Introduction to lattice-based cryptography

Alice Pellet-Mary

COSIC team, KU Leuven, Belgium

Cryptography, Network Security and Cybersecurity webminar, Session-VI

Lattice-based cryptography

lattices, ideal lattices, SVP, CVP, ...	(Ring) LWE (Ring) SIS, NTRU, ...	Regev encryption scheme signatures, trapdoors FHE, obfuscation, functional encryption, ...

+ maths
- crypto

+ crypto
- maths

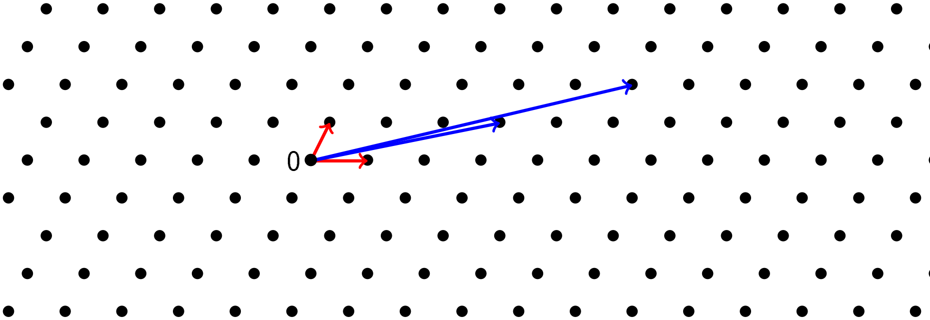
Outline of the talk

1. Lattices and lattice problems
2. Algorithmic problems for cryptography
3. Cryptographic primitives

Outline of the talk

1. Lattices and lattice problems
2. Algorithmic problems for cryptography
3. Cryptographic primitives

Lattices

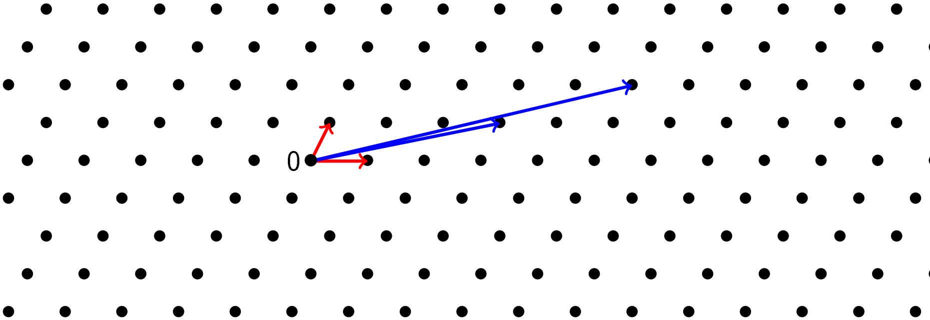


Lattice

A lattice L is a subset of $\mathbb{R}^n$ of the form $L = \{Bx \mid x \in \mathbb{Z}^n\}$, with $B \in \mathbb{R}^{n \times n}$ invertible. B is a basis of L, and n is its rank.

$\begin{pmatrix} 3 & 1 \\ 0 & 2 \end{pmatrix}$ and $\begin{pmatrix} 17 & 11 \\ 4 & 2 \end{pmatrix}$ are two bases of the above lattice.

Lattices



Lattice

A lattice L is a subset of $\mathbb{R}^n$ of the form $L = \{Bx \mid x \in \mathbb{Z}^n\}$, with $B \in \mathbb{R}^{n \times n}$ invertible. B is a basis of L, and n is its rank.

