CS-E4340 - Cryptography D:

Lecture 8: Introduction to Lattice-based Cryptography

Russell W. F. Lai

Introduction

Lattice-Based Cryptography and Lattices

What is lattice-based cryptography?

To build secure cryptographic systems based on hard problems over lattices.

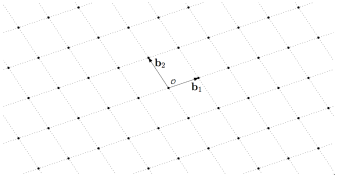
What is a lattice?

A lattice defined by a set of basis vectors

$$\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subseteq \mathbb{R}^n$$

is the set of all vectors in $\mathbb{R}^n$ in the linear span of $\mathbf{B}$ with integer coefficients, i.e.

$$\mathcal{L}(\mathbf{B}) := \{\mathbf{y} \in \mathbb{R}^n : \exists\, \mathbf{x} \in \mathbb{Z}^k,\ \mathbf{y} = \mathbf{B} \cdot \mathbf{x}\}.$$



Lattice-Based Cryptography and Lattices

What is lattice-based cryptography?

To build secure cryptographic systems based on hard problems over lattices.

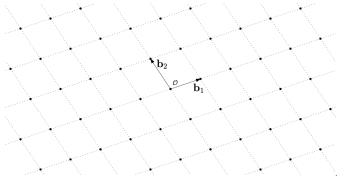
What is a lattice?

A lattice defined by a set of basis vectors

$$\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \subseteq \mathbb{R}^n$$

is the set of all vectors in $\mathbb{R}^n$ in the linear span of $\mathbf{B}$ with integer coefficients, i.e.

$$\mathcal{L}(\mathbf{B}) := \{\mathbf{y} \in \mathbb{R}^n : \exists\, \mathbf{x} \in \mathbb{Z}^k,\ \mathbf{y} = \mathbf{B} \cdot \mathbf{x}\}.$$



Which lattice problems are (believed to be) hard?

- Shortest Vector Problem (SVP):
  Given basis $\mathbf{B}$, find the shortest vector in $\mathcal{L}(\mathbf{B})$.
- Closest Vector Problem (CVP):
  Given basis $\mathbf{B}$ and target vector $\mathbf{t} \in \mathbb{R}^n$, find the vector in $\mathcal{L}(\mathbf{B})$ which is closest to $\mathbf{t}$.

... and their variants, e.g. approximation and promised versions, etc.

Note:

- These are worst-case problems.
- If we say that an algorithm can solve SVP, we mean that it solves SVP for all possible $\mathbf{B}$.
- There are easy instances, e.g. $\mathbf{B} = \mathbf{I}$, the identity matrix.

Which lattice problems are (believed to be) hard?

- Shortest Vector Problem (SVP):
  Given basis $\mathbf{B}$, find the shortest vector in $\mathcal{L}(\mathbf{B})$.
- Closest Vector Problem (CVP):
  Given basis $\mathbf{B}$ and target vector $\mathbf{t} \in \mathbb{R}^n$, find the vector in $\mathcal{L}(\mathbf{B})$ which is closest to $\mathbf{t}$.

... and their variants, e.g. approximation and promised versions, etc.

Note:

- These are worst-case problems.
- If we say that an algorithm can solve SVP, we mean that it solves SVP for all possible $\mathbf{B}$.
- There are easy instances, e.g. $\mathbf{B} = \mathbf{I}$, the identity matrix.

Which lattice problems are (believed to be) hard?

- To build crypto, we need problems which are hard on average.
- For example, you want your encryption to be secure when you sample your secret key at random.
- The two main average-case problems over lattices are:
  - Short Integer Solution (SIS)
  - Learning with Errors (LWE)

  They are in a sense average-case versions of SVP and CVP respectively.
- We will talk about them throughout the lecture.

Which lattice problems are (believed to be) hard?

- To build crypto, we need problems which are hard on average.
- For example, you want your encryption to be secure when you sample your secret key at random.
- The two main average-case problems over lattices are:
  - Short Integer Solution (SIS)
  - Learning with Errors (LWE)

  They are in a sense average-case versions of SVP and CVP respectively.
- We will talk about them throughout the lecture.

Why Lattice-Based Crypto?

