University of Illinois, Urbana Champaign
CS 598DK Special Topics in Cryptography

Instructor: Dakshita Khurana
Scribe: Andrew Miranti, Tanya Verma
Date: October 5, 2019

LECTURE
10

Encryption from Learning with Errors

10.1 Motivation & Introduction

All cryptographic primitives are based upon some problem that is believed to be difficult. While we have discussed a number of provably secure means to perform both public key and private key encryption already in class, these had some shortcomings. Since it is not known if $P = NP$, we cannot know with certainty if these problems are actually difficult for conventional nuPPT machines. Since it is commonly believed that $P \neq NP$, this uncertainty is tolerated if a cryptographic system would be secure in that case. Some, however, are weaker than this: RSA, for example, is based on the difficulty of the integer factorization problem. This problem is known to be solvable with quantum computing using Shor's algorithm, regardless of whether $P = NP$ [2].

Furthermore, prior methods discussed in class allow us to pass messages without fear of eavesdroppers, and sign messages against impersonation, but they lack the ability to compute using the ciphertext.

Definition 10.1. Homomorphic Encryption: A cryptographic system in which an operation over the ciphertexts corresponds to an operation over the plain texts, i.e.

$$Dec(Enc(m_0)\#Enc(m_1)) = m_1 @ m_2$$

for some operations $\#, @$.

Definition 10.2. Fully Homomorphic Encryption: A cryptographic system in which enough operations are supported that one can compute a ciphertext of the result of any arbitrary function over the plaintexts using only the ciphertexts.[7]

In this lecture we introduce a new difficult computational problem which addresses both these issues. It is thought to be resistant to quantum computing, and it admits a fully homomorphic encryption scheme. Consider the problem of solving systems of linear equations with known coefficients such as the following:

$$\begin{gathered}
a_{11}s_1 + a_{12}s_2 + ...a_{1n}s_n = b_1 \mod q \\
a_{21}s_1 + a_{22}s_2 + ...a_{2n}s_n = b_2 \mod q \\
... \\
a_{m1}s_1 + a_{m2}s_2 + ...a_{mn}s_n = b_m \mod q
\end{gathered}$$

In which all a_{ij}, b_i and q are known, and the s_j variables are the unknowns. This problem is known to be efficiently solvable using gaussian elimination. However, what if we add small (we will more formally define this term later) randomized error terms to each equation?

$$\begin{gathered}
a_{11}s_1 + a_{12}s_2 + ...a_{1n}s_n + e_1 = b_1 \mod q \\
a_{21}s_1 + a_{22}s_2 + ...a_{2n}s_n + e_2 = b_2 \mod q \\
... \\
a_{m1}s_1 + a_{m2}s_2 + ...a_{mn}s_n + e_m = b_m \mod q
\end{gathered}$$

In which the e_i terms are randomly selected from some distribution with mean zero and low standard deviation, most typically a discretized gaussian distribution. This gives us a new, much more difficult problem: Learning with Errors.

10.2 Lattice Cryptography

In this section, we briefly try to understand what a lattice is and how are they important for cryptography.[5][6]

Definition 10.3. A **lattice** is a set of points $\mathcal{L} = \{a_1v_1 + a_2v_2 + ... + a_nv_n \mid a_i \in \mathbb{Z}\}$ for linearly independent vectors $v_1...v_n \in \mathbb{R}^n$. $v_1...v_n$ form the basis of $\mathcal{L}$.

The basis of a lattice is not unique. This non-uniqueness of the basis ends up being highly relevant to cryptography because it helps hide the structure of the lattice, since two very different looking sets of basis vectors can yield the same lattice.

The main computational problems that arise from lattices are the **Shortest Vector Problem (SVP)** and the **Closest Vector Problem (CVP)**. These problems are supposed to be hard to solve by a quantum computer, because no one has made much progress in this area. This is why these problems allow us to develop quantum-resistant cryptography.

