The Learning with Errors Problem: Introduction and Basic Cryptography

The learning with errors (LWE) problem was introduced in its current form in a seminal work of Oded Regev for which he won the Gödel prize in 2018. In its typical form, the LWE problem asks to solve a system of noisy linear equations. That is, it asks to find $\mathbf{s} \in \mathbb{Z}_q^n$ given

$$\left\{(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) : \ \mathbf{s} \leftarrow \mathbb{Z}_q^n, \mathbf{a}_i \leftarrow \mathbb{Z}_q^n, e_i \leftarrow \chi\right\}_{i=1}^m \tag{1}$$

where:

- $\mathbb{Z}_q$ denotes the finite ring of integers modulo q, $\mathbb{Z}_q^n$ denotes the vector space of dimension n over $\mathbb{Z}_q$;
- χ is a probability distribution over $\mathbb{Z}$; and
- $a \leftarrow \mathcal{D}$ denotes that a is chosen according to the finite probability distribution $\mathcal{D}$, $a \leftarrow S$ denotes that a is chosen uniformly at random from the (finite) set S.

In this first lecture, we will present various perspectives on the LWE (and the closely related "short integer solutions" or SIS) problem, basic theorems regarding the different variants of these problems, their basic cryptographic applications, and the asymptotically best known algorithms for them.

We will shortly derive LWE in a different way, "from first principles", starting from a different view, that of finding special solutions to systems of linear equations.

1 Solving Systems of Linear Equations

Consider the problem of solving a system of linear equations

$$\mathbf{A}\mathbf{e} = \mathbf{b} \bmod q \tag{2}$$

where $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{b} \in \mathbb{Z}_q^n$ and $\mathbf{e}$ are the variables. This can be accomplished in polynomial time with Gaussian elimination. However, slight variations of this problem become hard for Gaussian elimination and indeed, we believe, for all polynomial-time algorithms. This course is concerned with two such problems, very related to each other, called the SIS problem and the LWE problem.

1.1 The Random Regime and SIS

Assume that we now ask for solutions to equation 2 where $\mathbf{e}$ lies in some subset $S \subseteq \mathbb{Z}_q^m$. Typically we will think of subsets S that are defined geometrically, that is either $S = [-B \dots B]^m$ is the set of all solutions where each coordinate can only take a bounded value (absolute value bounded by some number $B \ll q$) or $S = \mathsf{Ball}_R^2$, the Euclidean ball of radius R. That is, we are asking for short solutions to systems of linear equations and hence this is called the SIS (short integer solutions) problem.

The SIS problem $\mathsf{SIS}(n, m, q, B)$ as we will study is parameterized by the number of variables m, the number of equations n, the ambient finite field $\mathbb{Z}_q$, and the bound on the absolute value of the solutions B. Namely, we require that each coordinate $e_i \in [-B, -B+1, \dots, B-1, B]$.

To define an average-case problem, we need to specify the probability distributions for $\mathbf{A}$ and $\mathbf{b}$. We will, for the most part of this course, take $\mathbf{A}$ to be uniformly random in $\mathbb{Z}_q^{n\times m}$. There are two distinct ways to define $\mathbf{b}$. The first is in the random regime where we simply choose $\mathbf{b}$ from the uniform distribution over $\mathbb{Z}_q^n$.

Here, using a simple probabilistic argument, one can show that (B-bounded) solutions are very likely to exist if $(2B+1)^m \gg q^n$, or $m = \Omega(\frac{n\log q}{\log B})$. *We call this the random regime or the SIS regime.* Thus, roughly speaking, in the SIS regime, m is large enough that we are guaranteed solutions (even exponentially many of them) when $\mathbf{A}$ and $\mathbf{b}$ are chosen to be uniformly random. The problem then is to actually find a solution.

1.2 The Planted Regime and LWE

When $m \ll \frac{n\log q}{\log B}$, one can show again that there are likely to be no B-bounded solutions for a uniformly random $\mathbf{b}$ and thus, we have to find a different, sensible, way to state this problem. To do this, we first pick a B-bounded vector $\mathbf{e}$ and compute $\mathbf{b}$ as $\mathbf{Ae} \bmod q$. In a sense, we *plant* the solution $\mathbf{e}$ inside $\mathbf{b}$. The goal now is to recover $\mathbf{e}$ (which is very likely to be unique) given $\mathbf{A}$ and $\mathbf{b}$. *We call this the planted regime or the LWE regime.*

But why is this LWE when it looks so different from Equation 1?

