Modern Cryptography

Post-Quantum Cycles - Lattice-based Cryptography on LWE and R-LWE problems

Dr. Sara Ricci

Brno University of Technology
ricci@vut.cz

Table of contents

Table of contents

Lattice problems

Problems on lattices:

- Shortest Vector Problem (SVP)
- Closest Vector Problem (CVP)
- Shortest Independent Vectors Problem (SIVP)

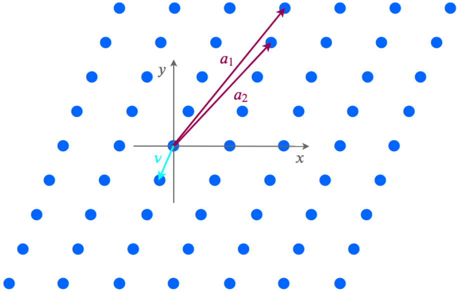
Problems combined with lattices:

- Learning With Error (LWE) problem
- Ring-Learning With Error (R-LWE) problem
- Small Integer Solution (SIS) problem

Lattice problems: SVP

Note

Given $\mathcal{B} = \{b_1, \ldots, b_n\}$ linearly indepenedent vectors, then $\mathcal{L}(\mathcal{B})$ is the set of all $t = \sum_i x_i b_i$ with $x_i \in \mathbb{Z}$.



Shortest Vector Problem (SVP)

Given a lattice $\mathcal{L}$, find the shortest non zero vector v in $\mathcal{L}$ with minimum $\|v\|_2$.

SVP and solving algorithms

Note

There are no known polynomial-time algorithms for solving the SVP

The algorithms that solves SVP

- deterministic (exactly), in $n^{O(n)}$ [Kannan '83]
- randomized (approximation), in $2^{O(n)}$ [AKS '01]

Faster algorithms are very unlikely because

- solve SVP exactly is NP-hard [Ajtai '98]

Nevertheless worst case hardness is not enough for cryptography.

Theorem (Ajtai '98)

Reduction of worst-case SVP to average-case SVP

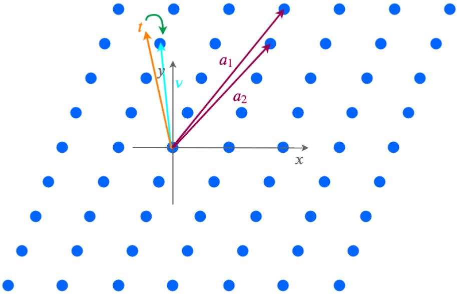
Note (informally)

Some worst-case SVP instances can be reduced to a SVP instance that can be generated with the solution (i.e., with the shortest vector).

Lattice problems: CVP

Note

Given $\mathcal{B} = \{b_1, \ldots, b_n\}$ linearly indepenedent vectors, then $\mathcal{L}(\mathcal{B})$ is the set of all $t = \sum_i x_i b_i$ with $x_i \in \mathbb{Z}$.



Closest Vector Problem (CVP)

Given a lattice $\mathcal{L}$ and a vector $t \in \mathbb{R}^n$, find $v \in \mathcal{L}$ with minimum $\|t - v\|_2$

CVP and solving algorithms

The algorithms that solves SVP

- deterministic (exactly), in $n^{O(n)}$ [Kannan '83]
- randomized (approximation), in $2^{O(n)}$ [AKS '01]

... and for CVP

- deterministic (exactly), in $n^{O(n)}$ [Micciancio,Voulgaris '10]

Note

SVP can be solved by a polynomial number of call to a CVP solver. Hence SVP hardness implies CVP hardness (not vice versa).

However, it is proven that

- solve SVP exactly is NP-hard [Ajtai '98]
- solve CVP exactly is NP-hard [van Emde Boas '81]

Beyond exact solutions

If we consider:

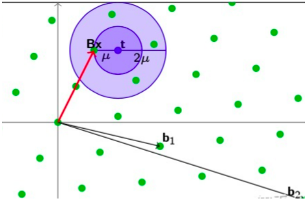
- $\text{dist}(t, \mathcal{L})$ be the minimum $\|t - v\|_2$ for $v \in \mathcal{L}$
- $\gamma > 1$ some approximation factor

The problem becomes:

Approx-CVP$_\gamma$

Given t and $\mathcal{L}$, find v' such that

$$\|t - v'\|_2 \leq \gamma \, \text{dist}(t, \mathcal{L})$$



Beyond exact solutions

If we consider:

- $\text{dist}(t, \mathcal{L})$ be the minimum $\|t - v\|_2$ for $v \in \mathcal{L}$
- $\gamma > 1$ some approximation factor

The problem becomes:

Approx-CVP$_\gamma$

Given t and $\mathcal{L}$, find v' such that

$$\|t - v'\|_2 \leq \gamma \, \text{dist}(t, \mathcal{L})$$

The algorithms that solves CVP

- deterministic (exactly), in $n^{O(n)}$ [Micciancio,Voulgaris '10]
- Approx-CVP$_\gamma$, in $2^{O(n)}$ [LLL '82, Babai '86]

Babai's Closest Vertex Algorithm

Note

If $\mathcal{B}$ is an orthogonal basis of $\mathcal{L}$, approx-CVP can be solved in polynomial time.

Note

Babai's algorithm also works with sufficiently orthogonal bases, i.e., it solves CVP and appr-CVP.

Theorem (Babai's Closest Vertex Algorithm)

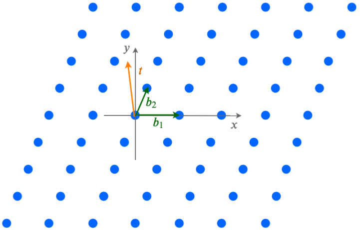
Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice with basis $\mathbf{b_1}, \ldots, \mathbf{b_n}$, and let $t \in \mathbb{R}^n$ be an arbitrary vector. If the vectors in the basis are sufficiently orthogonal to one another, then the following algorithm solves CVP.

> *Write $t = t_1\mathbf{b_1} + t_2\mathbf{b_2} + \cdots + t_n\mathbf{b_n}$ with $t_1, \ldots, t_n \in \mathbb{R}$.*
> *Set $a_i = \lceil t_i \rfloor$ for $i = 1, 2, \ldots, n$.*
> *Return the vector $v = a_1\mathbf{b_1} + a_2\mathbf{b_2} + \cdots + a_n\mathbf{b_n}$.*

Babai's Closest Vertex Algorithm: Example

Write $t = t_1\mathbf{b_1} + t_2\mathbf{b_2} + \cdots + t_n\mathbf{b_n}$ with $t_1, \ldots, t_n \in \mathbb{R}$.
Set $a_i = \lceil t_i \rfloor$ for $i = 1, 2, \ldots, n$.
Return the vector $\nu = a_1\mathbf{b_1} + a_2\mathbf{b_2} + \cdots + a_n\mathbf{b_n}$.

Find the closest vector to $t = (-1.1, 6.2)$



Let $\mathbf{b_1} = (5, 0)$ and $\mathbf{b_2} = (1, 2)$ be the basis of $\mathcal{L}$.

$$t_1 \begin{pmatrix} 5 \\ 0 \end{pmatrix} + t_2 \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} -1.1 \\ 6.2 \end{pmatrix}$$

that is

$$t_2 = 3.1, \quad t_1 = \frac{-1.1 - t_2}{5} = -0.84$$

and

$$a_2 = \lceil 3.1 \rfloor = 3, \quad a_1 = \lceil -0.84 \rfloor = -1$$

We obtain: $\nu = -\mathbf{b_1} + 3\mathbf{b_2} = (-2, 6)$

Babai's Closest Vertex Algorithm: Example

Write $t = t_1 \mathbf{b_1} + t_2 \mathbf{b_2} + \cdots + t_n \mathbf{b_n}$ with $t_1, \ldots, t_n \in \mathbb{R}$.
Set $a_i = \lceil t_i \rfloor$ for $i = 1, 2, \ldots, n$.
Return the vector $\nu = a_1 \mathbf{b_1} + a_2 \mathbf{b_2} + \cdots + a_n \mathbf{b_n}$.

Find the closest vector to $t = (-1.1, 6.2)$

Let $\mathbf{b_1} = (9, 8)$ and $\mathbf{b_2} = (8, 6)$ be the basis of $\mathcal{L}$.

$$t_1 \begin{pmatrix} 9 \\ 8 \end{pmatrix} + t_2 \begin{pmatrix} 8 \\ 6 \end{pmatrix} = \begin{pmatrix} -1.1 \\ 6.2 \end{pmatrix}$$