We represent a lattice by any of its basis

Algorithmic problems on lattices

Input: any basis of any lattice

Example of problems:

- Testing equality of lattices
- Testing inclusion of lattices
- Intersecting two lattices
- Computing a short vector of a lattice
- Computing a lattice vector close to a target

Algorithmic problems on lattices

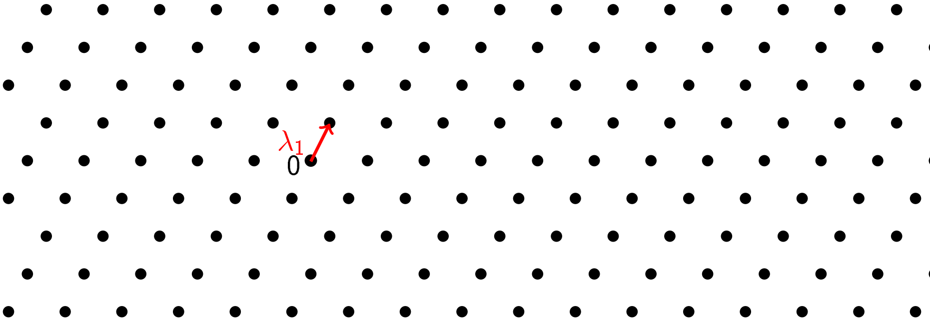
Input: any basis of any lattice

Example of problems:

- Testing equality of lattices $\Rightarrow$ easy
- Testing inclusion of lattices $\Rightarrow$ easy
- Intersecting two lattices $\Rightarrow$ easy
- Computing a short vector of a lattice $\Rightarrow$ hard
- Computing a lattice vector close to a target $\Rightarrow$ hard

easy: polynomial time   hard: no polynomial time algorithm known

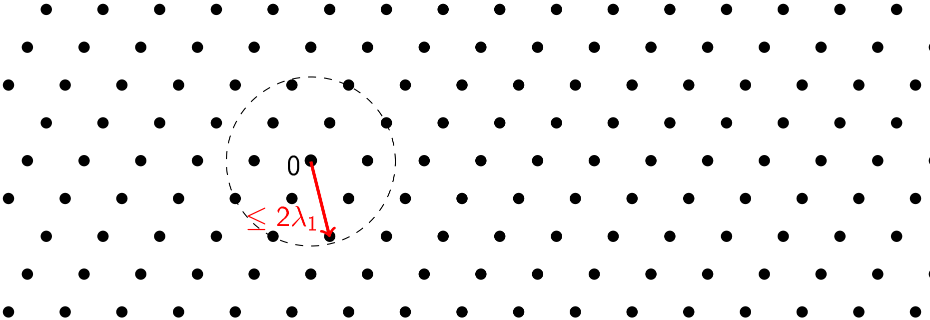
Shortest and Closest vector problems



Shortest Vector Problem (SVP)

Find a shortest (in Euclidean norm) non-zero vector.
Its Euclidean norm is denoted λ_1.

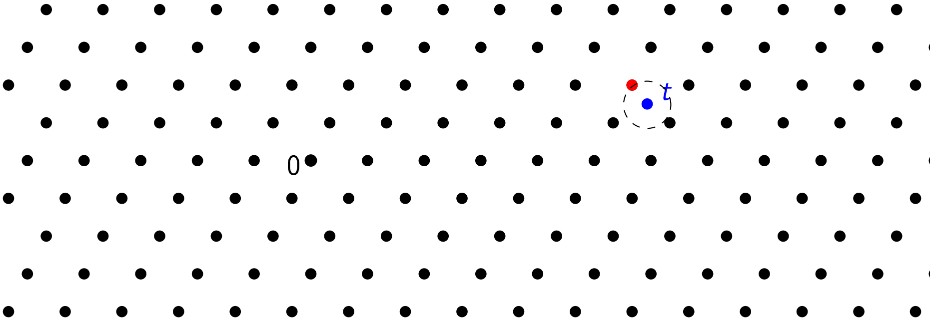
Shortest and Closest vector problems



Approximate Shortest Vector Problem (approx-SVP)

Find a short (in Euclidean norm) non-zero vector.
(e.g. of norm $\leq 2\lambda_1$).