The **Shortest Vector Problem** can be summarized as follows: If you are given an arbitrary lattice, which means you are given the basis of a lattice since a lattice is fully specified by a basis, can you find a short vector in this lattice? Restating this, can you find a combination of these vectors $v_1...v_n$ that somehow cancel and become short?

For crypto, we consider the approximate variant of this problem. We don't need to find the shortest vector, just something in the factor γ of the shortest.

The **Closest Vector Problem** states that given a basis B and a point v, find a lattice point that is at most γ times further than the closest lattice point.

A variant of this is **Bounded Distance Decoding**, which involves finding the closest lattice point, given that v is already "pretty close".

10.3 Learning with Errors

Definition 10.4. Learning with Errors (LWE): the problem of solving linear equations with small error terms.

Learning with errors can be thought as two closely related sub-problems: Search LWE and Decision LWE. Search LWE is relatively intuitive: given the values of all a_{ij}, b_i and q, find the variables s_j. LWE is useful for cryptography because it has been shown to be as hard to solve as any of the lattice problems described above, i.e., Shortest Vector Problem and the Closest Vector Problem.

For the sake of brevity, we will restate this problem in terms of linear algebra. Let $\mathbf{A}$ matrix with n rows and m columns, whose entries are drawn uniformly at random from the space of integers modulo q (noted as Z_q). Let $\mathbf{s}$ be a vector of length n with entries drawn uniformly at random from the same space. Let $\mathbf{e}$ be a vector of length m with entries drawn from the χ distribution. Compute $\mathbf{b}$ (all arithmetic operations from now on are performed mod q):

$$\mathbf{s}^T\mathbf{A} + \mathbf{e}^T = \mathbf{b}^T$$

With these values, we state the following assumption for the difficulty of solving this LWE problem.

$$\forall nuPPTAdv, P(Adv(\mathbf{A}, \mathbf{b}, q) = \mathbf{s}) = negl(n)$$

The probability is over the choices of $\mathbf{A}$, $\mathbf{s}$, $\mathbf{e}$ and the internal randomness of Adv.

But this assumption seems insufficient to be used as a foundational assumption for a cryptographic system because although finding **all** the coefficients s_j is difficult, it is not obvious just from this assumption that the adversary cannot learn any value of them, or gain some information about their values. This makes this form of LWE difficult to use as there is no single bit of information here that is guaranteed to be secure. There is another, equivalently strong LWE assumption with which it is simpler to prove security properties.

Another way of phrasing the difficulty of LWE is known as decisional LWE. This is the problem of, given a matrix of $\mathbf{A_{n,m}}$, and a vector $\mathbf{b_m}$, determining whether or not these are the coefficients matrix and results vector of *some* LWE problem, or if their entries are simply drawn uniformly at random from Z_q. It can be shown that the decisional and search LWE assumptions are actually of equal strength - that is, if an adversary can defeat one form, it can also defeat the other.

Notably, adding additional equations (increasing m does not make cryptosystems based on this scheme less secure. There are variants of these problems where, instead of the adversary getting only a fixed number of equations for a LWE problem, they are provided with an oracle that gives them as many equations as they can ask for (that is, some polynomial in n since the adversary is a nuPPT machine). Increasing m thus does not make the problem easier for the adversary, though if m is too small ($m < n$) then it becomes trivial to find working values for $\mathbf{s}$, but this solution is not unique since the system is underdetermined. Typically, however, we will assume $m = O(n \log q)$ as a good middle ground.[1]

10.4 Secret Key Encryption with LWE

We now present a definition of single bit secret key encryption using LWE. (Let $\leftarrow$ mean sampling from a given distribution). All randomness is based on the r parameter.