This is because the SIS problem in the planted regime is simply LWE in disguise. For, given an LWE instance $(\mathbf{A}, \mathbf{y}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T)$, let $\mathbf{A}^\perp \in \mathbb{Z}_q^{n\times m}$ be a full-rank set of vectors in the right-kernel of $\mathbf{A}$. That is,

$$\mathbf{A}^\perp \cdot \mathbf{A}^t = 0 \bmod q$$

Then,

$$\mathbf{b} := \mathbf{A}^\perp \cdot \mathbf{y} = \mathbf{A}^\perp \cdot (\mathbf{A}^t\mathbf{s} + \mathbf{e}) = \mathbf{A}^\perp \cdot \mathbf{e} \bmod q$$

so $(\mathbf{A}^\perp, \mathbf{b})$ is an SIS instance $\mathsf{SIS}(m-n, m, q, B)$ whose solution is the LWE error vector. Furthermore, this is in the planted regime since the LWE error vector $\mathbf{e}$ is unique given $(\mathbf{A}, \mathbf{y})$.

The reader should also notice that we can run the reduction in reverse, creating an LWE instance from an SIS instance. If the SIS instance is in the planted regime, this (reverse) reduction will produce an LWE instance.

2 Basic Theorems

We start with some basic structural theorems on LWE and SIS.

2.1 Normal Forms: SIS and LWE

The normal form for SIS is where the matrix $\mathbf{A}$ is systematic, that is of the form $\mathbf{A} = [\mathbf{A}'||\mathbf{I}]$ where $\mathbf{A}' \in \mathbb{Z}_q^{n\times(m-n)}$.

Lemma 1. *Normal-form SIS is as hard as SIS.*

Proof. To reduce from normal-form SIS to SIS, simply multiply the input to normal-form SIS (nfSIS), denoted $[\mathbf{A}'||\mathbf{I}]$, on the left by a random matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{n\times n}$. We will leave it to the reader to verify that the resulting matrix denoted $\mathbf{A} := \mathbf{B}[\mathbf{A}'||\mathbf{I}]$ is uniformly random. Furthermore, a solution to SIS on input $(\mathbf{A}, \mathbf{Bb}')$ gives us a solution to nfSIS on input $(\mathbf{A}', \mathbf{b}')$.

In the other direction, to reduce from SIS to normal-form SIS, write $\mathbf{A}$ as $[\mathbf{A}'||\mathbf{B}]$ and generate $[\mathbf{B}^{-1}\mathbf{A}'||\mathbf{I}]$ as the normal-form SIS instance. Again, a solution to the normal form instance $(\mathbf{B}^{-1}\mathbf{A}', \mathbf{B}^{-1}\mathbf{b})$ gives us a solution to SIS on input $(\mathbf{A}, \mathbf{b})$. □

The corresponding version of LWE is called short-secret LWE where both the entries of $\mathbf{s}$ and that of $\mathbf{e}$ are chosen from the error distribution χ. The proof of the following lemma follows along the lines of that for normal form SIS and is left as an exercise. (Indeed, an astute reader will observe that short-secret LWE is nothing but normal-form SIS in disguise.)

Lemma 2. *There is a polynomial-time reduction from* $\mathsf{ssLWE}(n, m, q, \chi)$ *to* $\mathsf{LWE}(n, m, q, \chi)$ *and one from* $\mathsf{LWE}(n, m, q, \chi)$ *to* $\mathsf{ssLWE}(n, m + n, q, \chi)$.

2.2 Decision vs. Search for LWE

In the decisional version of LWE, the problem is to distinguish between $(\mathbf{A}, \mathbf{y}^T := \mathbf{s}^T\mathbf{A} + \mathbf{e}^T \bmod q)$ and a uniformly random distribution. One can show, through a reduction that runs in $\mathsf{poly}(q)$ time, that the two problems are equivalent.

We will continue to see more structural theorems about LWE through the course, but this suffices for now.

3 Basic Cryptographic Applications

3.1 Private-Key Encryption

A private-key encryption scheme has three algorithms: a probabilistic key generation Gen which, on input a security parameter λ, generates a private key sk; a probabilistic encryption algorithm Enc which, on input sk and a message m chosen from a message space $\mathcal{M}$, generates a ciphertext c; and a deterministic decryption algorithm Dec which, on input sk and the ciphertext c, outputs a message m'.

Correctness requires that for every sk generated by Gen and every $m \in \mathcal{M}$,

$$\mathsf{Dec}(sk, \mathsf{Enc}(sk, m)) = m$$

The notion of security for private-key encryption is semantic security or equivalently, CPA-security, as defined in the Pass-Shelat lecture notes (see References at the end of the notes.) In a nutshell, this says that no probabilistic polynomial time (p.p.t.) adversary which gets oracle access to either the Left oracle or the Right oracle can distinguish between the two. Here, the Left (resp. the Right) oracle take as input a pair of messages $(m_L, m_R) \in \mathcal{M}^2$ and outputs an encryption of m_L (resp. m_R).