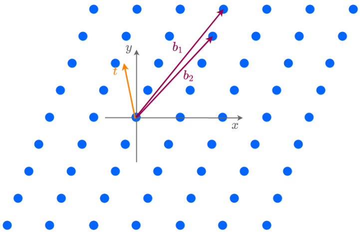
that is

$$t_2 = \frac{6.2 - 8t_1}{6}, \quad t_1 = \frac{3.3 + 4 * 6.2}{5} = 5.62$$

and

$$a_2 = \lceil -6.46 \rfloor = -7, \quad a_1 = \lceil 5.62 \rfloor = 6$$

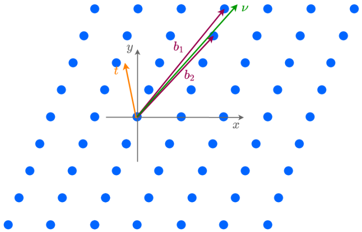
We obtain: $\nu = 6\mathbf{b_1} - 7\mathbf{b_2} = (6, 12)$



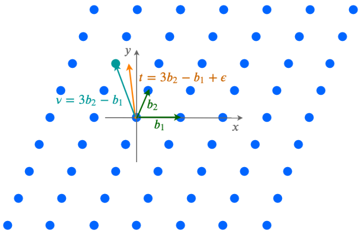
Babai's Closest Vertex Algorithm: Example

Write $t = t_1\mathbf{b_1} + t_2\mathbf{b_2} + \cdots + t_n\mathbf{b_n}$ with $t_1, \ldots, t_n \in \mathbb{R}$.
Set $a_i = \lceil t_i \rfloor$ for $i = 1, 2, \ldots, n$.
Return the vector $\nu = a_1\mathbf{b_1} + a_2\mathbf{b_2} + \cdots + a_n\mathbf{b_n}$.

Find the closest vector to $t = (-1.1, 6.2)$



If $\mathbf{b_1} = (9, 8)$ and $\mathbf{b_2} = (8, 6)$, then
$\nu = 6\mathbf{b_1} - 7\mathbf{b_2} = (6, 12)$



If $\mathbf{b_1} = (5, 0)$ and $\mathbf{b_2} = (1, 2)$, then
$\nu = -\mathbf{b_1} + 3\mathbf{b_2} = (-2, 6)$

Table of contents

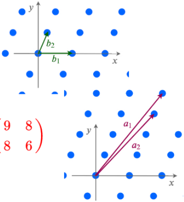
GGH public-key cryptosystem: example


Alice


Bob

Key Generation

Choose a good basis: $V = [v_1|v_2] = \begin{pmatrix} 5 & 1 \\ 0 & 2 \end{pmatrix}$



Choose U such that $\det(U) = \pm 1$

$U = \begin{pmatrix} 1 & 1 \\ 4 & 3 \end{pmatrix}$

Compute a bad basis: $W = UV = [w_1|w_2] = \begin{pmatrix} 9 & 8 \\ 8 & 6 \end{pmatrix}$

Publish the public key W

Encryption

Choose a message: $m = (x_1, x_2) = (1, -1)$

Choose a small random vector: $r = \begin{pmatrix} 1.1 \\ -0.5 \end{pmatrix}$

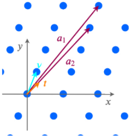
Compute: $t = x_1 w_1 + x_2 w_2 + r = \begin{pmatrix} 2.1 \\ 1.5 \end{pmatrix}$

Send t to Alice

Decryption

Use Babai's algorithm to solve CVP

$t = t_1 v_1 + t_2 v_2$
$a_1 = \lceil t_1 \rfloor = \lceil 0.27 \rfloor = 0$
$a_2 = \lceil t_2 \rfloor = \lceil 0.75 \rfloor = 1$
$\nu = a_1 v_1 + a_2 v_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$



$W^{-1} = \begin{pmatrix} -3/5 & 4/5 \\ 12/15 & -9/10 \end{pmatrix}$

$m = \nu^T W^{-1} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$

GGH public-key cryptosystem

Alice	Bob
Key Creation	
Choose a good basis $v_1, \ldots, v_n$ Choose an integer matrix U satisfying $\det(U) = \pm 1$. Compute a bad basis $w_1, \ldots, w_n$ as rows of $W = UV$. Publish the public key $w_1, \ldots, w_n$.	
Encryption	
	Choose small plaintext vector $m = (x_1, \ldots, x_n)$. Choose random small vector r. Use Alice public key to compute $t = x_1 w_1 + \cdots + x_n w_n + r$. Send the ciphertext t to Alice.
Decryption	
Use Babai's algorithm to compute the vector ν closest to t. Compute $\nu^T W^{-1}$ to recover m.	