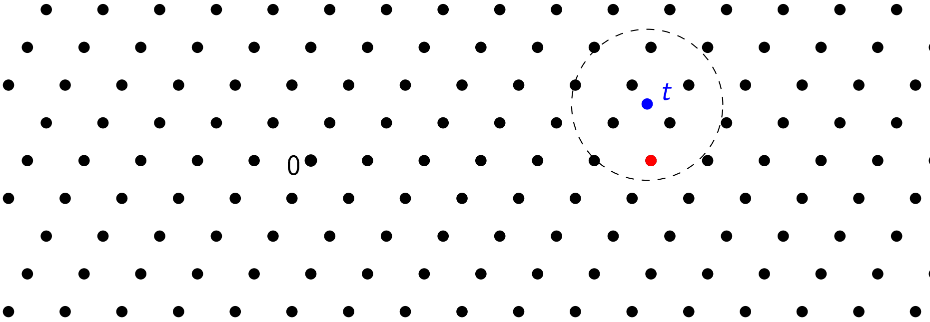
Shortest and Closest vector problems



Closest Vector Problem (CVP)

Given a target point t, find a point of the lattice closest to t.

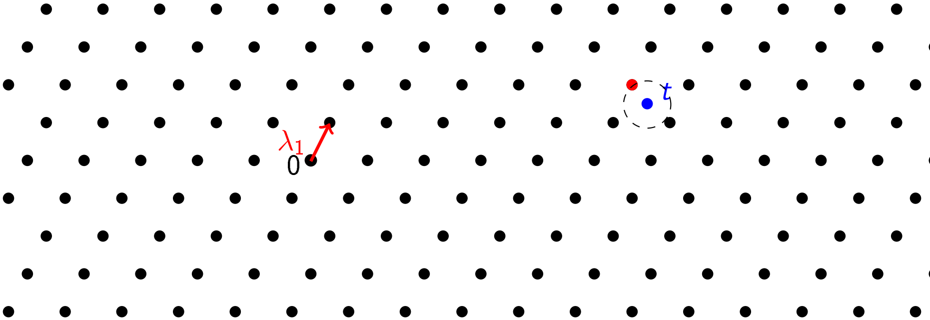
Shortest and Closest vector problems



Approximate Closest Vector Problem (approx-CVP)

Given a target point t, find a point of the lattice close to t.

Shortest and Closest vector problems

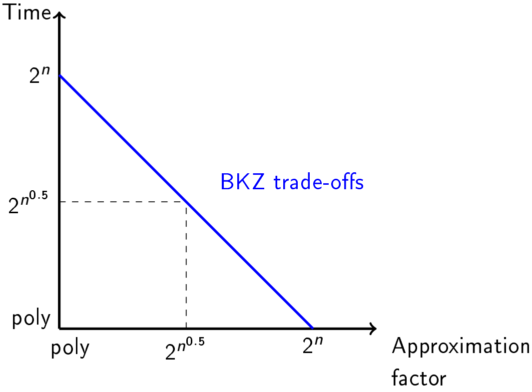


SVP and CVP are hard to solve when n increases

- even with a quantum computer
- even if we allow small approximation factor $(\gamma = \text{poly}(n))$

Hardness of SVP and CVP

Best Time/Approximation trade-off for SVP, CVP (even quantumly):
BKZ algorithm [Sch87,SE94]

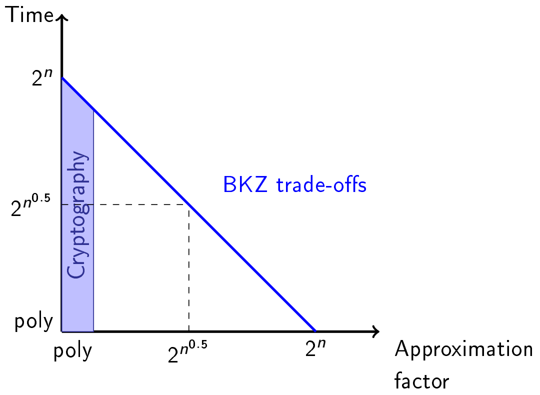


[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS.

