Lattice-Related Hard Problems

Dr. Essam Ghadafi

CyBOK Mapping

The lecture maps to the following CyBOK Knowledge Areas:

- Systems Security $\rightarrow$ Cryptography
- Infrastructure Security $\rightarrow$ Applied Cryptography

Outline

- Worst-case vs Average-case Problems
  - Definition and key differences
  - Examples in cryptography
- SVP and its Variants (Shortest Vector Problem)
  - Problem description, Hardness and Variants
- CVP and its Variants (Closest Vector Problem)
  - Problem description, Hardness and Variants
- SIS (Short Integer Solution) Problem
  - Problem description, Hardness and Variants
- LWE (Learning With Errors) Problem
  - Problem description, Hardness and Variants

Worst-Case vs. Average-Case Problems

- Average-Case Problems: The attacker must succeed in solving some random instances of the problem

  Example: **Average-Case Factoring Problem:** Given a composite integer $N = p \times q$ from a distribution D_N over products of two large primes, find p and q

Worst-Case vs. Average-Case Problems

- Average-Case Problems: The attacker must succeed in solving some random instances of the problem

  Example: **Average-Case Factoring Problem:** Given a composite integer $N = p \times q$ from a distribution D_N over products of two large primes, find p and q

- Worst-Case Problems: The attacker must succeed in solving all instances of the problem

  Example: **Worst-Case Factoring Problem:** Given a composite integer $N = p \times q$, where p and q are large primes, find p and q

Worst-Case vs. Average-Case Problems

- Average-Case Problems: The attacker must succeed in solving some random instances of the problem

  Example: **Average-Case Factoring Problem:** Given a composite integer $N = p \times q$ from a distribution D_N over products of two large primes, find p and q

- Worst-Case Problems: The attacker must succeed in solving all instances of the problem

  Example: **Worst-Case Factoring Problem:** Given a composite integer $N = p \times q$, where p and q are large primes, find p and q

Some problems are hard in the worst-case but easy on average. Basing security on worst-case hardness provides stronger guarantees

Shortest Vector Problem (SVP)

Intuition: Which point in the point grid is closest to the origin point (centre of the grid)?

Shortest Vector Problem (SVP)

Intuition: Which point in the point grid is closest to the origin point (centre of the grid)?

The Shortest Vector Problem (SVP):

- Given a basis $\mathbf{B}$ for a lattice $L(\mathbf{B})$, find the shortest non-zero vector $\mathbf{v}$ in the lattice
  Mathematically:

$$||\mathbf{v}|| = \min_{\mathbf{w} \in L(\mathbf{B}) \setminus \{\mathbf{0}\}} ||\mathbf{w}||$$

equivalently

$$||\mathbf{v}|| = \lambda_1(L(\mathbf{B}))$$

Remember the 1st successive minimum λ_1 denotes the shortest non-zero vector in the lattice

Shortest Vector Problem (SVP)

How hard is the problem?

- Hard because finding the shortest vector is NP-hard

- Easy to solve when the basis vectors $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$ are *orthogonal*, i.e. $\mathbf{b}_i^T \mathbf{b}_j = 0$ for all $i, j \in \{1, \ldots, n\}$ where $i \neq j$. In this case the shortest vector is the shortest base vector $\mathbf{b}_i$

$$\|\mathbf{v}_{\mathsf{SVP}}\| = \min_{1 \leq i \leq n} \|\mathbf{b}_i\|$$

Approximate SVP (γ-SVP)

A related variant to SVP

The goal is to find a vector $\mathbf{v}$ that is at most γ times the length of the shortest vector where $\gamma \geq 1$

Approximate SVP (γ-SVP)

A related variant to SVP

The goal is to find a vector $\mathbf{v}$ that is at most γ times the length of the shortest vector where $\gamma \geq 1$

- Given a basis $\mathbf{B}$ for a lattice $L(\mathbf{B})$, find a vector $\mathbf{v} \in L(\mathbf{B}) \setminus \{\mathbf{0}\}$ where $||\mathbf{v}|| \leq \gamma \cdot \lambda_1(L(\mathbf{B}))$

  Remember the 1st successive minimum λ_1 denotes the shortest non-zero vector in the lattice

Approximate SVP (γ-SVP)

A related variant to SVP

The goal is to find a vector $\mathbf{v}$ that is at most γ times the length of the shortest vector where $\gamma \geq 1$

- Given a basis $\mathbf{B}$ for a lattice $L(\mathbf{B})$, find a vector $\mathbf{v} \in L(\mathbf{B}) \setminus \{\mathbf{0}\}$ where $||\mathbf{v}|| \leq \gamma \cdot \lambda_1(L(\mathbf{B}))$

  Remember the 1st successive minimum λ_1 denotes the shortest non-zero vector in the lattice

