FIT5124 Advanced Topics in Security

Lecture 1: Lattice-Based Crypto. I

Ron Steinfeld
Clayton School of IT
Monash University

March 2016

Acknowledgements: Some figures sourced from Oded Regev's Lecture Notes on 'Lattices in Computer Science', Tel Aviv University, Fall 2004, and Vinod Vaikuntanathan's course on Lattices in Computer Science, MIT.

First Module In a Nutshell

Lattice-Based Cryptography is a cutting-edge cryptographic 'technology'. Has several interesting properties:

- Very fast Public-Key Cryptographic Operations (useful for performance-critical applications).
- Provable Security Guarantees
- Believed 'Post Quantum Computer' Security
- Allows more powerful cryptographic functionalities (in some cases not previously possible), e.g.
  - Fully Homomorphic Encryption (FHE): communication-efficient privacy-preserving computation protocols (later in unit!)

This Lecture: Brief introduction to lattices, hard computational problems, and some related mathematics (more to be introduced gradually in following lectures).

Lecture Outline

Lecture Outline: Motivation and Intro. to Lattice-Based Cryptography

- Lattice-Based Crypto: Brief History
- Lattices: Concepts and intro. to the mathematics
- Lattices: Hard Computational Problems – SVP
- Random Crypto. Lattices: SIS Problem
- SIS Application: Collision-Resistant Hash Function

Following Lectures:

- Cryptanalysis: How Secure is lattice-based crypto? How to choose parameters?
- How to use Lattice-based crypto to build encryption and signature schemes?
- How to make lattice-based crypto. efficient?

Motivation: Why study Lattice-Based Crypto?

Lattice-Based Cryptography has several interesting properties:

- Computational Efficiency: High-speed crypto algorithms
- Novel and Powerful Cryptographic Functionalities (e.g. Fully Homomorphic Encryption – FHE)
- Strong Provable Security Guarantees
- Believed Post Quantum Security

Motivation: Post Quantum World

Today:

- Public-key crypto is essential for secure web transactions.
- Deployed public-key cryptosystems based on Factorization or Discrete-Logarithm problems.

But:

- Shor (1994) showed Fact/DL solvable efficiently on large scale **quantum computer**.
- Quantum computer technology is currently primitive ($15 = 3 \times 5$), but for how long?

Lattice-based crypto seems to resist quantum attacks!

Motivation: Efficiency

Popular cryptosystems are relatively inefficient;
For security level 2^n:

- RSA – key length $\widetilde{O}(n^3)$, computation $\widetilde{O}(n^6)$.
- ECC – key length $\widetilde{O}(n)$, computation $\widetilde{O}(n^2)$.

Structured ('Ring based') Lattices – key length and computation $\widetilde{O}(n)$ **asymptotically, as** n **grows towards infinity**.

In Practice, for typical security parameter $n \approx 100$, with best current schemes, typically have:

- Structured Lattice crypto. **Computation** ≈ 100 times faster than RSA
- Structured Lattice crypto. **ciphertext/key length** $\approx$ RSA key/ciphertext length

Motivation: Provable Security Guarantees

Brief History of Lattice-Based Crypto

- 1978: Knapsack public-key cryptosystem (Merkle-Hellman).
  - Trapdoor One-way Function: $f(x_1, \ldots, x_n) = \sum_{i \leq n} g_i \cdot x_i$.
  - Public: persumably hard knapsack set $(g_1, \ldots, g_n)$.
  - Secret Trapdoor: easy knapsack $(g'_1, \ldots, g'_n)$, $g'_i > 2 \cdot g'_{i-1}$.
  - Public-Secret Relation: $g_i = a \cdot g'_i \bmod q$, $i = 1, \ldots, n$.
- 1982: Poly-time secret recovery attack (Shamir).
- 1980s:

```
for(i = 1; i < N; i++) {
  repair;
  attack;
}
```

Problem with Heuristic Designs: Special random instances – shortcut attacks can exist!