GGH public-key cryptosystem: a bit more

Note

GGH is a probabilistic cryptosystem since a single plaintext leads to many different ciphertexts due to the choice of the noise r.

Note (Attacks)

No asymptotically good attack to GGH is known. Known attacks break the cryptosystem in practice for moderately large values of the security parameter.

Note (Security)

Therefore, it is enough to increase the security parameter n (dimension of the lattice) to make the cryptosystem secure. However, this makes the cryptosystem impractical.

so how much big n?

Here is the link to the lattice challenge, i.e. SVP computation.

Summary

GGH scheme is secure but imprатical.

Table of contents

Problems combined with lattice

	Base	Ring	Module
Random error	Learning With Errors (LWE)	Ring-Learning With Errors(RLWE)	Module Learning With Errors (MLWE)
Rounding	Learning With Rounding (LWR)	Ring-Learning With Rounding(RLWR)	Module Learning With Rounding (MLWR)

Are all equivalent?

In theory: yes, but not strictly (one may need to change the parameters in order to have the same level of security).

In practice: At the moment there does not exist any attack which exploits the additional structures.

Error or Rounding?: in “error” we add a small noise, in “rounding” we round with a modulus.

Solving a system of equations

Given a system of equations

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 \equiv 1 \mod 7 \\ s_1 + 4s_2 + 2s_3 + 6s_4 \equiv 2 \mod 7 \\ 2s_1 + s_2 + 3s_3 + s_4 \equiv 4 \mod 7 \\ 3s_1 + 4s_2 + 4s_3 + 6s_4 \equiv 0 \mod 7 \end{cases}$$

We want to find $s = (s_1, \dots, s_4) \in \mathbb{Z}_7^4$

Solving a system of equations

Given a system of equations

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 \equiv 1 \mod 7 \\ s_1 + 4s_2 + 2s_3 + 6s_4 \equiv 2 \mod 7 \\ 2s_1 + s_2 + 3s_3 + s_4 \equiv 4 \mod 7 \\ 3s_1 + 4s_2 + 4s_3 + 6s_4 \equiv 0 \mod 7 \end{cases}$$

We want to find $s = (s_1, \ldots, s_4) \in \mathbb{Z}_7^4$

Note

This problem is easy to solve via Gaussian Elimination (GE)

$$\begin{pmatrix} 1 & 5 & 3 & 2 \\ 1 & 4 & 2 & 6 \\ 2 & 1 & 3 & 1 \\ 3 & 4 & 4 & 6 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 0 \end{pmatrix} \overset{GE}{\Rightarrow} \begin{pmatrix} 1 & 5 & 3 & 2 \\ 0 & 1 & 1 & -4 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

Solution

In this case, $s = (-1, -1, 0, 0)$.

Solving a system of equations

Given a system of equations

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 \equiv 1 \mod 7 \\ s_1 + 4s_2 + 2s_3 + 6s_4 \equiv 2 \mod 7 \\ 2s_1 + s_2 + 3s_3 + s_4 \equiv 4 \mod 7 \\ 3s_1 + 4s_2 + 4s_3 + 6s_4 \equiv 0 \mod 7 \end{cases}$$

We want to find $s = (s_1, \ldots, s_4) \in \mathbb{Z}_7^4$

Note

This problem is easy to solve via Gaussian Elimination (GE)

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 = 1 \mod 7 \\ s_2 + s_3 + (-4)s_4 = -1 \mod 7 \\ s_3 + 4s_4 = 0 \mod 7 \\ 2s_4 = 0 \mod 7 \end{cases}$$

Solution

In this case, $s = (-1, -1, 0, 0)$.

Passing to LWE problem (Regev, 2005)

Given a system of equations

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 \equiv 1 \mod 7 \\ s_1 + 4s_2 + 2s_3 + 6s_4 \equiv 2 \mod 7 \\ 2s_1 + s_2 + 3s_3 + s_4 \equiv 4 \mod 7 \\ 3s_1 + 4s_2 + 4s_3 + 6s_4 \equiv 0 \mod 7 \end{cases}$$

We want to find $s = (s_1, \ldots, s_4) \in \mathbb{Z}_7^4$

Note

If we add a small error $e \in \{-1, 0, 1\}$,

$$\begin{cases} s_1 + 5s_2 + 3s_3 + 2s_4 \approx 2\ (= 1 + 1) \mod 7 \\ s_1 + 4s_2 + 2s_3 + 6s_4 \approx 2 \mod 7 \\ 2s_1 + s_2 + 3s_3 + s_4 \approx 3 \mod 7 \\ 3s_1 + 4s_2 + 4s_3 + 6s_4 \approx -1 \mod 7 \end{cases}$$

the Gaussian elimination does not work anymore, i.e., we cannot find s.

