Lattice algorithms for variants of LWE

Shi Bai

Florida Atlantic University, Boca Raton.

PQCifris. 04/2022.

Outline

- LWE and SIS problems
- Algorithms for LWE
- Lattices and LWE
  - embed secret into a lattice point
  - concrete complexity
- Summary and discussion

1. Introduction to LWE/SIS.

LWE

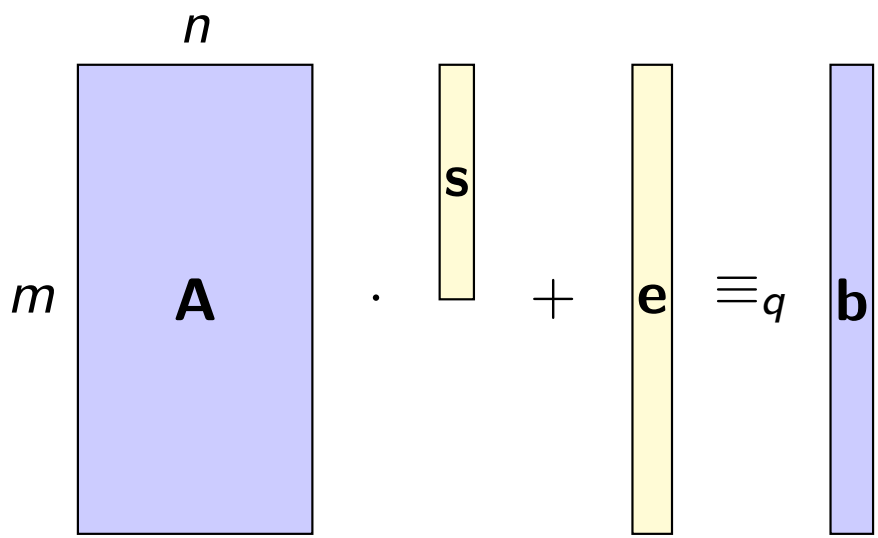
Matrix-form LWE: input security parameter λ, choose parameters n, q, m and two distribution $D_{\mathbf{s}}$, $D_{\mathbf{e}}$.

- sample uniform $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$.
- sample $\mathbf{s}$ according to $D_{\mathbf{s}}$.
- sample "small" $\mathbf{e}$ according to $D_{\mathbf{e}}$.

Problem: given $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b} = \mathbf{As} + \mathbf{e} \pmod{\mathbf{q}}$, recover $\mathbf{s}$ (or $\mathbf{e}$).

E.g. $n = \theta(\lambda)$; $q \approx 1.5n^2$; $m \approx 1.1n \log q$; $D_{\mathbf{s}}$ uniform on $\mathbb{Z}_q^n$; $D_{\mathbf{e}}$ discrete Gaussian with deviation $\sigma \approx 2\sqrt{n}$. Denote $\sigma = \alpha q$.

LWE



- Search version: Given $(\mathbf{A}, \mathbf{b})$, find $\mathbf{s}$ (or $\mathbf{e}$).
- Decisional version: Given samples $(\mathbf{A}, \mathbf{b})$ (either LWE or uniform), decide whether they are LWE samples or uniformly random samples.

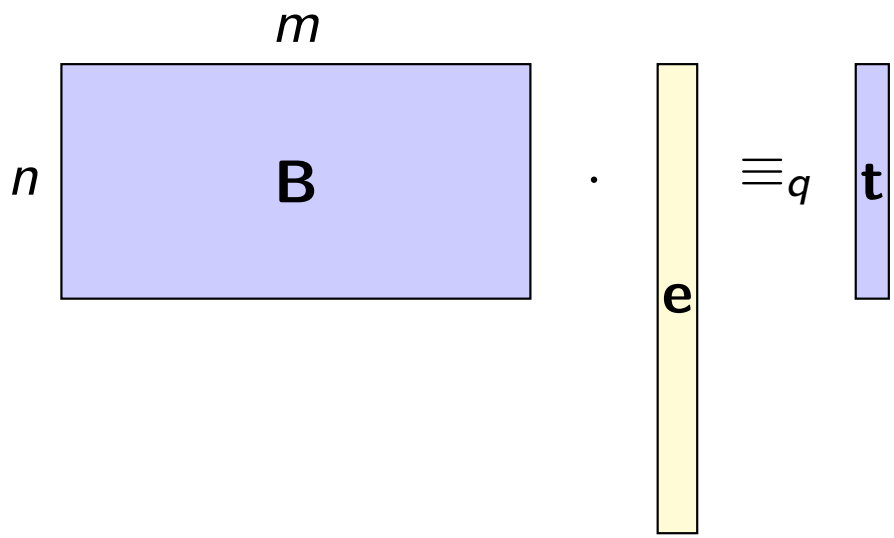
ISIS (inhomogeneous short integer solution)