Some common reasons, you may have more:

- (Conjectured) Post-Quantum Security: Quantum computers don't seem to help solving lattice problems much faster.
- Security from Worst-case Hardness: The average-case problems (SIS and LWE) are actually as hard as some worst-case problems (variants of SVP and CVP).
- Algorithmic Simplicity and Parallelism: We only do linear algebra over (relatively) small integers. Linear algebra can be parallelised.

Lecture Overview

In this lecture, we will

- † introduce the Short Integer Solution (SIS) and Learning with Errors (LWE) assumptions
- † construct a collision-resistant hash from SIS
- † construct two public-key encryption schemes from LWE

Notation

Our Playground: $\mathbb{Z}$ and $\mathbb{Z}_q$

- $\mathbb{Z} := \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ equipped with $+$ and $\times$
- $q \in \mathbb{N}$: A modulus. We assume it to be prime for simplicity.
- $[a, b] := \{a, a+1, \ldots, b\}$
- $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ is the ring of integers modulo q.
- We will use the balanced representation of $\mathbb{Z}_q$

$$[-\lceil q/2 \rceil + 1, \lfloor q/2 \rfloor].$$

- We will abuse notation and treat $\mathbb{Z}_q = [-\lceil q/2 \rceil + 1, \lfloor q/2 \rfloor]$.

Matrices and Vectors

- **A**: Matrices are denoted by bold captital letters
- **x**: Vectors are denoted by bold lower-case letters
- $\mathbf{x} = (x_i)_i$: The i-th entry of $\mathbf{x}$ is denoted by x_i
- $\|\cdot\|$: A norm over $\mathbb{R}^n$. By default, we use the infinity norm

$$\|\mathbf{x}\| = \max_i |x_i|.$$

Short Integer Solution (SIS) and Learning with Errors (LWE)

The Preimage and Decoding Problems

Suppose we have a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with $m > n$.

We can ask the following questions:

1. Preimage Problem: Given $\mathbf{v} \in \mathbb{Z}_q^n$, find $\mathbf{u} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$.
2. Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Both problems can be solved efficiently, e.g. by Gaussian elimination.

- What if the preimage $\mathbf{u}$ is restricted to fall in a small set $S \subseteq \mathbb{Z}^m$, e.g. $S = \{0,1\}^m$?
- What if the the codeword $\mathbf{b}$ is noisy, i.e. $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e}^\intercal \bmod q$?

The Preimage and Decoding Problems

Suppose we have a full-rank matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with $m > n$.

We can ask the following questions:

1. Preimage Problem: Given $\mathbf{v} \in \mathbb{Z}_q^n$, find $\mathbf{u} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$.
2. Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Both problems can be solved efficiently, e.g. by Gaussian elimination.

† What if the preimage $\mathbf{u}$ is restricted to fall in a small set $S \subseteq \mathbb{Z}^m$, e.g. $S = \{0,1\}^m$?

† What if the the codeword $\mathbf{b}$ is noisy, i.e. $\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q$?

Short Integer Solution (SIS)

- Preimage Problem: Given $\mathbf{v} \in \mathbb{Z}_q^n$, find $\mathbf{u} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$.
- SIS = Preimage problem with bounded-norm constraint

$\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ Problem

Given $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$. Find $\mathbf{u} \in \mathbb{Z}^m$ such that

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \leq \beta.$$

- $\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ Assumption: The $\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ problem is hard for PPT adversaries.
- Typically, we consider $q = \Theta(\beta \cdot n \cdot \log n)$ and $m = \Theta(n \cdot \log_\beta q)$.

Short Integer Solution (SIS)

- Preimage Problem: Given $\mathbf{v} \in \mathbb{Z}_q^n$, find $\mathbf{u} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$.
- SIS = Preimage problem with bounded-norm constraint

$\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ Problem

Given $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$. Find $\mathbf{u} \in \mathbb{Z}^m$ such that

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \leq \beta.$$

- $\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ Assumption: The $\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ problem is hard for PPT adversaries.
- Typically, we consider $q = \Theta(\beta \cdot n \cdot \log n)$ and $m = \Theta(n \cdot \log_\beta q)$.

Short Integer Solution (SIS)

† Preimage Problem: Given $\mathbf{v} \in \mathbb{Z}_q^n$, find $\mathbf{u} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q$.

