Lattice Exercises - Solutions

Mohammad Ferry Husnil Arif

September 21, 2025

Exercise 1 - Easy or difficult?

For each problem, determine if it is easy (polynomial complexity) or difficult to solve (exponential complexity) and justify by giving the algorithm if it exists (for $m \geq n$):

1. Given a lattice basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$ and a vector $\mathbf{v} \in \mathbb{Z}^m$, decide if $\mathbf{v} \in \Lambda(\mathbf{B})$.

   Solution: This problem is **easy** (polynomial complexity). To check if $\mathbf{v} \in \Lambda(\mathbf{B})$, we need to determine if there exists an integer vector $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathbf{B}\mathbf{x} = \mathbf{v}$.

   Since $\mathbf{B}$ generates a lattice, it has full column rank. We solve the system $\mathbf{B}\mathbf{x} = \mathbf{v}$ over $\mathbb{Q}$ using standard Gaussian elimination. If the system has no solution over $\mathbb{Q}$, then certainly $\mathbf{v} \notin \Lambda(\mathbf{B})$. If a unique solution $\mathbf{x} \in \mathbb{Q}^n$ exists (which is guaranteed for full column rank), we check whether all components of $\mathbf{x}$ are integers. If $\mathbf{x} \in \mathbb{Z}^n$, then $\mathbf{v} \in \Lambda(\mathbf{B})$; otherwise $\mathbf{v} \notin \Lambda(\mathbf{B})$.

   The complexity is $O(mn^2)$ for Gaussian elimination over $\mathbb{Q}$, plus $O(n)$ for checking integrality.

2. Given $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{Z}^{m \times n}$, decide if $\Lambda(\mathbf{B}_1) = \Lambda(\mathbf{B}_2)$.

   Solution: This problem is **easy** (polynomial complexity). To check if $\Lambda(\mathbf{B}_1) = \Lambda(\mathbf{B}_2)$, we verify that each basis generates the same lattice by checking mutual containment.

   Two lattices are equal if and only if each is contained in the other. Therefore:

   (a) Check if each column of $\mathbf{B}_1$ belongs to $\Lambda(\mathbf{B}_2)$ using the algorithm from Exercise 1.1

   (b) Check if each column of $\mathbf{B}_2$ belongs to $\Lambda(\mathbf{B}_1)$ using the same algorithm

   (c) If both conditions hold, then $\Lambda(\mathbf{B}_1) = \Lambda(\mathbf{B}_2)$

   Since we perform $2n$ membership tests, each taking $O(mn^2)$ time, the total complexity is $O(mn^3)$.

3. Given an integer matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, compute a basis for the lattice $\{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{x}^T\mathbf{A} = \mathbf{0}\}$.

Solution: This problem is **easy** (polynomial complexity). Note that the lattice $\{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{x}^T\mathbf{A} = \mathbf{0}\}$ is equivalent to the orthogonal lattice $\Lambda_q^\perp(\mathbf{A}^T)$ where $\mathbf{A}^T \in \mathbb{Z}_q^{n\times m}$. A detailed algorithm for computing a basis of such orthogonal lattices is provided in Exercise 5.3, which shows how to efficiently construct a basis with the algorithm runs in polynomial time.

4. Given an integer matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, compute a basis for the lattice $\{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{x}^T\mathbf{A} = \mathbf{0}\}$ such that each vector of this basis has an euclidean norm bounded by $q/2\sqrt{n}$.

   Solution: This problem is **difficult** (exponential complexity). While finding *some* basis for the lattice $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}_q^m : \mathbf{x}^T\mathbf{A} = \mathbf{0} \bmod q\}$ is easy (as shown in Exercise 1.3), finding a basis with all vectors having Euclidean norm bounded by $\beta = q/(2\sqrt{n})$ is computationally hard.

   This is precisely the Short Integer Solution (SIS) problem. By Proposition 5.7 from [GPV08], solving $\mathrm{SIS}_{q,m,\beta}$ with $\beta = q/(2\sqrt{n})$ is as hard as approximating SIVP (Shortest Independent Vectors Problem) in the worst case to within $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$ factors.

   Specifically, with $\beta = q/(2\sqrt{n})$, we get:

$$\gamma = \frac{q}{2\sqrt{n}} \cdot \tilde{O}(\sqrt{n}) = \frac{q}{2} \cdot \tilde{O}(1)$$

   Since q is typically polynomial in n (i.e., $q = \mathrm{poly}(n)$), we have $\gamma = \mathrm{poly}(n)$. According to the complexity of lattice problems, solving SIVP_γ with $\gamma = \mathrm{poly}(n)$ requires time $2^{\Omega(n)}$, which is exponential.

5. Given a basis $\mathbf{C}$, check if $\Lambda(\mathbf{C})$ is cyclic (i.e., for every lattice vector $\mathbf{x} \in \Lambda(\mathbf{C})$, all the vectors obtained by cyclically rotating the coordinates of $\mathbf{x}$ also belong to the lattice).

   Solution: This problem is **easy** (polynomial complexity). To check if $\Lambda(\mathbf{C})$ is cyclic, we need to verify that for every lattice vector $\mathbf{v} \in \Lambda(\mathbf{C})$, its cyclic rotation $\mathbf{T}(\mathbf{v})$ also belongs to the lattice, where $\mathbf{T}$ is the cyclic permutation matrix that shifts coordinates: $(v_1, v_2, \ldots, v_n) \mapsto (v_n, v_1, \ldots, v_{n-1})$.