Matrix-form ISIS: input security parameter λ, choose parameters n, q, m and a distribution $D_{\mathbf{e}}$.

- sample uniform $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$.
- sample "small" $\mathbf{e}$ according to $D_{\mathbf{e}}$.

Problem: given $(\mathbf{B}, \mathbf{t})$ where $\mathbf{t} = \mathbf{Be} \pmod{\mathbf{q}}$, recover $\mathbf{e}$.

ISIS



LWE variants

Variants of LWE $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$.

Distribution of $\mathbf{s}$, $\mathbf{e}$ ("small" means standard):

- $\mathbf{s}$ uniform, $\mathbf{e}$ small.
- Normal form: $\mathbf{s}$ small, $\mathbf{e}$ small.
- variant 1: $\mathbf{s}$ uniform, $\mathbf{e}$ tiny. (e.g. binary error LWE)
- variant 2: $\mathbf{s}$ tiny, $\mathbf{e}$ tiny. (e.g. binary secret-error LWE)
- variant 3: $\mathbf{s}$ tiny, $\mathbf{e}$ small. (e.g. binary secret LWE)
- variant 4: $\mathbf{s}$ sparse, $\mathbf{e}$ small. (e.g. sparse secret LWE)
- More variants: # samples restricted; Modulus q large.

Lots of applications: signatures (Dilithium, qTESLA), KE (Newhope, Kyber) and HE schemes (HElib, SEAL) and many others.

Combinations of distributions, number of samples, e.g. give variants with various difficulty – **concrete security level** needs to be carefully analyzed.

2. Algorithms for LWE

Three types of algorithms

- Algebraic: Arora-Ge algorithm and variants.
- Combinatoric: Blum-Kalai-Wasserman (BKW) algorithm and variants.
- Geometric: phrase the LWE instance as some lattice problem and solve this problem using lattice solvers (e.g. lattice reduction or sieving).

More attacks on structured LWE problem (of the aforementioned variants) using algebraic structure: Elias-Lauter-Ozman-Stange '15; Chen-Lauter-Stange '16, '17; Castryck-Iliashenko-Vercauteren '16; Peikert '16.

We will focus on LWE in general q-ary lattice in this talk.

Algebraic algorithms

Algebraic attacks, e.g. Arora-Ge algorithm and variants.

- Binary error LWE:
  - Poly. with samples $m = O(n^2)$.
  - Subexp. with samples $m = O(n \log\log n)$ (Albrecht-Cid-Faugere-Perret 14').
  - $m \gtrapprox n$ for the reduction to SIVP-γ with poly. γ in dimension $\Theta(n/\log n)$ (Micciancio-Peikert '13).
- Tiny error LWE:
  - $2^{\tilde{O}(\alpha^2 q^2)}$; thus already subexp for $\alpha q < \sqrt{n}$ with enough (same) samples.
- Small (standard) error LWE:
  - $2^{O(n \log\log n)}$ for $\alpha q = \sqrt{n}$. Slower than sieving/BKW.
  - $2^{O(n)}$ using Gröbner basis (Albrecht-Cid-Faugere-Perret 14').

For full power, need # Samples $\approx n^{2w}$ where w bounds the width of error.

Combinatoric algorithms

Combinatoric attacks: BKW-like algorithms (Blum-Kalai-Wasserman '99).

- LPN ($q = 2$):
  - $2^{O(n/\log n)}$ samples/time.
  - $2^{O(n/\log\log n)}$ time with $n^{1+\epsilon}$ samples (Lyubashevsky '05).
- Binary secret LWE:
  - $2^{O(n/\log\log n)}$ time with $poly(n)$ samples (Kirchner-Fouque '15).
- LWE: $2^{O(n)}$ (Albrecht-Faugere-Fitzpatric-Perret '14).