Private-Key Encryption from LWE.

- $\mathsf{Gen}(1^\lambda)$: Compute $n = n(\lambda)$, $q = q(\lambda)$ and $\chi = \chi(\lambda)$ in a way we will describe later in this lecture. Let the private key sk be a uniformly random vector

$$sk := \mathbf{s} \leftarrow \mathbb{Z}_q^n \ .$$

- $\mathsf{Enc}(sk, m)$: We will work with the message space $\mathcal{M} := \{0,1\}$. Larger message spaces can be handled by encrypting each bit of the message independently. The ciphertext is

$$c := (\mathbf{a}, b) := (\mathbf{a}, \mathbf{s}^T\mathbf{a} + e + m\lfloor q/2 \rceil \bmod q)$$

  where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and $e \leftarrow \chi$ is chosen from the LWE error distribution.

- $\mathsf{Dec}(sk, c = (\mathbf{a}, b))$: Output 0 if

$$\left| b - \mathbf{s}^T\mathbf{a} \bmod q \right| < q/4$$

  and 1 otherwise.

Lemma 3. *The scheme above is correct if the support of the error distribution* $\mathsf{Supp}(\chi) \subseteq (-q/4, q/4)$ *and CPA-secure under the LWE assumption* $\mathsf{LWE}(n, m = \mathsf{poly}(n), q, \chi)$.

Correctness and security are immediate and left as an exercise to the reader.

We left the issue of how to pick n, q and χ open, and indeed, they need to be chosen appropriately for the scheme to be secure. Correctness and security give us constraints on these parameters (see Lemma 3 above), but do not tell us how to completely specify them. To fully specify the parameters, we need to ensure security against attackers "running in 2^λ time" (this is the meaning of the security parameter λ that we will use throughout this course) and to do that, we need to evaluate the efficacy of various attacks on LWE which we will do (at least, asymptotically) in Section **??**.

3.2 Public-Key Encryption

A public-key encryption scheme is the same as private-key encryption except for two changes: first, the key generation algorithm Gen outputs a public key pk as well as a private key sk; and second, the encryption algorithm requires only the public key pk to encrypt. Security requires that a p.p.t. adversary which is given pk (and thus can encrypt as many messages as it wants on its own) cannot distinguish between an encryption of any two messages $m_0, m_1 \in \mathcal{M}$ of its choice.

Public-Key Encryption from LWE. There are many ways of doing this; we will present the cleanest one due to Lyubashevsky-Peikert-Regev.

- $\mathsf{Gen}(1^\lambda)$: Compute $n = n(\lambda)$, $q = q(\lambda)$ and $\chi = \chi(\lambda)$ in a way we will describe later in this lecture. Let the private key sk be a random vector $sk := \mathbf{s} \leftarrow \chi^n$ is chosen from the error distribution and the public key is

$$pk := (\mathbf{A}, \mathbf{y}^T := \mathbf{s}^T\mathbf{A} + \mathbf{e}^T) \in \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n$$

  where $\mathbf{A}$ is a uniformly random n-by-n matrix and $\mathbf{e} \leftarrow \chi^n$ is chosen from the error distribution.

- $\mathsf{Enc}(sk, m)$: We will work with the message space $\mathcal{M} := \{0,1\}$ as above. The ciphertext is

$$c := (\mathbf{a}, b) := (\mathbf{A}\mathbf{r} + \mathbf{x}, \mathbf{y}^T\mathbf{r} + x' + m\lfloor q/2 \rceil \bmod q)$$

  where $\mathbf{r}, \mathbf{x} \leftarrow \chi^n$ and $x' \leftarrow \chi$ are chosen from the LWE error distribution.

- $\mathsf{Dec}(sk, c = (\mathbf{a}, b))$: Output 0 if

$$\left| b - \mathbf{s}^T \mathbf{a} \bmod q \right| < q/4$$

and 1 otherwise.

Lemma 4. *The scheme above is correct if* $\mathsf{Supp}(\chi) \subseteq (-\sqrt{q/4(2n+1)}, \sqrt{q/4(2n+1)})$ *and CPA-secure under the LWE assumption* $\mathsf{LWE}(n, m = 2(n+1), q, \chi)$.

Proof. For correctness, note that the decryption algorithm computes

$$b - \mathbf{s}^T \mathbf{a} \bmod q = \mathbf{s}^T \mathbf{x} + \mathbf{e}^T \mathbf{r} + x'$$

whose absolute value, as long as $\mathsf{Supp}(\chi) \subseteq (-\sqrt{q/4(2n+1)}, \sqrt{q/4(2n+1)})$ is at most

$$q/4(2n+1) \cdot (2n+1) = q/4 \ .$$

For security, we proceed by the following sequence of hybrid experiments.