Learning with errors is hard.

LWE problem: ingredients

LWE ingredients:

- integers $n, q = poly(n), m$,
- an error probability distribution χ over $\mathbb{Z}_q$,
- m random vectors $a_i \in \mathbb{Z}_q^n$
- a secret $s \in \mathbb{Z}_q^n$ and $b_1, \ldots, b_m \in \mathbb{Z}_q$ such that

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2 \\ 13 & 14 & 14 & 6 \\ 6 & 10 & 13 & 1 \\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}}_{\mathbf{s}} + \underbrace{\begin{pmatrix} 1 \\ 0 \\ -1 \\ \vdots \\ 1 \end{pmatrix}}_{\mathbf{e}} = \underbrace{\begin{pmatrix} 8 \\ 16 \\ 3 \\ \vdots \\ 3 \end{pmatrix}}_{\mathbf{b}}$$

Is finding s even solvable?

Note

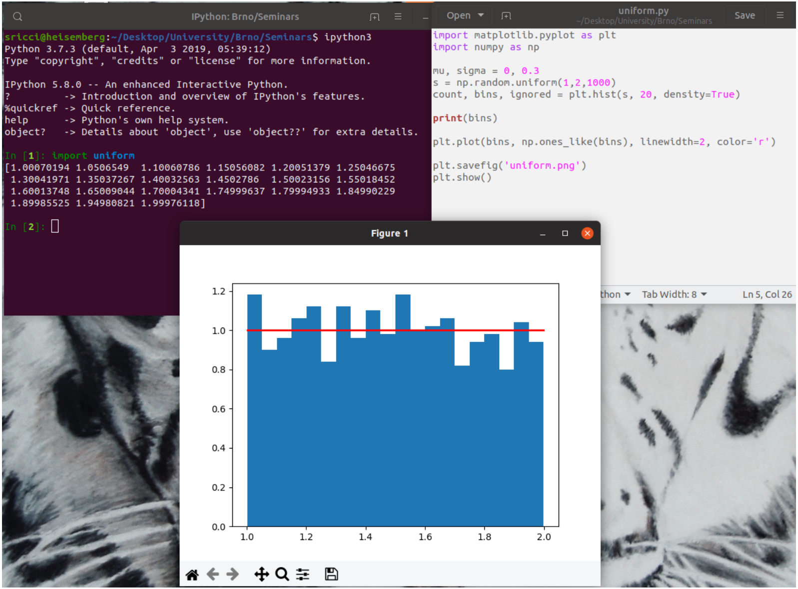
Short answer: it depends on the choice of the parameters.

- if $n = 1$ and $e \in \{-1, 0, 1\}$, i.e., χ is an uniform distribution over $\mathbb{Z}_2$, LWE can be solved (not easily) by linearization.
- if χ is uniform over $\mathbb{Z}_q$, there is no information on the related LWE problem solution.

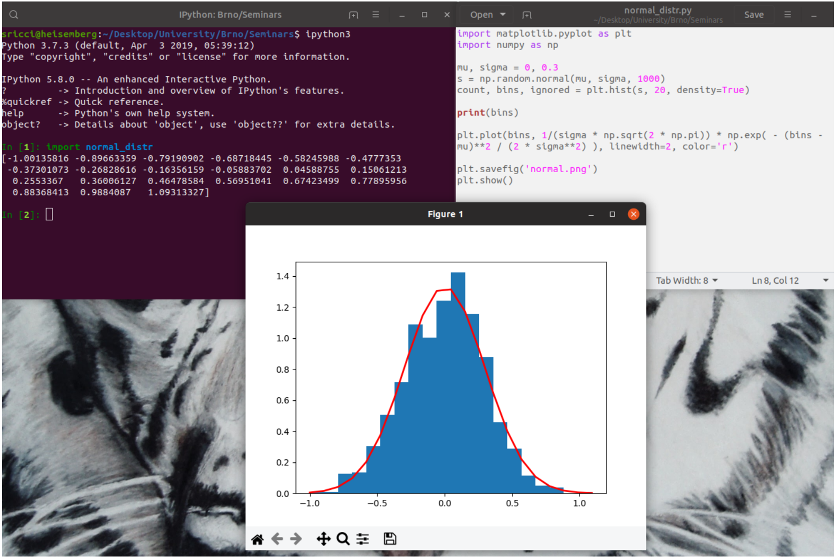
Note

It is important to choose (smartly) the error, i.e., the probability distribution χ that creates the error.