Also meet-in-the-middle type algorithms: useful for sparse $\mathbf{s}$ or $\mathbf{e}$.

Geometric algorithms

Geometric methods: turn the LWE into a problem on lattices (tools: lattice reduction and lattice sieving).

Feature: # LWE samples are usually small.

Quick summary (asymptotic running-time): when q is polynomial in n,

- LWE: $2^{O(n)}$.
- Binary secret-error LWE: $2^{O(n)}$.
- Binary secret but small error LWE: $2^{O(n)}$.
- Binary error but small/uniform secret LWE: $2^{O(n)}$.

Thus, these lattice algorithms are mostly relevant in terms of concrete security levels.

3. Lattices and LWE

Euclidean lattice

An integral lattice can be defined as the $\mathbb{Z}$-linear combination of n independent vectors $\mathbf{b}_i \in \mathbb{Z}^n$

$$L(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} \mathbb{Z}\mathbf{b}_i \right\}.$$

Let $\mathbf{B} = [\mathbf{b}_1\, \mathbf{b}_2\, \cdots\, \mathbf{b}_n]$ then $\Lambda = L(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

Euclidean lattice

An integral lattice can be defined as the $\mathbb{Z}$-linear combination of n independent vectors $\mathbf{b}_i \in \mathbb{Z}^n$

$$L(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n) = \left\{ \sum_{i=1}^{n} \mathbb{Z}\mathbf{b}_i \right\}.$$

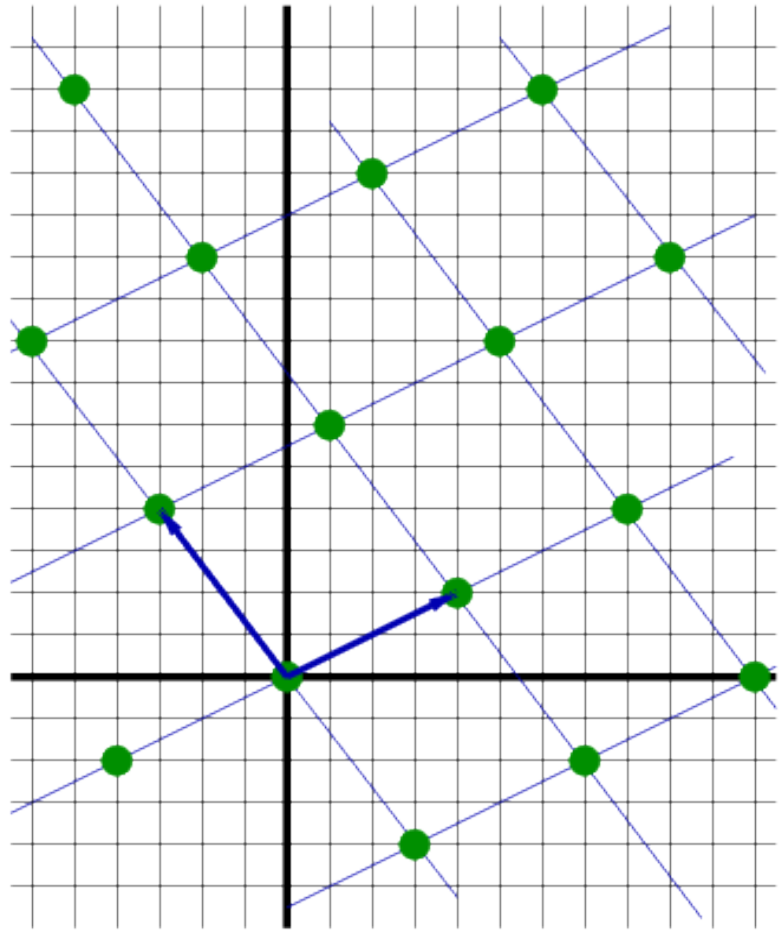
Let $\mathbf{B} = [\mathbf{b}_1\, \mathbf{b}_2\, \cdots\, \mathbf{b}_n]$ then $\Lambda = L(\mathbf{B}) = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

The volume of a lattice Λ is $|\det(\mathbf{B})|$, which is independent of the choice of the basis.