Hybrid $0.m$**.** The adversary gets pk and $\mathsf{Enc}(pk, m)$ where $m \in \{0, 1\}$.

Hybrid $1.m$**.** Feed the adveresary with a "fake" public key $\widetilde{pk}$ computed as

$$\widetilde{pk} = (\mathbf{A}, \mathbf{y}) \leftarrow \mathbb{Z}_q^{n \times n} \times \mathbb{Z}_q^n$$

and $\mathsf{Enc}(\widetilde{pk}, m)$. This is indistinguishable from Hybrid 0 by the hardness of $\mathsf{ssLWE}(n, n, q, \chi)$ and therefore, by Lemma 2, $\mathsf{LWE}(n, 2n, q, \chi)$.

Hybrid $2.m$**.** Feed the adversary with $\widetilde{pk}$ and $\widetilde{\mathsf{Enc}}(\widetilde{pk}, m)$ computed as

$$\widetilde{\mathsf{Enc}}(\widetilde{pk}, m) = (\mathbf{a}, b' + m\lfloor q/2 \rceil \bmod q)$$

where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ is uniformly random. This is indistinguishable from Hybrid 1 by $\mathsf{ssLWE}(n, n+1, q, \chi)$ or by Lemma 2, $\mathsf{LWE}(n, 2n+1, q, \chi)$, since the entire ciphertext can easily be rewritten as

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{y}^T \end{pmatrix} \mathbf{r} + \begin{pmatrix} \mathbf{x} \\ x' \end{pmatrix} + \begin{pmatrix} 0 \\ m\lfloor q/2 \rceil \end{pmatrix} \bmod q$$

which, since $\mathbf{y}$ is now uniformly random, is $n+1$ ssLWE samples and therefore can be indistinguishably replaced by

$$\begin{pmatrix} \mathbf{a} \\ b' \end{pmatrix} + \begin{pmatrix} 0 \\ m\lfloor q/2 \rceil \end{pmatrix} \bmod q$$

where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{b}' \leftarrow \mathbb{Z}_q$.

Hybrid $3.m$**.** Feed the adversary with uniformly random numbers from the appropriate domains. Follows from the previous expression for the fake ciphertext (random + anything = random).

For every $m \in \mathcal{M}$, Hybrid $0.m$ is computationally indistinguishable from Hybrid $3.m$. Furthermore, Hybrid 3 is completely independent of m. Therefore, Hybrids 0.0 and 0.1 are computationally indistinguishable from each other, establishing semantic security or CPA-security.

□

There are many ways to improve the *rate* of this encryption scheme, that is, lower the ratio of (#bits in ciphertext)/(#bits in plaintext) and indeed, even achieve a rate close to 1. We will see that in the first problem set.

We can also use these techniques as building blocks to construct several other cryptographic systems such as oblivious transfer protocols. We will see this as well in the first problem set.

This public-key encryption scheme has its origins in earlier works of Ajtai and Dwork (1997) and Regev (2004). We will see some of this in the first problem set as well.

3.3 Collision-Resistant Hashing

A collision resistant hashing scheme $\mathcal{H}$ consists of an ensemble of hash functions $\{\mathcal{H}_n\}_{n\in\mathbb{N}}$ where each $\mathcal{H}_n$ consists of a collection of functions that map n bits to $m < n$ bits. So, each hash function compresses its input, and by pigeonhole principle, it has collisions. That is, inputs $x \neq y$ such that $h(x) = h(y)$. Collision-resistance requires that every p.p.t. adversary who gets a hash function $h \leftarrow \mathcal{H}_n$ chosen at random fails to find a collision except with negligible probability.

Collision-Resistant Hashing from SIS. Here is a hash family $\mathcal{H}_n$ that is secure under $\mathsf{SIS}(n, m, q, B)$ where $n \log q > m \log(B + 1)$. Each hash function $h_{\mathbf{A}}$ is parameterized by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, takes as input $\mathbf{e} \in [0, \ldots, B]^m$ and outputs

$$h_{\mathbf{A}}(\mathbf{e}) = \mathbf{A}\mathbf{e} \bmod q$$

A collision gives us $\mathbf{e}, \mathbf{e}' \in [0, \ldots, B]^m$ where $\mathbf{A}\mathbf{e} = \mathbf{A}\mathbf{e}' \bmod q$ which in turn says that $\mathbf{A}(\mathbf{e} - \mathbf{e}') = 0 \bmod q$. Since each entry of $\mathbf{e} - \mathbf{e}'$ is in $[-B, \ldots, B]$, this gives us a solution to $\mathsf{SIS}(n, m, q, B)$.

References

The primary reference for the cryptographic definitions in this lecture is lecture notes by Pass and Shelat, available at this url.