In this lecture, we introduce the (average-case) *learning with errors* (LWE) problem of Regev [Reg05], recall its hardness based on worst-case lattice problems, and describe a simple public-key cryptosystem whose security can be proved based on the hardness of LWE. Since its introduction in 2005, LWE has served as the foundation for a huge variety of rich cryptographic constructions—several of which we still only know how to obtain from LWE-like problems.

1 History

The first public-key encryption scheme with a security proof based on a worst-case hardness assumption was given by Ajtai and Dwork in 1997 [AD97]. The security of their scheme is based on the conjectured (worst-case) hardness of the *unique-SVP* problem, which is defined roughly as follows: given a lattice whose shortest vector is "unique" by a significant factor—i.e., the minimum distance λ_1 is significantly smaller than the second successive minimum λ_2—find a shortest nonzero lattice vector. One way to construct such a lattice is to start with a lattice that has a large minimum distance λ_1, then augment it with a much shorter vector. While the Ajtai–Dwork encryption scheme is conceptually quite natural, it is very inefficient, and its full security proof is quite complex.

The next major encryption scheme with a worst-case hardness proof was developed by Regev in 2005 [Reg05]. This work introduced a clean and simple average-case problem called *learning with errors* (LWE), and proved that LWE is hard if certain lattice problems are hard in the worst case for *quantum* algorithms. (The quantum hypothesis was relaxed to a classical one in [Pei09], at the expense of worse parameters or a non-standard worst-case hardness assumption.) Moreover, Regev gave an encryption scheme whose security can be proved based on the assumption that LWE is hard. Regev's encryption scheme is more efficient than Ajtai–Dwork, and LWE itself is also substantially simpler and more versatile (as evidenced by the great number of other applications it has enabled). Interestingly, though, the average-case problems underlying the Ajtai–Dwork and Regev cryptosystems actually turn out to be *equivalent* (though with a significant loss in parameters) [Reg10].

In the past 15 years or so, many rich cryptographic constructions, such as identity-based encryption, attribute-based encryption, fully homomorphic encryption, and secure computation protocols, have been built on LWE.

2 Learning With Errors Problem

Learning With Errors is closely related to the SIS problem, which we have seen in previous lectures. Recall that the SIS problem is: given uniformly random vectors

$$\begin{pmatrix} \vdots \\ \mathbf{a}_1 \\ \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ \mathbf{a}_2 \\ \vdots \end{pmatrix}, \cdots, \begin{pmatrix} \vdots \\ \mathbf{a}_m \\ \vdots \end{pmatrix} \in \mathbb{Z}_q^n$$

defining the matrix $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{Z}_q^{n \times m}$, find a "short" nonzero vector $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0}$.

LWE is a kind of "dual" to the SIS problem, involving a "noisy" random linear system defined by $\mathbf{A}$. It comes in two versions: *search*, where the goal is to solve the system, and *decision*, where the goal is to distinguish such a system from a completely random one (that likely has no solution). We consider each one in turn.

2.1 Search Version

In the search version of LWE, a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$ is first chosen (either uniformly at random, or from some other specified distribution). The goal is then to find this vector $\mathbf{s}$, given uniformly random $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{Z}_q^n$ as above, along with "noisy" (mod-q) inner products

$$\begin{aligned} b_1 &= \langle \mathbf{s}, \mathbf{a}_1 \rangle + e_1 \in \mathbb{Z}_q, \\ b_2 &= \langle \mathbf{s}, \mathbf{a}_2 \rangle + e_2 \in \mathbb{Z}_q, \\ &\vdots \\ b_m &= \langle \mathbf{s}, \mathbf{a}_m \rangle + e_m \in \mathbb{Z}_q, \end{aligned}$$

where the error terms e_i are chosen independently from some specified error distribution. Stated more compactly using matrix notation, the goal is to find $\mathbf{s}$ given

$$\begin{aligned} \mathbf{A} &= (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathbb{Z}_q^{n \times m}, \\ \mathbf{b}^t &= \mathbf{s}^t \mathbf{A} + \mathbf{e}^t. \end{aligned}$$

Definition 2.1 (Learning With Errors, Search). For a positive integer modulus q, dimensions n, m, and an error distribution χ over $\mathbb{Z}$ (or, equivalently $\mathbb{Z}_q$), the $\mathsf{LWE}_{n,q,\chi,m}$ problem is defined as follows: given uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t \in \mathbb{Z}_q^{1 \times m}$ where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is uniformly random and $\mathbf{e} \leftarrow \chi^m$, find $\mathbf{s}$.