† SIS = Preimage problem with bounded-norm constraint

$\mathrm{SIS}_{n,m,q,\beta,\mathbf{v}}$ Problem

Given $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$. Find $\mathbf{u} \in \mathbb{Z}^m$ such that

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \leq \beta.$$

† $\mathrm{SIS}_{n,m,q,\beta,\mathbf{v}}$ Assumption: The $\mathrm{SIS}_{n,m,q,\beta,\mathbf{v}}$ problem is hard for PPT adversaries.

† Typically, we consider $q = \Theta(\beta \cdot n \cdot \log n)$ and $m = \Theta(n \cdot \log_\beta q)$.

How does the hardness scale with all the parameters?

$\mathsf{SIS}_{n,m,q,\beta,\mathbf{v}}$ Problem

Given $\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times m}$. Find $\mathbf{u} \in \mathbb{Z}^m$ such that

$$\mathbf{A} \cdot \mathbf{u} = \mathbf{v} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{u}\| \leq \beta.$$

- $n \uparrow \implies$ hardness $\uparrow$: More constraints
- $\beta \downarrow \implies$ hardness $\uparrow$: More restrictive constraint
- $m \uparrow \implies$ hardness $\downarrow$: Higher degree of freedom

Learning with Errors (LWE)

- Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.
- (Search-)LWE = Noisy version of the decoding problem

Search-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Search-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q,$$

recover $\mathbf{s}$.

† By default, we assume χ is the uniform distribution over $\mathbb{Z}_\beta$ for some $\beta \ll q$.

† Search-LWE$_{n,m,q,\chi}$ Assumption: The search-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

† Why Learning with Errors? Given "LWE samples" $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ where

$$b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i \bmod q \qquad \text{and} \qquad e_i \leftarrow\!\$\ \chi,$$

learn the linear function $\langle \mathbf{s}, \cdot \rangle \bmod q$.

Search-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Search-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q,$$

recover $\mathbf{s}$.

† By default, we assume χ is the uniform distribution over $\mathbb{Z}_\beta$ for some $\beta \ll q$.

† Search-LWE$_{n,m,q,\chi}$ Assumption: The search-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

† Why Learning with Errors? Given "LWE samples" $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ where

$$b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i \bmod q \quad \text{and} \quad e_i \leftarrow\!\$\ \chi,$$

learn the linear function $\langle \mathbf{s}, \cdot \rangle \bmod q$.

Search-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Search-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q,$$

recover $\mathbf{s}$.

† By default, we assume χ is the uniform distribution over $\mathbb{Z}_\beta$ for some $\beta \ll q$.

† Search-LWE$_{n,m,q,\chi}$ Assumption: The search-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

† Why Learning with Errors? Given "LWE samples" $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ where

$$b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i \bmod q \quad \text{and} \quad e_i \leftarrow\!\$\ \chi,$$

learn the linear function $\langle \mathbf{s}, \cdot \rangle \bmod q$.

Search-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Search-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where

$$\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e} \bmod q,$$

recover $\mathbf{s}$.

† By default, we assume χ is the uniform distribution over $\mathbb{Z}_\beta$ for some $\beta \ll q$.

† Search-LWE$_{n,m,q,\chi}$ Assumption: The search-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

† Why Learning with Errors? Given "LWE samples" $\{(\mathbf{a}_i, b_i)\}_{i=1}^m$ where

$$b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i \bmod q \qquad \text{and} \qquad e_i \leftarrow\!\$\ \chi,$$

learn the linear function $\langle \mathbf{s}, \cdot \rangle \bmod q$.

Decision-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Decision-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where either

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q \qquad \textit{or} \qquad \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^m.$$

Distinguish which one is the case.

† Decision-LWE$_{n,m,q,\chi}$ Assumption: The decision-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

Decision-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Decision-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where either

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q \qquad \textit{or} \qquad \mathbf{b} \leftarrow\!\!\$\ \mathbb{Z}_q^m.$$

Distinguish which one is the case.

† Decision-LWE$_{n,m,q,\chi}$ Assumption: The decision-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

Decision-Learning with Errors (LWE)

† Decoding Problem: Given $\mathbf{b} \in \mathbb{Z}_q^m$ such that $\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} \bmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, recover $\mathbf{s}$.

Decision-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\!\$\ \mathbb{Z}_q^{n \times m}$, $\mathbf{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where either

$$\mathbf{b}^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e} \bmod q \qquad \textit{or} \qquad \mathbf{b} \leftarrow\!\!\$\ \mathbb{Z}_q^m.$$

Distinguish which one is the case.

