Lattice Cryptography: Introduction and Open Problems

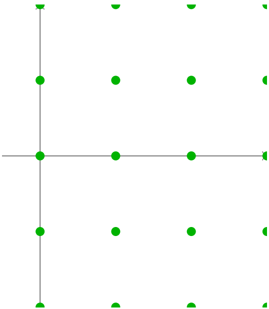
Daniele Micciancio

Department of Computer Science and Engineering
University of California, San Diego

August 2015

Point Lattices

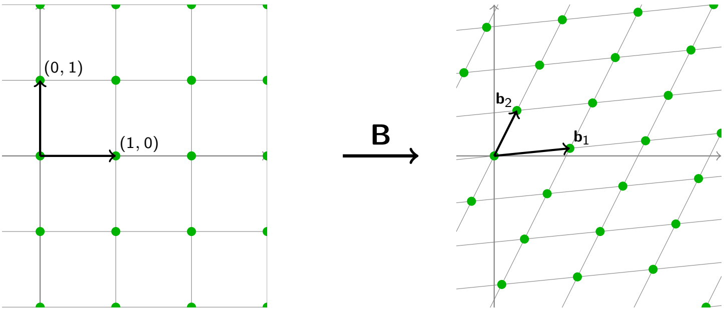
- The simplest example of lattice is $\mathbb{Z}^n = \{(x_1, \ldots, x_n) \colon x_i \in \mathbb{Z}\}$



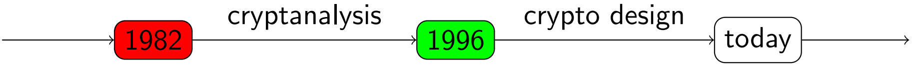
Point Lattices

- The simplest example of lattice is $\mathbb{Z}^n = \{(x_1, \ldots, x_n) \colon x_i \in \mathbb{Z}\}$
- Other lattices are obtained by applying a linear transformation

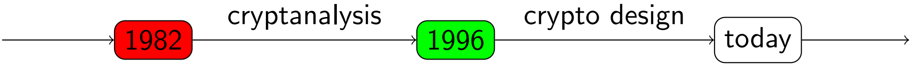
$$\mathbf{B} \colon \mathbf{x} = (x_1, \ldots, x_n) \mapsto \mathbf{B}\mathbf{x} = x_1 \cdot \mathbf{b}_1 + \cdots + x_n \cdot \mathbf{b}_n$$



Lattice Cryptography



- Lenstra, Lenstra, Lovasz (1982) : The “LLL” paper
  “Factoring Polynomials with Rational Coefficients”
  - Algorithmic breakthrough
  - Efficient approximate solution of lattice problems
  - Exponential approximation factor, but very good in practice
  - Killer App: Cryptanalysis



- Lenstra, Lenstra, Lovasz (1982) : The "LLL" paper
  "Factoring Polynomials with Rational Coefficients"
  - Algorithmic breakthrough
  - Efficient approximate solution of lattice problems
  - Exponential approximation factor, but very good in practice
  - Killer App: Cryptanalysis
- Ajtai (1996) : "Generating Hard Instances of Lattice Problems"
  - Marks the beginning of the modern use of lattices in the design of cryptographic functions

Ajtai's paper (quotes)

- "cryptography ... generation of a specific instance of a problem in NP which is thought to be difficult".
    - "NP-hard problems"
    - "very famous question (e.g., prime factorization)."

  "Unfortunately 'difficult to solve' means ... in the worst case"
- "no guidance about how to create [a hard instance]"
- "possible solution"
    1. "find a set of randomly generated problems", and
    2. "show that if there is an algorithm which [works] with a positive probability, then there is also an algorithm which solves the famous problem in the worst case."
- "In this paper we give such a class of random problems."

Example: Discrete Logrithm (DLOG)

- p: a prime
- $\mathbb{Z}_p^*$: multiplicative group
- $g \in \mathbb{Z}_p^*$: generator of (prime order sub-)group $G = \{g^i : i \in \mathbb{Z}\} \subseteq \mathbb{Z}_p^*$
- Input: $h = g^i \bmod p$

DLOG Problem

Given p, g, h, recover i (modulo $q = o(g)$)

Example: Discrete Logrithm (DLOG)

- p: a prime
- $\mathbb{Z}_p^*$: multiplicative group
- $g \in \mathbb{Z}_p^*$: generator of (prime order sub-)group $G = \{g^i : i \in \mathbb{Z}\} \subseteq \mathbb{Z}_p^*$
- Input: $h = g^i \bmod p$

DLOG Problem

Given p, g, h, recover i (modulo $q = o(g)$)

Random Self Reducibility

If you can solve DLOG for random g and h (with some probability), then you can solve it for any g, h in the worst-case.

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h
2. Compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbb{Z}_q^*$.

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h
2. Compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbb{Z}_q^*$.
3. Notice:
   - $g', h' \in G$ are (almost) uniformly random
   - $h' = h^{ab} = g^{iab} = (g')^{ib}$

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h
2. Compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbb{Z}_q^*$.
3. Notice:
   - $g', h' \in G$ are (almost) uniformly random
   - $h' = h^{ab} = g^{iab} = (g')^{ib}$
4. Find $j = DLOG(g', h') = ib$

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h
2. Compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbb{Z}_q^*$.
3. Notice:
   - $g', h' \in G$ are (almost) uniformly random
   - $h' = h^{ab} = g^{iab} = (g')^{ib}$
4. Find $j = DLOG(g', h') = ib$
5. Output $j/b \pmod{q}$.

DLOG: Random Self Reducibility (RSR)

1. Given arbitrary g, h
2. Compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbb{Z}_q^*$.
3. Notice:
   - $g', h' \in G$ are (almost) uniformly random
   - $h' = h^{ab} = g^{iab} = (g')^{ib}$
4. Find $j = DLOG(g', h') = ib$
5. Output $j/b \pmod{q}$.

Conclusion

We know how to choose $g, h \in G$.
But, how do we choose G?

DLOG vs Lattices (1)

Lattice Assumption

The complexity of solving lattice problems in n-dimensional lattices grows superpolynomially (or exponentially) in n.

DLOG vs Lattices (1)

Lattice Assumption

The complexity of solving lattice problems in n-dimensional lattices grows superpolynomially (or exponentially) in n.

- Similarly, one may conjecture that the complexity of DLOG grows superpolynomially in $n = \log p$ or $n = \log |G|$.

DLOG vs Lattices (1)

Lattice Assumption

The complexity of solving lattice problems in n-dimensional lattices grows superpolynomially (or exponentially) in n.

- Similarly, one may conjecture that the complexity of DLOG grows superpolynomially in $n = \log p$ or $n = \log |G|$.
- This is not the same:
  - For any n, there are (exponentially) many primes p.
  - Typically, p is chosen at random among all n-bit primes
  - Assumption is still average-case: DLOG is hard for random p.

DLOG vs Lattices (1)

Lattice Assumption

The complexity of solving lattice problems in n-dimensional lattices grows superpolynomially (or exponentially) in n.