Motivation: Provable Security Guarantees

- 1996: One-Way Func./Encryption with worst case to average case security proof (Ajtai/Ajtai-Dwork) – Introduction of SIS problem.
  - Proof that no shortcut attacks exist – any attack implies solving hard worst-case instances of lattice problems!
- 1996: Efficient ($\widetilde{O}(n)$ time/space) and Practical but heuristic security NTRU encryption (Hoffstein et al) – ideal lattices.
- 2002: Efficient lattice-based one-way function with security proof – ideal lattices (Micciancio).
- 2005: Lattice-Based public-key encryption with security proof – Introduction of LWE Problem (Regev).
- 2005-2015: Many Developments, e.g.
  - Improved Techniques/Proofs (Fourier analysis, Gaussians), Crypto. Hash Functions, Trapdoor signatures, ID-Based Encryption (IBE), Attribute-Based Encryption (ABE), Zero-Knowledge Proofs, Oblivious Transfer, Fully-Homomorphic Encryption (FHE), Cryptographic Multilinear Maps, Program Obfuscation,...

Lattices: Basic Concepts

Point lattices: an area of math. combinining matrix/vector algebra (linear algebra) and integer variables. Both geometry ad algebra play a role.

Before we begin: **Notations**

$\mathbb{Z}$: Set of integers, : $\mathbb{R}$: Set of real numbers $\mathbb{Z}_q$: Ring of integers modulo q

vectors – by default columns: $\vec{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$, with coordinates b_i, $i = 1, \ldots, n$. Convert to a row vector using transpose: $\vec{b}^T = [b_1 b_2 \cdots b_n]$.

Measures of length (aka norm) for vectors:

- Euclidean norm (aka 'length', '2-norm'): $\|\vec{b}\| = \sqrt{\sum_{i=1}^{n} b_i^2}$.
- Infinity norm (aka 'max' norm): $\|\vec{b}\|_\infty = \max_i |b_i|$.

Lattices: Basic Concepts

Definition

An n-dimensional (full-rank) lattice $L(B)$ is the set of all integer linear combinations of some basis set of linearly independent vectors $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{R}^n$:

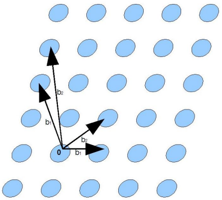
$$L(B) = \{c_1 \cdot \vec{b}_1 + c_2 \cdot \vec{b}_2 + \cdots + c_n \cdot \vec{b}_n : c_i \in \mathbb{Z}, i = 1, \ldots, n\}.$$

- Call $n \times n$ matrix $B = (\vec{b}_1, \ldots, \vec{b}_n)$ a basis for $L(B)$.

Example in 2 Dimensions ($n = 2$)

$$\vec{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{b}_2 = \begin{bmatrix} 1.2 \\ 1 \end{bmatrix},$$

$$\vec{b}'_1 = \begin{bmatrix} -0.6 \\ 2 \end{bmatrix}, \vec{b}'_2 = \begin{bmatrix} -0.4 \\ 3 \end{bmatrix}$$



Lattices: Basic Concepts

Definition

An n-dimensional (full-rank) lattice $L(B)$ is the set of all integer linear combinations of some basis set of linearly independent vectors $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{R}^n$:

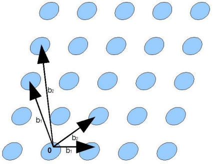
$$L(B) = \{c_1 \cdot \vec{b}_1 + c_2 \cdot \vec{b}_2 + \cdots + c_n \cdot \vec{b}_n : c_i \in \mathbb{Z}, i = 1, \ldots, n\}.$$

- Call $n \times n$ matrix $B = (\vec{b}_1, \ldots, \vec{b}_n)$ a basis for $L(B)$.
- L is discrete group in $\mathbb{R}^n$, under addition.

Example in 2 Dimensions ($n = 2$)

$$\vec{b}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \vec{b}_2 = \begin{bmatrix} 1.2 \\ 1 \end{bmatrix},$$

$$\vec{b}'_1 = \begin{bmatrix} -0.6 \\ 2 \end{bmatrix}, \vec{b}'_2 = \begin{bmatrix} -0.4 \\ 3 \end{bmatrix}$$