† Decision-LWE$_{n,m,q,\chi}$ Assumption: The decision-LWE$_{n,m,q,\chi}$ problem is hard for PPT adversaries.

How does the hardness scale with all the parameters?

Decision-LWE$_{n,m,q,\chi}$ Problem

Let $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n$, and $\mathbf{e} \leftarrow\!\$\ \chi^m$. Given $(\mathbf{A}, \mathbf{b})$ where either

$$\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e} \bmod q \qquad \textit{or} \qquad \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^m.$$

Distinguish which one is the case.

- $n \uparrow \implies$ hardness $\uparrow$: Longer secret
- Entropy of $\chi/q \uparrow \implies$ hardness $\uparrow$: More noise
- $m \uparrow \implies$ hardness $\downarrow$: More samples

Search- v.s. Decision-LWE

- Decision-LWE is hard $\implies$ Search-LWE is hard: Trivial –
  - Given $(\mathbf{A}, \mathbf{b})$, give it to the search oracle.
  - If it returns $\mathbf{s}$, $(\mathbf{A}, \mathbf{b})$ is probably not random.
  - Otherwise, $(\mathbf{A}, \mathbf{b})$ is probably random.
- Search-LWE is hard $\implies$ Decision-LWE is hard: Surprising! Come to the advanced course.

Search- v.s. Decision-LWE

- Decision-LWE is hard $\implies$ Search-LWE is hard: Trivial –
  - Given $(\mathbf{A}, \mathbf{b})$, give it to the search oracle.
  - If it returns $\mathbf{s}$, $(\mathbf{A}, \mathbf{b})$ is probably not random.
  - Otherwise, $(\mathbf{A}, \mathbf{b})$ is probably random.
- Search-LWE is hard $\implies$ Decision-LWE is hard: Surprising! Come to the advanced course.

Search- v.s. Decision-LWE

- Decision-LWE is hard $\implies$ Search-LWE is hard: Trivial –
  - Given $(\mathbf{A}, \mathbf{b})$, give it to the search oracle.
  - If it returns $\mathbf{s}$, $(\mathbf{A}, \mathbf{b})$ is probably not random.
  - Otherwise, $(\mathbf{A}, \mathbf{b})$ is probably random.
- Search-LWE is hard $\implies$ Decision-LWE is hard: Surprising! Come to the advanced course.

Collision-Resistant Hash Functions

Ajtai's Collision-Resistant Hash

We construct a (very simple) hash function family $\{f_{\mathbf{A}}\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$.

- Each function $f_{\mathbf{A}}$ is parametrised by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
- The input space is $\mathbb{Z}_\beta^m$ for some $\beta \in \mathbb{N}$ with $\beta \ll q$.

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q.$$

Ajtai's Collision-Resistant Hash

We construct a (very simple) hash function family $\{f_{\mathbf{A}}\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$.

- Each function $f_{\mathbf{A}}$ is parametrised by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$.
- The input space is $\mathbb{Z}_\beta^m$ for some $\beta \in \mathbb{N}$ with $\beta \ll q$.

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q.$$

Collision-Resistance

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q.$$

† Suppose an adversary can efficiently find a collision for a random $f_{\mathbf{A}}$, i.e.

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{y} \bmod q \qquad \text{and} \qquad \mathbf{x}, \mathbf{y} \in \mathbb{Z}_{\beta}^{m} \qquad \text{and} \qquad \mathbf{x} \neq \mathbf{y}.$$

† We have

$$\mathbf{A} \cdot (\mathbf{x} - \mathbf{y}) = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{x} - \mathbf{y}\| \leq \beta$$

i.e. we solve $\mathrm{SIS}_{n,m,q,\beta}$.

† $\mathrm{SIS}_{n,m,q,\beta}$ assumption $\implies \{f_{\mathbf{A}}\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$ is collision-resistant.

† Sometimes we call $f_{\mathbf{A}}$ a "SIS function".

Collision-Resistance

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q.$$

† Suppose an adversary can efficiently find a collision for a random $f_{\mathbf{A}}$, i.e.