- The larger γ, the easier the problem. 1-SVP is SVP

SVP variants

- Short Independent Vectors Problem (SIVP) denoted by (SIVP$_\gamma$):
  Input: Basis $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$ for $L(\mathbf{B})$
  Task: Find n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in L(\mathbf{B})$ such that $\|\mathbf{v}_i\| \leq \gamma \lambda_n(L(\mathbf{B}))$ for all $i = 1, \ldots, n$

SVP variants

- Short Independent Vectors Problem (SIVP) denoted by (SIVP$_\gamma$):
  Input: Basis $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$ for $L(\mathbf{B})$
  Task: Find n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in L(\mathbf{B})$ such that $\|\mathbf{v}_i\| \leq \gamma \lambda_n(L(\mathbf{B}))$ for all $i = 1, \ldots, n$

- GAP SVP (GapSVP$_\gamma$):
  Input: Basis $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$ for $L(\mathbf{B})$
  Task: Decide whether $\lambda_1(L(\mathbf{B})) \leq 1$ or $\lambda_1(L(\mathbf{B})) > \gamma$

Closest Vector Problem (CVP)

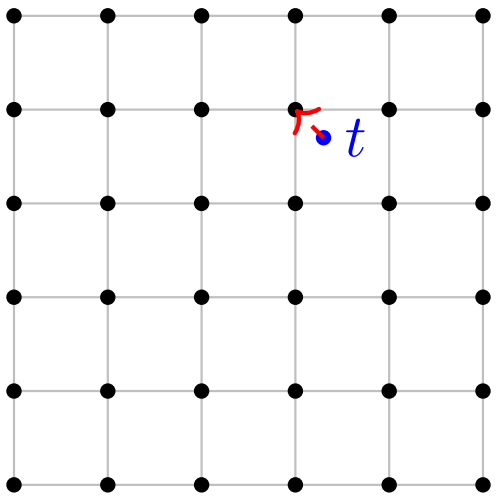
Intuition: From a location (which may not be a point) on the grid, find the nearest grid point

Closest Vector Problem (CVP)

Intuition: From a location (which may not be a point) on the grid, find the nearest grid point

Closest Vector Problem (CVP)

- Given a basis $\mathbf{B}$ for a lattice $L(\mathbf{B})$ and a target point $\mathbf{t}$, find the closest lattice vector $\mathbf{v} \in L(\mathbf{B})$
  Mathematically: $\min_{\mathbf{v} \in L(\mathbf{B})} ||\mathbf{v} - \mathbf{t}||$

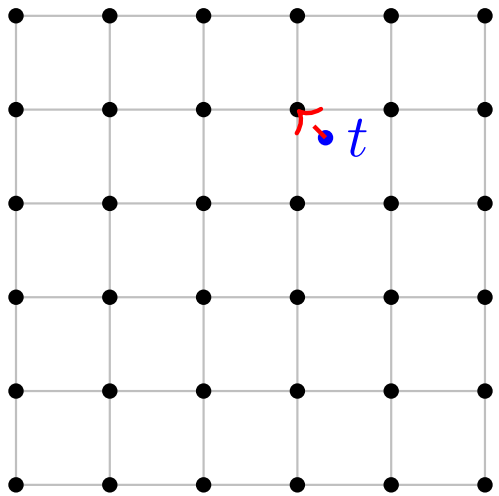


Closest Vector Problem (CVP)

Intuition: From a location (which may not be a point) on the grid, find the nearest grid point

Closest Vector Problem (CVP)

- Given a basis $\mathbf{B}$ for a lattice $L(\mathbf{B})$ and a target point $\mathbf{t}$, find the closest lattice vector $\mathbf{v} \in L(\mathbf{B})$
  Mathematically: $\min_{\mathbf{v} \in L(\mathbf{B})} ||\mathbf{v} - \mathbf{t}||$



- Harder than SVP and is NP-hard
  - SVP is a special case where $\mathbf{t} = \mathbf{0}$. If one can solve CVP, one can solve SVP. i.e. SVP $\leq$ CVP

Approximate CVP (γ-CVP)

The Approximate CVP (γ-CVP) is a variant of CVP

Instead of finding the closest vector $\mathbf{v} \in L(\mathbf{B})$ to $\mathbf{t}$, the task is instead to find $\mathbf{v} \in L(\mathbf{B})$ where the distance between $\mathbf{v}$ and $\mathbf{t}$ is at most γ times the distance between $\mathbf{t}$ and the closest vector in the lattice

Mathematically:

$$\min_{\mathbf{v} \in L(\mathbf{B})} \|\mathbf{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{v}' \in L(\mathbf{B})} \|\mathbf{v}' - \mathbf{t}\|$$

Approximate CVP (γ-CVP)