Lattices: Basic Concepts

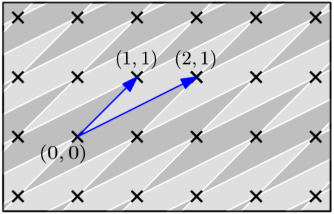
Definition

For an n-dim. lattice basis $B = (\vec{b}_1, \ldots, \vec{b}_n) \in \mathbb{R}^{n \times n}$, the fundamental paralellepiped (FP) of B, denoted $P(B)$, is the set of all real-valued $[0, 1)$-linear combinations of some basis set of linearly independent vectors $\vec{b}_1, \ldots, \vec{b}_n \in \mathbb{R}^n$:

$$P(B) = \{c_1 \cdot \vec{b}_1 + c_2 \cdot \vec{b}_2 + \cdots + c_n \cdot \vec{b}_n : 0 \leq c_i < 1, i = 1, \ldots, n\}.$$

- The translated FPs (in grey in example below) tile the whole n-dim. real vector space $\mathrm{span}(B) = \mathbb{R}^n$ spanned by B.

Example in 2 Dimensions ($n = 2$)



Lattices: Basic Concepts

- There are (infinitely!) many different bases for a lattice.
- Question: Given a lattice L with basis B, how can we tell if B' is another basis for L?
- Geometric Ans.: count L points contained in $P(B')$

Lemma

There is exactly one L point contained in $P(B')$ (the $\vec{0}$ vector) if and only if B' is a basis of L.

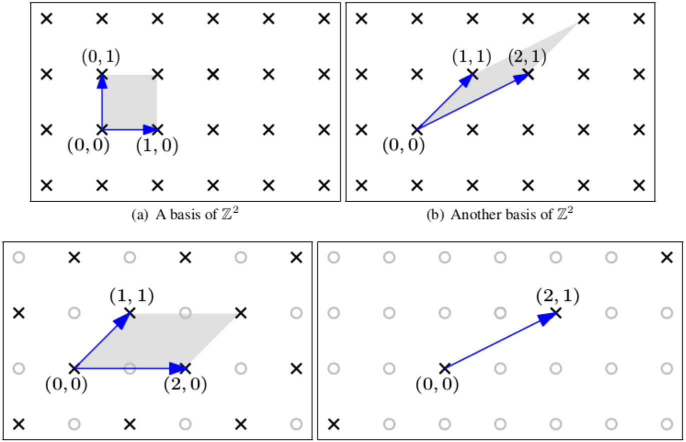
- Algebraic Ans.: Look at determinant of the matrix relating B' to B

Lemma

B' is a basis of $L(B)$ if and only if $B' = B \cdot U$ for some $n \times n$ integer matrix U with $\det(U) = \pm 1$ (we call such a U a unimodular matrix).

Lattices: Basic Concepts

Multiple Bases / FP Examples in 2 dim.



(a) A basis of $\mathbb{Z}^2$

(b) Another basis of $\mathbb{Z}^2$

(c) Not a basis of $\mathbb{Z}^2$

(d) Not a full-rank lattice

Lattices: Basic Concepts

Definition

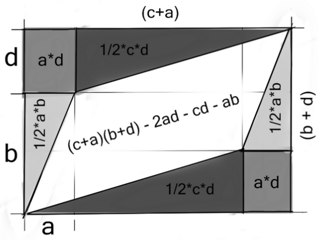
For an n-dim. lattice $L(B)$, the determinant of $L(B)$, denoted $\det L(B)$ is the n-dim. volume of the FP $P(B)$.

Lemma (Equivalent algebraic def. of lattice determinant)

For an n-dim. lattice $L(B)$, we have $\det(L(B)) = |\det(B)|$.

Example of algebraic-geometric relation in 2-dim.:

$$B = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$



- Consequence: For a large n-dim ball S, number of L points in $S \approx \mathrm{vol}(S)/\det(L)$

Lattices: Basic Concepts

Why is the determinant $\det(L(B)) = |\det(B)|$ a property of the lattice L and not dependent on the particular basis B? Recall:

Lemma (Relation of lattice bases)

Any two bases B, B' of a given lattice L are related by $B' = B \cdot U$ for some matrix $U \in \mathbb{Z}^{n \times n}$ with $\det U \in \{-1, 1\}$.

As a consequence, any two bases of L have the same (absolute) determinant:

$$|\det(B')| = |\det(B \cdot U)| = |\det(B) \cdot \det(U)| = |\det(B)| \cdot |\det(U)| = |\det(B)|.$$

Hence, the determinant (FP volume) is a lattice property, invariant of the basis used.