Lattice minimum

Lattice minimum

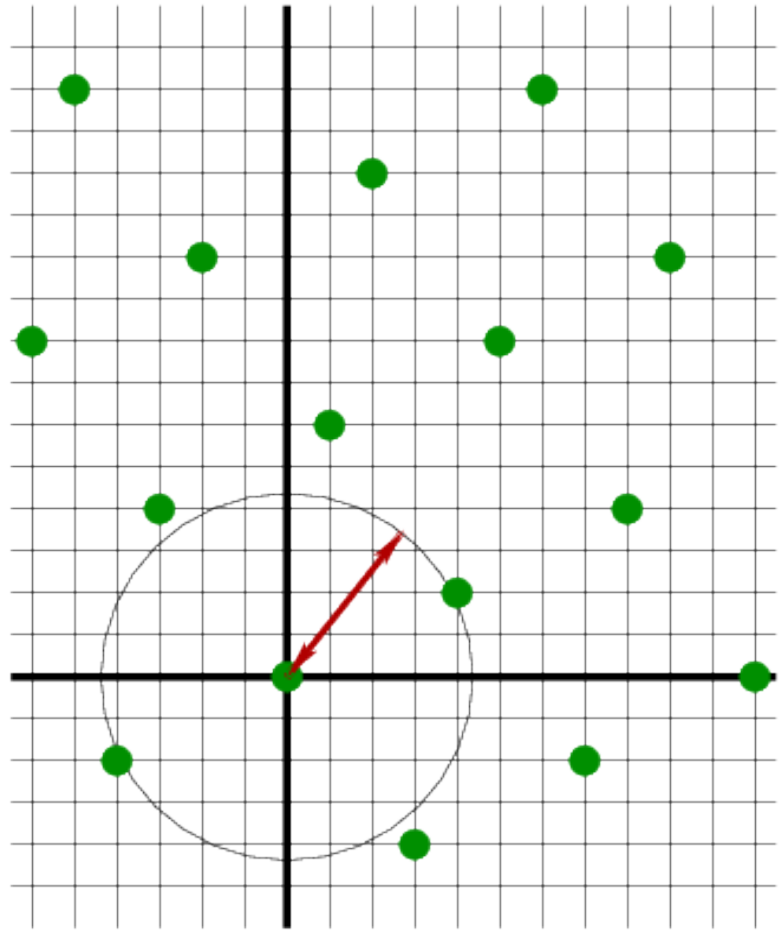
$\lambda_1(\Lambda) = \min\,(\ \|\mathbf{b}\|\colon \mathbf{b} \in \Lambda \setminus \mathbf{0}\ )$



Lattice minimum

Lattice minimum

$\lambda_1(\Lambda) = \min\,(\ \|\mathbf{b}\| \colon \mathbf{b} \in \Lambda \setminus \mathbf{0}\ )$

