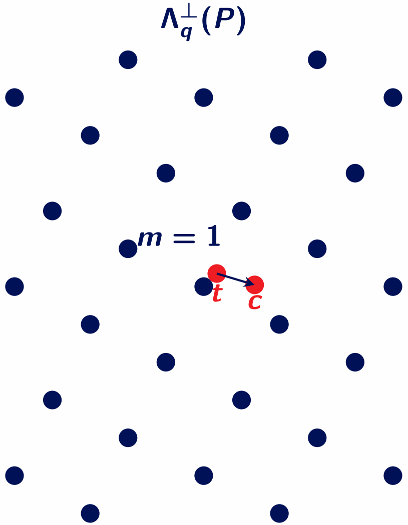
$\textcolor{green}{\mathsf{sk}} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P).$

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^\perp(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^\top.$

1 bit encryption scheme from LWE



KeyGen:

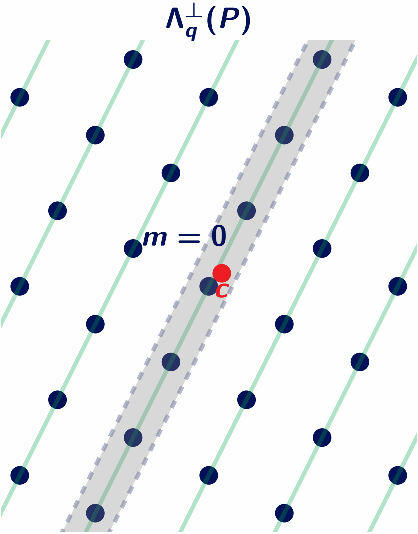
$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^\perp(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^\top$.



KeyGen:

$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

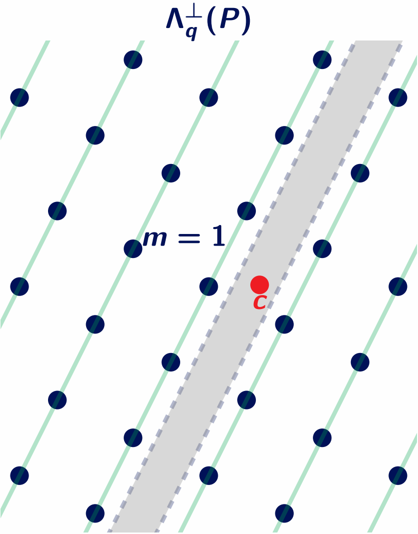
Generate: BDD instance $t = v + e'$ in $\Lambda_q^{\perp}(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^{\top}$.

Decrypt(c, sk):

Compute: $x = \left\langle c, \begin{pmatrix} e \\ 1 \end{pmatrix} \right\rangle \bmod q$.

Output: $m' = \begin{cases} 0 & \text{, if } -\frac{q}{4} \leq x \leq \frac{q}{4} \\ 1 & \text{, else} \end{cases}$.



KeyGen:

$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^\perp(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^\top$.

Decrypt(c, sk):

Compute: $x = \left\langle c, \begin{pmatrix} e \\ 1 \end{pmatrix} \right\rangle \bmod q$.

Output: $m' = \begin{cases} 0 & \text{, if } -\frac{q}{4} \leq x \leq \frac{q}{4} \\ 1 & \text{, else} \end{cases}$.

1 bit encryption scheme from LWE

security proof

dec-LWE $\Rightarrow$

$\Lambda_q^\perp(P) \approx$ random q-lattice

(no information about sk)

dec-LWE $\Rightarrow$

$t \approx$ uniform random target

$c \approx$ uniform random target

(no information about m)

KeyGen:

$\mathsf{pk} = (A, b = As + e), P = \begin{pmatrix} A & b \\ 0 & 1 \end{pmatrix}$.

$\mathsf{sk} = e$, short vector $\begin{pmatrix} e \\ 1 \end{pmatrix} \in \Lambda_q(P)$.

Encrypt(m, pk) :

Generate: BDD instance $t = v + e'$ in $\Lambda_q^\perp(P)$

Output: $c = t + \left\lfloor \frac{q}{2} \right\rceil \cdot m \cdot (0, \ldots, 0, 1)^\top$.

Decrypt(c, sk):

Compute: $x = \left\langle c, \begin{pmatrix} e \\ 1 \end{pmatrix} \right\rangle \bmod q$.