- Similarly, one may conjecture that the complexity of DLOG grows superpolynomially in $n = \log p$ or $n = \log |G|$.
- This is not the same:
  - For any n, there are (exponentially) many primes p.
  - Typically, p is chosen at random among all n-bit primes
  - Assumption is still average-case: DLOG is hard for random p.
- We do not know how to reduce $DLOG(\mathbb{Z}_p^*)$ to $DLOG(\mathbb{Z}_q^*)$. RSR provides no guidance on how to choose p.

DLOG vs Lattices (2)

Alternative assumption

DLOG(p_n) is hard when p_n is the smallest prime $> 2^n$.

- Equivalent to worst-case family of problems (indexed by n)
- Ad-hoc: problem definition seems rather arbitrary

DLOG vs Lattices (2)

Alternative assumption

DLOG(p_n) is hard when p_n is the smallest prime $> 2^n$.

- Equivalent to worst-case family of problems (indexed by n)
- Ad-hoc: problem definition seems rather arbitrary

There is more:

- Lattice problems in dimension n reduce to lattice problems in dimension $m > n$:

$$\boxed{\mathbf{B}} \Longrightarrow \begin{array}{|c|c|}\hline \mathbf{B} & \mathbf{O} \\ \hline \mathbf{O} & \infty \\ \hline \end{array}$$

- No such reduction for DLOG:

$$DLOG(p_n) \overset{?}{\Longrightarrow} DLOG(p_{n+1})$$

DLOG vs Lattices (3)

- Other (natural) representations:

$$G = (\mathbb{Z}_p^*, \cdot) \equiv (\mathbb{Z}_{p-1}, +)$$

  but "DLOG" in $(\mathbb{Z}_{p-1}, +)$ is easy.

- Other (still natural) groups:

$$G = \mathbb{Z}_{pq}^*$$

DLOG vs Lattices (3)

- Other (natural) representations:

$$G = (\mathbb{Z}_p^*, \cdot) \equiv (\mathbb{Z}_{p-1}, +)$$

  but "DLOG" in $(\mathbb{Z}_{p-1}, +)$ is easy.

- Other (still natural) groups:

$$G = \mathbb{Z}_{pq}^*$$

Question

Assume one of $DLOG(\mathbb{Z}_p)$ and $DLOG(\mathbb{Z}_{p \cdot q})$ is polynomial time solvable, and one is not. Which group family would you choose?

DLOG vs Lattices (3)

- Other (natural) representations:

$$G = (\mathbb{Z}_p^*, \cdot) \equiv (\mathbb{Z}_{p-1}, +)$$

  but "DLOG" in $(\mathbb{Z}_{p-1}, +)$ is easy.
- Other (still natural) groups:

$$G = \mathbb{Z}_{pq}^*$$

Question

Assume one of $DLOG(\mathbb{Z}_p)$ and $DLOG(\mathbb{Z}_{p \cdot q})$ is polynomial time solvable, and one is not. Which group family would you choose?

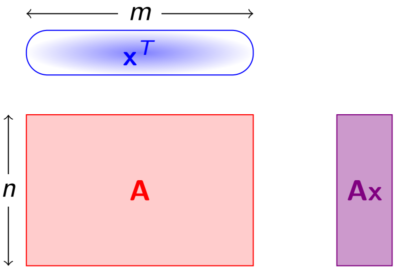
Chinese Reminder Theorem (CRT): $\mathbb{Z}_{pq} \approx \mathbb{Z}_p \times \mathbb{Z}_q$

$$DLOG(\mathbb{Z}_p^*) \Longrightarrow DLOG(\mathbb{Z}_{pq}^*).$$

Reduction in the other direction requires factoring.

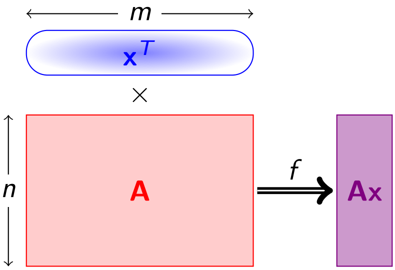
Ajtai's one-way function (SIS)

- Parameters: $m, n, q \in \mathbb{Z}$
- Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
- Input: $\mathbf{x} \in \{0,1\}^m$



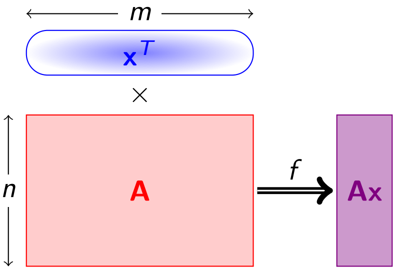
Ajtai's one-way function (SIS)

- Parameters: $m, n, q \in \mathbb{Z}$
- Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
- Input: $\mathbf{x} \in \{0,1\}^m$
- Output: $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$



Ajtai's one-way function (SIS)

- Parameters: $m, n, q \in \mathbb{Z}$
- Key: $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$
- Input: $\mathbf{x} \in \{0,1\}^m$
- Output: $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$



Theorem (A'96)

For $m > n \lg q$, if lattice problems (SIVP) are hard to approximate in the worst-case, then $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}\mathbf{x} \bmod q$ is a one-way function.

Applications: OWF [A'96], Hashing [GGH'97], Commit [KTX'08], ID schemes [L'08], Signatures [LM'08,GPV'08,...,DDLL'13] ...

Relation to lattices

- The kernel set $\Lambda^\perp(\mathbf{A})$ is a lattice

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \colon \mathbf{Az} = \mathbf{0} \pmod q\}$$

- Collisions $\mathbf{Ax} = \mathbf{Ay} \pmod q$ can be represented by a single vector $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \{-1, 0, 1\}$ such that

$$\mathbf{z} = \quad \mathbf{x} - \quad \mathbf{y}$$

Relation to lattices

- The kernel set $\Lambda^\perp(\mathbf{A})$ is a lattice

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \colon \mathbf{A}\mathbf{z} = \mathbf{0} \pmod q\}$$

- Collisions $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{y} \pmod q$ can be represented by a single vector $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \{-1, 0, 1\}$ such that

$$\mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y} = \mathbf{0} \bmod q$$

Relation to lattices

- The kernel set $\Lambda^\perp(\mathbf{A})$ is a lattice

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \colon \mathbf{Az} = \mathbf{0} \pmod{q}\}$$

- Collisions $\mathbf{Ax} = \mathbf{Ay} \pmod{q}$ can be represented by a single vector $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \{-1, 0, 1\}$ such that

$$\mathbf{Az} = \mathbf{Ax} - \mathbf{Ay} = \mathbf{0} \bmod q$$

- Collisions are lattice vectors $\mathbf{z} \in \Lambda^\perp(\mathbf{A})$ with small norm $\|\mathbf{z}\|_\infty = \max_i |z_i| = 1$.