Uniform distribution: example



Normal distribution: example



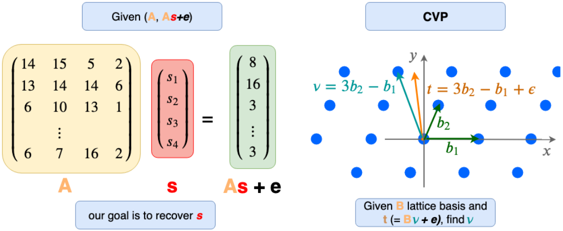
LWE problem: formulation

Problem (Computational-LWE problem)

Given $n, m, q \in \mathbb{Z}$ and χ a distribution in $\mathbb{Z}_q$ (typically "rounded" normal distribution).
Input: *a pair $(A, As + e)$, where*

- $A \in_{UR} \mathbb{Z}_q^{m \times n}$.
- *e chosen in $\mathbb{Z}_q^m$ according to χ^m.*

Goal: *For a vector $s \in_{UR} \mathbb{Z}_q^n$, given arbitrarily many samples $(A, As + e)$, compute s.*



LWE problem: formulation

Problem (Computational-LWE problem)

Given $n, m, q \in \mathbb{Z}$ and χ a distribution in $\mathbb{Z}_q$ (typically "rounded" normal distribution).
Input: *a pair $(A, As + e)$, where*

- $A \in_{UR} \mathbb{Z}_q^{m \times n}$.
- e *chosen in $\mathbb{Z}_q^m$ according to χ^m.*

Goal: *For a vector $s \in_{UR} \mathbb{Z}_q^n$, given arbitrarily many samples $(A, As + e)$, compute s.*

Note (Hardness)

Breaking the problem (or finding an efficient algorithm for LWE) implies having an efficient quantum algorithm for approximating SVP.

Theorem (Regev)

The LWE problems are as hard as worst-case assumptions in general lattices when χ is a discrete Gaussian with standard deviation $\sigma = \alpha q$ for some fixed real number $0 < \alpha < 1$.

LWE problem in matrix form

$$\begin{aligned}
14s_1 + 15s_2 + 5s_3 + 2s_4 &\approx 8 \bmod 17\\
13s_1 + 14s_2 + 14s_3 + 6s_4 &\approx 16 \bmod 17\\
6s_1 + 10s_2 + 13s_3 + 1s_4 &\approx 3 \bmod 17\\
10s_1 + 4s_2 + 12s_3 + 16s_4 &\approx 12 \bmod 17\\
&\vdots\\
6s_1 + 7s_2 + 16s_3 + 2s_4 &\approx 3 \bmod 17
\end{aligned}$$

In this case: $\mathbf{s} \in \mathbb{Z}_{17}^4$

and the equation is correct up to some small additive error (say, **±1**)

our goal is to recover ***s***

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2\\ 13 & 14 & 14 & 6\\ 6 & 10 & 13 & 1\\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s_1\\ s_2\\ s_3\\ s_4 \end{pmatrix}}_{\mathbf{s}} + \underbrace{\begin{pmatrix} 1\\ 0\\ -1\\ \vdots\\ 1 \end{pmatrix}}_{\mathbf{e}} = \underbrace{\begin{pmatrix} 8\\ 16\\ 3\\ \vdots\\ 3 \end{pmatrix}}_{\mathbf{As}+\mathbf{e}}$$

LWE problem: all matrices!

Given (A, As+e)

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2 \\ 13 & 14 & 14 & 6 \\ 6 & 10 & 13 & 1 \\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}}_{\mathbf{s}} = \underbrace{\begin{pmatrix} 8 \\ 16 \\ 3 \\ \vdots \\ 3 \end{pmatrix}}_{\mathbf{As} + \mathbf{e}}$$

our goal is to recover *s*

Given (A, As'+e)

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2 \\ 13 & 14 & 14 & 6 \\ 6 & 10 & 13 & 1 \\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s'_1 \\ s'_2 \\ s'_3 \\ s'_4 \end{pmatrix}}_{\mathbf{s'}} = \underbrace{\begin{pmatrix} 5 \\ 13 \\ 16 \\ \vdots \\ 1 \end{pmatrix}}_{\mathbf{As'} + \mathbf{e}}$$

our goal is to recover *s'*

Given (A, AS+E)