Lattices: Basic Concepts

Sometimes, useful to remove from each basis vector its components along the previous basis vectors:

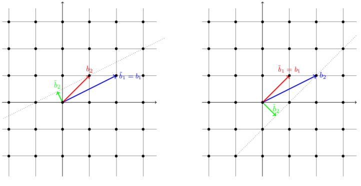
Definition

For a lattice basis $B = (\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n)$, its Gram-Schmidt Orthogonalization (GSO) is the matrix of vectors $B^* = (\vec{b}^*_1, \vec{b}^*_2, \ldots, \vec{b}^*_n)$ defined by $\vec{b}^*_1 = \vec{b}_1$ and for $i \geq 2$,

$$\vec{b}^*_i = \vec{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \cdot \vec{b}^*_j, \text{ where } \mu_{i,j} = \frac{\langle \vec{b}_i, \vec{b}^*_j \rangle}{\langle \vec{b}^*_j, \vec{b}^*_j \rangle}.$$

Example of GSOs in 2-Dimensions:

$$B = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \tilde{B} = \begin{bmatrix} 1 & 0.5 \\ 1 & 0.5 \end{bmatrix}$$



Lattices: Basic Concepts

Can view GSO transformation as re-writing the coordinates of $\vec{b}_i$'s in a rotated coordinate system along $\vec{b}_i^*$s:

$$\begin{bmatrix} | & \cdots & | \\ \vec{b}_1 & \ddots & \vec{b}_n \\ | & \cdots & | \end{bmatrix} = \begin{bmatrix} | & \cdots & | \\ \vec{b}_1^* & \ddots & \vec{b}_n^* \\ | & \cdots & | \end{bmatrix} \cdot \begin{bmatrix} 1 & \mu_{2,1} & \cdots & \mu_{n,1} \\ 0 & 1 & \cdots & \mu_{n,2} \\ 0 & 0 & \cdots & \mu_{n,3} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

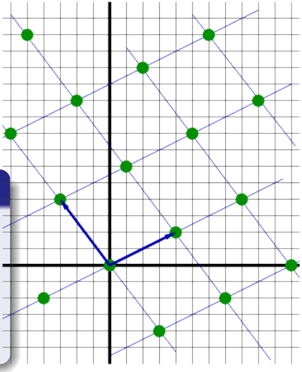
$$= \begin{bmatrix} | & \cdots & | \\ \frac{\vec{b}_1^*}{\|\vec{b}_1^*\|} & \ddots & \frac{\vec{b}_n^*}{\|\vec{b}_n^*\|} \\ | & \cdots & | \end{bmatrix} \cdot \begin{bmatrix} \|\vec{b}_1^*\| & \|\vec{b}_1^*\| \cdot \mu_{2,1} & \cdots & \|\vec{b}_1^*\| \cdot \mu_{n,1} \\ 0 & \|\vec{b}_2^*\| & \cdots & \|\vec{b}_2^*\| \cdot \mu_{n,2} \\ 0 & 0 & \cdots & \|\vec{b}_3^*\| \cdot \mu_{n,3} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \|\vec{b}_n^*\| \end{bmatrix}$$

- ith column of Bottom RHS matrix = coordinates of $\vec{b}_i$ in the rotated coordinate system
- From last row, every non-zero lattice vector has length $\geq \|\vec{b}_n^*\|$.
- Because $\vec{b}_i^*$'s are orthogonal, the FP of B^* is a n-dimensional cube of side lengths $\|\vec{b}_i^*\|$:
  $\det L(B) = |\det(B)| = |\det(B^*)| = \prod_{i=1}^n \|\vec{b}_i^*\|$.

Lattices Background: Shortest Vector Problem (SVP)

For crypto. security, need computationally hard lattice problems.
Many problems related to geometry of lattices seem to be hard!

The most basic geometric quantity about a lattice is its minimum (aka Minkowski first minimum).