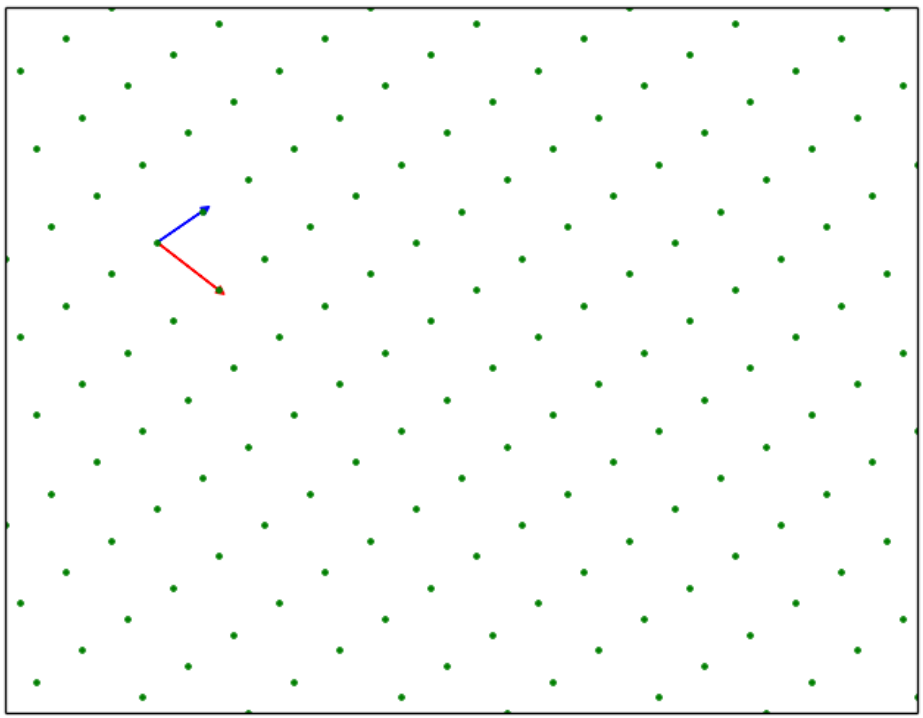


Computational problems for lattices

Shortest vector problem (SVP)

Input: $\mathbf{B} \in \mathbb{Z}^{n \times n}$ a basis matrix of Λ.
Output: $\mathbf{s} \in \Lambda \setminus \mathbf{0}$ shortest.

The difficulty **heavily** depends on the "shape" of the input basis $\mathbf{B}$. In cryptography, a "bad" $\mathbf{B}$ is given.

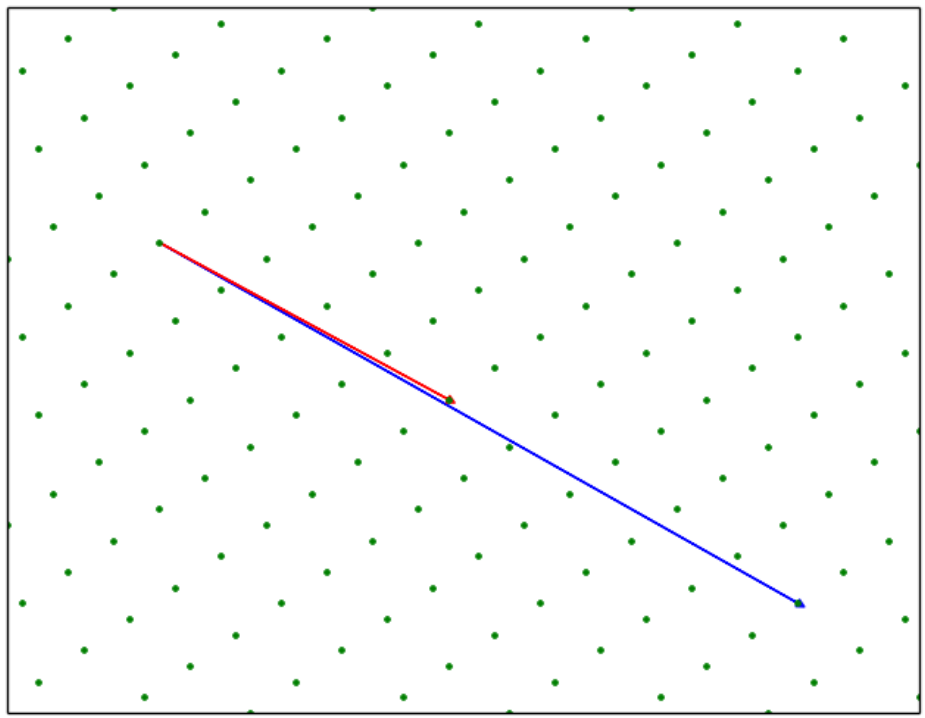


Computational problems for lattices

Shortest vector problem (SVP)

Input: $\mathbf{B} \in \mathbb{Z}^{n \times n}$ a basis matrix of Λ.
Output: $\mathbf{s} \in \Lambda \setminus \mathbf{0}$ shortest.

The difficulty **heavily** depends on the "shape" of the input basis $\mathbf{B}$. In cryptography, a "bad" $\mathbf{B}$ is given.



SVP-γ

Cryptography needs to use a relaxed version: SVP-γ s.t. γ depends on n.

- SVP-1: enumeration or sieving.
- SVP-γ: Block Korkine-Zolotarev (BKZ) reduction.
  - γ is exponential in n: Lenstra-Lenstra-Lovász (LLL) algorithm.
  - γ is polynomial/sub-exponential in n: cryptography.