[SE94] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematical programming.

Hardness of SVP and CVP

Best Time/Approximation trade-off for SVP, CVP (even quantumly):
BKZ algorithm [Sch87,SE94]



[Sch87] C.-P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. TCS.

[SE94] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. Mathematical programming.

Exact SVP in practice

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

Exact SVP in practice

- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 80$ or $n = 100 \rightsquigarrow$ a few minutes on a personal laptop

Exact SVP in practice

- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 80$ or $n = 100 \rightsquigarrow$ a few minutes on a personal laptop
- up to $n = 170 \rightsquigarrow$ a few days on a big computer with optimized code

Exact SVP in practice

- $n = 2 \rightsquigarrow$ easy, very efficient in practice
- up to $n = 80$ or $n = 100 \rightsquigarrow$ a few minutes on a personal laptop
- up to $n = 170 \rightsquigarrow$ a few days on a big computer with optimized code
- from $n = 500$ to $n = 1000 \rightsquigarrow$ cryptography

An example of lattice reduction algorithm

The Lagrange-Gauss algorithm:

- For lattices of rank $n = 2$ only
- Solves exact SVP
- Polynomial time

An example of lattice reduction algorithm

The Lagrange-Gauss algorithm:

- For lattices of rank $n = 2$ only
- Solves exact SVP
- Polynomial time

video

An example of lattice reduction algorithm

The Lagrange-Gauss algorithm:

- For lattices of rank $n = 2$ only
- Solves exact SVP
- Polynomial time

video

But remember: when n is large, solving exact SVP is hard

Outline of the talk

1. Lattices and lattice problems
2. Algorithmic problems for cryptography
3. Cryptographic primitives

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

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 (or some basis of a lattice) it might be easy

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 (or some basis of a lattice) it might be easy

For crypto, we need problems that are hard on average

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

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]

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

Find $x \in \{-B, \cdots, B\}^m \setminus \{0\}$ s.t. $x \, A = 0 \bmod q$

[Ajt96] M. 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]

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

Find $x \in \{-B, \cdots, B\}^m \setminus \{0\}$ s.t. $x \cdot A = 0 \bmod q$

Solving SIS with non-negligible probability (e.g., $\geq 2^{-80}$) $\overset{\sim}{\iff}$ Solving SVP in any lattice of rank n

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

SIS is a lattice problem

SIS (Short Integer Solution)

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

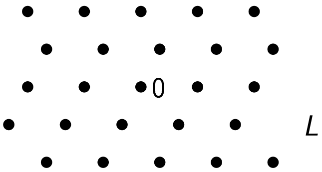
Find $x \in \{-B, \cdots, B\}^m \setminus \{0\}$ s.t. $\boxed{x}\,\boxed{A} = 0 \bmod q$

SIS is a lattice problem

SIS (Short Integer Solution)

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

Find $x \in \{-B, \cdots, B\}^m \setminus \{0\}$ s.t. $\boxed{x}\ \boxed{A} = 0 \bmod q$



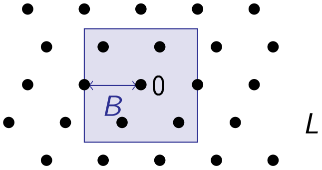
$$L = \{x \in \mathbb{Z}^m \,|\, xA = 0 \bmod q\}$$

SIS is a lattice problem

SIS (Short Integer Solution)

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

Find $x \in \{-B, \cdots, B\}^m \setminus \{0\}$ s.t. $x A = 0 \bmod q$



$$L = \{x \in \mathbb{Z}^m \mid xA = 0 \bmod q\}$$

SIS ≈ SVP in L

The LWE problem

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

LWE (Learning With Errors) [Reg05]

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

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

Recover s or e

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

The LWE problem

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

LWE (Learning With Errors) [Reg05]