We sometimes drop the m subscript when the number of noisy inner products is effectively unlimited, i.e., when m can be chosen by the adversary (subject to the adversary's own running time).

Notice that without the error, LWE is easy: if the columns of $\mathbf{A}$ generate all of $\mathbb{Z}_q^n$ (which is highly likely for large enough m), we can simply use Gaussian elimination to find a right-inverse $\mathbf{A}^- \in \mathbb{Z}_q^{m \times n}$ of $\mathbf{A}$, then compute $\mathbf{s}^t = \mathbf{b}^t \mathbf{A}^-$. But with the error in place, this strategy does not work, because the $\mathbf{e}^t \mathbf{A}^-$ term conceals $\mathbf{s}$.

Error distribution. A typical choice of the error distribution χ, and one for which the worst-case-hardness theorems apply (see Theorem 2.2 below), is a *discrete Gaussian* $D_{\mathbb{Z},r}$ where $r = \alpha q \geq \sqrt{n}$ for some $\alpha < 1$. Recall that $D_{\mathbb{Z},r}$ is a Gaussian of width r restricted to the integers, which assigns probability proportional to $\exp(-\pi z^2/r^2)$ to each $z \in \mathbb{Z}$. We can think of $D_{\mathbb{Z},r}$ as being concentrated on an interval of width about r (centered at zero), which covers about an α-fraction of $\mathbb{Z}_q$, so α can be thought of as the *error rate* relative to q. Applications frequently take a sufficiently small $\alpha = 1/\operatorname{poly}(n)$, although smaller error rates are sometimes used as well.

The following is a central result from [Reg05]. Notice that the approximation factor for the worst-case lattice problem varies inversely with α, i.e., the smaller the error rate, the weaker the hardness guarantee.

Theorem 2.2. *For $\chi = D_{\mathbb{Z},\alpha q}$ where $\alpha q \geq \sqrt{n}$ and any $m = \operatorname{poly}(n)$, the search-$\mathsf{LWE}_{n,q,\chi,m}$ problem is hard on the average (even for quantum algorithms) if $\mathsf{SIVP}_{\tilde{O}(n/\alpha)}$ (among others) on n-dimensional lattices is hard in the worst case for* quantum *algorithms.*

Real-world practical constructions sometimes use an error distribution that is narrower and/or simpler to sample from, e.g., a (narrow) binomial distribution or a uniform distribution over an interval. While LWE still appears to be hard for such error distributions (and sensibly chosen other parameters), what we have been able to prove is much more limited.

2.2 Decision Version

In the decision version of LWE, the goal is more modest: to *distinguish* an LWE instance from a uniformly random one. That is, given $(\mathbf{A}, \mathbf{b})$ that is sampled either according to Definition 2.1 or uniformly at random (with no secret $\mathbf{s}$ or error $\mathbf{e}$), determine which is the case.

More formally, a distinguisher $\mathcal{D}$ solves the decision version if

$$\Pr_{\mathbf{s},\mathbf{A},\mathbf{e}}[\mathcal{D}(\mathbf{A}, \mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t) \text{ accepts}] - \Pr_{\text{uniform } \mathbf{A},\mathbf{b}}[\mathcal{D}(\mathbf{A}, \mathbf{b}^t) \text{ accepts}] \geq \frac{1}{\text{poly}(n)}.$$

Later, we will see that, perhaps surprisingly, the decision version is actually *no easier than* the search version (under mild conditions on the parameters).

2.3 LWE as a Lattice Problem

Analogously to how SIS can be seen as an approximate shortest vector problem on a certain class of random lattices, LWE can also be seen as a certain lattice problem, called *bounded-distance decoding*, on the *duals* of these SIS lattices.