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{y} \bmod q \qquad \text{and} \qquad \mathbf{x}, \mathbf{y} \in \mathbb{Z}_\beta^m \qquad \text{and} \qquad \mathbf{x} \neq \mathbf{y}.$$

† We have

$$\mathbf{A} \cdot (\mathbf{x} - \mathbf{y}) = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{x} - \mathbf{y}\| \leq \beta$$

i.e. we solve $\mathrm{SIS}_{n,m,q,\beta}$.

† $\mathrm{SIS}_{n,m,q,\beta}$ assumption $\implies \left\{f_{\mathbf{A}}\right\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$ is collision-resistant.

† Sometimes we call $f_{\mathbf{A}}$ a "SIS function".

Collision-Resistance

$$f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q.$$

† Suppose an adversary can efficiently find a collision for a random $f_{\mathbf{A}}$, i.e.

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{y} \bmod q \qquad \text{and} \qquad \mathbf{x}, \mathbf{y} \in \mathbb{Z}_\beta^m \qquad \text{and} \qquad \mathbf{x} \neq \mathbf{y}.$$

† We have

$$\mathbf{A} \cdot (\mathbf{x} - \mathbf{y}) = \mathbf{0} \bmod q \qquad \text{and} \qquad 0 < \|\mathbf{x} - \mathbf{y}\| \leq \beta$$

i.e. we solve $\mathrm{SIS}_{n,m,q,\beta}$.

† $\mathrm{SIS}_{n,m,q,\beta}$ assumption $\implies \{f_{\mathbf{A}}\}_{\mathbf{A} \in \mathbb{Z}_q^{n \times m}}$ is collision-resistant.

† Sometimes we call $f_{\mathbf{A}}$ a "SIS function".

Regularity

† The SIS function $f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A} \cdot \mathbf{x} \bmod q$ is more useful than hashing.

† Regularity: If q is prime and $m > n \cdot \log_\beta q + 2\lambda$, then the statistical distance between the following distributions is at most $2^{-\lambda}$:

$$\left\{ (\mathbf{A}, \mathbf{y}) : \begin{array}{c} \mathbf{A} \leftarrow\$ \mathbb{Z}_q^{n \times m} \\ \mathbf{x} \leftarrow\$ \mathbb{Z}_\beta^m \\ \mathbf{y} := \mathbf{A} \cdot \mathbf{x} \bmod q \end{array} \right\} \quad \text{and} \quad \left\{ (\mathbf{A}, \mathbf{y}) : \begin{array}{c} \mathbf{A} \leftarrow\$ \mathbb{Z}_q^{n \times m} \\ \mathbf{y} \leftarrow\$ \mathbb{Z}_q^n \end{array} \right\}$$

† Note: For $f_{\mathbf{A}}$ to be compressing, we anyway want $|\mathbb{Z}_\beta^m| > |\mathbb{Z}_q^n|$, i.e. $m > n \cdot \log_\beta q$.

Public-Key Encryption

Public-Key Encryption (PKE)

What is a public-key encryption scheme?

Like a symmetric-key encryption scheme, but now encryption takes a public key.

- Key Generation: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- Encryption: $\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
- Decryption: $x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

Correctness:

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) = x.$$

Public-Key Encryption (PKE)

What is a public-key encryption scheme?

Like a symmetric-key encryption scheme, but now encryption takes a public key.

- † Key Generation: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- † Encryption: $\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
- † Decryption: $x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

Correctness:

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) = x.$$

Public-Key Encryption (PKE)

What is a public-key encryption scheme?

Like a symmetric-key encryption scheme, but now encryption takes a public key.

- † Key Generation: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- † Encryption: $\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, x)$
- † Decryption: $x \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

Correctness:

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, x)) = x.$$

IND-CPA-Security of PKE

Same as IND-CPA-security of symmetric-key encryption, but simpler
– no need to provide encryption oracle (it's public-key!)

IND-CPA-Security of PKE Π: For any PPT adversary $\mathcal{A}$,

$$\left|\Pr\left[\text{IND-CPA}^0_{\Pi,\mathcal{A}}(1^\lambda)=1\right]-\Pr\left[\text{IND-CPA}^1_{\Pi,\mathcal{A}}(1^\lambda)=1\right]\right|\leq \text{negl}(\lambda)$$

where

$\text{IND-CPA}^b_{\Pi,\mathcal{A}}(1^\lambda)$

$(\text{pk},\text{sk})\leftarrow \text{KGen}(1^\lambda)$

$(x_0,x_1)\leftarrow \mathcal{A}(\text{pk})$

$\text{ctxt}^*\leftarrow \text{Enc}(\text{pk},x_b)$

$b'\leftarrow \mathcal{A}(\text{ctxt}^*)$

return b'

IND-CPA-Security of PKE

Same as IND-CPA-security of symmetric-key encryption, but simpler
– no need to provide encryption oracle (it's public-key!)