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

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

Recover s or e

Solving LWE with non-negligible probability (e.g., $\geq 2^{-80}$) $\stackrel{\sim}{\Longleftrightarrow}$ Solving SVP in any lattice of rank n

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

LWE is a lattice problem

LWE (Learning With Errors)

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

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

Recover s or e

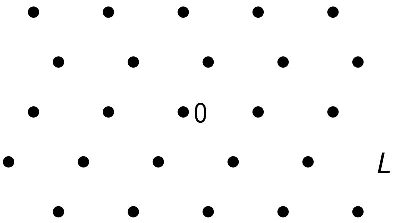
LWE is a lattice problem

LWE (Learning With Errors)

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

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

Recover s or e



$$L = \{x \in \mathbb{Z}^n \,|\, \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

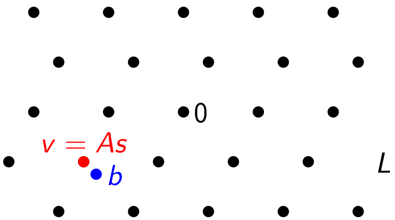
LWE is a lattice problem

LWE (Learning With Errors)

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

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

Recover s or e



$$L = \{x \in \mathbb{Z}^n \mid \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

$b = v + e$,
where $v \in L$ and e small

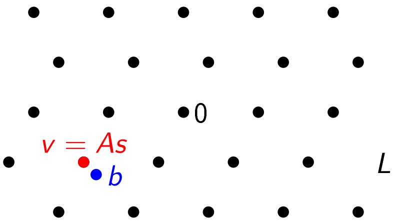
LWE is a lattice problem

LWE (Learning With Errors)

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

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

Recover s or e



$$L = \{x \in \mathbb{Z}^n \,|\, \exists s \in \mathbb{Z}^n, As = x \bmod q\}$$

$$b = v + e,$$
where $v \in L$ and e small

LWE $\approx$ CVP in L

Summary on SIS and LWE

SIS and LWE are average-case problems

Summary on SIS and LWE

SIS and LWE are average-case problems
$\Rightarrow$ 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)

SIS $\overset{\sim}{\longleftrightarrow}$ average case SVP

LWE $\overset{\sim}{\longleftrightarrow}$ average case CVP

Decision variant of LWE

decision-LWE

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

Given A and b, where

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

Guess whether b is uniform or not.

Decision variant of LWE

decision-LWE

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

Given A and b, where

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

Guess whether b is uniform or not.

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

Decision variant of LWE

decision-LWE

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

Given A and b, where

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

Guess whether b is uniform or not.

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

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

Outline of the talk

Collision-resistant hash functions

$\mathcal{G} = \{H : S \to S'\}$ is a family of collision-resistant hash functions if

- it is compressing: $|S'| < |S|$
- it is collision-resistant: $\forall$ PPT adversary $\mathcal{A}$,

$$\Pr_{H \leftarrow \mathrm{Uniform}(\mathcal{G})}\Big[(x_1, x_2) \leftarrow \mathcal{A}(H) \,|\, x_1 \neq x_2,\ H(x_1) = H(x_2)\Big] \leq \mathrm{negl}$$

SIS-based collision-resistant hash functions

$\mathcal{G} = \{H_A \,|\, A \in \mathbb{Z}_q^{m \times n}\}$, where

$$H_A : \{0,1\}^m \to \{0, \cdots, q-1\}^n$$
$$x \mapsto x \, A \mod q$$

SIS-based collision-resistant hash functions

$\mathcal{G} = \{H_A \mid A \in \mathbb{Z}_q^{m \times n}\}$, where

$$H_A : \{0,1\}^m \to \{0, \cdots, q-1\}^n$$

$$x \mapsto x \, A \mod q$$

- compressing: $m > n \log q$

SIS-based collision-resistant hash functions

$\mathcal{G} = \{H_A \,|\, A \in \mathbb{Z}_q^{m \times n}\}$, where

$$H_A : \{0,1\}^m \to \{0, \cdots, q-1\}^n$$

$$x \mapsto x \cdot A \bmod q$$

- compressing: $m > n \log q$
- collision-resistance: $\mathcal{A}$ breaking $\mathcal{G} \Rightarrow \mathcal{A}_{SIS}$ breaking SIS