The Approximate CVP (γ-CVP) is a variant of CVP

Instead of finding the closest vector $\mathbf{v} \in L(\mathbf{B})$ to $\mathbf{t}$, the task is instead to find $\mathbf{v} \in L(\mathbf{B})$ where the distance between $\mathbf{v}$ and $\mathbf{t}$ is at most γ times the distance between $\mathbf{t}$ and the closest vector in the lattice

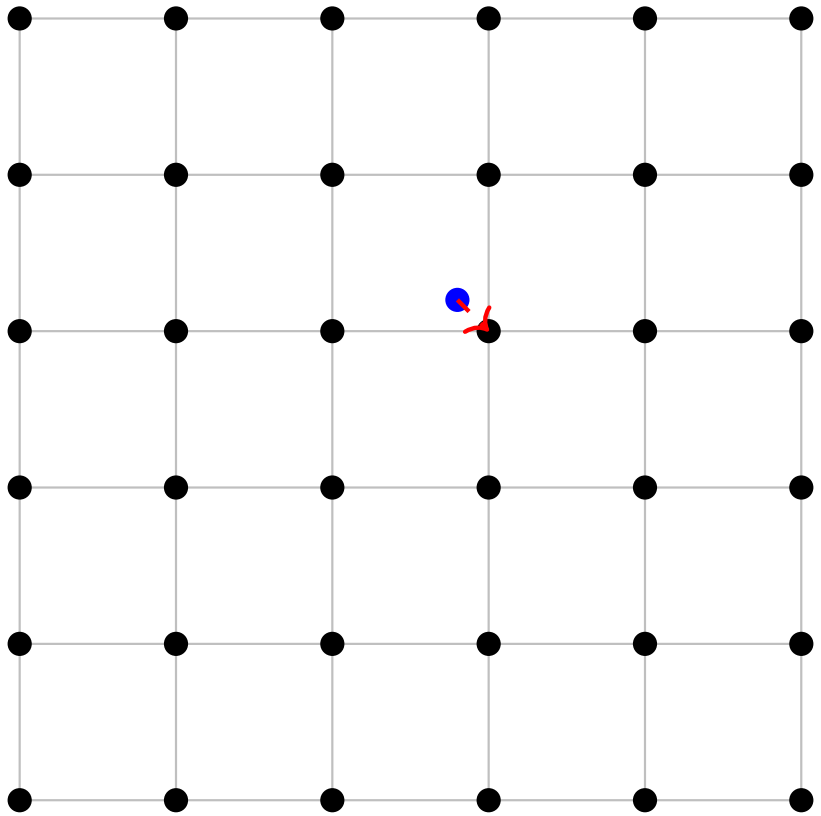
Mathematically:

$$\min_{\mathbf{v} \in L(\mathbf{B})} \|\mathbf{v} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{v}' \in L(\mathbf{B})} \|\mathbf{v}' - \mathbf{t}\|$$

- At least as hard as γ-SVP, i.e. γ-SVP $\leq \gamma$-CVP

Bounded Distance Decoding (BDD)

The Bounded Distance Decoding (BDD) problem is a special case of CVP where the target $\mathbf{t}$ is close to a lattice point

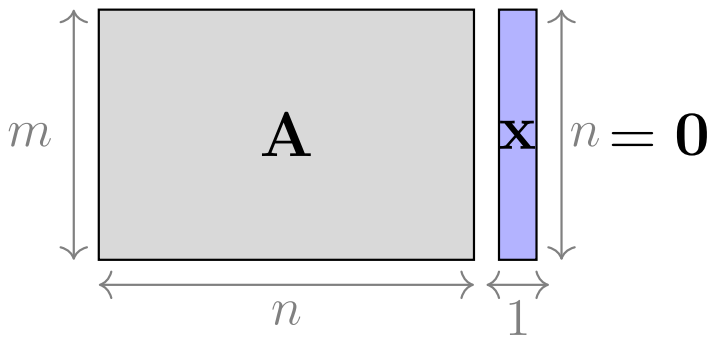


Short Integer Solution (SIS)

The SIS problem introduced by Ajtai [1] is parametrised by parameters m, n, q, γ and is denoted by $\text{SIS}_{m,n,q,\gamma}$

- Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, find a short non-zero vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{x}\| \leq \gamma$ and

$$\mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q$$

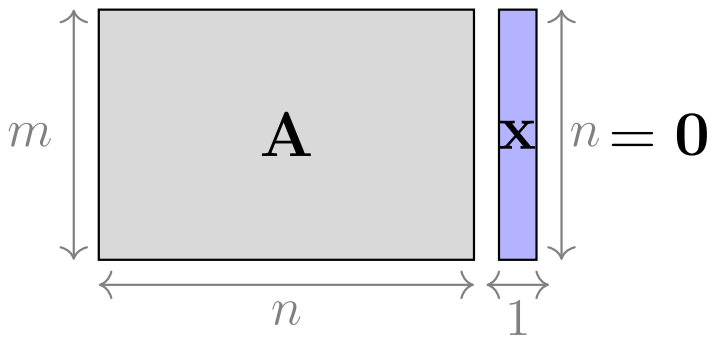


Short Integer Solution (SIS)