Output: $m' = \begin{cases} 0 & \text{, if } -\frac{q}{4} \le x \le \frac{q}{4} \\ 1 & \text{, else} \end{cases}$.

Summary on SIS and LWE

SIS and LWE are average-case problems

Summary on SIS and LWE

SIS and LWE are average-case problems

$\Longrightarrow$ Good for crypto

(negligible probability to sample a weak key)

Summary on SIS and LWE

SIS and LWE are average-case problems

$\Rightarrow$ Good for crypto

(negligible probability to sample a weak key)

family of random q-ary lattices

Summary on SIS and LWE

SIS and LWE are average-case problems

$\Rightarrow$ Good for crypto

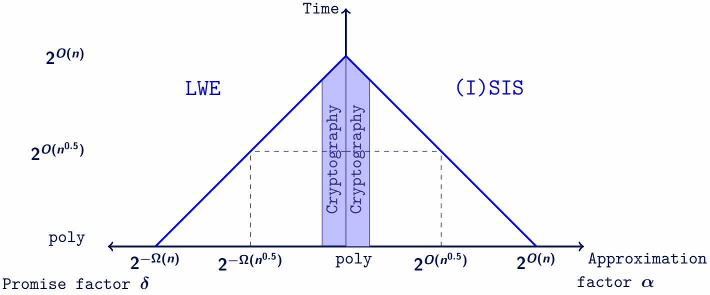
(negligible probability to sample a weak key)

family of random q-ary lattices

(I)SIS $\overset{\sim}{\longleftrightarrow}$ average-case SVP/CVP

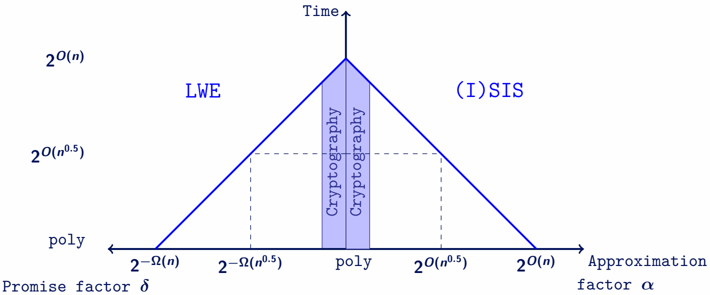
LWE $\overset{\sim}{\longleftrightarrow}$ average case BDD/uSVP

LWE vs SIS



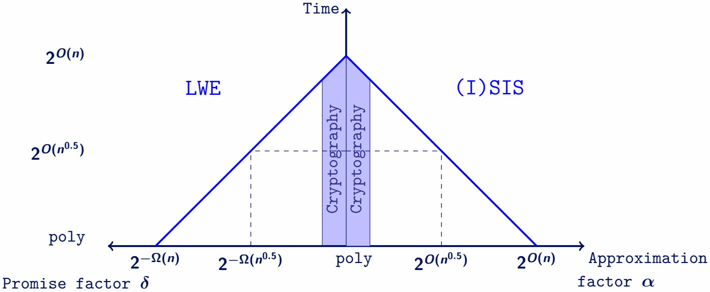
Time
$2^{O(n)}$
$2^{O(n^{0.5})}$
poly
LWE
(I)SIS
Cryptography
Cryptography
$2^{-\Omega(n)}$
$2^{-\Omega(n^{0.5})}$
poly
$2^{O(n^{0.5})}$
$2^{O(n)}$
Promise factor δ
Approximation factor α

LWE vs SIS



decision-LWE $\overset{\sim}{\Longleftrightarrow}$ (search) LWE $\underset{\text{quantum}}{\overset{\sim}{\Longleftrightarrow}}$ SIS

LWE vs SIS



decision-LWE $\overset{\sim}{\Longleftrightarrow}$ (search) LWE $\underset{\text{quantum}}{\overset{\sim}{\Longleftrightarrow}}$ SIS

Exercise

Prove that decision-LWE $\leq$ SIS

Hint: See decryption of LWE encryption scheme

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)
- as hard as worst-case lattice problems

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)
- as hard as worst-case lattice problems
  - no major flaw in the design
  - but cryptographic constructions choose smaller parameters than the ones needed for the reductions

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)
- as hard as worst-case lattice problems
  - no major flaw in the design
  - but cryptographic constructions choose smaller parameters than the ones needed for the reductions
- best known algorithm has time $2^{\Omega(m)}$ (for well chosen parameters q and B)