$$KeyGen(1^n, r) = \mathbf{s} \leftarrow Z_q^n$$

$$Enc(\mathbf{s}, \mu, r) = (\mathbf{a}, \mathbf{s}^T\mathbf{a} + e + \mu\lfloor\frac{q}{2}\rfloor \mod q) : \mathbf{a} \xleftarrow{uniform} Z_q^n e \leftarrow \chi$$

Recall that e is not public, and is not part of the secret key. Thus, the decryption algorithm does not know it. However, the decryption algorithm does know the distribution. We require for this algorithm to work that the χ distribution has a mean of zero and, with overwhelming probability falls into the range $[\frac{-q}{4}, \frac{q}{4}]$. If we require perfect correctness, then we can round e into this range. With this requirement set, we can use the following for our decryption:

$$Dec(\mathbf{s}, \mathbf{a}, b) = \begin{cases} 0 & \text{if } b - \mathbf{s}^T\mathbf{a} \mod q \in [\frac{-q}{4}, \frac{q}{4}] \\ 1 & \text{if } b - \mathbf{s}^T\mathbf{a} \mod q \in [\frac{q}{4}, \frac{3q}{4}] \end{cases}$$

To see why this works, recall that

$$b = \mathbf{s}^T\mathbf{a} + e + \mu\lfloor\frac{q}{2}\rfloor \mod q$$

Thus,

$$b - \mathbf{s}^T\mathbf{a} = e + \mu\lfloor\frac{q}{2}\rfloor \mod q$$

Since e is in the range $[\frac{-q}{4}, \frac{q}{4}]$, if μ was zero then this value will be in the range $[\frac{-q}{4}, \frac{q}{4}]$. Otherwise it will be in the range $[\frac{q}{4}, \frac{3q}{4}]$. Thus, Dec can distinguish $\mu = 0$ and $\mu = 1$ even though it does not know e. The requirement also generalizes to $||\mathbf{e}|| < \frac{q}{4}$ when encrypting multiple messages, which guarantees that each component of $\mathbf{e}$ is in the appropriate range with overwhelming probability.

10.5 Secret Key Encryption Proof of Security

Single bit message security:
We want to prove that $Enc(s, 0, r) \approx Enc(s, 1, r)$ (where $\approx$ means computationally indistinguishable to nuPPT machines). Note that we have made explicit the r randomness parameter in this statement of the problem.

$$
\begin{aligned}
Enc(s,0,r) &= (\mathbf{a}, \mathbf{s}^T\mathbf{a} + e \mod q) \\
&\approx (\mathbf{a}, b) \text{ (chosen uniformly from } Z_q \text{ - by decisional LWE assumption} \\
&= (\mathbf{a}, b + \mu\lfloor\frac{q}{2}\rfloor \mod q) \text{ constant offset of uniform distribution is identical to uniform} \\
&\approx (\mathbf{a}, \mathbf{s}^T\mathbf{a} + \mu\lfloor\frac{q}{2}\rfloor + e \mod q)\text{by decisional LWE assumption} \\
&= Enc(s,1,r)\,\square
\end{aligned}
$$

Multi bit message security:
We want to prove that $Enc(s,0,r_1)Enc(s,0,r_2)...Enc(s,0,r_m) \approx Enc(s,1,r_1)Enc(s,1,r_2)...Enc(s,1,r_m)$

By definition of Enc (note the different randomness parameters gives us different a and e vectors.:

$$
\begin{aligned}
Enc(s,0,r_1) &= (\mathbf{a^1}, \mathbf{b^1}) = (\mathbf{a}^1, \mathbf{s}^T\mathbf{a}^1 + e^1) \mod q \\
Enc(s,0,r_2) &= (\mathbf{a^2}, \mathbf{b^2}) = (\mathbf{a}^2, \mathbf{s}^T\mathbf{a}^2 + e^2) \mod q \\
&... \\
Enc(s,0,r_m) &= (\mathbf{a^m}, \mathbf{b^m}) = (\mathbf{a}^m, \mathbf{s}^T\mathbf{a}^m + e^m) \mod q
\end{aligned}
$$