SIS-based collision-resistant hash functions

$\mathcal{G} = \{H_A \mid A \in \mathbb{Z}_q^{m \times n}\}$, where

$$H_A : \{0,1\}^m \to \{0, \cdots, q-1\}^n$$

$$x \mapsto x A \mod q$$

- compressing: $m > n \log q$
- collision-resistance: $\mathcal{A}$ breaking $\mathcal{G} \Rightarrow \mathcal{A}_{SIS}$ breaking SIS

  $\mathcal{A}_{SIS}(A)$:
  - $(x_1, x_2) \leftarrow \mathcal{A}(H_A)$ $(x_1 A = x_2 A \bmod q)$
  - output $x_1 - x_2$ $(\in \{-B, \cdots, B\}^m$ since $B \geq 1)$

Encryption scheme

$\mathsf{KeyGen}(1^\lambda) = (\mathrm{sk}, \mathrm{pk})$
$\mathsf{Enc}(\mathrm{pk}, m \in \{0, 1\}) = c$
$\mathsf{Dec}(\mathrm{sk}, c) = \overline{m}$

Encryption scheme

$\mathsf{KeyGen}(1^\lambda) = (\mathrm{sk}, \mathrm{pk})$
$\mathsf{Enc}(\mathrm{pk}, m \in \{0,1\}) = c$
$\mathsf{Dec}(\mathrm{sk}, c) = \overline{m}$

- Correction: $\forall (sk, pk) \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\forall m \in \{0,1\}$,

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

- CPA security: $\forall$ PPT adversary $\mathcal{A}$,

$$\Big| \Pr_{(sk,pk)\leftarrow \mathsf{KeyGen}(1^\lambda)} \Big[\mathcal{A}(\mathrm{pk}, c) = 1 \mid c \leftarrow \mathsf{Enc}(\mathrm{pk}, 0)\Big] - \Pr_{(sk,pk)\leftarrow \mathsf{KeyGen}(1^\lambda)} \Big[\mathcal{A}(\mathrm{pk}, c) = 1 \mid c \leftarrow \mathsf{Enc}(\mathrm{pk}, 1)\Big]\Big| = \mathrm{negl}$$

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):

- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n \times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b := A s + e)$

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):

- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n\times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b := A s + e)$

Enc(pk, m):

- sample $s' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^{n+1})$
- output $c = s' \cdot \begin{bmatrix} A & b \end{bmatrix} + e' + \lfloor \frac{q}{2} \rceil \, [0 \ldots 0 m] \mod q$

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):
- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n\times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b := A s + e)$

Enc(pk, m):
- sample $s' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^{n+1})$
- output $c = s' \cdot [A \mid b] + e' + \lfloor \frac{q}{2} \rceil \, (0 \ldots 0 m) \mod q$

Security: $b \approx b \leftarrow \text{Uniform}(\mathbb{Z}_q^n)$ (by decision-LWE)

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):

- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n\times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b \leftarrow \text{Uniform}(\mathbb{Z}_q^n))$

Enc(pk, m):

- sample $s' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^{n+1})$
- output $c = s' \cdot (A \mid b) + e' + \lfloor \frac{q}{2} \rceil (0 \ldots 0 m) \mod q$

Security: $b \approx b \leftarrow \text{Uniform}(\mathbb{Z}_q^n)$ (by decision-LWE)

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):
- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n \times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b \leftarrow \text{Uniform}(\mathbb{Z}_q^n))$

Enc(pk, m):
- sample $s' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^{n+1})$
- output $c = s' \cdot (A \mid b) + e' + \lfloor \frac{q}{2} \rceil (0 \ldots 0 m) \bmod q$