Algorithms for SVP-γ and algorithms for SVP-1 are reciprocal.

SVP-γ

Cryptography needs to use a relaxed version: SVP-γ s.t. γ depends on n.

- SVP-1: enumeration or sieving.
- SVP-γ: Block Korkine-Zolotarev (BKZ) reduction.
  - γ is exponential in n: Lenstra-Lenstra-Lovász (LLL) algorithm.
  - γ is polynomial/sub-exponential in n: cryptography.

Algorithms for SVP-γ and algorithms for SVP-1 are reciprocal.

(1) Best algorithm for SVP-γ: BKZ-β whose complexity is dominated by SVP-1 in smaller dim β.

(2) Best algorithm for SVP-1 at dimension β.

- Enumeration $2^{O(\beta \log \beta)}$ (Kannan-Fincke-Pohst '83).
- Sieving $2^{O(\beta)}$ (Ajtai-Kumar-Sivakumar '01).

Linking (1) and (2), e.g. determine "required" β given SVP-γ in dim n? For the BKZ algorithm to succeed in recovering the shortest, we need:

$$\gamma^{1/n} \geq \left(\frac{\beta}{2\pi e}\right)^{1/(2(\beta-1))}.$$

Use the smallest such β.

Computational problems: uSVP-γ

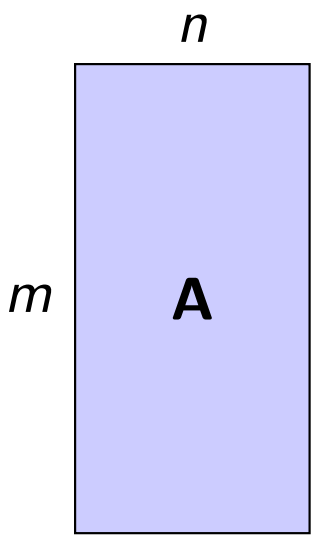
The LWE is closed related to the uSVP problem.

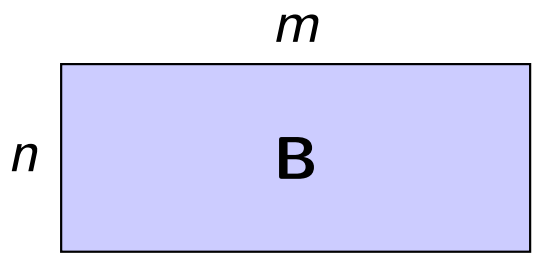
uSVP-γ

Let $\gamma \geq 1$. Given a input basis $\mathbf{B}$ such that $\frac{\lambda_2(\Lambda(B))}{\lambda_1(\Lambda(B))} \geq \gamma$, the goal is to find a non-zero lattice vector of norm $\lambda_1(\Lambda(B))$. We call the γ as the gap of the uSVP-γ problem.

Note: $\mathbf{B}$ is special in some sense, e.g., it has a gap. We get such $\mathbf{B}$ from an LWE instance.

Two q-ary lattices: $m > n$





$$\Lambda_q^{\perp}(\mathbf{B}) = \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{Be} = \mathbf{0} \pmod{}$$

$$\Lambda_q(\mathbf{A}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{Ax} \pmod{q}, \forall \mathbf{x} \in \mathbb{Z}^n\}$$

Sometimes, $\mathbf{B} = \mathbf{A}^T$. In such case,

$$\Lambda_q^{\perp}(\mathbf{A}^T) = q \cdot \Lambda_q(\mathbf{A})^*$$

where $\Lambda_q(\mathbf{A})^*$ is the dual lattice of $\Lambda_q(\mathbf{A})$.

Refer them as: image/column lattice and kernel lattice.

Direct decoding attack

Main idea: $\mathbf{e}$ is short.

The image lattice $\Lambda_q(\mathbf{A}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{A}\mathbf{x} \pmod q, \forall \mathbf{x} \in \mathbb{Z}^n\}$.

- Solve CVP/BDD on $\Lambda_q(\mathbf{A})$ given target point $\mathbf{b}$.
- The lattice has rank m and volume q^{m-n}.
- Convert to uSVP using Kannan's embedding.
- The concrete security depends on the number of samples m given.
- For best asymptotics, $m \approx -\frac{n \log q}{\log \alpha}$.