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)
- as hard as worst-case lattice problems
  - no major flaw in the design
  - but cryptographic constructions choose smaller parameters than the ones needed for the reductions
- best known algorithm has time $2^{\Omega(m)}$ (for well chosen parameters q and B)
  - by transforming LWE and (I)SIS into SVP/CVP instances

[Pei16] Peikert. A decade of lattice cryptography. Foundations and trends in theoretical computer science

Recap

(decision) LWE / SIS:

- lattice problems over random q-ary lattices
- all somewhat equivalent (quantumly)
- as hard as worst-case lattice problems
  - no major flaw in the design
  - but cryptographic constructions choose smaller parameters than the ones needed for the reductions
- best known algorithm has time $2^{\Omega(m)}$ (for well chosen parameters q and B)
  - by transforming LWE and (I)SIS into SVP/CVP instances
- useful survey [Pei16]

[Pei16] Peikert. A decade of lattice cryptography. Foundations and trends in theoretical computer science

Algebraic lattices

Motivation

- A lattice of dimension $\boldsymbol{n}$ is described by some basis $\boldsymbol{B} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{n}}$

Motivation

- A lattice of dimension $\boldsymbol{n}$ is described by some basis $\boldsymbol{B} \in \mathbb{R}^{n \times n}$
  $\Rightarrow \boldsymbol{n}^2$ coefficients, $(n = 1000, \; n^2 = 10^6)$

Motivation

- A lattice of dimension $\boldsymbol{n}$ is described by some basis $\boldsymbol{B} \in \mathbb{R}^{\boldsymbol{n} \times \boldsymbol{n}}$

  $\Rightarrow \boldsymbol{n}^2$ coefficients, $(n = 1000,\ n^2 = 10^6)$
- Storage: multiple MB or GB of data

Motivation

- A lattice of dimension $\boldsymbol{n}$ is described by some basis $\boldsymbol{B} \in \mathbb{R}^{n \times n}$

  $\Rightarrow n^2$ coefficients, $(n = 1000,\ n^2 = 10^6)$
- Storage: multiple MB or GB of data
- Efficiency: matrix-matrix product $O(n^3)$, matrix-vector $O(n^2)$

Motivation

- A lattice of dimension $\boldsymbol{n}$ is described by some basis $\boldsymbol{B} \in \mathbb{R}^{n \times n}$

  $\Rightarrow \boldsymbol{n}^2$ coefficients, $(n = 1000,\ n^2 = 10^6)$
- Storage: multiple MB or GB of data
- Efficiency: matrix-matrix product $O(n^3)$, matrix-vector $O(n^2)$

(we ignore here the dependency on the size of each coefficient)

Motivation

- A lattice of dimension n is described by some basis $B \in \mathbb{R}^{n\times n}$

  $\Rightarrow n^2$ coefficients, $(n = 1000,\ n^2 = 10^6)$

- Storage: multiple MB or GB of data
- Efficiency: matrix-matrix product $O(n^3)$, matrix-vector $O(n^2)$

(we ignore here the dependency on the size of each coefficient)

Idea: add (algebraic) structure

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 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

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

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

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

- 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

($K = \mathbb{Q}[X]/P(X)$, $\alpha_1, \cdots, \alpha_d$ 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 $x \in I$

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 $\mathcal{O}_K = \mathbb{Z}[X]/(X^2 + 1)$

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

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 via the coefficient embedding Σ:

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

$\mathcal{O}_K$ is a lattice via the coefficient embedding Σ:

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

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

$\mathcal{O}_K$ is a lattice via the coefficient embedding Σ:

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

$\Sigma(\mathcal{O}_K)$ 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:

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

$\mathcal{O}_K$ is a lattice via the coefficient embedding Σ:

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

$\Sigma(\mathcal{O}_K)$ 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:

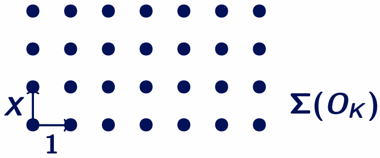
- $\langle g \rangle = g \cdot \mathcal{O}_K = g \cdot 1 \cdot \mathbb{Z} + g \cdot X \cdot \mathbb{Z} + \cdots + g \cdot X^{d-1} \cdot \mathbb{Z}$
- $\Sigma(\langle g \rangle) = \Sigma(g) \cdot \mathbb{Z} + \cdots + \Sigma(g \cdot X^{d-1}) \cdot \mathbb{Z}$