The SIS problem introduced by Ajtai [1] is parametrised by parameters m, n, q, γ and is denoted by $\text{SIS}_{m,n,q,\gamma}$

- Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, find a short non-zero vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{x}\| \leq \gamma$ and

$$\mathbf{A}\mathbf{x} \equiv \mathbf{0} \bmod q$$



- Generally, $\gamma < qn$, as otherwise, $\mathbf{s} = (q, 0, \ldots, 0)$ is a valid solution
- The bigger m, the harder the problem
- The problem is trivial when $n < m$

Short Integer Solution (SIS)

SIS is used in some hash functions and digital signatures

Short Integer Solution (SIS)

SIS is used in some hash functions and digital signatures

The Inhomogeneous Short Integer Solution (ISIS) problem is similar to SIS, except the right-hand side is $\mathbf{y} \in \mathbb{Z}_q^n$ instead of $\mathbf{0}$

ISIS is parametrised by m, n, q, γ and denoted by $\text{ISIS}_{m,n,q,\gamma}$

- Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a random vector $\mathbf{y} \in \mathbb{Z}_q^m$ find a short vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{x}\| \leq \gamma$ and

$$\mathbf{A}\mathbf{x} \equiv \mathbf{y} \bmod q$$

Short Integer Solution (SIS)

SIS is used in some hash functions and digital signatures

The Inhomogeneous Short Integer Solution (ISIS) problem is similar to SIS, except the right-hand side is $\mathbf{y} \in \mathbb{Z}_q^n$ instead of $\mathbf{0}$

ISIS is parametrised by m, n, q, γ and denoted by $\text{ISIS}_{m,n,q,\gamma}$

- Given a random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a random vector $\mathbf{y} \in \mathbb{Z}_q^m$ find a short vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\|\mathbf{x}\| \leq \gamma$ and

$$\mathbf{A}\mathbf{x} \equiv \mathbf{y} \bmod q$$

  - ISIS is as hard as SIS
    - SIS requires solving $\mathbf{A}\mathbf{x} = \mathbf{0}$ vs. solving $\mathbf{A}\mathbf{x} - \mathbf{y} = \mathbf{0}$ in ISIS

SIS as a Lattice Problem

SIS is a lattice problem as well

We use the matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ to define the lattice:

$$L(\mathbf{A}) = \{\mathbf{e} \in \mathbb{Z}^n : \mathbf{Ae} = \mathbf{0} \bmod q\}$$

SIS is then defined as finding a short vector $\mathbf{e}$ in this lattice

LWE – Setting the scene

Consider a system of m linear equations of the form:

$$\sum_{i=1}^{n} a_{1,i} s_i = b_1 \bmod q$$

$$\vdots$$

$$\sum_{i=1}^{n} a_{m,i} s_i = b_m \bmod q$$

where $a_{j,i} \in \mathbb{Z}_q$ are known coefficients, $b_j \in \mathbb{Z}_q$ are known results, and $s_i \in \mathbb{Z}_q$ are the unknowns we seek to solve for.

LWE – Setting the scene

Consider a system of m linear equations of the form:

$$\sum_{i=1}^{n} a_{1,i} s_i = b_1 \bmod q$$

$$\vdots$$

$$\sum_{i=1}^{n} a_{m,i} s_i = b_m \bmod q$$

where $a_{j,i} \in \mathbb{Z}_q$ are known coefficients, $b_j \in \mathbb{Z}_q$ are known results, and $s_i \in \mathbb{Z}_q$ are the unknowns we seek to solve for.

We can represent the system as

$$\mathbf{As} = \mathbf{b} \bmod q$$

LWE – Setting the scene

Consider a system of m linear equations of the form:

$$\sum_{i=1}^{n} a_{1,i} s_i = b_1 \bmod q$$

$$\vdots$$

$$\sum_{i=1}^{n} a_{m,i} s_i = b_m \bmod q$$

where $a_{j,i} \in \mathbb{Z}_q$ are known coefficients, $b_j \in \mathbb{Z}_q$ are known results, and $s_i \in \mathbb{Z}_q$ are the unknowns we seek to solve for.

We can represent the system as

$$\mathbf{As} = \mathbf{b} \bmod q$$

This system of equations is straightforward to solve (in poly time), e.g., using Gaussian elimination

Learning with Errors (LWE)

LWE introduced by Regev [6] adds small noise to linear equations, making them hard to solve even for quantum computers

LWE and its variants have been used to construct various types of cryptosystems, including:

- Public-Key Encryption
- Key Exchange
- Identity-Based Encryption
- Zero-Knowledge Proofs
- ...