Asymptotic running-time with above m:

$$\left(\frac{n \log q}{\log^2 \alpha}\right)^{O\left(\frac{n \log q}{\log^2 \alpha}\right)}.$$

Note: binary error LWE does not change the asymptotics $2^{O(n)}$.

Direct decoding attack

Kannan's embedding: BDD $\rightarrow$ uSVP.

In LWE/BDD $\mathbf{b} = \mathbf{As} + \mathbf{e} + \mathbf{c}q$, thus $\mathbf{b}$ is close to the lattice point $\mathbf{As} + \mathbf{c}q$ in $\Lambda_q(\mathbf{A})$ where $\mathbf{e}$ is the small "shift". Let $\mathbf{L}$ be the basis of $\Lambda_q(\mathbf{A})$.

Construct

$$\mathbf{L}' = \begin{pmatrix} \mathbf{L} & \mathbf{b} \\ \mathbf{0} & 1 \end{pmatrix}.$$

We have

$$\mathbf{L}' \cdot \begin{pmatrix} * \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{e} \\ 1 \end{pmatrix}$$

where * is (negative) coefficients in generating the lattice point $\mathbf{As} + \mathbf{c}q$.

Direct decoding attack: example

Given n, q, α in LWE. Try each m up to some bound.

Let $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}$ be its matrix notation.

- BDD problem has error $\mathbf{e}$ with length $\|\mathbf{e}\|$.
- uSVP problem has gap $\gamma \approx \frac{\sqrt{\frac{m}{2\pi e}} q^{(m-n)/m}}{\|\mathbf{e}\|}$.
- To solve this uSVP problem, we need a BKZ-β algorithm where

$$\gamma^{1/m} \geq \left(\frac{\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}$$

.

- From above, we can compute smallest β. Running-time (with sieving as SVP-core) is $\approx 2^{0.265\beta}$.

We choose the smallest such β for some m.

This is close to the method in Lattice Estimator*.

*https://github.com/malb/lattice-estimator

Solving an ISIS-like problem

LWE $\rightarrow$ ISIS-like $\rightarrow$ BDD $\rightarrow$ uSVP, solve by BKZ.

$$\mathbf{A}^{\perp}\mathbf{b} = \mathbf{A}^{\perp}(\mathbf{A}\mathbf{s} + \mathbf{e}) = \mathbf{A}^{\perp} \cdot \mathbf{e} \pmod{q}.$$

Find $\mathbf{e}$ by solving a ISIS-like problem.

General way to solve ISIS: $\mathbf{B} \cdot \mathbf{e} = \mathbf{t} \pmod{q}$.

- Find arbitrary (not necessarily short) $\mathbf{y}$ such that $\mathbf{B} \cdot \mathbf{y} = \mathbf{t} \pmod{q}$.
- Kernel lattice $\Lambda_q^{\perp}(\mathbf{B}) = \{\mathbf{x} \in \mathbb{Z}^m \mid \mathbf{B}\mathbf{x} = \mathbf{0} \pmod{q}\}$.
- Call BDD/CVP with target point $\mathbf{y}$ in the kernel lattice. This gives $\mathbf{v}$ closest to $\mathbf{y}$.
- $\mathbf{B}\mathbf{v} - \mathbf{B}\mathbf{y} = \mathbf{B}\mathbf{e} = \mathbf{t} \pmod{q}$.

The lattice has rank m and volume q^{m-n}. Only $\mathbf{e}$ is used.

Another ISIS-like problem

Given $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \pmod{q}$, re-write as

$$\mathbf{b} = [\mathbf{A}|\mathbf{I}_m] \cdot \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix} = \mathbf{A}' \cdot \mathbf{s}' \pmod{q}.$$

- Find short $\begin{bmatrix} \mathbf{s} \\ \mathbf{e} \end{bmatrix}$ in the kernel lattice of $\mathbf{A}'$.
- The information of $\mathbf{s}$ is retained.
- The lattice has rank $m + n$ and volume q^m.
- If $\mathbf{s}$ and $\mathbf{e}$ are not balanced, re-balance the lattice (B.-Galbraith, '14).

These methods are sometimes equivalent, but not always, depending on the parameters given.