IND-CPA-Security of PKE Π: For any PPT adversary $\mathcal{A}$,

$$\left|\Pr\left[\text{IND-CPA}^0_{\Pi,\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\text{IND-CPA}^1_{\Pi,\mathcal{A}}(1^\lambda) = 1\right]\right| \leq \text{negl}(\lambda)$$

where

$\text{IND-CPA}^b_{\Pi,\mathcal{A}}(1^\lambda)$

$(\text{pk}, \text{sk}) \leftarrow \text{KGen}(1^\lambda)$

$(x_0, x_1) \leftarrow \mathcal{A}(\text{pk})$

$\text{ctxt}^* \leftarrow \text{Enc}(\text{pk}, x_b)$

$b' \leftarrow \mathcal{A}(\text{ctxt}^*)$

return b'

Regev's Public-Key Encryption

Observation:

- Given LWE samples $\begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \end{pmatrix}$.
- For short $\mathbf{r}$,

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r} = \begin{pmatrix} \mathbf{A} \cdot \mathbf{r} \\ \mathbf{s}^\mathsf{T} \cdot \mathbf{A} \cdot \mathbf{r} + \mathbf{e}^\mathsf{T} \cdot \mathbf{r} \end{pmatrix}$$

  is still an LWE sample with secret $\mathbf{s}$ and slightly larger error $\mathbf{e}^\mathsf{T} \cdot \mathbf{r}$.

Idea:

- Public key = LWE samples $(\mathbf{A}, \mathbf{b})$
- Secret key = LWE secret $\mathbf{s}$
- Encrypt: Combine LWE samples to get new one, use it to one-time-pad message
- Decrypt: Remove the one-time-pad using LWE secret

Regev's Public-Key Encryption

Observation:

- Given LWE samples $\begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} = \begin{pmatrix} \mathbf{A} \\ \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \end{pmatrix}$.
- For short $\mathbf{r}$,

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r} = \begin{pmatrix} \mathbf{A} \cdot \mathbf{r} \\ \mathbf{s}^\mathsf{T} \cdot \mathbf{A} \cdot \mathbf{r} + \mathbf{e}^\mathsf{T} \cdot \mathbf{r} \end{pmatrix}$$

  is still an LWE sample with secret $\mathbf{s}$ and slightly larger error $\mathbf{e}^\mathsf{T} \cdot \mathbf{r}$.

Idea:

- Public key = LWE samples $(\mathbf{A}, \mathbf{b})$
- Secret key = LWE secret $\mathbf{s}$
- Encrypt: Combine LWE samples to get new one, use it to one-time-pad message
- Decrypt: Remove the one-time-pad using LWE secret

Regev's Public-Key Encryption

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, $\mathsf{sk} = \mathbf{s}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e} \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ \text{and}$$

$$\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{r} \leftarrow\!\$\ \mathbb{Z}_\beta^m$$
$$\mathbf{c}_0 = \mathbf{A} \cdot \mathbf{r} \bmod q$$
$$c_1 = \mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q$$

† Decryption:

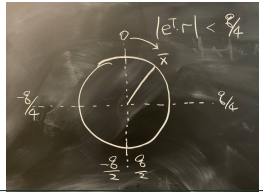
$$\bar{x} = c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness if $q > m \cdot \beta^2 + 1$:

$$\begin{aligned} \bar{x} &= c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q \\ &= (\mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x) - \mathbf{s}^\mathsf{T} \cdot (\mathbf{A} \cdot \mathbf{r}) \bmod q \\ &= \mathbf{e}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q \end{aligned}$$

$$\mathbf{e}^\mathsf{T} \cdot \mathbf{r} = \sum_{i=1}^{m} e_i \cdot r_i$$

$$|\mathbf{e}^\mathsf{T} \cdot \mathbf{r}| \le m \cdot \beta^2 / 4 < q/4$$



Regev's Public-Key Encryption

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, $\mathsf{sk} = \mathbf{s}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e} \leftarrow\!\$\ \mathbb{Z}_\beta^m, \text{ and}$$

$$\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{r} \leftarrow\!\$\ \mathbb{Z}_\beta^m$$
$$\mathbf{c}_0 = \mathbf{A} \cdot \mathbf{r} \bmod q$$
$$c_1 = \mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q$$