Learning With Errors (LWE)

Definition: Let m, n, q be positive integers, and let χ be a probability distribution over $\mathbb{Z}_q$

The (search) LWE problem denoted by ($\text{LWE}_{m,n,q,\chi}$) is:

- Randomly choose a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$
- Choose a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Choose a random error/noise vector $\mathbf{e}$ according to χ^m
  - i.e., each coordinate e_i is independently drawn from χ

Learning With Errors (LWE)

Definition: Let m, n, q be positive integers, and let χ be a probability distribution over $\mathbb{Z}_q$

The (search) LWE problem denoted by ($\text{LWE}_{m,n,q,\chi}$) is:

- Randomly choose a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$
- Choose a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$
- Choose a random error/noise vector $\mathbf{e}$ according to χ^m
  - i.e., each coordinate e_i is independently drawn from χ

Input: $(\mathbf{A}, \mathbf{b})$ where $\mathbf{As} + \mathbf{e} \equiv \mathbf{b} \mod q$ where $\mathbf{b} \in \mathbb{Z}_q^m$
Task: Recover $\mathbf{s}$

Decisional LWE

The (decision) LWE problem denoted by (D-LWE$_{m,n,q,\chi}$) is:

- $\mathbf{A}$, $\mathbf{s}$, and $\mathbf{e}$ are all chosen as in the search LWE problem
- **Challenge Generation:**
  - **Case 0** (real): Compute $\mathbf{b}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$
  - **Case 1** (fake): Choose $\mathbf{b}_1 \in \mathbb{Z}_q^m$ uniformly at random

Input: $(\mathbf{A}, \mathbf{b}_b)$ where $b \in \{0, 1\}$ is chosen uniformly at random
Task: Guess the case b

Decisional LWE

The (decision) LWE problem denoted by (D-LWE$_{m,n,q,\chi}$) is:

- $\mathbf{A}$, $\mathbf{s}$, and $\mathbf{e}$ are all chosen as in the search LWE problem
- **Challenge Generation:**
  - **Case 0** (real): Compute $\mathbf{b}_0 = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$
  - **Case 1** (fake): Choose $\mathbf{b}_1 \in \mathbb{Z}_q^m$ uniformly at random

Input: $(\mathbf{A}, \mathbf{b}_b)$ where $b \in \{0, 1\}$ is chosen uniformly at random
Task: Guess the case b

Interestingly, LWE can be reduced to D-LWE

- An algorithm solving D-LWE can also solve LWE. The reverse implication is trivial
  - This means equivalence of the two variants of LWE

Hardness of LWE

- If one can solve $\mathsf{SIS}_{m,n,q,\gamma}$, one can solve $\mathsf{LWE}_{m,n,q,\chi}$, i.e. $\mathsf{LWE}_{m,n,q,\chi} \leq \mathsf{SIS}_{m,n,q,\gamma}$

Hardness of LWE

- If one can solve $\mathsf{SIS}_{m,n,q,\gamma}$, one can solve $\mathsf{LWE}_{m,n,q,\chi}$, i.e. $\mathsf{LWE}_{m,n,q,\chi} \leq \mathsf{SIS}_{m,n,q,\gamma}$
- If one can solve $\mathsf{LWE}_{m,n,q,\chi}$, one can solve GapSVP_{γ}
  - e.g. [6, 2]: $\mathsf{GapSVP}_{\gamma}, \mathsf{SIVP}_{\gamma} \leq \mathsf{LWE}_{m,n,q,\chi}$

    Note: In [6] the reduction is *quantum*, while in [2] it is *classical* but only for *polynomial modulus* q, with some losses in dimension and noise

Ring-LWE – Setting the Scene

Ring-LWE (R-LWE) [4] is a LWE variant adapted to polynomial rings (instead of integers) to improve efficiency

Remember: A polynomial is of the form $\sum_{i=0}^{d} c_i x^i$, where:

- x is the indeterminate, d is the degree of the polynomial (i.e. the highest power of x), and $c_i \in \mathbb{Z}$ are the coefficients

Polynomials can be represented by their coefficient vector $\mathbf{c} = (c_0, c_1, \ldots, c_d)$

Ring-LWE – Setting the Scene

- Polynomial Ring R: Let $R = \mathbb{Z}[x]/f(x)$ for some monic polynomial $f(x)$ of degree $d = n$, (e.g. $f(x) = x^n + 1$)
  - Elements of R are polynomials of degree $< n$ with integer coefficients
    - R is a set of polynomials reduced modulo $f(x)$

Ring-LWE – Setting the Scene

- Polynomial Ring R: Let $R = \mathbb{Z}[x]/f(x)$ for some monic polynomial $f(x)$ of degree $d = n$, (e.g. $f(x) = x^n + 1$)
  - Elements of R are polynomials of degree $< n$ with integer coefficients
    - R is a set of polynomials reduced modulo $f(x)$
- Polynomial Ring R_q: Let $R_q = \mathbb{Z}_q[x]/f(x)$
  - Elements of R_q are polynomials of degree $< n$ with coefficients in in $\mathbb{Z}_q$