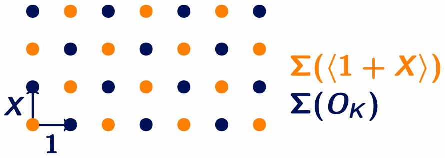
$\Sigma(\langle g \rangle)$ is a lattice of rank d in $\mathbb{Z}^d$ with basis $(\Sigma(g \cdot X^i))_{0 \leq i < d}$

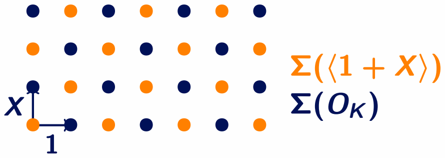
(this is also true for non principal ideals)

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

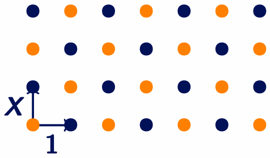
Ideal lattices (2)




X
1
$\Sigma(\langle 1 + X \rangle)$
$\Sigma(\mathcal{O}_K)$



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



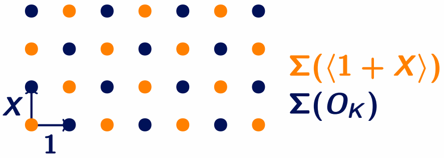
$\Sigma(\langle 1+X \rangle)$

$\Sigma(\mathcal{O}_K)$

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

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

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



We have

$$g \cdot X = \sum_{i=0}^{d-1} g_i X^{i+1} = g_{d-1}X^d + \sum_{i=0}^{d-2} g_i X^{i+1}$$

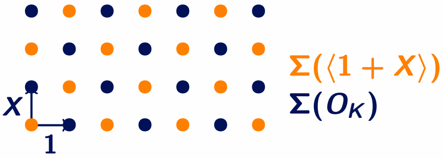
$$= -g_{d-1} + \sum_{i=0}^{d-2} g_i X^{i+1} \bmod X^d + 1$$

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

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

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

Ideal lattices (2)



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

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

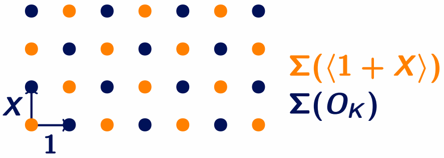
$$\begin{pmatrix} g_0 & -g_{d-1} & \cdots & -g_1 \\ g_1 & g_0 & \cdots & -g_2 \\ \vdots & \vdots & \ddots & \vdots \\ g_{d-1} & g_{d-2} & \cdots & g_0 \end{pmatrix}$$

We have

$$g\cdot X = \sum_{i=0}^{d-1} g_i X^{i+1} = g_{d-1}X^d + \sum_{i=0}^{d-2} g_i X^{i+1}$$

$$= -g_{d-1} + \sum_{i=0}^{d-2} g_i X^{i+1} \bmod X^d + 1$$

Ideal lattices (2)



We have

$$g\cdot X=\sum_{i=0}^{d-1} g_i X^{i+1} = g_{d-1}X^d+\sum_{i=0}^{d-2} g_i X^{i+1}$$

$$= -g_{d-1}+\sum_{i=0}^{d-2} g_i X^{i+1} \bmod X^d+1$$

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

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

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

Storage: n^2 coefficients $\rightarrow$ n

Time: $O(n^2)\rightarrow O(n\log(n))$

(fast polynomial multiplication via FFT)

Module lattices

(Free) module:

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

(Free) module:

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

- 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 $\mathbb{Z}^n$

(Free) module:

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

- 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)

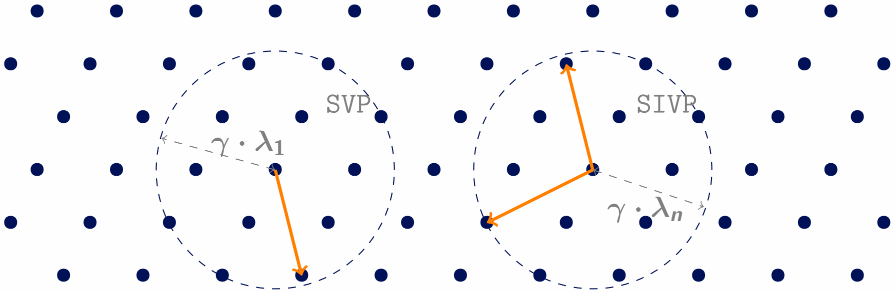
In $K = \mathbb{Q}[X]/(X^d+1)$:

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

basis of a
principal ideal lattice

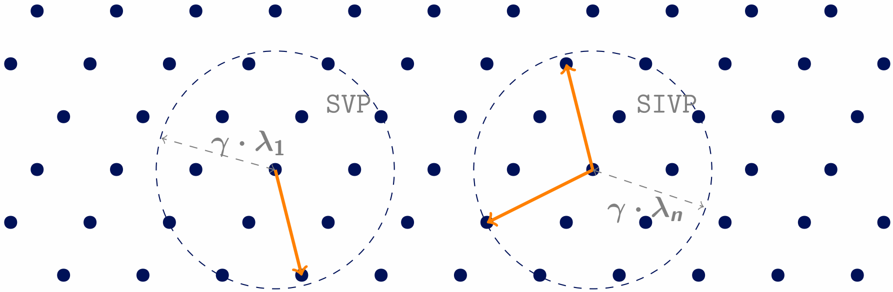
$$d \updownarrow \quad \begin{pmatrix} M_{b_{11}} & M_{b_{12}} & \cdots & M_{b_{1k}} \\ M_{b_{21}} & M_{b_{22}} & \cdots & M_{b_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ M_{b_{k1}} & M_{b_{k2}} & \cdots & M_{b_{kk}} \end{pmatrix} \quad \updownarrow dk$$

basis of a free module lattice
of rank k



γ-SVP
shortest vector problem

γ-SIVP
shortest independent vector problem



γ-SVP
shortest vector problem

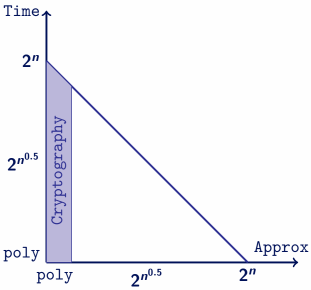
γ-SIVP
shortest independent vector problem

Notations:

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

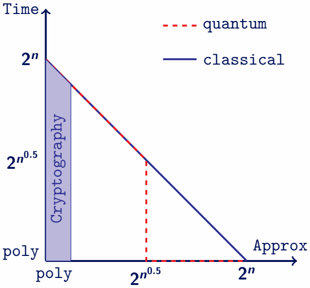
(worst-case: we want algorithms for all ideal/module lattices)

Asymptotics:



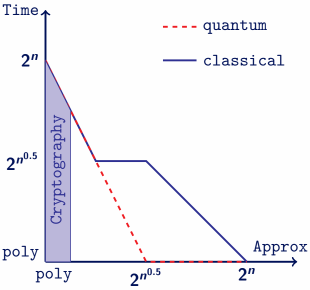
SVP and mod-SVP$_k$

($k \geq 2$)



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.

Ring and Module-LWE

(search) mod-LWE$_k$

Parameters: q and B

Problem: Sample

- $A \leftarrow \mathcal{U}((\mathcal{O}_K/q\mathcal{O}_K)^{m\times k})$
- secret $s \in (\mathcal{O}_K/q\mathcal{O}_K)^k$
- error $e \in \mathcal{O}_K^m$ 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

- $A \leftarrow \mathcal{U}((\mathcal{O}_K/q\mathcal{O}_K)^{m\times k})$
- secret $s \in (\mathcal{O}_K/q\mathcal{O}_K)^k$
- error $e \in \mathcal{O}_K^m$ 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)

$\boxed{\text{RLWE = mod-LWE}_1}$

mod-LWE vs mod-SIVP

$$\text{mod-uSVP}_{m+1} \geq \text{mod-BDD}_m \geq \text{mod-LWE}_k \underset{\text{quantumly!}}{\geq} \text{mod-SIVP}_k$$

mod-LWE vs mod-SIVP

$\text{mod-uSVP}_{m+1} \geq \text{mod-BDD}_m \geq \text{mod-LWE}_k \geq \text{mod-SIVP}_k$

quantumly!

How large should m be?