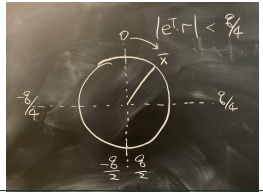
† Decryption:

$$\bar{x} = c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness if $q > m \cdot \beta^2 + 1$:

$$\begin{aligned} \bar{x} &= c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q \\ &= (\mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x) - \mathbf{s}^\mathsf{T} \cdot (\mathbf{A} \cdot \mathbf{r}) \bmod q \\ &= \mathbf{e}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q \end{aligned}$$

$$\mathbf{e}^\mathsf{T} \cdot \mathbf{r} = \sum_{i=1}^{m} e_i \cdot r_i$$
$$|\mathbf{e}^\mathsf{T} \cdot \mathbf{r}| \leq m \cdot \beta^2/4 < q/4$$



IND-CPA-Security

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, $\mathsf{sk} = \mathbf{s}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e} \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ \text{and}$$

$$\mathbf{b}^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}^\mathsf{T} \bmod q$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\begin{aligned} \mathbf{r} &\leftarrow\!\$\ \mathbb{Z}_\beta^m \\ \mathbf{c}_0 &= \mathbf{A} \cdot \mathbf{r} \bmod q \\ c_1 &= \mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q \end{aligned}$$

† Decryption:

$$\begin{aligned} \bar{x} &= c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q \\ x &= \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

IND-CPA-Security if

† $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, where χ is the uniform distribution over $\mathbb{Z}_\beta$, and

† $m > (n+1) \cdot \log_\beta q + 2\lambda$.

Note: $\mathbf{r} \mapsto \begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r}$ is a SIS function

IND-CPA-Security

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m} \text{ and } \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^n$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\begin{aligned} \mathbf{r} &\leftarrow\!\$\ \mathbb{Z}_\beta^m \\ \mathbf{c}_0 &= \mathbf{A} \cdot \mathbf{r} \bmod q \\ c_1 &= \mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q \end{aligned}$$

† Decryption:

$$\begin{aligned} \bar{x} &= c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q \\ x &= \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

IND-CPA-Security if

† $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, where χ is the uniform distribution over $\mathbb{Z}_\beta$, and

† $m > (n+1) \cdot \log_\beta q + 2\lambda$.

Note: $\mathbf{r} \mapsto \begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r}$ is a SIS function

IND-CPA-Security

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m} \text{ and } \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^n$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\begin{aligned} \mathbf{r} &\leftarrow\!\$\ \mathbb{Z}_\beta^m \\ \mathbf{c}_0 &= \mathbf{A} \cdot \mathbf{r} \bmod q \\ c_1 &= \mathbf{b}^\mathsf{T} \cdot \mathbf{r} + \lfloor q/2 \rfloor \cdot x \bmod q \end{aligned}$$

† Decryption:

$$\begin{aligned} \bar{x} &= c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q \\ x &= \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

IND-CPA-Security if

† $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, where χ is the uniform distribution over $\mathbb{Z}_\beta$, and

† $m > (n+1) \cdot \log_\beta q + 2\lambda$.

Note: $\mathbf{r} \mapsto \begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r}$ is a SIS function

IND-CPA-Security

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m} \text{ and } \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^n$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{d}_0 \leftarrow\!\$\ \mathbb{Z}_q^n,\ d_1 \leftarrow\!\$\ \mathbb{Z}_q$$
$$\mathbf{c}_0 = \mathbf{d}_0$$
$$c_1 = d_1 + \lfloor q/2 \rfloor \cdot x \bmod q$$

(by regularity)

† Decryption:

$$\bar{x} = c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

IND-CPA-Security if

† $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, where χ is the uniform distribution over $\mathbb{Z}_\beta$, and

† $m > (n+1) \cdot \log_\beta q + 2\lambda$.