But this is identical to:

$$Enc(s, \mathbf{0}^m, \mathbf{r}) = (\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}) \text{ with } \mathbf{A} \xleftarrow{uniform} Z_q^{n,m}, \mathbf{s} \xleftarrow{uniform} Z_q^n, \mathbf{e} \leftarrow \chi^m$$

By the decisional LWE assumption:

$$(\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e}) \approx (\mathbf{A}, \mathbf{b}) \text{ with } \mathbf{b} \leftarrow Z_q^m$$

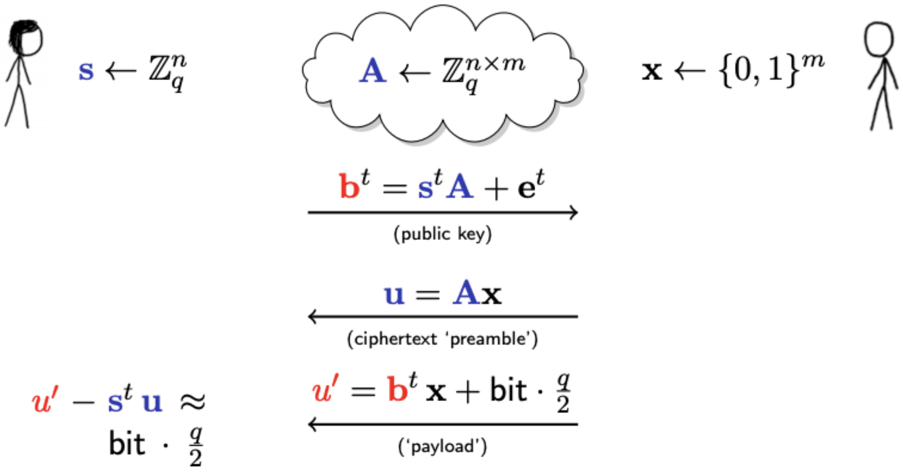
Since shifting a uniform distribution $\mod q$ by a constant does not change the distribution, it's easy to see that:

$$(\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{b} + \mu\lfloor\frac{q}{2}\rfloor\mathbf{1}^m)$$

In which $\mathbf{1}^m$ is the vector of all ones with length m.

$$
\begin{aligned}
(\mathbf{A}, \mathbf{b}) &= (\mathbf{A}, \mathbf{b} + \mu\lfloor\frac{q}{2}\rfloor\mathbf{1}^m) \\
&\approx (\mathbf{A}, \mathbf{s}^T\mathbf{A} + \mathbf{e} + \mu\lfloor\frac{q}{2}\rfloor\mathbf{1}^m) \\
&= Enc(s, \mathbf{1}^m, \mathbf{r})\,\square
\end{aligned}
$$

Thus, this secret key encryption scheme is also secure over an arbitrary number of messages.



$\mathbf{s} \leftarrow \mathbb{Z}_q^n$
$\mathbf{A} \leftarrow \mathbb{Z}_q^{n\times m}$
$\mathbf{x} \leftarrow \{0,1\}^m$
$\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t$
(public key)
$\mathbf{u} = \mathbf{A}\mathbf{x}$
(ciphertext 'preamble')
$u' - \mathbf{s}^t\,\mathbf{u} \approx \text{bit} \cdot \frac{q}{2}$
$u' = \mathbf{b}^t\,\mathbf{x} + \text{bit} \cdot \frac{q}{2}$
('payload')

FIGURE 10.1: Public Key Encryption[4]

10.6 Public Key Cryptography

We now present public key encryption using LWE as the foundational assumption.

We can model this encryption scheme using Alice and Bob as shown in Figure 10.1. Alice chooses a secret key $\mathbf{s}$ and a random $n \times m$ matrix $\mathbf{A}$. We require that $m > n \log q$ for reasons that will become clear during the proof of security. Alice's public key is $(\mathbf{A}, \mathbf{b}^T)$, where $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$.