- as small as possible
- but so that the closest point to $\boldsymbol{b}$ is $\boldsymbol{As}$

mod-LWE vs mod-SIVP

$$\text{mod-uSVP}_{m+1} \geq \text{mod-BDD}_m \geq \text{mod-LWE}_k \underset{\text{quantumly!}}{\geq} \text{mod-SIVP}_k$$

How large should m be?

- as small as possible
- but so that the closest point to b is As
- $m = k$ is not sufficient

mod-LWE vs mod-SIVP

mod-uSVP$_{m+1}$ $\geq$ mod-BDD$_m$ $\geq$ mod-LWE$_k$ $\geq$ mod-SIVP$_k$

quantumly!

How large should m be?

- as small as possible
- but so that the closest point to b is As
- $m = k$ is not sufficient
- $m = k + 1$ might be sufficient depending on B and q
  - we need roughly $m = k \cdot \frac{\log(q)}{\log(q/B)}$
  - for $k = 1$, $m = 2$ is possible if $B \lesssim \sqrt{q}$

(search) NTRU

Parameters: $q \geq B > 1$

Objective: Sample $f, g \in \mathcal{O}_K$ with coefficients in $\{-B, \cdots, B\}$.
Given $h = f \cdot g^{-1} \bmod q$, recover (f, g)

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

(search) NTRU

Parameters: $q \geq B > 1$

Objective: Sample $f, g \in \mathcal{O}_K$ with coefficients in $\{-B, \cdots, B\}$. Given $h = f \cdot g^{-1} \bmod q$, recover (f, g)

dec-NTRU

Parameters: q, B

Objective: distinguish between h as above and h uniform in $\mathcal{O}_K/(q\mathcal{O}_K)$

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

NTRU as a lattice

Recall: $h = f \cdot g^{-1} \bmod q$

Definition (NTRU Lattice)

$$\mathcal{L}^{h,q} := \{(a, b) \in R^2 : h \cdot b = a \bmod q\}$$

Recall: $h = f \cdot g^{-1} \bmod q$

Definition (NTRU Lattice)

$$\mathcal{L}^{h,q} := \{(a, b) \in R^2 : h \cdot b = a \bmod q\}$$

- $d = \deg(R)$, rank 2 module, dimension $n = 2d$, $\det(\mathcal{L}^{h,q}) = q^d$.

NTRU as a lattice

Recall: $h = f \cdot g^{-1} \bmod q$

Definition (NTRU Lattice)

$$\mathcal{L}^{h,q} := \{(a, b) \in R^2 : h \cdot b = a \bmod q\}$$

- $d = \deg(R)$, rank 2 module, dimension $n = 2d$, $\det(\mathcal{L}^{h,q}) = q^d$.
- $\mathrm{gh}(\mathcal{L}^{h,q}) \approx \sqrt{d/\pi e} \cdot \sqrt{q}$

Recall: $h = f \cdot g^{-1} \bmod q$

Definition (NTRU Lattice)

$$\mathcal{L}^{h,q} := \{(a, b) \in R^2 : h \cdot b = a \bmod q\}$$

- $d = \deg(R)$, rank 2 module, dimension $n = 2d$, $\det(\mathcal{L}^{h,q}) = q^d$.
- $\mathrm{gh}(\mathcal{L}^{h,q}) \approx \sqrt{d/\pi e} \cdot \sqrt{q}$

Short vector(s)

The rotations $(x^i \cdot f, x^i \cdot g)$ are unusually short vectors in $\mathcal{L}^{h,q}$.

NTRU as a lattice

Recall: $h = f \cdot g^{-1} \bmod q$

Definition (NTRU Lattice)

$$\mathcal{L}^{h,q} := \{(a, b) \in R^2 : h \cdot b = a \bmod q\}$$

- $d = \deg(R)$, rank 2 module, dimension $n = 2d$, $\det(\mathcal{L}^{h,q}) = q^d$.
- $\mathrm{gh}(\mathcal{L}^{h,q}) \approx \sqrt{d/\pi e} \cdot \sqrt{q}$

Short vector(s)

The rotations $(x^i \cdot f, x^i \cdot g)$ are unusually short vectors in $\mathcal{L}^{h,q}$.

$$\text{bad basis} = \begin{pmatrix} q & h \\ 0 & 1 \end{pmatrix}, \quad \text{good basis} = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

Two regimes of NTRU

If $\|(f, g)\| \geq \mathrm{poly}(\log n) \cdot \mathrm{gh}(\mathcal{L}^{h,q})$

If $\|(f, g)\| \leq \mathrm{gh}(\mathcal{L}^{h,q})$

Two regimes of NTRU

If $\|(f,g)\| \geq \mathrm{poly}(\log n) \cdot \mathrm{gh}(\mathcal{L}^{h,q})$

- h is statistically close to uniform mod q [SS11,WW18]
- dec-NTRU is statistically hard

If $\|(f,g)\| \leq \mathrm{gh}(\mathcal{L}^{h,q})$

[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.