Recall that we previously defined the q-ary SIS lattices $\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{0}\}$. We can similarly define the q-ary LWE lattices as follows:

$$\mathcal{L}(\mathbf{A}) = \{\mathbf{w} \in \mathbb{Z}^m : \mathbf{w}^t = \mathbf{s}^t\mathbf{A} \pmod q \text{ for some } \mathbf{s} \in \mathbb{Z}_q^n\} = \mathbf{A}^t \cdot \mathbb{Z}_q^n + q\mathbb{Z}^m.$$

Essentially, this lattice contains any $\mathbf{w}$ corresponding to an LWE vector *without* any error. We can then see LWE as a "noisy decoding" problem on such a lattice.

In the search version, we are given a random lattice $\mathcal{L}(\mathbf{A})$ and a point $\mathbf{b} \in \mathbb{Z}_q^m$ that is generated by adding some relatively small noise to a lattice point, and the goal is to find that lattice point. This is similar to the Closest Vector Problem (CVP), except that here the target $\mathbf{b}$ is guaranteed to be close to the lattice, whereas in CVP it can be arbitrary. In the decision version, the target $\mathbf{b}$ is either a noisy lattice point or uniformly random and independent of $\mathbf{A}$, and the goal is to determine which is the case.

Finally, observe that $\mathcal{L}(\mathbf{A})$ and $\mathcal{L}^\perp(\mathbf{A})$ are dual lattices up to scaling by q, i.e., $\mathcal{L}^\perp(\mathbf{A})^* = q^{-1}\mathcal{L}(\mathbf{A})$. This can be seen by showing containment in both directions:

- First, for any $\mathbf{w} \in \mathcal{L}(\mathbf{A})$ we have $\mathbf{w}^t = \mathbf{s}^t\mathbf{A} \pmod q$ for some $\mathbf{s} \in \mathbb{Z}_q^n$, so for all $\mathbf{z} \in \mathcal{L}^\perp(\mathbf{A})$ we have $\langle \mathbf{w}, \mathbf{z} \rangle = \mathbf{s}^t\mathbf{A}\mathbf{z} = \mathbf{s}^t \cdot \mathbf{0} = 0 \pmod q$, hence $q^{-1}\mathcal{L}(\mathbf{A}) \subseteq \mathcal{L}^\perp(\mathbf{A})^*$.
- Second, for any $\mathbf{z} \in (q^{-1}\mathcal{L}(\mathbf{A}))^* = q\mathcal{L}(\mathbf{A})^*$, the inner product of $\mathbf{z}$ with each row vector of $\mathbf{A}$ is in $q\mathbb{Z}$ by definition, so $\mathbf{A}\mathbf{z} = \mathbf{0} \in \mathbb{Z}_q^n$. Hence, $(q^{-1}\mathcal{L}(\mathbf{A}))^* \subseteq \mathcal{L}^\perp(\mathbf{A})$, so $\mathcal{L}^\perp(\mathbf{A})^* \subseteq q^{-1}\mathcal{L}(\mathbf{A})$.

2.4 LWE as a Knapsack Problem

We consider one final interpretation of LWE, as a knapsack-type problem. For secret $\mathbf{s} \in \mathbb{Z}^n$ and error vector $\mathbf{e} \in \mathbb{Z}^m$ (both chosen from the error distribution; see "normal form" below), an LWE instance $(\mathbf{A}, \mathbf{b})$ can be seen as the following knapsack:

$$\mathbf{b}^t = (\mathbf{s}^t, \mathbf{e}^t) \cdot \begin{pmatrix} \mathbf{A} \\ \mathbf{I}_m \end{pmatrix}.$$

We view the rows of the matrix as the knapsack weights, and the vector $(\mathbf{s}^t, \mathbf{e}^t)$ as the selections (which are small integers, not necessarily binary). Recall that the hardness of a knapsack problem is largely controlled

by its density, which we can quantify here as $\frac{\text{input length}}{\text{output length}} = \frac{(n+m)\log(\alpha q)}{m\log(q)}$. Typically, m is significantly larger than n, so the density is $\approx \frac{m\log(\alpha q)}{m\log(q)} = \frac{\log(\alpha q)}{\log(q)} < 1$. Note that since the density is less than one, the output length is larger than the input length, i.e., LWE "expands" the secret $\mathbf{s}, \mathbf{e}$ into a larger pseudorandom output. (By contrast, recall that for typical parameters SIS "compresses" the input $\mathbf{x}$ to $\mathbf{A}\mathbf{x}$, like a hash function).