Hybrid strategies: lattice reduction + combinatoric (meet-in-the-middle) algorithms.

Hybrid attacks:

- Hoffstein, Howgrave-Graham and Silverman '07; Howgrave-Graham '07 on NTRU.
- Wunderer '16 on uSVP/BDD from LWE.
- Buchmann, Göpfert, Player, Wunderer '16 on binary error LWE.
- Albrecht '17 on binary secret LWE.
- Sometimes a better algorithm (usually when $\mathbf{s}$ or $\mathbf{e}$ are sparse).

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

- Sieving: $2^{0.292\beta+o(\beta)}$ (classic) and $2^{0.265\beta+o(\beta)}$ (quantum).

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

- Sieving: $2^{0.292\beta+o(\beta)}$ (classic) and $2^{0.265\beta+o(\beta)}$ (quantum).
  - Simulated complexity $2^{0.296\beta-18.9}$. (Albrecht-Ducas-Herold-Kirshanova-Postlethwaite-Stevens, Eurocrypt, '19)

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

- Sieving: $2^{0.292\beta+o(\beta)}$ (classic) and $2^{0.265\beta+o(\beta)}$ (quantum).
  - Simulated complexity $2^{0.296\beta-18.9}$. (Albrecht-Ducas-Herold-Kirshanova-Postlethwaite-Stevens, Eurocrypt, '19)
- Enumeration: $2^{(0.187+o(1))\beta \log \beta}$ (classic) and $2^{(\frac{0.187}{2}+o(1))\beta \log \beta}$ (quantum).

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

- Sieving: $2^{0.292\beta+o(\beta)}$ (classic) and $2^{0.265\beta+o(\beta)}$ (quantum).
  - Simulated complexity $2^{0.296\beta-18.9}$. (Albrecht-Ducas-Herold-Kirshanova-Postlethwaite-Stevens, Eurocrypt, '19)
- Enumeration: $2^{(0.187+o(1))\beta\log\beta}$ (classic) and $2^{(\frac{0.187}{2}+o(1))\beta\log\beta}$ (quantum).
- Faster Enumeration: $2^{(0.125+o(1))\beta\log\beta}$ when $n=\Omega(\beta\log\log\beta)$ (Albrecht-B.-Fouque-Kirchner-Stehle-Wen, Crypto '20).
  The idea is to preprocess in a larger region/rank than the enumeration rank β. Thus m needs to be large enough.
  - Simulated complexity $2^{0.125\beta\log\beta-0.547\beta+10.4}$.

Concrete complexity of SVP-1

Solving SVP-1 on (sub)lattices of rank-β, we can use:

- Sieving: $2^{0.292\beta+o(\beta)}$ (classic) and $2^{0.265\beta+o(\beta)}$ (quantum).
  - Simulated complexity $2^{0.296\beta-18.9}$. (Albrecht-Ducas-Herold-Kirshanova-Postlethwaite-Stevens, Eurocrypt, '19)
- Enumeration: $2^{(0.187+o(1))\beta\log\beta}$ (classic) and $2^{(\frac{0.187}{2}+o(1))\beta\log\beta}$ (quantum).
- Faster Enumeration: $2^{(0.125+o(1))\beta\log\beta}$ when $n=\Omega(\beta\log\log\beta)$ (Albrecht-B.-Fouque-Kirchner-Stehle-Wen, Crypto '20).
  The idea is to preprocess in a larger region/rank than the enumeration rank β. Thus m needs to be large enough.
  - Simulated complexity $2^{0.125\beta\log\beta-0.547\beta+10.4}$.
- Lattice reduction with approximate enumeration oracles
  - $2^{0.125\beta\log\beta-0.654\beta+25.84}$ (Albrecht-B.-Li-Rowell, Crypto '21).

  The idea is to use approximated SVP solver.

Some questions

- A more extensive study of hybrid attacks to LWE.
- Removing gaps between reduction/best-attacks in variants of LWE.
- Estimating the running-time of BKZ-β more precisely.
- Sieving v.s. enumeration. Sieving outperforms enum in theory/practice, but memory is exponential.
- Better algorithm (strategies) for BKZ-like reduction?
- How to better use the algebraic structures in lattice reduction ?


THANK
YOU