Ring-LWE

Definition: Given polynomial rings R and R_q, and an error distribution χ over small elements of ring R

The (Search) Ring-LWE (R-LWE) problem, denoted by ($\text{R-LWE}_{q,n,\chi}$), is defined as follows:

- Sample polynomials $a_1, \ldots, a_n \in R_q$ independently and uniformly at random. Let $\mathbf{a} = (a_1, \ldots, a_n)$
- Sample error polynomials $e_1, \ldots, e_n$ independently from χ. Let $\mathbf{e} = (e_1, \ldots, e_n)$

Input: $(\mathbf{a}, \mathbf{b} = s\ \mathbf{a} + \mathbf{e} \bmod qR)$, for some fixed polynomial $s \in R_q$
Task: Recover s

Ring-LWE

Definition: Given polynomial rings R and R_q, and an error distribution χ over small elements of ring R

The (Search) Ring-LWE (R-LWE) problem, denoted by ($\text{R-LWE}_{q,n,\chi}$), is defined as follows:

- Sample polynomials $a_1, \ldots, a_n \in R_q$ independently and uniformly at random. Let $\mathbf{a} = (a_1, \ldots, a_n)$
- Sample error polynomials $e_1, \ldots, e_n$ independently from χ. Let $\mathbf{e} = (e_1, \ldots, e_n)$

Input: $(\mathbf{a}, \mathbf{b} = s\ \mathbf{a} + \mathbf{e} \bmod qR)$, for some fixed polynomial $s \in R_q$
Task: Recover s

Note: $\bmod\ qR$ means reducing polynomial coefficients modulo q, where qR is the ideal in R generated by q

Decisional Ring-LWE

The Decisional Ring-LWE (D-R-LWE) problem denoted by (D-R-LWE$_{q,n,\chi}$) can be defined similarly where the task is to distinguish $(\mathbf{a}, \mathbf{b})$, where

$$\mathbf{b} = s\mathbf{a} + \mathbf{e} \mod qR$$

from a uniform tuple $(\mathbf{a}, \mathbf{b}) \in R_q^n \times R_q^n$

Hardness of Ring-LWE

Lyubashevsky, et al. [4, 5] gave the following results

$$\text{Ideal-SIVP}_{\gamma}(\text{worst case}) \leq_{\text{quantum polytime}} \text{D-R-LWE}_{q,n,\chi}$$

- If you can solve D-R-LWE, there is a **quantum algorithm** that can solve Ideal-SIVP$_{\gamma}$(worst case)

Hardness of Ring-LWE

Lyubashevsky, et al. [4, 5] gave the following results

$$\text{Ideal-SIVP}_{\gamma}(\text{worst case}) \leq_{\text{quantum polytime}} \text{D-R-LWE}_{q,n,\chi}$$

- If you can solve D-R-LWE, there is a **quantum algorithm** that can solve Ideal-SIVP$_{\gamma}$(worst case)

$$\text{R-LWE}_{q,n,\chi} \leq \text{D-R-LWE}_{q,n,\chi}$$

- If you can solve Decisional R-LWE, there is a **classical algorithm** that can solve Search R-LWE

Ring-LWE vs. LWE

- R-LWE yield more efficient constructions
  - Multiplication in the polynomial ring is more efficient, e.g. using FFT-like techniques
  - Smaller element representations (Elements from R_q vs. elements from $\mathbb{Z}$)
    - Storage cost in standard LWE: $O(m \cdot n)$ integers (typically $O(n^2)$ since $m = O(n)$)
    - Storage cost in Ring-LWE: $O(n)$ integers

- Like LWE's relation to SIS, Ring-LWE retains worst-case hardness due to its close connection to Ring-SIS

Ring-LWE vs. LWE

- R-LWE yield more efficient constructions
  - Multiplication in the polynomial ring is more efficient, e.g. using FFT-like techniques
  - Smaller element representations (Elements from R_q vs. elements from $\mathbb{Z}$)
    - Storage cost in standard LWE: $O(m \cdot n)$ integers (typically $O(n^2)$ since $m = O(n)$)
    - Storage cost in Ring-LWE: $O(n)$ integers

- Like LWE's relation to SIS, Ring-LWE retains worst-case hardness due to its close connection to Ring-SIS

Note: R-LWE is defined over **structured lattices** (ideal lattices with algebraic structure), whereas LWE works over **general, unstructured lattices**

- R-LWE assumes hardness for a special class of lattices

Module-LWE — Intuition

Intuition: M-LWE is a bridge between (unstructured) LWE and structured Ring-LWE

- LWE: Elements are simple vectors over $\mathbb{Z}_q^n$ (unstructured)
- Ring-LWE: Elements are polynomials from a ring R_q (highly structured)