Let us consider some examples of how the LWE parameters affect the density and hardness. For the typical choice of $q = \text{poly}(n)$ and $\alpha q \geq \sqrt{n}$, the density is some constant in $(0, 1)$, which is believed to be hard. But for the extreme case $q = 2^n, \alpha q = \text{poly}(n)$, the density is $\Theta((\log n)/n)$, and the problem can be solved efficiently by the previously covered Lagarias–Odlyzko/Frieze attack (with somewhat tighter analysis). Interestingly, for these parameters the worst-case hardness result in Theorem 2.2 becomes vacuous, because it relies on lattice problems with approximation factors $\approx 1/\alpha \approx 2^n$, which are easy.

3 Properties of LWE

We now recall some simple, yet very powerful, properties of LWE that are especially relevant to cryptographic applications.

Confirming a candidate solution. It is possible to efficiently confirm a candidate solution $\mathbf{s}'$ to an LWE instance: simply test if the values $b_i - \langle \mathbf{s}', \mathbf{a}_i \rangle$ are sufficiently small, i.e., if $\mathbf{b}^t - (\mathbf{s}')^t\mathbf{A}$ is sufficiently short. When $\mathbf{s}' = \mathbf{s}$, this value will be small. When $\mathbf{s} \neq \mathbf{s}'$, then the value $\langle \mathbf{s} - \mathbf{s}', \mathbf{a} \rangle$ is "well-spread" enough to distinguish it from an LWE error vector.

Shifting the secret. We can "shift" the underlying secret $\mathbf{s}$ by any known vector $\mathbf{t} \in \mathbb{Z}_q^n$, using the following transformation on LWE samples:

$$(\mathbf{a}_i, b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i) \mapsto (\mathbf{a}_i, b_i' = b_i + \langle \mathbf{t}, \mathbf{a}_i \rangle = \langle \mathbf{s} + \mathbf{t}, \mathbf{a}_i \rangle + e_i).$$

We can use this property to "amplify" the success probability in attacking (search or decision) LWE. For example, given some distinguisher $\mathcal{D}$ that has $1/\text{poly}(n)$ advantage, we can construct a distinguisher that is essentially perfect. It calls $\mathcal{D}$ many times on additional groups of LWE samples, each with a re-randomized secret, so that each call to $\mathcal{D}$ is on a completely fresh, independent LWE instance. The distinguisher then outputs the majority decision of all the calls to $\mathcal{D}$. By independence and standard Chernoff bounds, this majority decision is correct with high probability.

(Hermite) normal form. LWE becomes no easier if we use a "short" secret whose entries are drawn from the error distribution χ, rather than uniformly at random (or any other distribution). The proof is an extension of the transformation from the previous lecture that put SIS in (Hermite) normal form, and also keeps track of the effect on the LWE secret.

Theorem 3.1. *LWE is no easier when* $\mathbf{s}$ *is drawn from the error distribution* χ^n*, versus any other distribution.*

Proof. We prove this theorem by reduction. For concreteness we consider the search-LWE problem; essentially the same argument works for the decision version. Suppose we have some inverter $\mathcal{I}$ that solves search-LWE when the secret is drawn from χ^n. We need to show that we can use $\mathcal{I}$ to efficiently solve search-LWE for any distribution of secret $\mathbf{s}$. The reduction is as follows:

1. Take LWE samples to get $(\bar{\mathbf{A}}, \bar{\mathbf{b}}^t = \mathbf{s}^t\bar{\mathbf{A}} + \bar{\mathbf{e}}^t)$ where $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times n}$ is square and invertible. With enough attempts this will happen with high probability.