Two regimes of NTRU

If $\|(f,g)\| \geq \mathrm{poly}(\log n)\cdot \mathrm{gh}(\mathcal{L}^{h,q})$

- h is statistically close to uniform mod q [SS11,WW18]
- dec-NTRU is statistically hard

If $\|(f,g)\| \leq \mathrm{gh}(\mathcal{L}^{h,q})$

- h is not statistically close to uniform mod q
- NTRU is a special case of mod-uSVP$_2$

[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.

Two regimes of NTRU

If $\|(f,g)\| \geq \mathrm{poly}(\log n) \cdot \mathrm{gh}(\mathcal{L}^{h,q})$

- h is statistically close to uniform mod q [SS11,WW18]
- dec-NTRU is statistically hard

If $\|(f,g)\| \leq \mathrm{gh}(\mathcal{L}^{h,q})$

- h is not statistically close to uniform mod q
- NTRU is a special case of mod-uSVP$_2$

uSVP regime $\Rightarrow$ short structured basis

$\Rightarrow$ efficient encryption/signature scheme

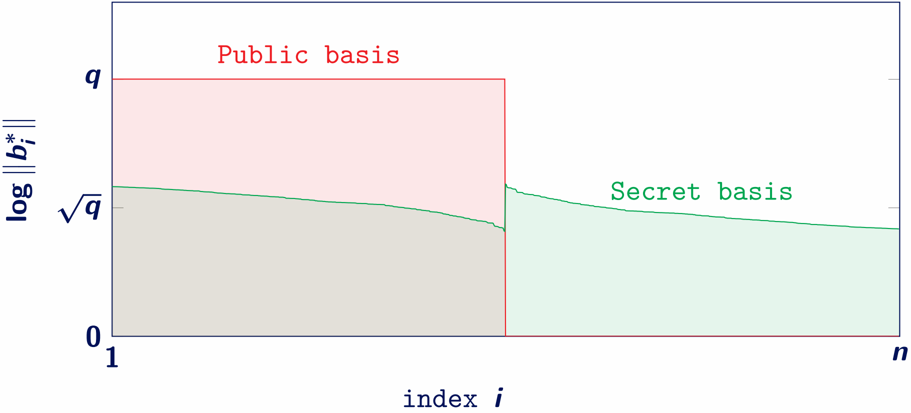
(e.g. NTRUEncrypt, NTRUSign, Falcon)

[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.

NTRU public vs secret basis

public and secret bases generated from the NTRU problem



Recap

- Algebraic structure reduces sizes and improves efficiency

Recap

- Algebraic structure reduces sizes and improves efficiency
- Can still define average-case problems

Recap

- Algebraic structure reduces sizes and improves efficiency
- Can still define average-case problems
- Most worst-case to average-case reductions still apply

Recap

- Algebraic structure reduces sizes and improves efficiency
- Can still define average-case problems
- Most worst-case to average-case reductions still apply
- Ideal lattices $=$ rank 1 modules can be vulnerable

Recap

- Algebraic structure reduces sizes and improves efficiency
- Can still define average-case problems
- Most worst-case to average-case reductions still apply
- Ideal lattices $=$ rank 1 modules can be vulnerable
- NIST candidates (e.g. Kyber, Dilithium, Falcon) use rank ≥ 2
  (seems safe so far, but arguably their weakest point)

Conclusion on lattice-based crypto

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 on lattice-based crypto

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 on lattice-based crypto

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 on lattice-based crypto

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 on lattice-based crypto

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 vs classical cryptography

Conclusion on lattice-based crypto

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 vs classical cryptography
- structured lattice problems are still young
  - more cryptanalysis is needed

Conclusion on lattice-based crypto

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 vs classical cryptography
- structured lattice problems are still young
  - more cryptanalysis is needed

Thank you