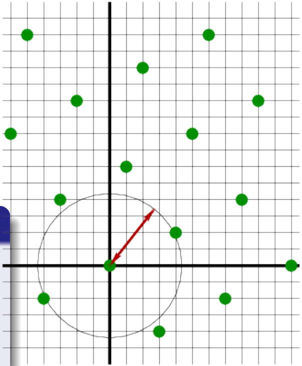


Definition

For an n-dim. lattice L, its minimum $\lambda(L)$ is the length of the shortest non-zero vector of L:
$\lambda(L) = \min(\|\vec{b}\| : \vec{b} \in L \setminus \mathbf{0})$

Lattices Background: Shortest Vector Problem (SVP)

For crypto. security, need computationally hard lattice problems.
Many problems related to geometry of lattices seem to be hard!



The most basic geometric quantity about a lattice is its minimum (aka Minkowski first minimum).

Definition

For an n-dim. lattice L, its minimum $\lambda(L)$ is the length of the shortest non-zero vector of L:
$\lambda(L) = \min(\|\vec{b}\| : \vec{b} \in L \setminus \mathbf{0})$

Lattices Background: Minkowski's Theorem

For a given lattice L, how large can the lattice minimum $\lambda(L)$ be?

Theorem (Minkowski's First Theorem)

For any n-dim. lattice L, we have $\lambda(L) \leq \sqrt{n} \cdot \det L^{1/n}$.

Proof Idea: An analogue of the Pigeon-hole principle.

Lattices Background: Shortest Vector Problem (SVP)

Finding a vector of approximately minimum length seems to be hard, as the dimension n grows.

γ-Shortest Vector Problem (γ-SVP)

Given basis B for n-dim. lattice, find $\vec{b} \in L$ with: $0 < \|\vec{b}\| \leq \gamma \cdot \lambda(L)$.

Hardness of γ-SVP increases as approximation factor γ decreases:

- For $\gamma \geq 2^{O(n)}$: Easy – LLL algorithm solves in $\mathcal{P}oly(n)$ time.
- For $\gamma \leq O(1)$: NP-Hard (under randomized reductions) – very unlikely $\mathcal{P}oly(n)$ time algorithm exists.
- For crypto, need $\gamma = O(n^c)$ for some constant $c \geq 1/2$:
  - Best known attack algorithm time $T = 2^{O(n)}$ (even 'quantumly'!)
  - Best known γ-Time tradeoff: $T = \min(2^{O(n)}, 2^{O(n \log n)/\log \gamma})$.
  - Seems harder than Integer Factorization and Discrete Log.

Lattices Background: Cryptographic Lattices – q-ary lattices and SVP

Hardness of γ-SVP problem instance strongly depends on the given lattice basis B:

- There are many easy instances of γ-SVP, even for $\gamma = 1$ ('NP hard' case). Simple example: $B = I$.

In crypto., need to generate random lattices bases for which γ-SVP is hard to solve 'on average'.

- How to generate such 'hard' random lattices?

One possible answer (Ajtai, 1996): Generate a random q-ary lattice!

Lattices Background: Cryptographic q-ary lattices and SIS Problem

Hardness of γ-SVP problem instance strongly depends on the given lattice basis B:

- There are many easy instances of γ-SVP, even for $\gamma = 1$ ('NP hard' case). Simple example: $B = I$.

In crypto., need to generate random lattices bases for which γ-SVP is hard to solve 'on average'.

- How to generate such 'hard' random lattices?

One possible answer (Ajtai '96): a random q-ary lattice!

Ajtai's Random q-ary 'perp' lattices

Given an integer q and a uniformly random matrix $A \in \mathbb{Z}_q^{n \times m}$, the q-ary perp lattice $L_q^{\perp}(A)$ is defined by:

$$L_q^{\perp}(A) = \{\vec{v} \in \mathbb{Z}^m : A \cdot \vec{v} = \vec{0} \bmod q\}.$$

Lattices Background: Cryptographic q-ary lattices and SIS Problem

- γ-SVP problem for random q-ary perp lattices seems to be hard on average
  - Ajtai proved it, assuming γ-SVP is hard in the worst-case – see end of this module!
- Hardness of this computational problem is security basis for most of lattice-based cryptography.
- Known in lattice-based cryptography as the Small Integer Solution (SIS) Problem.