2. Transform all subsequent samples as:

$$(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a}\rangle + e) \mapsto (\mathbf{a}' = -\bar{\mathbf{A}}^{-1}\mathbf{a}, b' = \langle \bar{\mathbf{b}}, \mathbf{a}'\rangle + b).$$

   Observe that

$$b' = (\mathbf{s}^t\bar{\mathbf{A}} + \bar{\mathbf{e}}^t) \cdot (-\bar{\mathbf{A}}^{-1} \cdot \mathbf{a}) + \langle \mathbf{s}, \mathbf{a}\rangle + e = \langle \bar{\mathbf{e}}, \mathbf{a}'\rangle + e,$$

   so the transform produces properly distributed LWE samples with secret $\bar{\mathbf{e}}$, which is distributed according to χ^n. (Note that $\mathbf{a}'$ is uniformly random, as needed, because $\mathbf{a}$ is uniformly random and $\bar{\mathbf{A}}$ is invertible.)

3. Give the transformed samples to $\mathcal{I}$, which will output $\bar{\mathbf{e}}$ by hypothesis. Then solve for $\mathbf{s}$ as $\mathbf{s}^t = (\bar{\mathbf{b}}^t - \bar{\mathbf{e}}^t) \cdot \bar{\mathbf{A}}^{-1}$. □

4 Search Versus Decision

It is easy to see that for any sensible parameters, the decision version of LWE reduces to the search version. Specifically, given an oracle that solves search-LWE, we can easily solve decision as follows: given $(\mathbf{A}, \mathbf{b}^t)$, query the oracle to get a candidate solution $\mathbf{s} \in \mathbb{Z}_q^n$. Test whether $\mathbf{b}^t - \mathbf{s}^t\mathbf{A}$ is "short enough" (relative to the error distribution), accepting if so and rejecting otherwise.

This distinguisher has good advantage, because when the input is an LWE instance, the oracle returns the underlying secret, so $\mathbf{b}^t - \mathbf{s}^t\mathbf{A}$ is distributed according to the error distribution. On the other hand, when the input is uniformly random, it can be shown that with high probability there *does not exist* any $\mathbf{s}$ for which $\mathbf{b}^t - \mathbf{s}^t\mathbf{A}$ is sufficiently short, so the distinguisher must reject (no matter what the oracle returns to it).

We will now show that the converse reduction also holds: given an oracle that solves decision-LWE, it is possible to efficiently solve search-LWE.

Theorem 4.1. *For any prime $q = \mathrm{poly}(n)$, search-LWE reduce to decision-LWE.*

We note that the two hypotheses on q (primality, and being polynomially bounded) have been significantly relaxed using additional ideas, e.g., in [Pei09].

Proof. Let $\mathcal{D}$ be an oracle that solves decision-LWE. By the amplification property, we can assume that $\mathcal{D}$ is essentially perfect, i.e., it returns the correct answer with advantage $1 - 2^{-n}$.

We use $\mathcal{D}$ to solve search-LWE via a reduction. Without loss of generality, it is enough to show how to find the first entry $s_1 \in \mathbb{Z}_q$ of the secret. In turn, to do this it is enough to show how to reliably test whether or not $s_1 = 0$. This is because we can shift s_1 by $0, 1, 2, \ldots, q-1 = \mathrm{poly}(n)$ and zero-test until we find the correct value.

To perform the test, the reduction takes input samples $(\mathbf{a}_i, b_i)$, and for each one chooses a fresh uniformly random $r_i \leftarrow \mathbb{Z}_q$, producing a transformed sample $(\mathbf{a}'_i = \mathbf{a}_i - (r, 0, \ldots, 0), b_i)$. It calls $\mathcal{D}$ on these modified samples; if $\mathcal{D}$ accepts, the reduction concludes that $\mathbf{s}_1 = 0$; otherwise it concludes that $\mathbf{s}_1 \neq 0$.