   The key observation is that we only need to check this property for the basis vectors of $\mathbf{C}$. This is because if $\mathbf{T}(\mathbf{c}_i) \in \Lambda(\mathbf{C})$ for all basis vectors $\mathbf{c}_i$ (columns of $\mathbf{C}$), then by linearity of $\mathbf{T}$:

   - For any $\mathbf{v} = \sum_{i=1}^n a_i\mathbf{c}_i \in \Lambda(\mathbf{C})$ where $a_i \in \mathbb{Z}$
   - We have $\mathbf{T}(\mathbf{v}) = \mathbf{T}\left(\sum_{i=1}^n a_i\mathbf{c}_i\right) = \sum_{i=1}^n a_i\mathbf{T}(\mathbf{c}_i)$
   - Since each $\mathbf{T}(\mathbf{c}_i) \in \Lambda(\mathbf{C})$ and lattices are closed under integer linear combinations, we get $\mathbf{T}(\mathbf{v}) \in \Lambda(\mathbf{C})$

   Therefore, our algorithm is:

   (a) For each column $\mathbf{c}_i$ of the basis matrix $\mathbf{C}$:

(b) Compute $\mathbf{T}(\mathbf{c}_i)$ (the cyclic rotation of $\mathbf{c}_i$)

(c) Check if $\mathbf{T}(\mathbf{c}_i) \in \Lambda(\mathbf{C})$ by solving the system $\mathbf{Cx} = \mathbf{T}(\mathbf{c}_i)$ for integer $\mathbf{x} \in \mathbb{Z}^n$

(d) If no integer solution exists for any $\mathbf{c}_i$, then the lattice is not cyclic

Step 3 uses the same algorithm as Exercise 1.1 (checking membership in a lattice). The total complexity is $O(n)$ times the complexity of Exercise 1.1, which gives us $O(mn^3)$.

6. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be a uniformly sampled matrix with $m \geq 4n \log q$, and $\mathbf{r}$ be uniformly sampled in $\{0, 1\}^m$. Given $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$, find $\mathbf{r}$.

Solution: This problem is **difficult** (exponential complexity). By Lemma 5.1 from [GPV08], when $m \geq 2n \log q$, for all but a q^{-n} fraction of matrices $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, the subset-sums of columns of $\mathbf{A}$ generate $\mathbb{Z}_q^n$ and stronger result in footnote 7 of [GPV08] state that a random subset-sum of $\mathbf{A}$'s columns is statistically close to uniform over $\mathbb{Z}_q^n$ for almost all $\mathbf{A}$.

In our case, with $m \geq 4n \log q$ (which exceeds the requirement), the syndrome $\mathbf{r}^T\mathbf{A}$ mod q is statistically close to uniform over $\mathbb{Z}_q^n$. This means it reveals essentially no information about $\mathbf{r}$ that could help narrow down the search space.

The only known algorithm is brute force:

(a) For each possible $\mathbf{r}' \in \{0, 1\}^m$:

(b) Compute $\mathbf{s}' = \mathbf{r}'^T\mathbf{A} \bmod q$

(c) If $\mathbf{s}' = \mathbf{r}^T\mathbf{A}$, output $\mathbf{r}'$ and halt

This algorithm has complexity $O(mn \cdot 2^m)$, which is exponential in m. The statistical closeness to uniform distribution ensures that no better algorithm exists, as the syndrome provides no useful structure to exploit.

7. Let $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ be a uniformly sampled matrix with $m \geq 4n \log q$, and $\mathbf{r}$ be uniformly sampled in $\{0, 1\}^n$. Given $(\mathbf{A}, \mathbf{Ar})$, find $\mathbf{r}$.

Solution: This problem is **easy** (polynomial complexity).

This is simply solving a linear system $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{b} = \mathbf{Ar}$ is given. Since $m \geq 4n \log q \gg n$, the system is overdetermined (more equations than unknowns). With high probability over the choice of random $\mathbf{A}$, the matrix has full column rank, ensuring at most one solution exists.

The algorithm is:

(a) Solve the linear system $\mathbf{Ax} = \mathbf{b}$ over $\mathbb{Z}_q$ to find $\mathbf{x} \in \mathbb{Z}_q^n$

(b) Check if $\mathbf{x} \in \{0, 1\}^n$

(c) If yes, output $\mathbf{x} = \mathbf{r}$; otherwise, no valid solution exists

Step 1 can be done using Gaussian elimination, taking polynomial time $O(mn^2)$. Since $\mathbf{A}$ is random with $m \gg n$, the solution (if it exists) is unique with overwhelming probability, and it must be the original $\mathbf{r}$ since $\mathbf{Ar} = \mathbf{b}$.

Exercise 2 - Solving LWE in dimension 2 and 3

Solve (in $\mathbb{Z}$) the following linear systems of equations with noise, knowing that in each equation, the noise is in $\{0, 1\}$:

1.

$$\begin{aligned} x_1 + x_2 &\simeq 3 \\ 2x_1 + x_2 &\simeq 4 \\ x_1 + 3x_2 &\simeq 4 \\ -x_1 + x_2 &\simeq 1 \\ 3x_1 + 2x_2 &\simeq 5 \end{aligned}$$

2.

$$\begin{aligned} 2x_1 + x_2 + x_3 &\simeq 10 \\ x_1 + 4x_2 + 3x_3 &\simeq 26 \\ 3x_1 + x_2 + 2x_3 &\simeq 13 \\ x_1 + 2x_2 + 2x_3 &\simeq 15 \\ 2x_1 + 2x_2 + x_3 &\simeq 15 \end{aligned}$$

Solution: We solve these noisy linear systems by reformulating them as Closest Vector Problem (CVP) instances and applying Kannan's embedding technique [Kan83].

For a system of noisy equations where each equation has the form $\mathbf{a}_i^T \mathbf{x} \simeq b_i$ with noise in $\{0, 1\}$, we can write:

$$\mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{e}$$

where $\mathbf{A}$ is the coefficient matrix, $\mathbf{b}$ is the vector of right-hand sides, and $\mathbf{e} \in \{0, 1\}^m$ is the unknown noise vector.

This is equivalent to finding the closest point in the lattice $\Lambda = \{\mathbf{A}\mathbf{y} : \mathbf{y} \in \mathbb{Z}^n\}$ to the target vector $\mathbf{b}$. The closest lattice point $\mathbf{A}\mathbf{x}^*$ will satisfy $\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_\infty \leq 1$, ensuring all noise components are in $\{0, 1\}$.

Following Kannan's embedding method, we construct an extended lattice with basis:

$$\mathbf{B}' = \begin{pmatrix} \mathbf{I}_n & \mathbf{A}^T \\ \mathbf{0} & -\mathbf{b}^T \end{pmatrix}$$

and append an additional column $(\mathbf{0}, \dots, 0, M)^T$ where M is a large embedding parameter.