Relation to lattices

- The kernel set $\Lambda^\perp(\mathbf{A})$ is a lattice

$$\Lambda^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \colon \mathbf{Az} = \mathbf{0} \pmod q\}$$

- Collisions $\mathbf{Ax} = \mathbf{Ay} \pmod q$ can be represented by a single vector $\mathbf{z} = \mathbf{x} - \mathbf{y} \in \{-1, 0, 1\}$ such that

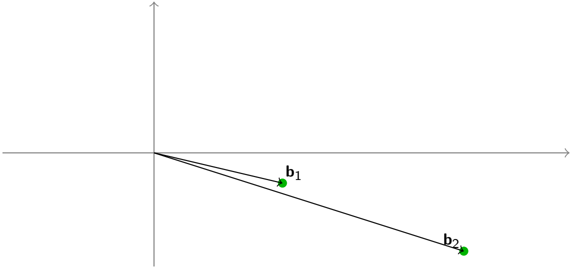
$$\mathbf{Az} = \mathbf{Ax} - \mathbf{Ay} = \mathbf{0} \bmod q$$

- Collisions are lattice vectors $\mathbf{z} \in \Lambda^\perp(\mathbf{A})$ with small norm $\|\mathbf{z}\|_\infty = \max_i |z_i| = 1$.
- ... there is a much deeper and interesting relation between breaking $f_{\mathbf{A}}$ and lattice problems.

Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

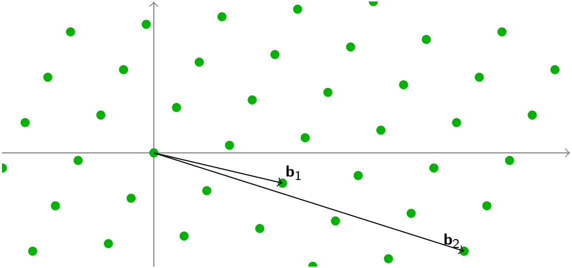
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

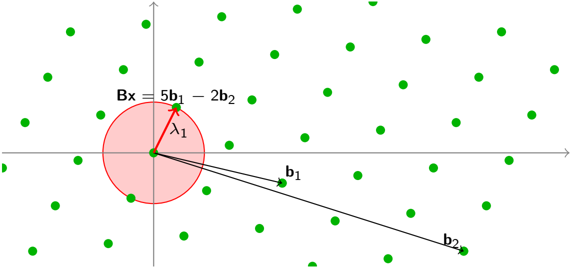
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP)

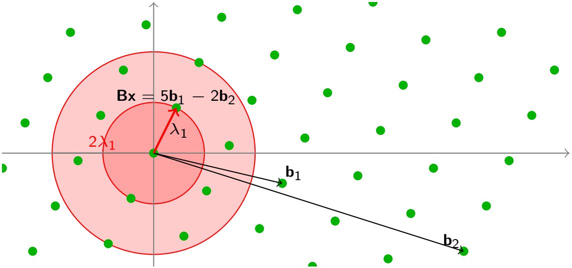
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \lambda_1$



Shortest Vector Problem

Definition (Shortest Vector Problem, SVP_γ)

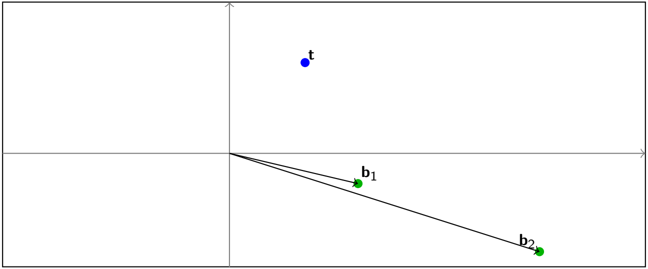
Given a lattice $\mathcal{L}(\mathbf{B})$, find a (nonzero) lattice vector $\mathbf{Bx}$ (with $\mathbf{x} \in \mathbb{Z}^k$) of length (at most) $\|\mathbf{Bx}\| \leq \gamma\lambda_1$



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

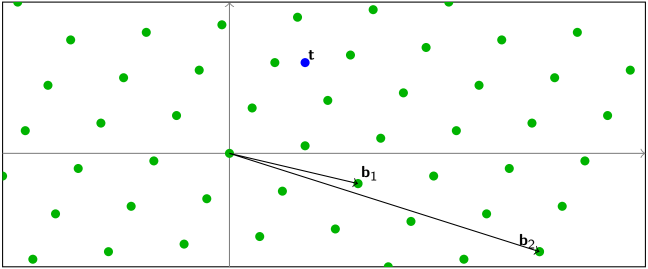
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

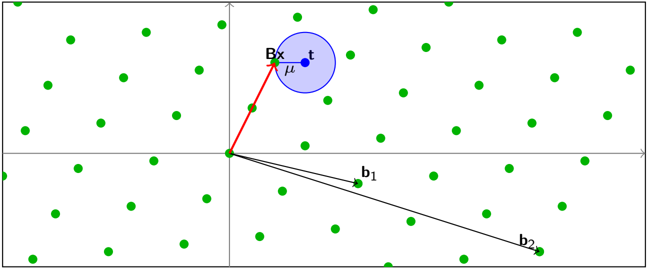
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP)

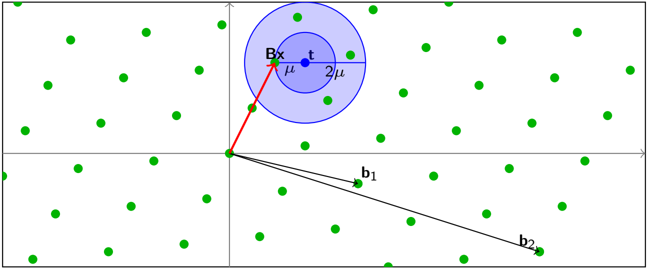
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \mu$ from the target



Closest Vector Problem

Definition (Closest Vector Problem, CVP_γ)

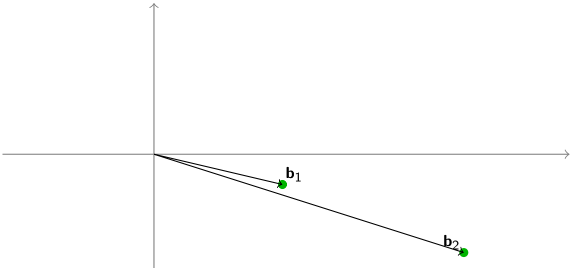
Given a lattice $\mathcal{L}(\mathbf{B})$ and a target point $\mathbf{t}$, find a lattice vector $\mathbf{Bx}$ within distance $\|\mathbf{Bx} - \mathbf{t}\| \leq \gamma\mu$ from the target



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

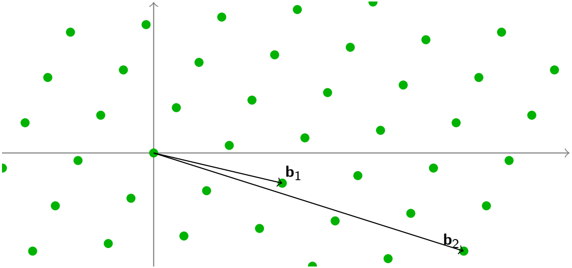
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