Security: $s' \cdot (A \mid b) + e' \approx b' \leftarrow \text{Uniform}(\mathbb{Z}_q^{n+1})$

(by transposing decision-LWE)

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):

- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n\times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b \leftarrow \text{Uniform}(\mathbb{Z}_q^n))$

Enc(pk, m):

- sample $s' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B, \cdots, B\}^{n+1})$
- output $c = b' + \lfloor \frac{q}{2} \rceil \, 0...0m \mod q$

Security: $s' \cdot (A \mid b) + e' \approx b' \leftarrow \text{Uniform}(\mathbb{Z}_q^{n+1})$

(by transposing decision-LWE)

LWE-based encryption (Regev's Encryption)

KeyGen(1^λ):
- sample $A \leftarrow \text{Uniform}(\mathbb{Z}_q^{n\times n})$
- sample s, $e \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- output $\mathrm{sk} = s$ and $\mathrm{pk} = (A, b \leftarrow \text{Uniform}(\mathbb{Z}_q^n))$

Enc(pk, m):
- sample $s' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^n)$
- sample $e' \leftarrow \text{Uniform}(\{-B,\cdots,B\}^{n+1})$
- output $c = b' + \lfloor \frac{q}{2} \rceil \, 0...0m \mod q$

Security: $(b' + \lfloor \frac{q}{2} \rceil \, 0...0m \mod q)$ uniform in $\mathbb{Z}_q^{n+1}$

$\Rightarrow$ independent of m

Decryption and correction

Reminder

$c = \boxed{s'} \cdot \boxed{A \mid b} + \boxed{e'} + \lfloor \frac{q}{2} \rceil \boxed{0 \dots 0 m} \bmod q$ and $\mathrm{sk} = \boxed{s}$

Dec(sk, c):

- $x = \boxed{c} \cdot \begin{pmatrix} s \\ -1 \end{pmatrix} \bmod q$ $(x \in [-q/2, q/2])$
- if $|x| < q/4$ output 0
- otherwise output 1

Decryption and correction

Reminder

$c = s' \cdot \begin{pmatrix} A & b \end{pmatrix} + e' + \lfloor \frac{q}{2} \rceil \begin{pmatrix} 0 \dots 0 m \end{pmatrix} \mod q$ and $\mathrm{sk} = s$

Dec(sk, c):

- $x = c \cdot \begin{pmatrix} s \\ -1 \end{pmatrix} \mod q$ $\quad (x \in [-q/2, q/2])$
- if $|x| < q/4$ output 0
- otherwise output 1

Correction:

$$c \cdot \begin{pmatrix} s \\ -1 \end{pmatrix} = s' A s - s' b + e' \begin{pmatrix} s \\ -1 \end{pmatrix} - m \cdot \lfloor \frac{q}{2} \rceil$$

Decryption and correction

Reminder

$c = \boxed{s'} \cdot \boxed{A \mid b} + \boxed{e'} + \lfloor \frac{q}{2} \rceil \boxed{0...0m} \mod q$ and $\mathrm{sk} = \boxed{s}$

Dec(sk, c):
- $x = \boxed{c} \cdot \begin{bmatrix} s \\ -1 \end{bmatrix} \mod q \quad (x \in [-q/2, q/2])$
- if $|x| < q/4$ output 0
- otherwise output 1

Correction:

$$\boxed{c} \cdot \begin{bmatrix} s \\ -1 \end{bmatrix} = \boxed{s'} \boxed{A} \boxed{s} - \boxed{s'} (\boxed{A} \boxed{s} + \boxed{e}) + \boxed{e'} \begin{bmatrix} s \\ -1 \end{bmatrix} - m \cdot \lfloor \frac{q}{2} \rceil$$

Decryption and correction

Reminder

$c = \boxed{s'} \cdot \boxed{A \mid b} + \boxed{e'} + \lfloor \frac{q}{2} \rceil \boxed{0 \dots 0 m} \mod q$ and $\mathrm{sk} = \boxed{s}$