The short vector in this extended lattice has the form $(\mathbf{x}^*, 1)$ which, when multiplied by $\mathbf{B}'$, gives us $(\mathbf{x}^*, \mathbf{A}\mathbf{x}^* - \mathbf{b}, M)$. Since $\mathbf{A}\mathbf{x}^* - \mathbf{b} = -\mathbf{e}$ where $\mathbf{e} \in \{0, 1\}^m$, we can directly verify that each component of $\mathbf{A}\mathbf{x}^* - \mathbf{b}$ is in $\{-1, 0\}$.

Implementation in SageMath:

For system 2.1:

```
A = matrix(ZZ, [
    [1, 1],
    [2, 1],
    [1, 3],
    [-1, 1],
    [3, 2]
])
v = vector(ZZ, [3, 4, 4, 1, 5])

# Build the block matrix
M = block_matrix([
    [identity_matrix(2), A.T],
    [zero_matrix(1, 2), -matrix(ZZ, v)]
])
M = M.augment(vector(ZZ, [0, 0, 2**64]))
M[:, 2:7] *= 2**32
M = M.LLL()
M[:, 2:7] /= 2**32

for row in M:
    if abs(row[-1]) == 2**64:
        x = row[0:2]
        assert all(num in [0, 1] for num in (v - A*x))
        print(f"{x=}")
```

This gives us $\mathbf{x} = (\mathbf{1}, \mathbf{1})$ with noise vector $\mathbf{e} = (1, 1, 0, 1, 0)$.

For system 2.2:

```
A = matrix(ZZ, [
    [2, 1, 1],
    [1, 4, 3],
    [3, 1, 2],
    [1, 2, 2],
    [2, 2, 1]
])
v = vector(ZZ, [10, 26, 13, 15, 15])

# Build the block matrix
M = block_matrix([
    [identity_matrix(3), A.T],
    [zero_matrix(1, 3), -matrix(ZZ, v)]
])
M = M.augment(vector(ZZ, [0, 0, 0, 2**64]))
M[:, 3:8] *= 2**32
M = M.LLL()
M[:, 3:8] /= 2**32

```

```python
for row in M:
    if abs(row[-1]) == 2**64:
        x = row[0:3]
        assert all(num in [0, 1] for num in (v - A*x))
        print(f"{x=}")
```

This gives us $\mathbf{x} = (\mathbf{2}, \mathbf{5}, \mathbf{1})$ with noise vector $\mathbf{e} = (0, 1, 0, 1, 0)$.

The scaling factors 2^{32} and 2^{64} are used to ensure numerical stability during LLL reduction while preserving the integer structure of the problem.

Verification: For both solutions, we verify that $\mathbf{Ax} + \mathbf{e} = \mathbf{b}$ where each component of $\mathbf{e}$ is indeed in $\{0, 1\}$.

Exercise 3 - Reduction

1. Let $n \geq 1$ be an integer, show that there is a reduction from $\text{LWE}_{n,q,\alpha}$ for m samples to $\text{SIS}_{q,m,\beta}$. On which condition on α and β does it work?

Solution: We show a reduction from $\text{LWE}_{n,q,\alpha}$ (decision version) to $\text{SIS}_{q,m,\beta}$.

First, let us define the two problems precisely:

- **$\text{LWE}_{n,q,\alpha}$ (Decision)**: Given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{v} \in \mathbb{Z}_q^m$, distinguish between:
  - Case 1: $\mathbf{v} = \mathbf{As} + \mathbf{e} \pmod q$ where $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$
  - Case 2: $\mathbf{v} \leftarrow U(\mathbb{Z}_q^m)$ (uniformly random)
- **$\text{SIS}_{q,m,\beta}$**: Given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, find a nonzero vector $\mathbf{w} \in \mathbb{Z}^m$ such that $\mathbf{A}^T\mathbf{w} = \mathbf{0} \pmod q$ and $\|\mathbf{w}\| \leq \beta$.

The Reduction: Given an LWE instance $(\mathbf{A}, \mathbf{v})$, we use the SIS solver to distinguish whether $\mathbf{v}$ is an LWE sample or uniformly random:

(a) Use the $\text{SIS}_{q,m,\beta}$ solver on $\mathbf{A}$ to obtain a short vector $\mathbf{w} \in \mathbb{Z}^m$ such that $\mathbf{A}^T\mathbf{w} = \mathbf{0} \pmod q$ and $\|\mathbf{w}\| \leq \beta$.

(b) Compute the inner product $\langle \mathbf{v}, \mathbf{w} \rangle \pmod q$.

(c) If $|\langle \mathbf{v}, \mathbf{w} \rangle| < q/10$, output "LWE sample"; otherwise output "uniform".

Analysis: The key observation is that:

- If $\mathbf{v} = \mathbf{As} + \mathbf{e}$, then

$$\langle \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{As} + \mathbf{e}, \mathbf{w} \rangle = \langle \mathbf{s}, \mathbf{A}^T\mathbf{w} \rangle + \langle \mathbf{e}, \mathbf{w} \rangle = 0 + \langle \mathbf{e}, \mathbf{w} \rangle \pmod q$$

- If $\mathbf{v}$ is uniform, then $\langle \mathbf{v}, \mathbf{w} \rangle$ is uniformly distributed over $\mathbb{Z}_q$.

To bound $|\langle \mathbf{e}, \mathbf{w} \rangle|$, we need know the bound $\|\mathbf{e}\|$ where $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$.

For negligible ϵ, by Lemma 3.1 from [GPV08], the smoothing parameter of $\mathbb{Z}^m$ satisfies:

$$\eta_\epsilon(\mathbb{Z}^m) \leq \mathrm{bl}(\mathbb{Z}^m) \cdot \omega(\sqrt{\log m}) = 1 \cdot \omega(\sqrt{\log m}) = \omega(\sqrt{\log m})$$

If we set $\alpha q \geq \omega(\sqrt{\log m})$, then $\alpha q \geq \eta_\epsilon(\mathbb{Z}^m)$.