The above strategy works because when $s_1 = 0$,

$$b_i = \langle \mathbf{s}, \mathbf{a}_i\rangle + e_i = \langle \mathbf{s}, \mathbf{a}'_i\rangle + e_i,$$

so $(\mathbf{a}'_i, b_i)$ is a properly distributed LWE sample (with secret $\mathbf{s}$), hence $\mathcal{D}$ accepts with high probability. On the other hand, when $s_1 \neq 0$, we have

$$b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i = \langle \mathbf{s}, \mathbf{a}' \rangle + s_1 \cdot r_i + e_i.$$

Since $s_1 \cdot r_i \in \mathbb{Z}_q$ is uniformly random (and fresh for each sample) because q is prime, the transformed samples $(\mathbf{a}'_i, b_i)$ are uniformly random, so $\mathcal{D}$ rejects with high probability. □

5 LWE-Based Public-Key Encryption

A public-key encryption scheme consists of three algorithms:

- $\mathsf{Gen}()$ outputs a public encryption key pk and a private decryption key sk.
- $\mathsf{Enc}(pk, \mu)$, given a public key and a message $\mu \in \{0, 1\}$, outputs a ciphertext c.
- $\mathsf{Dec}(sk, c)$, given a secret key and a ciphertext c, outputs a message $\mu \in \{0, 1\}$.

For correctness, we require that for any $\mu \in \{0, 1\}$, if $(pk, sk) \leftarrow \mathsf{Gen}()$ and $c \leftarrow \mathsf{Enc}(pk, \mu)$, then $\mathsf{Dec}(sk, c)$ outputs μ (either with certainty, or with high probability if such is acceptable).

For security, we desire "semantic security," which informally says that no efficient adversary, given a public key, can distinguish an encryption of zero from an encryption of one. More precisely, $(pk, \mathsf{Enc}(pk, 0))$ and $(pk, \mathsf{Enc}(pk, 1))$ should be indistinguishable, where in both cases $(pk, sk) \leftarrow \mathsf{Gen}()$.

5.1 Cryptosystem

Here we define the three algorithms of an LWE-based encryption scheme. (This scheme is a variant of the one originally given in [Reg05], and is essentially "dual" to it in its key properties.) It is parameterized by LWE parameters n, q, χ, m, which we instantiate below in the analysis.

- $\mathsf{Gen}()$: choose $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{x} \leftarrow \{0, 1\}^m$ uniformly at random, and let $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{Z}_q^n$. Output secret key $sk = \mathbf{x}$ and public key $pk = (\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times (m+1)}$.

  (Note that this creates an SIS instance with a known short solution, as described in the previous lecture.)

- $\mathsf{Enc}((\mathbf{A}, \mathbf{y}), \mu)$: choose $\mathbf{s} \leftarrow \chi^n$ and $\mathbf{e} \leftarrow \chi^{m+1}$, and output the ciphertext

$$\mathbf{c}^t = \mathbf{s}^t(\mathbf{A}, \mathbf{y}) + \mathbf{e}^t + (0, \ldots, 0, \mu \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^{1 \times (m+1)}.$$

- $\mathsf{Dec}(\mathbf{x}, \mathbf{c})$: Compute

$$d = \mathbf{c}^t \cdot \begin{pmatrix} -\mathbf{x} \\ 1 \end{pmatrix} \in \mathbb{Z}_q$$

  and output 0 if $d \in [-q/4, q/4] \pmod q$, otherwise output 1. (In other words, output 0 if d is closer modulo q to 0 than to $q/2$.)

Observe that, in encryption, the (scaled) message is included in, and concealed by, the last entry of an LWE vector relative to the public key $(\mathbf{A}, \mathbf{y})$. In decryption, the private key $\mathbf{x}$ is used to nearly annihilate this LWE vector, leaving behind just the (scaled) message plus some residual error. (This error is why we scale the message upon encryption, so that it can be recovered during decryption.)