Module-LWE — Intuition

Intuition: M-LWE is a bridge between (unstructured) LWE and structured Ring-LWE

- LWE: Elements are simple vectors over $\mathbb{Z}_q^n$ (unstructured)
- Ring-LWE: Elements are polynomials from a ring R_q (highly structured)
- M-LWE: Elements are vectors of polynomials in R_q^k, more structured than LWE but less structured than Ring-LWE
  - Interpolates between LWE and Ring-LWE

Module-LWE — Intuition

Intuition: M-LWE is a bridge between (unstructured) LWE and structured Ring-LWE

- LWE: Elements are simple vectors over $\mathbb{Z}_q^n$ (unstructured)
- Ring-LWE: Elements are polynomials from a ring R_q (highly structured)
- M-LWE: Elements are vectors of polynomials in R_q^k, more structured than LWE but less structured than Ring-LWE
  - Interpolates between LWE and Ring-LWE

  Ring-LWE is a special case of Module-LWE where $k = 1$

Module-LWE – Definition

Similarly to the setup for Ring-LWE, given

$$R = \mathbb{Z}[x]/f(x), \quad R_q = R/qR,$$

a module rank k, and an error distribution χ over small elements of R

Module-LWE – Definition

Similarly to the setup for Ring-LWE, given

$$R = \mathbb{Z}[x]/f(x), \quad R_q = R/qR,$$

a module rank k, and an error distribution χ over small elements of R

The (Search) Module-LWE (M-LWE) problem, denoted by $(\text{M-LWE}_{q,n,k,\chi})$, is:

- Sample elements $\mathbf{a}_1, \ldots, \mathbf{a}_n \in R_q^k$ independently and uniformly at random. Let $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in R_q^{n \times k}$.
- Independently sample error elements $e_1, \ldots, e_n \in R_q$ from χ. Let $\mathbf{e} = (e_1, \ldots, e_n) \in R_q^n$.

Module-LWE – Definition

Similarly to the setup for Ring-LWE, given

$$R = \mathbb{Z}[x]/f(x), \quad R_q = R/qR,$$

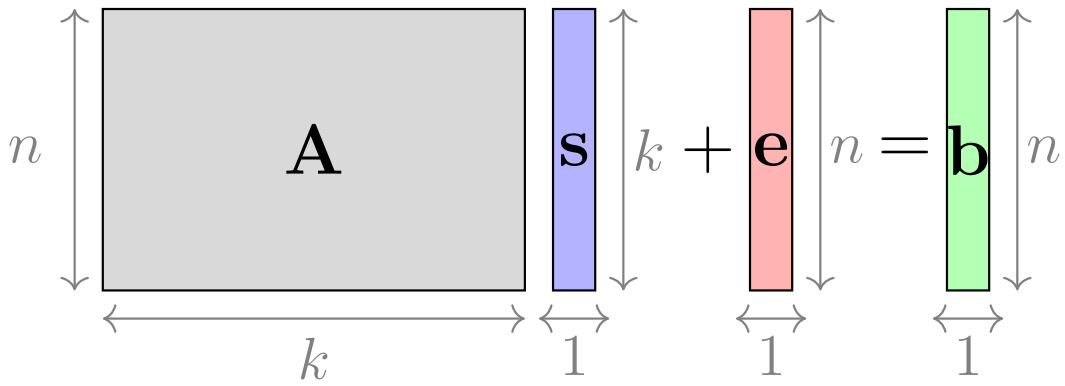
a module rank k, and an error distribution χ over small elements of R

The (Search) Module-LWE (M-LWE) problem, denoted by $(\text{M-LWE}_{q,n,k,\chi})$, is:

- Sample elements $\mathbf{a}_1, \ldots, \mathbf{a}_n \in R_q^k$ independently and uniformly at random. Let $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_n) \in R_q^{n \times k}$.
- Independently sample error elements $e_1, \ldots, e_n \in R_q$ from χ. Let $\mathbf{e} = (e_1, \ldots, e_n) \in R_q^n$.

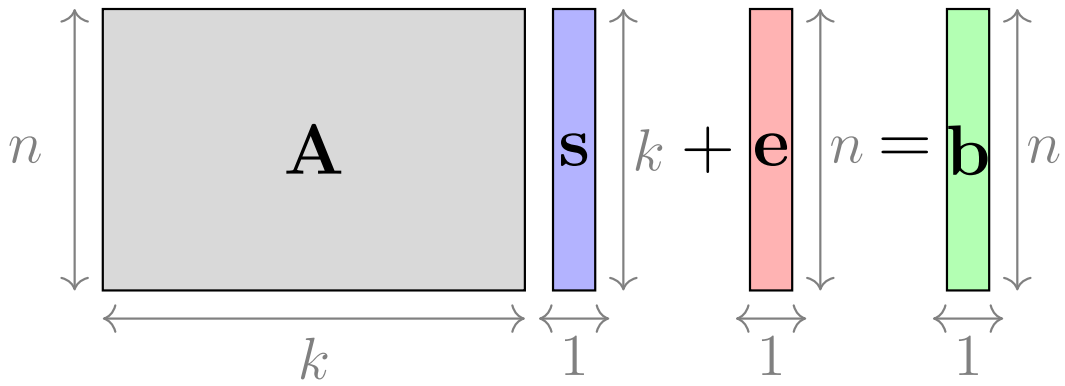
Input: $(\mathbf{A}, \mathbf{b} = \mathbf{As} + \mathbf{e}) \in R_q^{n \times k} \times R_q^n$, for some secret vector of polynomials $\mathbf{s} \in R_q^k$
Task: Recover $\mathbf{s}$