By Lemma 2.9 from [GPV08], when $s = \alpha q \geq \eta_\epsilon(\mathbb{Z}^m)$, for $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$ we have:

$$\Pr[\|\mathbf{e}\| > \alpha q \sqrt{m}] \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-m}$$

which is negligible. Therefore, with overwhelming probability, $\|\mathbf{e}\| \leq \alpha q \sqrt{m}$.

Consequently:

$$|\langle \mathbf{e}, \mathbf{w} \rangle| \leq \|\mathbf{e}\| \cdot \|\mathbf{w}\| \leq \alpha q \sqrt{m} \cdot \beta = \alpha \beta q \sqrt{m}$$

For the reduction to successfully distinguish between the two cases, we need $\alpha \beta q \sqrt{m} < q/10$, which gives us:

Parameter Condition: The reduction works when

$$\boxed{\alpha \beta < \frac{1}{10\sqrt{m}}}$$

assuming $\alpha q \geq \omega(\sqrt{\log m})$ hold.

Under this condition, LWE samples will have $|\langle \mathbf{v}, \mathbf{w} \rangle| = |\langle \mathbf{e}, \mathbf{w} \rangle| < q/10$, while uniform samples will have $\langle \mathbf{v}, \mathbf{w} \rangle$ distributed uniformly over $\mathbb{Z}_q$, allowing us to distinguish between the two cases.

Exercise 4 - Dual-Regev Encryption scheme

We first define the Dual-Regev encryption scheme.

Definition 1 (Dual-Regev Encryption)**.** *Let n, m, and q be integers such that q is prime and $m \geq O(n \log q)$, and let α, γ be real numbers.*

$\mathsf{DualRegev.KeyGen}(n, m)$*: Sample $\mathbf{A}$ uniform in $\mathbb{Z}_q^{m \times n}$, and $\mathbf{x}$ discrete Gaussian on $\mathbb{Z}^m$ of parameter γq. The secret key is $\mathsf{sk} = \mathbf{x}$ and the public key is $\mathsf{pk} = \mathbf{y}^T = \mathbf{x}^T \mathbf{A} \bmod q$ in $\mathbb{Z}_q^n$.*

$\mathsf{DualRegev.Enc}(M, \mathsf{pk})$*: Given $M \in \{0,1\}$, sample $\mathbf{s} \leftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \leftarrow D_{\mathbb{Z}^m, \alpha q}$ and $e' \leftarrow D_{\mathbb{Z}, \alpha q}$. The ciphertext is $(\mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{y}^T \mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$.*

$\mathsf{DualRegev.Dec}((\mathbf{b}, c), \mathsf{sk})$*: Given a ciphertext $(\mathbf{b}, c)$, compute ... ?*

1. Give the decryption algorithm, what do you compute, and how do you find M?

   Solution: The decryption algorithm works as follows:

   $\mathsf{DualRegev.Dec}((\mathbf{b}, c), \mathsf{sk} = \mathbf{x})$:

   (a) Compute $b' = c - \mathbf{x}^T\mathbf{b} \pmod q$
   (b) Output $M = 0$ if b' is closer to 0 than to $\lfloor q/2 \rfloor$ (i.e., if $|b'| < q/10$)
   (c) Output $M = 1$ if b' is closer to $\lfloor q/2 \rfloor$ than to 0 (i.e., if $|b' - \lfloor q/2 \rfloor| < q/10$)

   This works because:

   $$\begin{aligned} b' = c - \mathbf{x}^T\mathbf{b} &= (\mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M) - \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) \\ &= \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M - \mathbf{x}^T\mathbf{A}\mathbf{s} - \mathbf{x}^T\mathbf{e} \\ &= e' - \mathbf{x}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot M \end{aligned}$$

   where we used the fact that $\mathbf{y}^T = \mathbf{x}^T\mathbf{A} \pmod q$.

2. What is the condition between α, γ and q to make sure the scheme is correct?

   Solution: For correct decryption, we need $|e' - \mathbf{x}^T\mathbf{e}| < q/10$ to ensure we can distinguish between the cases $M = 0$ and $M = 1$.

   To analyze this, we define:

   $$\tilde{\mathbf{e}} = \begin{pmatrix} e' \\ -\mathbf{e} \end{pmatrix} \in \mathbb{Z}^{m+1}, \quad \tilde{\mathbf{x}} = \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \in \mathbb{Z}^{m+1}$$

   Then $e' - \mathbf{x}^T\mathbf{e} = \tilde{\mathbf{x}}^T\tilde{\mathbf{e}}$, and we can bound:

   $$|e' - \mathbf{x}^T\mathbf{e}| = |\tilde{\mathbf{x}}^T\tilde{\mathbf{e}}| \le \|\tilde{\mathbf{x}}\| \cdot \|\tilde{\mathbf{e}}\|$$

   Following the same approach as in Exercise 3, we use Lemma 3.1 from [GPV08] to establish that the smoothing parameter $\eta_\epsilon(\mathbb{Z}^m) \le \omega(\sqrt{\log m})$. Then, if we set $\alpha q \ge \omega(\sqrt{\log m})$ and $\gamma q \ge \omega(\sqrt{\log m})$, we can apply Lemma 2.9 from [GPV08] to obtain that with overwhelming probability:

   - $\|\tilde{\mathbf{e}}\| \le \alpha q\sqrt{m+1}$ (since $\tilde{\mathbf{e}}$ has distribution $D_{\mathbb{Z}^{m+1},\alpha q}$)
   - $\|\mathbf{x}\| \le \gamma q\sqrt{m}$ (since $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\gamma q}$)

   Since $\|\tilde{\mathbf{x}}\|^2 = 1 + \|\mathbf{x}\|^2$, we have:

   $$\|\tilde{\mathbf{x}}\| = \sqrt{1 + \|\mathbf{x}\|^2} \le \sqrt{1 + \gamma^2 q^2 m}$$

   Therefore:

   $$|e' - \mathbf{x}^T\mathbf{e}| \le \sqrt{1 + \gamma^2 q^2 m} \cdot \alpha q\sqrt{m+1}$$

For large $\gamma q\sqrt{m}$, we can approximate $\sqrt{1+\gamma^2 q^2 m} \approx \gamma q\sqrt{m}$, giving:

$$|e' - \mathbf{x}^T\mathbf{e}| \lesssim \gamma q\sqrt{m} \cdot \alpha q\sqrt{m+1} \approx \alpha\gamma q^2 m$$

For correctness, we require:

$$\alpha\gamma q^2 m < \frac{q}{10}$$

Correctness Condition:

$$\boxed{\alpha\gamma q < \frac{1}{10m}}$$

This condition ensures correct decryption with overwhelming probability, assuming $\alpha q \geq \omega(\sqrt{\log m})$ and $\gamma q \geq \omega(\sqrt{\log m})$.