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2 \\ 13 & 14 & 14 & 6 \\ 6 & 10 & 13 & 1 \\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s_1 & s'_1 \\ s_2 & s'_2 \\ s_3 & s'_3 \\ s_4 & s'_4 \end{pmatrix}}_{\mathbf{S}} = \underbrace{\begin{pmatrix} 8 & 5 \\ 16 & 13 \\ 3 & 16 \\ \vdots & \vdots \\ 3 & 1 \end{pmatrix}}_{\mathbf{AS} + \mathbf{E}}$$

our goal is to recover *S*

When the error is “enough small”?

Simplified problem

We want to encrypt one-bit message $m \in \{0, 1\}$.
If $p \in \mathbb{Z}_{17}$ is our noise, then

$$c = p + m$$

For which value of p are we able to decrypt?

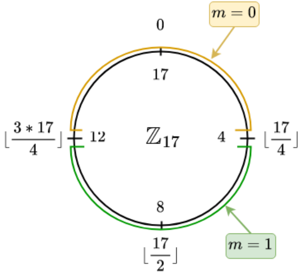
When the error is "enough small"?

Simplified problem

We want to encrypt one-bit message $m \in \{0, 1\}$.
If $p \in \mathbb{Z}_{17}$ is our noise, then

$$c = p + m$$

For which value of p are we able to decrypt?



Therefore, the encryption becomes

$$c = p + m\lfloor \frac{q}{2} \rfloor$$

and the decryption is

$$m = \begin{cases} 1 & \text{if } p \in [5, 11] \\ 0 & \text{if } p \in [-4, 3] \end{cases}$$

(Regev) one-bit encryption scheme

Public Parameter

Security parameter: n=3
Modulus : q=17
Number of eqs: 4

Key Generation

Private key: $s = \begin{pmatrix} 2 \\ 3 \\ -2 \end{pmatrix} \in_{UR} \mathbb{Z}_{17}^{3}$ $A = \begin{pmatrix} 5 & 3 & -1 \\ 6 & 1 & -13 \\ 1 & -11 & 1 \\ -7 & 4 & 0 \end{pmatrix} \in_{UR} \mathbb{Z}_{17}^{4\times 3}$

$e = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} \in_{\psi_\alpha} \mathbb{Z}_{17}^{4}$ $p = As + e = \begin{pmatrix} 21 \\ -11 \\ -33 \\ -2 \end{pmatrix} + e \bmod 17 = \begin{pmatrix} 5 \\ 5 \\ 1 \\ -1 \end{pmatrix}$

Public key: *(A,p)*

Encryption

Message: $b = 1$ is a bit $\{0,1\}$

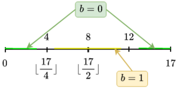
$$Enc_{A,p}(b) = (a', p') = \left(\sum_{\mathcal{I}} a_i, \sum_{\mathcal{I}} p_i + b\lfloor \frac{q}{2} \rfloor\right)$$
$$= (a_1 + a_4 = (-2, 7, -1), p_1 + p_4 = 12)$$

Ciphertext: (a', p')

Decryption

$e' = p' - a' * s^T = 12 - 2 = 10$

$Dec_s(a', p') = \begin{cases} 0 & \text{if } e' \sim 0 \\ 1 & \text{if } e' \sim \frac{17}{2} \end{cases} = 1$



(Generalized) Regev cryptosystem

Note

The one-bit scheme can be generalized to work with $m \in \mathbb{Z}_q$.

Efficiency:

- only multiplications and additions modulus q.
- parallelization.
- Public/Private key size: $(nl \log q, m(n+l) \log q)$.
- Operation encryption/decryption per bit: $(O(m(1+n/l)), O(n))$ (ignoring logarithmic factors).

Security:

- distinguishing between public keys (A, P) as generated by the cryptosystem and pairs (A, P) chosen uniformly at random implies a solution to the LWE problem.

Decryption errors:

- the error has to be "small enough".
- Using an appropriate set of parameters and en error correting code to encode m help.

Note

It is necessary to choose the parameters so that the LWE problem is hard.

LWE applications to cryptography ...

... include but are not limited to

- secret key encryption
- public key encryption
- key exchange
- digital signature
- (fully) homomorphic encryption
- identity-based encryption
- zero-knowledge proofs

Table of contents

R-LWE problem: why?

Note

Cryptographic schemes based LWE problems (and SIS) tend to require rather large key sizes ($\sim n^2$).