Module-LWE – A Closer Look



Note: : Some formulations explicitly sample $\mathbf{s}$ from a distribution χ_s (e.g. discrete Gaussian or binomial); others assume $\mathbf{s}$ is uniform in R_q^k

Module-LWE – A Closer Look



Note: : Some formulations explicitly sample $\mathbf{s}$ from a distribution χ_s (e.g. discrete Gaussian or binomial); others assume $\mathbf{s}$ is uniform in R_q^k

Comparison with Ring-LWE:

- Ring-LWE is a special case of M-LWE when $k = 1$

Hardness of Module-LWE

Langlois & Stehlé [3] showed that

$$\text{Module-SIVP}_{\gamma} \leq_{\text{quantum polytime}} \text{M-LWE}_{q,n,k,\chi}$$

- If $\text{M-LWE}_{q,n,k,\chi}$ can be solved efficiently, then worst-case $\text{Module-SIVP}_{\gamma}$ can also be solved efficiently (by a quantum polynomial-time algorithm)

Efficiency of Module-LWE

Storage cost in Module-LWE is $O(n \cdot k)$

Efficiency of Module-LWE

Storage cost in Module-LWE is $O(n \cdot k)$

NIST-standard CRYSTALS-Kyber (key encapsulation mechanism) and CRYSTALS-Dilithium (signature) both rely on Module-LWE

Efficiency of Module-LWE

Storage cost in Module-LWE is $O(n \cdot k)$

NIST-standard CRYSTALS-Kyber (key encapsulation mechanism) and CRYSTALS-Dilithium (signature) both rely on Module-LWE

Example:
In CRYSTALS-Kyber 512, the M-LWE parameters used are:
$n = k = 2$, $f(x) = X^{256} + 1$, and $q = 3329$

Decisional Module-LWE – Definition

The setup is similar to search M-LWE

Decisional Module-LWE – Definition

The setup is similar to search M-LWE

Decisional Module-LWE (D-M-LWE): The problem ($\text{D-M-LWE}_{q,n,k,\chi}$) is to **distinguish** between:

$$(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}),$$

for a secret $\mathbf{s} \in R_q^k$, an error vector $\mathbf{e} \in R_q^n$ with entries independently sampled from χ, and $\mathbf{A}$ is a uniformly random matrix over $R_q^{n \times k}$; and

$$(\mathbf{A}, \mathbf{b}) \in R_q^{n \times k} \times R_q^n$$

sampled uniformly at random

Main Takeaways

- Worst-Case vs. Average-Case Problems:
  - Worst-case: Focus on the hardest instances (e.g., SVP)
  - Average-case: Focus on typical instances (e.g., SIS)
- Shortest Vector Problem (SVP):
  - Finding the shortest non-zero vector in a lattice
- Closest Vector Problem (CVP):
  - Finding the closest lattice vector to a given point
- Short Integer Solution(SIS):
  - Finding short integer solutions to modular equations
- Learning With Errors (LWE):
  - Recovering a secret vector from noisy linear equations
- Ring Learning With Errors (R-LWE): A variant of LWE defined over polynomial rings
  - Recovering a secret polynomial from noisy polynomial equations
- Module Learning With Errors (M-LWE): A generalization of LWE and R-LWE that balances efficiency and security across modules

Additional Resources & Reading

- D. Micciancio. Efficient reductions among lattice problems. In ACM-SIAM symposium on Discrete algorithms (SODA), 2008.

References

[1] M. Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing (STOC)*, 1996.

[2] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, D. Stehlé. Classical hardness of learning with errors. In *ACM symposium on Theory of Computing (STOC)*, 2013.

[3] A. Langlois, D. Stehlé. Worst-Case to Average-Case Reductions for Module Lattices, In *Designs, Codes and Cryptography*, 2015.

[4] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. In *Eurocrypt*, 2010.

[5] V. Lyubashevsky, C. Peikert, O. Regev. On ideal lattices and learning with errors over rings. In *J. ACM*, 2013.

[6] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *J. ACM 56, 6*, 2009.