3. Show that the distribution of the public key is statistically close to the uniform distribution in $\mathbb{Z}_q^n$.

   Solution: The public key in the Dual-Regev encryption scheme is $\mathbf{y}^T = \mathbf{x}^T\mathbf{A} \bmod q$ where $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\gamma q}$. By Corollary 5.4 from [GPV08], for all but a $2q^{-n}$ fraction of $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and for parameter $\gamma q \geq \omega(\sqrt{\log m})$ (as required in part 4.2 for correctness), the distribution of $\mathbf{x}^T\mathbf{A} \bmod q$ for $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\gamma q}$ is statistically close to uniform over $\mathbb{Z}_q^n$.

4. Prove that the Dual-Regev encryption scheme is IND-CPA secure under the hardness of the LWE problem.

   Solution: We prove that the Dual-Regev encryption scheme is IND-CPA secure by reduction from the decisional LWE problem. We show that if there exists an adversary $\mathcal{A}$ that breaks the IND-CPA security of Dual-Regev with non-negligible advantage ε, then we can construct an algorithm $\mathcal{B}$ that solves the decisional LWE problem with the same advantage ε. The precise definition of IND-CPA and LWE protocol we give below.

IND-CPA

$\mathcal{C}$		$\mathcal{A}$
$b \leftarrow U(\{0,1\})$		
Generate $(\mathsf{pk}, \mathsf{sk})$		
	$\xrightarrow{\mathsf{pk}}$	
		Choose M_0, M_1
	$\xleftarrow{M_0, M_1}$	
$c \leftarrow \mathrm{Enc}(\mathsf{pk}, M_b)$		
	$\xrightarrow{\text{c}}$	
		Output b'
		$\mathcal{A}$ wins if $b = b'$

$$\mathsf{Adv}_{\mathcal{A}}^{\text{IND-CPA}} = \left|\Pr[\mathcal{A} \text{ wins}] - \frac{1}{2}\right|$$

LWE

$\mathcal{C}$		$\mathcal{A}$
$\mathbf{A} \leftarrow U(\mathbb{Z}_q^{m\times n})$		
RAND $(b=0)$: $\mathbf{b} \leftarrow U(\mathbb{Z}_q^m)$		
LWE $(b=1)$: $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$		
	$\xrightarrow{(\mathbf{A}, \mathbf{b})}$	
		output b'

$$\mathsf{Adv}_{\mathcal{A}}^{\text{lwe}} = \left|\Pr[\mathcal{A} \xrightarrow{\text{RAND}} 1] - \Pr[\mathcal{A} \xrightarrow{\text{LWE}} 1]\right|$$

Suppose there exists a PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of Dual-Regev with non-negligible advantage ε. We construct a PPT algorithm $\mathcal{B}$ that solves the decisional LWE problem with advantage ε. The exact algorithm can be seen below

Reduction Protocol

$\mathcal{C}$		$\mathcal{B}$		$\mathcal{A}$
$\tilde{\mathbf{A}} = (\mathbf{A}; \mathbf{u}^T) \leftarrow U(\mathbb{Z}_q^{(m+1)\times n})$				
RAND : $\tilde{\mathbf{b}} \leftarrow U(\mathbb{Z}_q^{m+1})$				
LWE : $\tilde{\mathbf{b}} = \tilde{\mathbf{A}}\mathbf{s} + \tilde{\mathbf{e}}$				
	$\xrightarrow{(\tilde{\mathbf{A}}, \tilde{\mathbf{b}})}$			
		Extract $\mathbf{A}, \mathbf{u}$ from $\tilde{\mathbf{A}}$		
			$\xrightarrow{(\mathbf{A}, \mathbf{u})}$	
				Choose M_0, M_1
			$\xleftarrow{M_0, M_1}$	
		$b \leftarrow U(\{0,1\})$		
		Extract $\mathbf{b}, (u^T s + e')$ from $\tilde{\mathbf{b}}$		
		$c = u^T s + e' + \lfloor q/2 \rceil \cdot M_b$		
			$\xrightarrow{(\mathbf{b}, c)}$	
				Output b'
			$\xleftarrow{b'}$	
		If $b = b'$ output 1		
		Else output 0		

The reduction $\mathcal{B}$ receives a decisional LWE challenge $(\tilde{\mathbf{A}}, \tilde{\mathbf{b}})$ where $\tilde{\mathbf{A}} \in \mathbb{Z}_q^{(m+1)\times n}$ is uniformly random, and $\tilde{\mathbf{b}} \in \mathbb{Z}_q^{m+1}$ is either:

- RAND case: $\tilde{\mathbf{b}} \leftarrow U(\mathbb{Z}_q^{m+1})$ (uniformly random)
- LWE case: $\tilde{\mathbf{b}} = \tilde{\mathbf{A}}\mathbf{s} + \tilde{\mathbf{e}}$ for some secret $\mathbf{s} \in \mathbb{Z}_q^n$ and error $\tilde{\mathbf{e}} \leftarrow D_{\mathbb{Z}^{m+1},\alpha q}$

$\mathcal{B}$ simulates the IND-CPA game for $\mathcal{A}$ as follows:

(a) **Key Generation:** $\mathcal{B}$ parses $\tilde{\mathbf{A}} = \begin{pmatrix} \mathbf{A} \\ \mathbf{u}^T \end{pmatrix}$ where $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{u} \in \mathbb{Z}_q^n$. It sends the public key $\mathsf{pk} = (\mathbf{A}, \mathbf{u})$ to $\mathcal{A}$.

(b) **Challenge:** $\mathcal{A}$ sends two messages $M_0, M_1 \in \{0, 1\}$. $\mathcal{B}$ chooses a random bit $b \leftarrow U(\{0, 1\})$.