5.2 Correctness

Lemma 5.1. *Letting $\chi = D_{\mathbb{Z},r}$ for some $r > 0$ and $q \geq r\sqrt{m+1} \cdot \log n$, the cryptosystem is correct with high probability $1 - n^{-\Omega(\log n)}$.*

Notice that the error rate $\alpha = r/q \lesssim 1/\sqrt{m} \lesssim 1/\sqrt{n}$, ignoring log factors. And recall that in order to get a worst-case hardness guarantee, we need to take $r \geq \sqrt{n}$, which requires that $q \gtrsim n$.

Proof. By construction, the secret key $\mathbf{x} \in \mathbb{Z}^m$ and public key $(\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times (m+1)}$ satisfy the relation

$$(\mathbf{A}, \mathbf{y}) \cdot \mathbf{x}' = \mathbf{0} \in \mathbb{Z}_q^n,$$

where $\mathbf{x} = \binom{-\mathbf{x}}{1} \in \mathbb{Z}^{m+1}$. Now, consider decryption of a ciphertext $\mathbf{c}^t \leftarrow \mathsf{Enc}((\mathbf{A}, \mathbf{y}), \mu)$. It computes

$$d = \left(\mathbf{s}^t(\mathbf{A}, \mathbf{y}) + \mathbf{e}^t + (\mathbf{0}, \mu \cdot \lfloor q/2 \rfloor)\right) \cdot \mathbf{x}' = \mathbf{s}^t \cdot \mathbf{0} + \mathbf{e}^t \cdot \mathbf{x}' + \mu \cdot \lfloor q/2 \rfloor,$$

so it suffices to show that $\langle \mathbf{e}, \mathbf{x}' \rangle \in \mathbb{Z}$ has magnitude less than $\lfloor q/4 \rfloor$ with high probability. By construction, $\|\mathbf{x}'\| \leq \sqrt{m+1}$, and $\mathbf{e}$ is drawn from $\chi^{m+1} = D_{\mathbb{Z}^{m+1},r}$.

By previously shown tail inequalities, we have $|\langle \mathbf{e}, \mathbf{x}' \rangle| \leq tr\sqrt{m+1}$ except with probability at most $2\exp(-\pi t^2)$, for any $t > 0$. The claim follows by taking a suitable $t = \Theta(\log n)$. □

5.3 Security

Lemma 5.2. *For any $m \geq (1+\delta)n \log_2 q$ (where $\delta > 0$ is any constant), the cryptosystem is semantically secure if decision-$\mathsf{LWE}_{n,q,\chi,m+1}$ is hard.*

Proof. We need to show that a public key together with an encryption of zero is indistinguishable from a public key together with an encryption of one. First, consider the public key $pk = \mathbf{A}' = (\mathbf{A}, \mathbf{y}) \in \mathbb{Z}_q^{n \times (m+1)}$. By regularity (see the previous lecture), its distribution is exponentially close to uniform. So we can treat $\mathbf{A}'$ as if it were exactly uniform, up to a negligible difference in distinguisher advantage.

Now, consider the ciphertext $\mathbf{c}^t = \mathbf{s}^t\mathbf{A}' + \mathbf{e}^t + \mu \cdot \lfloor q/2 \rfloor$. Ignoring the $\mu \cdot \lfloor q/2 \rfloor$ term, the remainder $\mathbf{b}^t = \mathbf{s}^t\mathbf{A}' + \mathbf{e}^t$ is distributed exactly as an LWE vector relative to $\mathbf{A}'$. So, by the assumed hardness of decision-LWE, $(\mathbf{A}', \mathbf{b})$ is computationally indistinguishable from uniformly random. Thus, for any fixed message bit μ, the public key-ciphertext pair $(pk = \mathbf{A}', \mathbf{c} \leftarrow \mathsf{Enc}(pk, \mu))$ is computationally indistinguishable from uniformly random, hence the two distributions for $\mu = 0, 1$ are computationally indistinguishable from each other, as needed. □

References

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293. 1997. Page 1.

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. 2009. Pages 1 and 5.

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):1–40, 2009. Preliminary version in STOC 2005. Pages 1, 2, and 6.

[Reg10] O. Regev. The learning with errors problem (invited survey). In *IEEE Conference on Computational Complexity*, pages 191–204. 2010. Page 1.