Note: $\mathbf{r} \mapsto \begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r}$ is a SIS function

IND-CPA-Security

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{b})$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m} \text{ and } \mathbf{b} \leftarrow\!\$\ \mathbb{Z}_q^n$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{c}_0 \leftarrow\!\$\ \mathbb{Z}_q^n \text{ and } c_1 \leftarrow\!\$\ \mathbb{Z}_q$$

† Decryption:

$$\bar{x} = c_1 - \mathbf{s}^\mathsf{T} \cdot \mathbf{c}_0 \bmod q$$

$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

IND-CPA-Security if

† $\mathsf{LWE}_{n,m,q,\chi}$ assumption holds, where χ is the uniform distribution over $\mathbb{Z}_\beta$, and

† $m > (n+1) \cdot \log_\beta q + 2\lambda$.

Note: $\mathbf{r} \mapsto \begin{pmatrix} \mathbf{A} \\ \mathbf{b}^\mathsf{T} \end{pmatrix} \cdot \mathbf{r}$ is a SIS function

Dual-Regev Public-Key Encryption

- Observation: In Regev's scheme,
  - pk = LWE samples
  - ctxt = SIS function outputs + (encoding of) x
- Idea: Flip the roles of pk and ctxt.

Dual-Regev Public-Key Encryption

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, $\mathsf{sk} = \mathbf{u}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{u} \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ \text{and}$$

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{u} \bmod q$$

† Encryption of $x \in \{0, 1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e}_0 \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ e_1 \leftarrow\!\$\ \mathbb{Z}_\beta$$
$$\mathbf{c}_0^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e}_0^\intercal \bmod q$$
$$c_1 = \mathbf{s}^\intercal \cdot \mathbf{v} + e_1 + \lfloor q/2 \rfloor \cdot x \bmod q$$

† Decryption:

$$\bar{x} = c_1 - \mathbf{c}_0^\intercal \cdot \mathbf{u} \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness and IND-CPA-security:

† Left as exercises.

Interesting property:

† For each $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, there are many functionally equivalent secret keys.

† Useful for constructing more advanced encryption schemes.

Dual-Regev Public-Key Encryption

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, $\mathsf{sk} = \mathbf{u}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{u} \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ \text{and}$$

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{u} \bmod q$$

† Encryption of $x \in \{0,1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e}_0 \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ e_1 \leftarrow\!\$\ \mathbb{Z}_\beta$$
$$\mathbf{c}_0^\intercal = \mathbf{s}^\intercal \cdot \mathbf{A} + \mathbf{e}_0^\intercal \bmod q$$
$$c_1 = \mathbf{s}^\intercal \cdot \mathbf{v} + e_1 + \lfloor q/2 \rfloor \cdot x \bmod q$$

† Decryption:

$$\bar{x} = c_1 - \mathbf{c}_0^\intercal \cdot \mathbf{u} \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness and IND-CPA-security:

† Left as exercises.

Interesting property:

† For each $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, there are many functionally equivalent secret keys.

† Useful for constructing more advanced encryption schemes.

Dual-Regev Public-Key Encryption

† Key Generation: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, $\mathsf{sk} = \mathbf{u}$, where

$$\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times m},\ \mathbf{u} \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ \text{and}$$

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{u} \bmod q$$

† Encryption of $x \in \{0, 1\}$: $\mathsf{ctxt} = (\mathbf{c}_0, c_1)$ where

$$\mathbf{s} \leftarrow\!\$\ \mathbb{Z}_q^n,\ \mathbf{e}_0 \leftarrow\!\$\ \mathbb{Z}_\beta^m,\ e_1 \leftarrow\!\$\ \mathbb{Z}_\beta$$
$$\mathbf{c}_0^\mathsf{T} = \mathbf{s}^\mathsf{T} \cdot \mathbf{A} + \mathbf{e}_0^\mathsf{T} \bmod q$$
$$c_1 = \mathbf{s}^\mathsf{T} \cdot \mathbf{v} + e_1 + \lfloor q/2 \rfloor \cdot x \bmod q$$

† Decryption:

$$\bar{x} = c_1 - \mathbf{c}_0^\mathsf{T} \cdot \mathbf{u} \bmod q$$
$$x = \begin{cases} 0 & |\bar{x}| < q/4 \\ 1 & \text{otherwise} \end{cases}$$

Correctness and IND-CPA-security:

† Left as exercises.

Interesting property:

† For each $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$, there are many functionally equivalent secret keys.

† Useful for constructing more advanced encryption schemes.