Problem

Small Integer Solution Problem – $SIS_{q,m,n,\beta}$: *Given n and a matrix A sampled uniformly in $\mathbb{Z}_q^{n\times m}$, find $\vec{v} \in \mathbb{Z}^m \setminus \{\vec{0}\}$ such that $A\vec{v} = \vec{0} \mod q$ and $\|\vec{v}\| \leq \beta$.*

Relation between SIS and γ-SVP

Problem

Small Integer Solution Problem – *$SIS_{q,m,n,\beta}$: Given n and a matrix A sampled uniformly in $\mathbb{Z}_q^{n \times m}$, find $\vec{v} \in \mathbb{Z}^m \setminus \{\vec{0}\}$ such that $A\vec{v} = \vec{0} \mod q$ and $\|\vec{v}\| \leq \beta$.*

Explicit relation of to γ-SVP:

- We have $\det(L_q^{\perp}(A)) = q^n$ (see week 2 tutorial).
- By Minkowski's Theorem, $\lambda(L_q^{\perp}(A)) \leq \sqrt{m} q^{n/m} \approx \sqrt{m}$ for $m \geq n \log q$.
- If Minkowski bound is good, then $SIS_{q,m,\beta} = \gamma$-SVP for $L_q^{\perp}(A)$, with $\gamma \approx \beta / \sqrt{m} q^{n/m}$ (practical refinement to Minkowski bound to be discussed next week).

Crypto. Application: Ajtai's Cryptographic Hash Function

How to use the hardness of SIS problem in cryptography?
First application: Collision-Resistant Hash Function (CRHF).

Definition

Ajtai's Hash Function $g_{q,m,n,d,A}$: Pick $A = (a_{i,j})$ uniformly random $n \times m$ matrix over $\mathbb{Z}_q$ (A = function 'public key'). Given input $\vec{x} \in \mathbb{Z}^m$ having 'small' coordinates ($\|\vec{x}\|_\infty \leq d$), hash function output is defined as

$$g_{q,m,n,d,A}(\vec{x}) = A \cdot \vec{x} \bmod q.$$

$$g(\vec{x}) = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & \cdots & a_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & \cdots & a_{n,m} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ \vdots \\ x_m \end{bmatrix} \bmod q$$

Collision Resistance Security from SIS Problem

- Choose parameters such that domain is larger than range – collisions for f exist: $(2d+1)^m > q^n$.
- e.g., for compression ratio 2, may have $d = 1$, $m = 2 \cdot n \log q / \log(3)$.

Q: Why is it collision-resistant, assuming that SIS is a hard problem?

A: Collision-Resistance Security Reduction from SIS

- We show how to build an efficient SIS algorithm S, given an efficient collision-finder algorithm CF for function g.

Collision Resistance Security from SIS Problem

Suppose there was an efficient collision-finder attack algorithm CF for function g:

- Given random key (A, q) for function g_A, CF runs in time T_B and outputs a collision pair $\vec{x}_1 \neq \vec{x}_2$.

Then, given a SIS instance (A, q), SIS algorithm S:

- Runs collision-finder CF on input (A, q). CF outputs $\vec{x}_1 \neq \vec{x}_2$.
- S outputs SIS problem solution $\vec{v} = \vec{x}_1 - \vec{x}_2$.

Why does S work?

- A collision $\vec{x}_1 \neq \vec{x}_2$ gives a 'short' non-zero vector in $L_q^{\perp}(A)$:

  $$A\vec{x}_1 = A\vec{x}_2 \bmod q \Rightarrow \vec{v} = \vec{x}_1 - \vec{x}_2 \in L_q^{\perp}(A) \setminus \{\vec{0}\}, \|\vec{v}\| \leq \beta,$$

  where $\beta = 2\sqrt{m} \cdot d$.
- S is efficient (run-time $T_S \approx T_{\mathrm{CF}}$) if CF is efficient.

We proved **Theorem:** Collision-Resistance of g is (at least) as hard as $SIS_{q,m,n,\beta}$ with $\beta = 2\sqrt{m} \cdot d$.

Security of Lattice-Based Cryptography

- **Q1:** How should we choose the parameters q, m, n, d of Ajtai's hash function?
- **Q2:** How hard (secure) is SIS Problem and related γ-SVP problem?

Next week: We attempt to answer these questions.