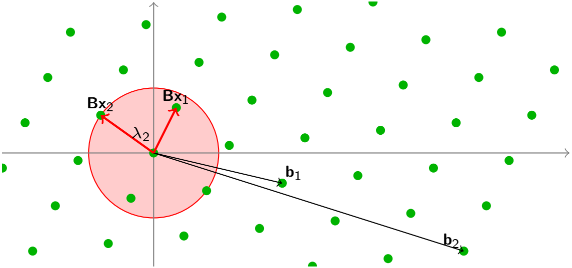
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP)

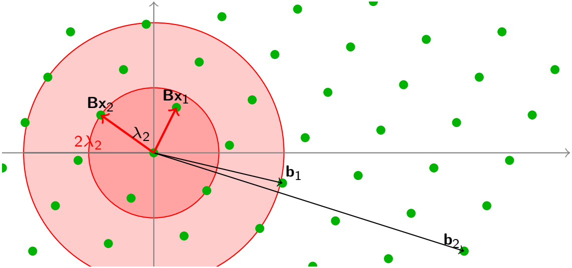
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \lambda_n$



Shortest Independent Vectors Problem

Definition (Shortest Independent Vectors Problem, SIVP_γ)

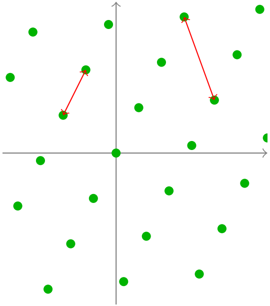
Given a lattice $\mathcal{L}(\mathbf{B})$, find n linearly independent lattice vectors $\mathbf{Bx}_1, \ldots, \mathbf{Bx}_n$ of length (at most) $\max_i \|\mathbf{Bx}_i\| \leq \gamma\lambda_n$



Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$

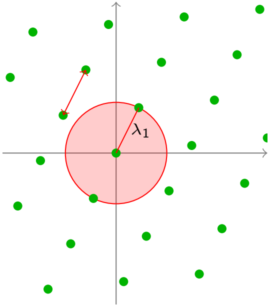


Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y} \in \mathcal{L}, \mathbf{x} \neq \mathbf{y}} \|\mathbf{x} - \mathbf{y}\|$$

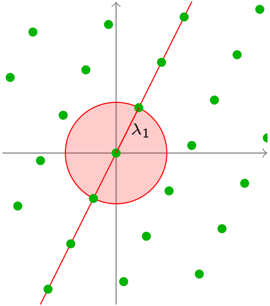
$$= \min_{\mathbf{x} \in \mathcal{L}, \mathbf{x} \neq \mathbf{0}} \|\mathbf{x}\|$$



Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$



- Successive minima ($i = 1, \ldots, n$)

$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

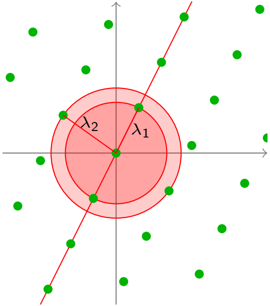
Minimum Distance and Successive Minima

- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$

- Successive minima ($i = 1, \ldots, n$)

$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$



Minimum Distance and Successive Minima

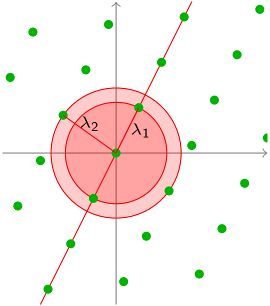
- Minimum distance

$$\lambda_1 = \min_{\mathbf{x},\mathbf{y}\in\mathcal{L},\mathbf{x}\neq\mathbf{y}} \|\mathbf{x}-\mathbf{y}\| = \min_{\mathbf{x}\in\mathcal{L},\mathbf{x}\neq\mathbf{0}} \|\mathbf{x}\|$$

- Successive minima ($i = 1, \ldots, n$)

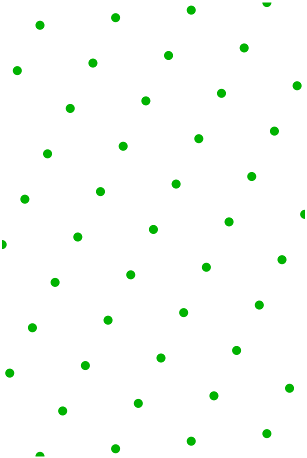
$$\lambda_i = \min\{r : \dim \operatorname{span}(\mathcal{B}(r) \cap \mathcal{L}) \geq i\}$$

- Examples
  - $\mathbb{Z}^n$: $\lambda_1 = \lambda_2 = \ldots = \lambda_n = 1$
  - Always: $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$



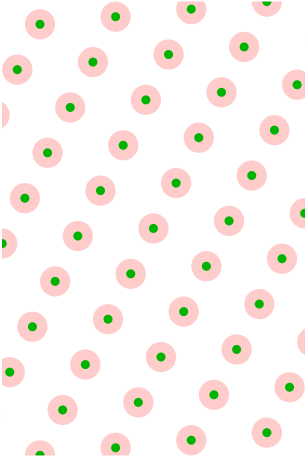
Blurring a lattice

Consider a lattice Λ, and



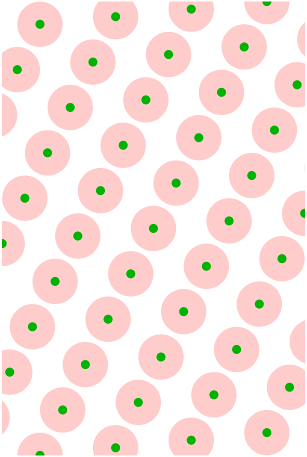
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered.



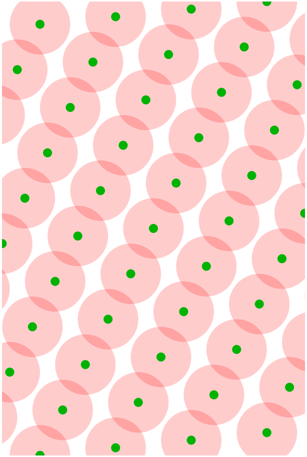
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered.



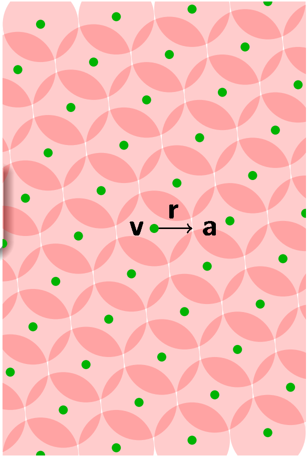
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered.



Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered.



How much noise is needed?

$\|\mathbf{r}\| \leq \quad \sqrt{n} \cdot \lambda_n/2$

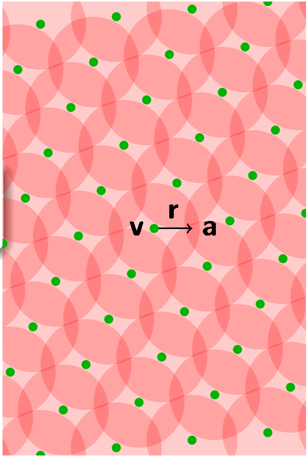
- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.

Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed?

$\|\mathbf{r}\| \leq \qquad \sqrt{n} \cdot \lambda_n/2$



- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.

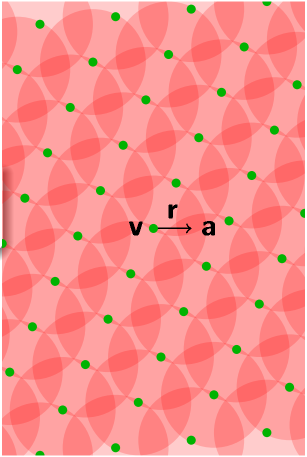
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed?

$\|\mathbf{r}\| \leq \qquad \sqrt{n} \cdot \lambda_n / 2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.

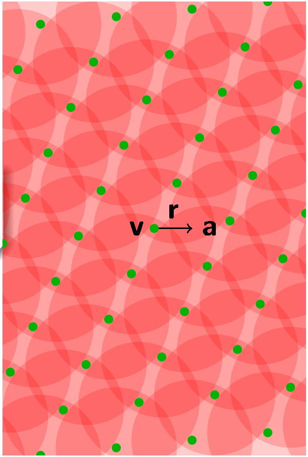


Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed?

$\|\mathbf{r}\| \leq \quad \sqrt{n} \cdot \lambda_n/2$



- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.

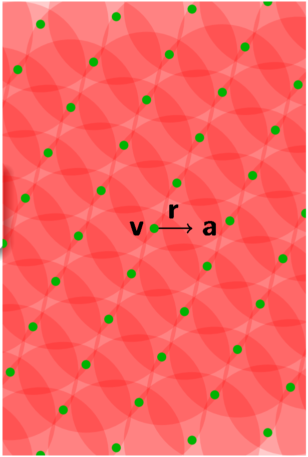
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed?

$\|\mathbf{r}\| \leq \quad \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.



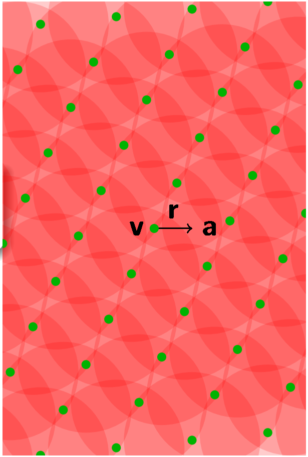
Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$

- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n/\Lambda$ is uniformly distributed.

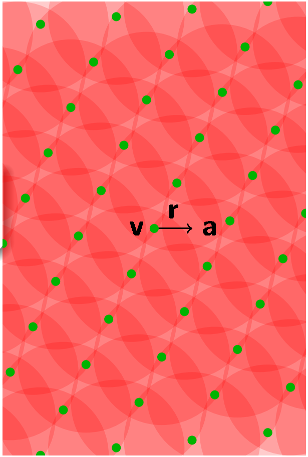


Blurring a lattice

Consider a lattice Λ, and add noise to each lattice point until the entire space is covered. Increase the noise until the space is uniformly covered.

How much noise is needed? [MR]

$\|\mathbf{r}\| \leq (\log n) \cdot \sqrt{n} \cdot \lambda_n/2$



- Each point in $\mathbf{a} \in \mathbb{R}^n$ can be written $\mathbf{a} = \mathbf{v} + \mathbf{r}$ where $\mathbf{v} \in \mathcal{L}$ and $\|\mathbf{r}\| \approx \sqrt{n}\lambda_n$.
- $\mathbf{a} \in \mathbb{R}^n/\Lambda$ is uniformly distributed.
- Think of $\mathbb{R}^n \approx \frac{1}{q}\Lambda$ [GPV'07]

Average-case hardness (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i \in \frac{1}{q}\Lambda$, where
  - $\mathbf{v}_i \in \Lambda$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \approx \frac{1}{q}\Lambda^m \equiv \mathbb{Z}_q^{n \times m}$
- Assume we can find a short lattice vector $\mathbf{z} \in \mathbb{Z}^m$

$$\mathbf{Az} = \mathbf{0}$$

Average-case hardness (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i \in \frac{1}{q}\Lambda$, where
  - $\mathbf{v}_i \in \Lambda$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \approx \frac{1}{q}\Lambda^m \equiv \mathbb{Z}_q^{n\times m}$
- Assume we can find a short lattice vector $\mathbf{z} \in \mathbb{Z}^m$

$$\sum(\mathbf{v}_i + \mathbf{r}_i)z_i = \sum \mathbf{a}_i z_i = \mathbf{A}\mathbf{z} = \mathbf{0}$$

Average-case hardness (sketch)

- Generate random points $\mathbf{a}_i = \mathbf{v}_i + \mathbf{r}_i \in \frac{1}{q}\Lambda$, where
  - $\mathbf{v}_i \in \Lambda$ is a random lattice point
  - $\mathbf{r}_i$ is a random error vector of length $\|\mathbf{r}_i\| \approx \sqrt{n}\lambda_n$
- $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \approx \frac{1}{q}\Lambda^m \equiv \mathbb{Z}_q^{n \times m}$
- Assume we can find a short lattice vector $\mathbf{z} \in \mathbb{Z}^m$

$$\sum (\mathbf{v}_i + \mathbf{r}_i) z_i = \sum \mathbf{a}_i z_i = \mathbf{A}\mathbf{z} = \mathbf{0}$$

- Rearranging the terms yields a lattice vector

$$\sum \mathbf{v}_i z_i = -\sum \mathbf{r}_i z_i$$

of length at most $\|\sum \mathbf{r}_i z_i\| \approx \sqrt{m} \cdot \max \|\mathbf{r}_i\| \approx n \cdot \lambda_n$

Shortcomings of Ajtai's function

Expressivity:

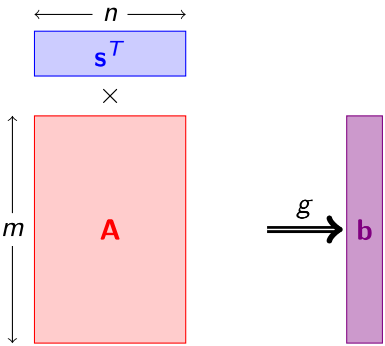
- Ajtai's proof requires $m > n \log q$
- The function $f_{\mathbf{A}} : \{0,1\}^m \to \mathbb{Z}_q^n$ is not injective
- Enough for one-way functions, collision resistant hashing, some digital siguatures, commitments, identification, etc.
- ... but (public key) encryption seem to require stronger assumptions.
- 1996: Ajtai-Dwork cryptosystem, based on the "unique" Shortest Vector Problem.

Efficiency:

- The matrix/key $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ requires $\Omega(n^2)$ storage (and computation)
- 1996: NTRU Cryptosystem, efficient, but not supported by security proof from worst-case lattice problems.