From a practical point of view, it will be good to reduce the key size to almost linear size.

R-LWE problem: why?

Note

Cryptographic schemes based LWE problems (and SIS) tend to require rather large key sizes ($\sim n^2$).

From a practical point of view, it will be good to reduce the key size to almost linear size.

- Need: more compactness.
- Idea: replace a random matrix with a structured "circulant" matrix with a random column

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \Rightarrow \begin{pmatrix} a_1 & a_4 & a_3 & a_2 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_2 & a_1 & a_4 \\ a_4 & a_3 & a_2 & a_1 \end{pmatrix}$$

- We need to memorize only one column.

Ideal lattice: on the other side

Note

It is possible to achieve this goal assuming that there is some structure in the LWE samples.
The structure is the Ideal lattice, i.e. the group $\mathbb{Z}_q^n$ is replaced by the ring $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$.

$(a_0, a_1, \dots, a_{n-1}) \in \mathbb{Z}_q^n \quad \Longleftrightarrow \quad a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \in \mathbb{Z}_q[x]/\langle x^n + 1\rangle$

operations modulus q

$15 \bmod 7 = 2(7) + 1 \equiv 1 \bmod 7$

each number in $\mathbb{Z}$ can be represented as a reminder/class $[0]_7, [1]_7, \dots, [6]_7$

operations modulus x^n+1 in $\mathbb{Z}_q$

$x^3 + 3x \bmod x^2 + 1 = x(x^2 + 1) + 2x \equiv 2x \bmod x^2 + 1$

each polynomial in $\mathbb{Z}_q[x]$ has a unique representation $a_0 + a_1 x + a_2 x^2 \in \mathbb{Z}_7[x]/\langle x^2 + 1\rangle$

LWE

Given (**A**, **As+e**)

$$\underbrace{\begin{pmatrix} 14 & 15 & 5 & 2 \\ 13 & 14 & 14 & 6 \\ 6 & 10 & 13 & 1 \\ & \vdots & & \\ 6 & 7 & 16 & 2 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}}_{\mathbf{s}} \approx \underbrace{\begin{pmatrix} 8 \\ 16 \\ 3 \\ \vdots \\ 3 \end{pmatrix}}_{\mathbf{As}+\mathbf{e}}$$

our goal is to recover **s**

R-LWE

Given (**A**, **As+e**)

$$\underbrace{\begin{matrix} 2 - x + 2x^2 - x^3 \\ -1 - x - x^2 \\ -x + x^2 - 3x^3 \end{matrix}}_{\mathbf{A}} \quad \underbrace{1 - x + x^3}_{\mathbf{s}} \approx \underbrace{\begin{matrix} 3 - 4x + 2x^2 - 3x^3 \\ 1 + x - x^2 \\ \dots \end{matrix}}_{\mathbf{As}+\mathbf{e}}$$

our goal is to recover **s**

Summary

Problems on lattice:

- CVP and SVP problems are NP-hard, i.e., secure against quantum attacks.
- Cryptographic protocols are based on hard problems that are easy to solve if you have a trapdoor (e.g., a good basis of a lattice)
- CVP and appr-CVP can be solved by Babai's algorithm with a lattice good basis
- GGH cryptosystem is based on the CVP problem and uses Babai's algorithm to decrypt the message.

LWE and R-LWE problems:

- solving a system of equations is easy (Gaussian elimination).
- if a small error is added, then Gaussian elimination does not work anymore
- LWE problem is NP-Hard with the right choice of parameters.
- LWE has too big key size (matrix A and secret s)
- R-LWE adds a structure to A that allows memorizing only one column of A, i.e. smaller keys.
- R-LWE problem is based on polynomials.

References

- Books:
  - Bernstein, D.J., Buchmann, J., Dahmen, E.: *Post-Quantum Cryptography*. Springer (2008)
  - Hoffstein, J., Pipher, J. C., Silverman, J. H., Silverman, J. H.: *An introduction to mathematical cryptography*. New York: springer (2008).

- Articles:
  - Regev, O.: *The learning with errors problem.* (2010).
  - Bai, .S, Galbraith, S.D.: *An improved compression technique for signatures based on learning with errors.* (2014).
  - Chen Z, Wang J, Chen L, Song X.: *A Regev-type fully homomorphic encryption scheme using modulus switching.* (2014).

Thank you for attention!

ricci@vut.cz
https://axe.utko.feec.vutbr.cz/

SCAN ME