(c) **Ciphertext Generation:** $\mathcal{B}$ parses $\tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{b} \\ v \end{pmatrix}$ where $\mathbf{b} \in \mathbb{Z}_q^m$ and $v \in \mathbb{Z}_q$. It computes:

$$c = v + \lfloor q/2 \rfloor \cdot M_b$$

and sends the ciphertext $(\mathbf{b}, c)$ to $\mathcal{A}$.

(d) **Output:** $\mathcal{A}$ outputs a bit b'. If $b = b'$, then $\mathcal{B}$ outputs 1 (guessing LWE); otherwise, it outputs 0 (guessing RAND).

Analysis:

Case 1: LWE instance. When $\tilde{\mathbf{b}} = \tilde{\mathbf{A}}\mathbf{s} + \tilde{\mathbf{e}}$, we have:

$$\tilde{\mathbf{b}} = \begin{pmatrix} \mathbf{A} \\ \mathbf{u}^T \end{pmatrix} \mathbf{s} + \begin{pmatrix} \mathbf{e} \\ e' \end{pmatrix} = \begin{pmatrix} \mathbf{A}\mathbf{s} + \mathbf{e} \\ \mathbf{u}^T\mathbf{s} + e' \end{pmatrix}$$

Therefore, $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ and $v = \mathbf{u}^T\mathbf{s} + e'$. The ciphertext is:

$$(\mathbf{b}, c) = (\mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{u}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M_b)$$

This is exactly a valid Dual-Regev encryption of M_b under public key $(\mathbf{A}, \mathbf{u})$ with randomness $\mathbf{s}$ and error terms $\mathbf{e}, e'$. Since $\mathbf{u}$ is uniformly random (as part of $\tilde{\mathbf{A}}$), by the result from Exercise 4.3, the public key distribution is statistically close to that of the real Dual-Regev scheme.

Therefore, $\mathcal{A}$ receives a perfect simulation of the IND-CPA game and outputs $b' = b$ with probability $\frac{1}{2} + \varepsilon$.

Case 2: RAND instance. When $\tilde{\mathbf{b}}$ is uniformly random, both $\mathbf{b}$ and v are uniformly random and independent. In particular, v is uniform over $\mathbb{Z}_q$, so:

$$c = v + \lfloor q/2 \rfloor \cdot M_b$$

is uniformly distributed over $\mathbb{Z}_q$ regardless of the value of M_b. The ciphertext reveals no information about b, so $\mathcal{A}$ can only guess randomly. Thus, $\Pr[b' = b] = \frac{1}{2}$.

Advantage Calculation:

$$\begin{aligned} \mathsf{Adv}_{\mathcal{B}}^{\text{lwe}} &= |\Pr[\mathcal{B} \to 1 \mid \text{LWE}] - \Pr[\mathcal{B} \to 1 \mid \text{RAND}]| \\ &= |\Pr[b' = b \mid \text{LWE}] - \Pr[b' = b \mid \text{RAND}]| \\ &= \left| \left(\frac{1}{2} + \varepsilon \right) - \frac{1}{2} \right| \\ &= \varepsilon \end{aligned}$$

Since ε is non-negligible by assumption, $\mathcal{B}$ solves the decisional LWE problem with non-negligible advantage, contradicting the hardness of LWE. Therefore, no such adversary $\mathcal{A}$ can exist, and the Dual-Regev encryption scheme is IND-CPA secure under the LWE assumption.

Exercise 5

Let $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ be a matrix specifying the q-ary lattice $\Lambda_q^{\perp}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\}$. You may assume throughout this problem that q is prime (but it is not a necessary hypothesis).

Note that $\mathbf{A}$ is the transpose of the matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ used during the lecture.

Note: The solutions to parts 1-3 follow closely the development in [Pei22], particularly the results on equivalent lattice representations and canonical basis construction for SIS lattices.

1. Describe an efficient algorithm that finds an n-by-n submatrix of $\mathbf{A}$ which is invertible over $\mathbb{Z}_q$ if one exists. (For uniformly random matrix $\mathbf{A}$ and typically used m, it can be shown that such a submatrix exists with high probability). Also argue that this invertible submatrix can be moved to the first n columns of $\mathbf{A}$, without essentially changing the lattice.

   Solution: To find an $n \times n$ invertible submatrix of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$:

   Algorithm:

   (a) Compute the reduced row echelon form (RREF) of $\mathbf{A}$ over $\mathbb{Z}_q$

   (b) Identify the pivot columns (columns with leading non-zero entries)

   (c) If there are at least n pivot columns, the first n pivot columns form an invertible $n \times n$ submatrix

   Since q is prime, $\mathbb{Z}_q$ is a field, so the pivot columns are linearly independent. An $n \times n$ matrix over a field is invertible if and only if its columns are linearly independent.

   To move this invertible submatrix to the first n columns, let the pivot columns have indices $\{i_1, \ldots, i_n\}$. Construct a permutation matrix $\mathbf{P}$ that moves these columns to positions $1, \ldots, n$. Then $\mathbf{A}' = \mathbf{A}\mathbf{P}$ has the form $[\mathbf{H}|\mathbf{B}]$ where $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ is the invertible submatrix and $\mathbf{B} \in \mathbb{Z}_q^{n\times(m-n)}$ contains the remaining columns.

   To show this doesn't essentially change the lattice, we state and prove the following lemma:

   Lemma 2 ([Pei22, Lemma 1.3])**.** *For any invertible matrix $\mathbf{T} \in \mathbb{Z}_q^{m\times m}$, we have*

$$\Lambda_q^{\perp}(\mathbf{A} \cdot \mathbf{T}) = \mathbf{T}^{-1} \cdot \Lambda_q^{\perp}(\mathbf{A})$$

Proof. We show both set containments.