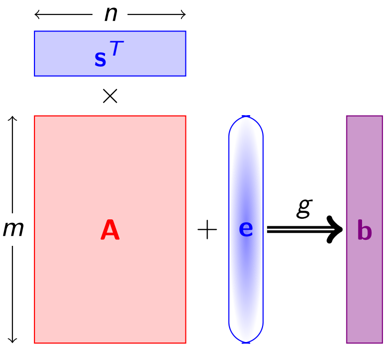
Learning with errors (LWE)

- $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$.
- $g_{\mathbf{A}}(\mathbf{s} \quad) = \mathbf{A}\mathbf{s} \qquad \bmod q$



Learning with errors (LWE)

- $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$.
- $g_{\mathbf{A}}(\mathbf{s};\mathbf{e}) = \mathbf{As} + \mathbf{e} \bmod q$
- Learning with Errors: Given $\mathbf{A}$ and $g_{\mathbf{A}}(\mathbf{s},\mathbf{e})$, recover $\mathbf{s}$.

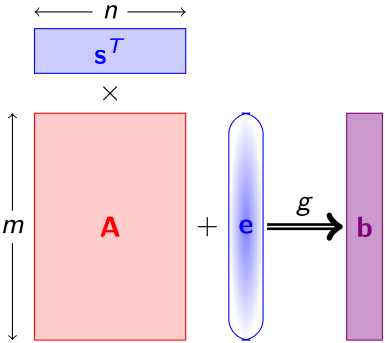


Learning with errors (LWE)

- $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathcal{E}^m$.
- $g_{\mathbf{A}}(\mathbf{s}; \mathbf{e}) = \mathbf{As} + \mathbf{e} \bmod q$
- Learning with Errors: Given $\mathbf{A}$ and $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$, recover $\mathbf{s}$.

Theorem (Regev'05)

The function $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is hard to invert on the average, assuming SIVP is hard to approximate in the worst-case even for quantum computers.

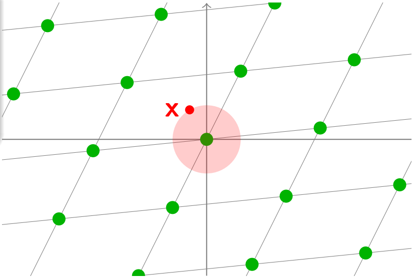


SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$

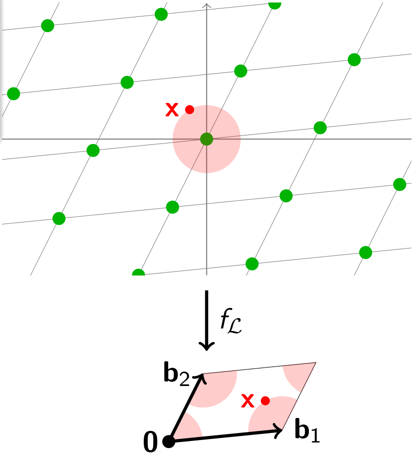
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$



SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

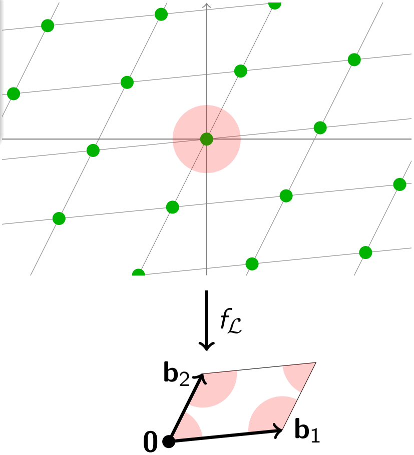


SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective

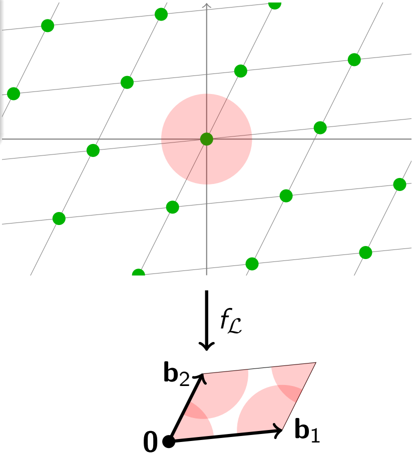


SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective

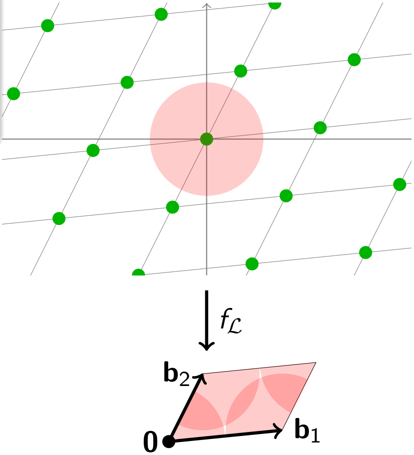


SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $f_{\mathcal{L}}$ is surjective

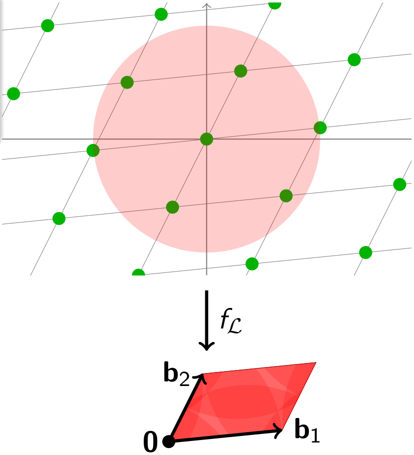


SIS/LWE as CVP

Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $f_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $f_{\mathcal{L}}(\mathbf{x})$ is almost uniform



SIS/LWE as CVP

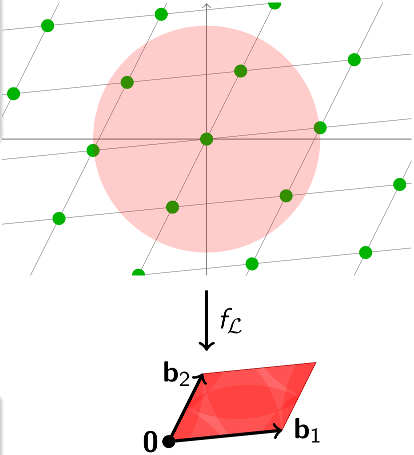
Candidate OWF

Key: a hard lattice $\mathcal{L}$
Input: $\mathbf{x}$, $\|\mathbf{x}\| \leq \beta$
Output: $f_{\mathcal{L}}(\mathbf{x}) = \mathbf{x} \bmod \mathcal{L}$

- $\beta < \lambda_1/2$: $f_{\mathcal{L}}$ is injective
- $\beta > \lambda_1/2$: $f_{\mathcal{L}}$ is not injective
- $\beta \geq \mu$: $f_{\mathcal{L}}$ is surjective
- $\beta \gg \mu$: $f_{\mathcal{L}}(\mathbf{x})$ is almost uniform

Question

Are these functions cryptographically hard to invert?



Special Versions of CVP

Definition (Closest Vector Problem (CVP))

Given $(\mathcal{L}, \mathbf{t}, d)$, with $\mu(\mathbf{t}, \mathcal{L}) \leq d$, find a lattice point within distance d from $\mathbf{t}$.