Dec(sk, c):

- $x = \boxed{c} \cdot \begin{pmatrix} s \\ \text{-1} \end{pmatrix} \mod q$ $(x \in [-q/2, q/2])$
- if $|x| < q/4$ output 0
- otherwise output 1

Correction:

$$\boxed{c} \cdot \begin{pmatrix} s \\ \text{-1} \end{pmatrix} = \boxed{s'}\,\boxed{A}\,\boxed{s} - \boxed{s'}(\boxed{A}\,\boxed{s} + \boxed{e}) + \boxed{e'} \begin{pmatrix} s \\ \text{-1} \end{pmatrix} - m \cdot \lfloor \frac{q}{2} \rceil$$

$$= -\boxed{s'}\,\boxed{e} + \boxed{e'} \begin{pmatrix} s \\ \text{-1} \end{pmatrix} - m \cdot \lfloor \frac{q}{2} \rceil$$

Decryption and correction

Reminder

$c = s' \cdot \begin{pmatrix} A & b \end{pmatrix} + e' + \lfloor \frac{q}{2} \rceil \begin{pmatrix} 0 \ldots 0 m \end{pmatrix} \bmod q$ and $\mathrm{sk} = s$

Dec(sk, c):

- $x = c \cdot \begin{pmatrix} s \\ -1 \end{pmatrix} \bmod q$ $(x \in [-q/2, q/2])$
- if $|x| < q/4$ output 0
- otherwise output 1

Correction:

$$c \cdot \begin{pmatrix} s \\ -1 \end{pmatrix} = s' A s - s'(A s + e) + e' \begin{pmatrix} s \\ -1 \end{pmatrix} - m \cdot \lfloor \frac{q}{2} \rceil$$

$$= \quad \text{small} \quad - m \cdot \lfloor \frac{q}{2} \rceil$$

NIST post-quantum standardization process

Objective: new standard for post-quantum encryption (and signature)

NIST post-quantum standardization process

Objective: new standard for post-quantum encryption (and signature)

- Started in 2017 $\rightsquigarrow$ 48 encryption candidates

NIST post-quantum standardization process

Objective: new standard for post-quantum encryption (and signature)

- Started in 2017 $\rightsquigarrow$ 48 encryption candidates
- Since August 2020 (round 3) $\rightsquigarrow$ 4 candidates left
    - 3 of them are based on lattices

Conclusion

Structured lattices

Reminder

Lattices are represented by a basis B.

Structured lattices

Reminder

Lattices are represented by a basis B.

By default: $B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} \rightsquigarrow n^2$ storage

Structured lattices

Reminder

Lattices are represented by a basis B.

By default: $B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix}$ $\rightsquigarrow$ n^2 storage

Structured basis: $B = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ b_n & b_1 & \cdots & b_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_2 & b_3 & \cdots & b_1 \end{pmatrix}$ $\rightsquigarrow$ n storage (e.g., RLWE)

Structured lattices

Reminder

Lattices are represented by a basis B.

By default: $$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} \rightsquigarrow n^2 \text{ storage}$$

Structured basis: $$B = \begin{pmatrix} b_1 & b_2 & \cdots & b_n \\ b_n & b_1 & \cdots & b_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_2 & b_3 & \cdots & b_1 \end{pmatrix} \rightsquigarrow n \text{ storage (e.g., RLWE)}$$

- schemes more efficient
- are they still secure?

Take-away

- Wide range of possible questions related to lattice-based crypto

Take-away

- Wide range of possible questions related to lattice-based crypto
- Promising way to construct post-quantum crypto

Take-away

- Wide range of possible questions related to lattice-based crypto
- Promising way to construct post-quantum crypto
- Young research area $\rightsquigarrow$ many things to do

Take-away

- Wide range of possible questions related to lattice-based crypto
- Promising way to construct post-quantum crypto
- Young research area $\rightsquigarrow$ many things to do

Questions?