($\subseteq$) Let $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A} \cdot \mathbf{T})$. Then $(\mathbf{A} \cdot \mathbf{T})\mathbf{x} = \mathbf{0} \pmod q$. Let $\mathbf{y} = \mathbf{T}\mathbf{x}$. Then

$$\mathbf{A}\mathbf{y} = \mathbf{A}(\mathbf{T}\mathbf{x}) = (\mathbf{A} \cdot \mathbf{T})\mathbf{x} = \mathbf{0} \pmod q$$

so $\mathbf{y} \in \Lambda_q^\perp(\mathbf{A})$. Since $\mathbf{x} = \mathbf{T}^{-1}\mathbf{y}$, we have $\mathbf{x} \in \mathbf{T}^{-1} \cdot \Lambda_q^\perp(\mathbf{A})$.

($\supseteq$) Let $\mathbf{x} \in \mathbf{T}^{-1} \cdot \Lambda_q^\perp(\mathbf{A})$. Then $\mathbf{x} = \mathbf{T}^{-1}\mathbf{y}$ for some $\mathbf{y} \in \Lambda_q^\perp(\mathbf{A})$. We have

$$(\mathbf{A} \cdot \mathbf{T})\mathbf{x} = (\mathbf{A} \cdot \mathbf{T})(\mathbf{T}^{-1}\mathbf{y}) = \mathbf{A}\mathbf{y} = \mathbf{0} \pmod q$$

so $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A} \cdot \mathbf{T})$. ■

For a permutation matrix $\mathbf{P}$ is invertible matrix. Therefore, $\Lambda_q^\perp(\mathbf{A}\mathbf{P}) = \mathbf{P}^{-1} \cdot \Lambda_q^\perp(\mathbf{A})$ is simply a coordinate permutation of $\Lambda_q^\perp(\mathbf{A})$, preserving all essential geometric properties like determinant and successive minima.

2. Prove that the invertible submatrix can be replaced by the identity matrix $\mathbf{I}_n$, possibly changing the rest of $\mathbf{A}$ as well, without changing the lattice.

Solution: Given $\mathbf{A} = [\mathbf{H}|\mathbf{A}']$ where $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ is invertible and $\mathbf{A}' \in \mathbb{Z}_q^{n\times(m-n)}$, we can transform it to $[\mathbf{I}_n|\tilde{\mathbf{A}}]$ without changing the lattice.

Lemma 3 ([Pei22, Lemma 1.2])**.** *Let* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ *be invertible. Then*

$$\Lambda_q^\perp(\mathbf{H} \cdot \mathbf{A}) = \Lambda_q^\perp(\mathbf{A})$$

Proof. ($\subseteq$) Let $\mathbf{x} \in \Lambda_q^\perp(\mathbf{H} \cdot \mathbf{A})$. Then $(\mathbf{H} \cdot \mathbf{A})\mathbf{x} = \mathbf{0} \pmod q$, which gives $\mathbf{H}(\mathbf{A}\mathbf{x}) = \mathbf{0} \pmod q$. Since $\mathbf{H}$ is invertible over $\mathbb{Z}_q$, multiplying both sides by $\mathbf{H}^{-1}$ yields $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$, so $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$.

($\supseteq$) Let $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$. Then $\mathbf{A}\mathbf{x} = \mathbf{0} \pmod q$. Therefore, $(\mathbf{H}\cdot\mathbf{A})\mathbf{x} = \mathbf{H}(\mathbf{A}\mathbf{x}) = \mathbf{H}\cdot\mathbf{0} = \mathbf{0} \pmod q$, so $\mathbf{x} \in \Lambda_q^\perp(\mathbf{H} \cdot \mathbf{A})$. ■

Using Lemma 3, we can left-multiply $\mathbf{A} = [\mathbf{H}|\mathbf{A}']$ by $\mathbf{H}^{-1}$ to obtain:

$$\mathbf{H}^{-1} \cdot \mathbf{A} = \mathbf{H}^{-1} \cdot [\mathbf{H}|\mathbf{A}'] = [\mathbf{H}^{-1}\mathbf{H}|\mathbf{H}^{-1}\mathbf{A}'] = [\mathbf{I}_n|\tilde{\mathbf{A}}]$$

where $\tilde{\mathbf{A}} = \mathbf{H}^{-1}\mathbf{A}' \in \mathbb{Z}_q^{n\times(m-n)}$.

By Lemma 3, we have:

$$\Lambda_q^\perp([\mathbf{I}_n|\tilde{\mathbf{A}}]) = \Lambda_q^\perp(\mathbf{H}^{-1} \cdot \mathbf{A}) = \Lambda_q^\perp(\mathbf{A})$$

Therefore, the lattice remains unchanged when we replace the invertible submatrix $\mathbf{H}$ with the identity matrix $\mathbf{I}_n$ (and update the remaining columns accordingly).

3. Using the previous parts, describe how to efficiently compute a basis of $\Lambda_q^\perp(\mathbf{A})$.

   Hint: if $\mathbf{A} = [\mathbf{I}_n|\tilde{\mathbf{A}}]$, then show that the n columns of $\begin{pmatrix} q\mathbf{I}_n \\ \mathbf{0} \end{pmatrix}$ are vectors in $\Lambda_q^\perp(\mathbf{A})$. Find $m - n$ more columns and prove that all m columns together form a basis $\mathbf{B}$ of $\Lambda_q^\perp(\mathbf{A})$, i.e. that $\mathbf{B} \cdot \mathbb{Z}^m = \Lambda_q^\perp(\mathbf{A})$.

   Solution: Following the canonical basis construction from [Pei22], we construct a basis for $\Lambda_q^\perp(\mathbf{A})$ when $\mathbf{A} = [\mathbf{I}_n|\tilde{\mathbf{A}}]$ where $\tilde{\mathbf{A}} \in \mathbb{Z}_q^{n\times(m-n)}$.

   Consider the following matrix:

$$\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\tilde{\mathbf{A}} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{pmatrix} \in \mathbb{Z}^{m\times m}$$

   where $-\tilde{\mathbf{A}}$ represents any integer matrix whose entries reduce to $-\tilde{\mathbf{A}} \pmod q$ (e.g., with entries in $\{0, 1, \ldots, q-1\}$).

   We verify that $\mathbf{B}$ is a basis of $\Lambda_q^\perp(\mathbf{A})$:

   1. Linear Independence: The matrix $\mathbf{B}$ is upper triangular with non-zero diagonal entries (q in the first n positions and 1 in the remaining $m - n$ positions), hence its columns are linearly independent.