- If d is arbitrary, then one can find the closest lattice vector by binary search on d.
- Bounded Distance Decoding (BDD): If $d < \lambda_1(\mathcal{L})/2$, then there is at most one solution. Solution is the closest lattice vector.
- Absolute Distance Decoding (ADD): If $d \geq \rho(\mathcal{L})$, then there is always at least one solution. Solution may not be closest lattice vector.

Computational problems on random lattices

Ajtai's class of random lattices an their duals:

$$\begin{aligned} \mathbf{A} &\in \mathbb{Z}^{n\times m} \\ \Lambda_q^{\perp}(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\} \\ \Lambda_q(\mathbf{A}) &= \mathbf{A}^T\mathbb{Z}^n + q\mathbb{Z}^m \end{aligned}$$

Inverting Ajtai's function $\mathbf{A}\mathbf{x} = \mathbf{b}$

- Solution $\mathbf{x}$ always exist, but it is hard to find
- Average case version of ADD on random $\Lambda_q^{\perp}(\mathbf{A})$

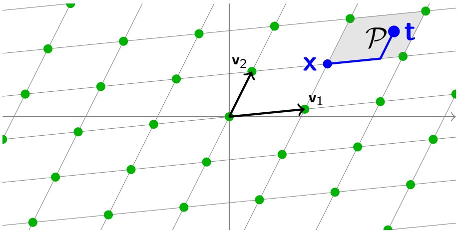
Solving LWE $\mathbf{s}\mathbf{A} + \mathbf{x} = \mathbf{b}$

- For small enough $\mathbf{x}$, solution is unique
- Average case version of BDD on random dual lattice $\Lambda_q(\mathbf{A})$.

ADD reduces to SIVP

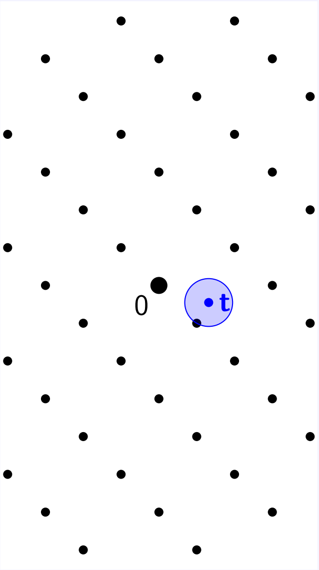
ADD input: $\mathcal{L}$ and arbitrary $\mathbf{t}$

- Compute short vectors $\mathbf{V} = \text{SIVP}(\mathcal{L})$
- Use $\mathbf{V}$ to find a lattice vector within distance $\sum_i \frac{1}{2}\|\mathbf{v}_i\| \leq (n/2)\lambda_n \leq n\rho$ from $\mathbf{t}$



BDD reduces to SIVP

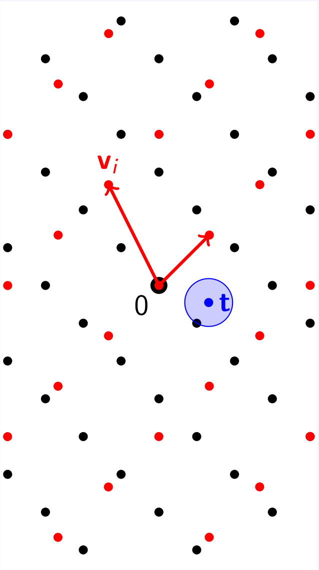
BDD input: $\mathbf{t}$ close to $\mathcal{L}$



BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

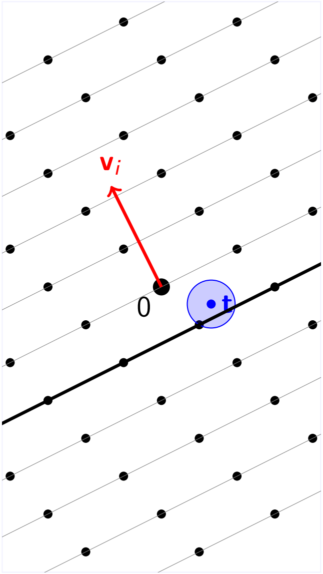
- Compute $\mathbf{V} = \text{SIVP}(\mathcal{L}^*)$



BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

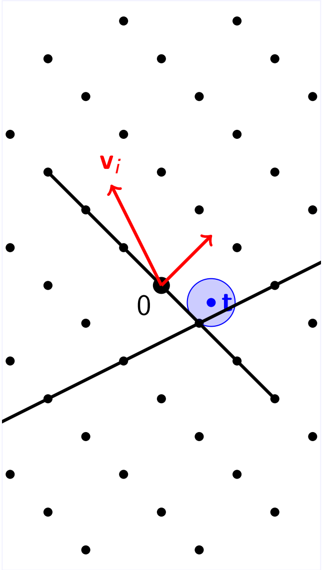
- Compute $\mathbf{V} = \text{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$



BDD reduces to SIVP

BDD input: $\mathbf{t}$ close to $\mathcal{L}$

- Compute $\mathbf{V} = \text{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to $\mathbf{t}$
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$

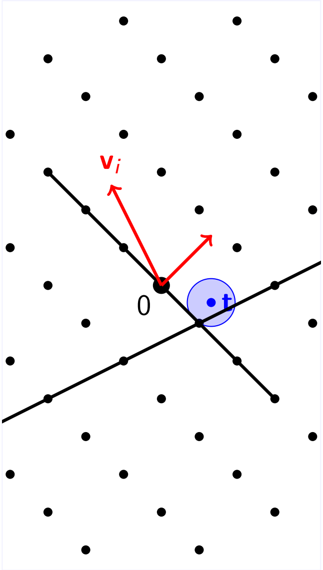


BDD reduces to SIVP

BDD input: **t** close to $\mathcal{L}$

- Compute $\mathbf{V} = \text{SIVP}(\mathcal{L}^*)$
- For each $\mathbf{v}_i \in \mathcal{L}^*$, find the layer $L_i = \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{v}_i = c_i\}$ closest to **t**
- Output $L_1 \cap L_2 \cap \cdots \cap L_n$
- Output is correct as long as

$$\mu(\mathbf{t}, \mathcal{L}) \leq \frac{\lambda_1}{2n} \leq \frac{1}{2\lambda_n^*} \leq \frac{1}{2\|\mathbf{v}_i\|}$$



Special Versions of SVP and SIVP

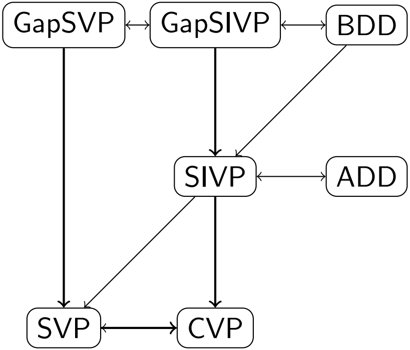
- GapSVP: compute (or approximate) the value λ_1 without necessarily finding a short vector
- GapSIVP: compute (or approximate) the value λ_n without necessarily finding short linearly independent vectors
- Transference Theorem $\lambda_1 \approx 1/\lambda_n^*$: GapSVP can be (approximately) solved by solving GapSIVP in the dual lattice, and vice versa

Problems

- **Exercise:** Computing λ_1 (or λ_n) exactly is as hard as SVP (or SIVP)
- **Open Problem:** Reduce approximate SVP (or SIVP) to approximate GapSVP (or GapSIVP)

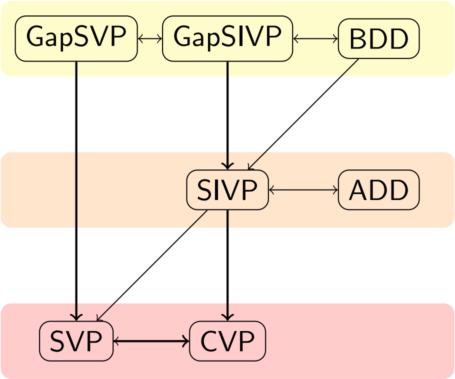
Relations among lattice problems

- SIVP $\approx$ ADD [MG'01]
- SVP $\leq$ CVP [GMSS'99]
- SIVP $\leq$ CVP [M'08]
- BDD $\lesssim$ SIVP
- CVP $\lesssim$ SVP [L'87]
- GapSVP $\approx$ GapSIVP [LLS'91,B'93]
- GapSVP $\lesssim$ BDD [LM'09]



Relations among lattice problems

- SIVP $\approx$ ADD [MG'01]
- SVP $\leq$ CVP [GMSS'99]
- SIVP $\leq$ CVP [M'08]
- BDD $\lesssim$ SIVP
- CVP $\lesssim$ SVP [L'87]
- GapSVP $\approx$ GapSIVP [LLS'91,B'93]
- GapSVP $\lesssim$ BDD [LM'09]



Open Problems

- Does the ability to approximate λ_1 helps in solving SVP?
- Does the ability to approximate λ_n helps in solving SIVP?
- Is there a reduction from CVP/SVP to SIVP?
  - Yes, for the exact version of the problems [M. 08]
  - Open for approximation version
- Is there a classical (nonquantum) reduction from SIVP/ADD to GapSVP/BDD?

Idea

Use structured matrix

$$\mathbf{A} = [\mathbf{A}^{(1)} \mid \ldots \mid \mathbf{A}^{(m/n)}]$$

where $\mathbf{A}^{(i)} \in \mathbb{Z}_q^{n\times n}$ is circulant

$$\mathbf{A}^{(i)} = \begin{bmatrix} a_1^{(i)} & a_n^{(i)} & \cdots & a_2^{(i)} \\ a_2^{(i)} & a_1^{(i)} & \cdots & a_3^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ a_n^{(i)} & a_{n-1}^{(i)} & \cdots & a_1^{(i)} \end{bmatrix}$$

- "Generalized Compact Knapsacks and Efficient One-Way Functions" (Micciancio, FOCS 2002)
- Efficient version of Ajtai's connection:
  - $O(n \log n)$ space and time complexity
  - Provable security: guidance on how to choose random instances.

Theorem

"CyclicSIS" is hard to invert on average, assuming the worst-case hardness of lattice problems over "cyclic" lattices.

Ideal Lattices and Algebraic number theory

- Isomorphism: $\mathbf{A}^{cyc} \leftrightarrow \mathbb{Z}[X]/(X^n - 1)$
- Cyclic SIS:

$$f_{\mathbf{a}_1,\ldots,\mathbf{a}_k}(\mathbf{u}_1,\ldots,\mathbf{u}_k) = \sum_i \mathbf{a}_i(X) \cdot \mathbf{u}_i(X) \pmod{X^n - 1}$$

  where $a_i, u_i \in R = \mathbb{Z}[X]/(X^n - 1)$.
- More generally, use $R = \mathbb{Z}[X]/p(X)$ for some monic polynomial $p(X) \in \mathbb{Z}[X]$
- If $p(X)$ is irreducible, then finding collisions to $f_{\mathbf{a}}$ for random $\mathbf{a}$ is as hard as solving lattice problems in the worst case in ideal lattices
- Can set R to the ring of integers of $K = Q[X]/p(X)$.

How to choose $p(X)/R$?

RingSIS (Lyubashevsky, PhD Thesis, UCSD 2008)

- define $f_{\mathbf{a}}(\mathbf{u}) = \sum_i \mathbf{a}_i(X) \cdot u_i(X)$
- Notice: no reduction modulo $p(X)$!
- If $f_{\mathbf{a}}(\mathbf{u}) = f_{\mathbf{a}}(\mathbf{u}')$ in $\mathbb{Z}[X]$, then $f_{\mathbf{a}}(\mathbf{u}) = f_{\mathbf{a}}(\mathbf{u}') \pmod{p(X)}$.
- Conclusion: breaking f is at least as hard as solving lattices problems in ideal lattices for *any* $p(X)$.

How to choose $p(X)/R$?

RingSIS (Lyubashevsky, PhD Thesis, UCSD 2008)

- define $f_{\mathbf{a}}(\mathbf{u}) = \sum_i \mathbf{a}_i(X) \cdot u_i(X)$
- Notice: no reduction modulo $p(X)$!
- If $f_{\mathbf{a}}(\mathbf{u}) = f_{\mathbf{a}}(\mathbf{u}')$ in $\mathbb{Z}[X]$, then $f_{\mathbf{a}}(\mathbf{u}) = f_{\mathbf{a}}(\mathbf{u}') \pmod{p(X)}$.
- Conclusion: breaking f is at least as hard as solving lattices problems in ideal lattices for *any* $p(X)$.

RingLWE:

- Most applications require not only hardness of inverting $f_{\mathbf{a}}$, but also pseudorandomness of output $f_{\mathbf{a}}(\mathbf{u})$
- [Lyubashevsky,Peikert,Regev'10]: For cyclotomic $p(X)$, hardness of inverting $f_{\mathbf{a}}$ implies pseudorandomness of $f_{\mathbf{a}}(\mathbf{u})$.
- [Lauter'15] constructs polynomial rings where inverting $f_{\mathbf{a}}$ is conceivably hard, but $f_{\mathbf{a}}(\mathbf{u})$ is easily distinguished from random.

Classical Hardness of LWE

- [P'09, BLPRS'13] There is a classical reduction from GapSVP to LWE when $q = 2^{O(n)}$, or LWE dimension $d = O(n^2)$

Open Problems

- Is there a more efficient reduction from GapSVP to LWE?
- Is there a classical reduction from SIVP to LWE?
- Is there a reduction from SVP/SIVP to LWE on ideal lattices?

More Open Problems – Tonight 7:30pm

- Bring your own open problems to share!
- Send email to

  daniele@cs.ucsd.edu

  with estimated time for scheduling.
- . . . or, just talk to me over lunch or coffee break.

More Open Problems – Tonight 7:30pm

- Bring your own open problems to share!
- Send email to

  daniele@cs.ucsd.edu

  with estimated time for scheduling.
- . . . or, just talk to me over lunch or coffee break.

Thank you!