   2. Columns belong to the lattice: For each column $\mathbf{b}_j$ of $\mathbf{B}$, we verify that $\mathbf{A}\mathbf{b}_j = \mathbf{0} \pmod q$:

   - For $j \le n$: The j-th column is $(0, \ldots, 0, q, 0, \ldots, 0)^T$ with q in position j.

$$[\mathbf{I}_n|\tilde{\mathbf{A}}] \cdot \mathbf{b}_j = q \cdot \mathbf{e}_j = \mathbf{0} \pmod q$$

   - For $j > n$: The j-th column has the form $(-\tilde{\mathbf{a}}_{j-n}, \mathbf{e}_{j-n})^T$ where $\tilde{\mathbf{a}}_{j-n}$ is the $(j-n)$-th column of $\tilde{\mathbf{A}}$.

$$[\mathbf{I}_n|\tilde{\mathbf{A}}] \cdot \mathbf{b}_j = -\tilde{\mathbf{a}}_{j-n} + \tilde{\mathbf{a}}_{j-n} = \mathbf{0} \pmod q$$

   Complete Algorithm:

   (a) Find an invertible $n \times n$ submatrix of $\mathbf{A}$ using RREF (part 1)

   (b) Use column permutation to move it to the first n columns: $\mathbf{A}' = \mathbf{A}\mathbf{P}$

   (c) Transform to systematic form: $[\mathbf{I}_n|\tilde{\mathbf{A}}] = \mathbf{H}^{-1}\mathbf{A}'$ (part 2)

   (d) Output the basis $\mathbf{B} = \begin{pmatrix} q\mathbf{I}_n & -\tilde{\mathbf{A}} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{pmatrix}$

   (e) Transform back: the basis for the original lattice is $\mathbf{P}\mathbf{B}$

4. Recall that the SIS problem is to find a short nonzero solution to $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$ for uniformly random $\mathbf{A}$. Using the previous parts, prove that the following problem is

at least as hard as SIS: given uniformly random $\mathbf{A}'$, find a short nonzero solution to $\mathbf{A}'\mathbf{z} = \mathbf{e} \bmod q$ where $\mathbf{e} \in \mathbb{Z}^n$ is any short vector of the attacker's choice.

Hint: the number of columns needed could not be the same in $\mathbf{A}$ and $\mathbf{A}'$.

Solution: We prove that the Inhomogeneous SIS (ISIS) problem is at least as hard as SIS by giving a reduction from SIS to ISIS.

ISIS Problem: Given uniformly random $\mathbf{A}' \in \mathbb{Z}_q^{n\times m'}$, find a short nonzero $\mathbf{z}' \in \mathbb{Z}^{m'}$ such that $\mathbf{A}'\mathbf{z}' = \mathbf{e} \pmod q$ where $\mathbf{e} \in \mathbb{Z}^n$ is any short vector of the attacker's choice, and $\|\mathbf{z}'\| \leq \beta'$.

Reduction: Given a SIS instance with uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and bound β, we construct an algorithm that uses an ISIS solver to find a short nonzero $\mathbf{z}$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \pmod q$ and $\|\mathbf{z}\| \leq \beta$.

(a) **Partition the matrix:** Choose some $m' < m$ and partition $\mathbf{A} = [\mathbf{A}_1|\mathbf{A}_2]$ where $\mathbf{A}_1 \in \mathbb{Z}_q^{n\times m'}$ and $\mathbf{A}_2 \in \mathbb{Z}_q^{n\times(m-m')}$. Since $\mathbf{A}$ is uniformly random, both $\mathbf{A}_1$ and $\mathbf{A}_2$ are uniformly random over their respective domains.

(b) **Sample a short vector:** Sample a random short vector $\mathbf{z}_2 \in \mathbb{Z}^{m-m'}$ with $\|\mathbf{z}_2\| \leq \beta_2$ for some parameter $\beta_2 > 0$.

(c) **Compute target vector:** Compute $\mathbf{e} = -\mathbf{A}_2\mathbf{z}_2 \pmod q$.

(d) **Call ISIS solver:** Use the ISIS solver on instance $(\mathbf{A}_1, \mathbf{e})$ to find $\mathbf{z}_1 \in \mathbb{Z}^{m'}$ such that $\mathbf{A}_1\mathbf{z}_1 = \mathbf{e} \pmod q$ and $\|\mathbf{z}_1\| \leq \beta_1$ for some parameter $\beta_1 > 0$.

(e) **Construct SIS solution:** Output $\mathbf{z} = \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \in \mathbb{Z}^m$.

Correctness: We verify that $\mathbf{z}$ is a valid SIS solution:

$$\begin{aligned}
\mathbf{A}\mathbf{z} &= [\mathbf{A}_1|\mathbf{A}_2] \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \\
&= \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 \\
&= \mathbf{e} + \mathbf{A}_2\mathbf{z}_2 \\
&= -\mathbf{A}_2\mathbf{z}_2 + \mathbf{A}_2\mathbf{z}_2 \\
&= \mathbf{0} \pmod q
\end{aligned}$$

For the norm bound, we have:

$$\|\mathbf{z}\| = \left\| \begin{pmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{pmatrix} \right\| = \sqrt{\|\mathbf{z}_1\|^2 + \|\mathbf{z}_2\|^2} \leq \sqrt{\beta_1^2 + \beta_2^2}$$

To ensure $\|\mathbf{z}\| \leq \beta$, we need to choose β_1 and β_2 such that:

$$\beta_1^2 + \beta_2^2 \leq \beta^2$$

This reduction shows that if we can efficiently solve ISIS with bound β_1 (finding short solutions to inhomogeneous systems), then we can efficiently solve SIS with bound β. Therefore, ISIS is at least as hard as SIS.

References

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, pages 197–206, New York, NY, USA, 2008. ACM.

[Kan83] Ravi Kannan. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, STOC '83, pages 99–108, New York, NY, USA, 1983. ACM.

[Pei22] Chris Peikert. Lattices in cryptography: Lecture 12 - sis lattices & applications. University of Michigan, Fall 2022, 2022. Course lecture notes. Scribe: Jacob Alperin-Sheriff.