In order to encrypt a message to Alice, Bob chooses an m-dimensional binary vector $\mathbf{r}$. Bob then multiples $\mathbf{A}$ to the left of $\mathbf{r}$, basically yielding a subset sum of the columns of $\mathbf{A}$, since $\mathbf{r}$ is a binary vector. $\mathbf{u} = \mathbf{Ar}$ is equivalent to a collision resistant hash function, which takes in an input (in this case, $\mathbf{r}$), and yields a seemingly random vector.

Now, Bob computes $\mathbf{b}^T\mathbf{r}$, and adds the product of his message bit μ and $\lfloor\frac{q}{2}\rfloor$, yielding the main ciphertext, $u' = \mathbf{b}^T\mathbf{r} + \mu\lfloor\frac{q}{2}\rfloor$. Essentially, he is going to be sending either $\mathbf{b}^T\mathbf{r}$ for $\mu = 0$, or something very far from $\mathbf{b}^T\mathbf{r}$, for $\mu \neq 0$.

In order to decrypt the message, Alice subtracts information dependent on her secret key $\mathbf{s}^T\mathbf{u}$ from u', which is either approximately 0 or $\frac{q}{2}$, depending on the value of the message bit μ.

DEFINITION 10.5. Public Key Encryption Using LWE:

$$\text{KeyGen}(\mathbf{1}^n) : (pk = (\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T), sk = \mathbf{s})$$
$$\text{with } \mathbf{s} \xleftarrow{uniform} Z_q^n, \mathbf{A} \xleftarrow{uniform} Z_q^{n,m}, \mathbf{e} \leftarrow \chi^m$$

In this case, we require that $||\mathbf{e}|| < \frac{q}{4m}$ - which means that the sum of all elements of $\mathbf{e} < \frac{q}{4}$. This fact will be important for correctness.

$$\mathrm{Enc}(pk,\mu) : (\mathbf{u} = \mathbf{A}\mathbf{r}, u' = \mathbf{b}^T\mathbf{r} + \mu\lfloor\frac{q}{2}\rfloor)$$
$$\text{with } \mathbf{r} \xleftarrow{uniform} \{0,1\}^m$$

$$\mathrm{Dec}(sk,(\mathbf{u},u')) = \begin{cases} 0 & \text{if } u' - \mathbf{s}^T\mathbf{u} \mod q \in [\frac{-q}{4}, \frac{q}{4}] \\ 1 & \text{if } u' - \mathbf{s}^T\mathbf{u} \mod q \in [\frac{q}{4}, \frac{3q}{4}] \end{cases}$$

10.7 Public Key Encryption: Proof of Security

Before we begin the proof, we will note a feature of public key cryptography. In this space, single message security implies multiple message security. To see why, consider the case of an adversary can distinguish two messages given ciphertexts $Enc(pk, m_{0,0}), ...Enc(pk, m_{0,j})$ and $Enc(pk, m_{1,0}), ...Enc(pk, m_{1,j})$ and wants to distinguish the messages m_0, m_1 given their ciphertexts. In this case, the adversary can simply place these two ciphertexts into different sets and encrypt an arbitrary number of random messages to make enlarge these sets (this is possible, because the adversary has the encryption key, unlike in private key crypto where the adversary cannot encrypt messages themselves), and use the distinguisher we assumed existed to distinguish the sets (and thus, the messages m_0, m_1). As a result, it is sufficient to prove that the public key crypto system we described above is secure for a single message.

Let's try to understand why the above encryption scheme is secure. If we have an adversary reading the transcript above, they can see Alice's public key $(\mathbf{A}, \mathbf{b})$, and they can see the ciphertext $(\mathbf{u}, u')$. To prove security, we observe two key points:

The first thing to note is that LWE is a hard problem, which implies that it is difficult to distinguish $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T$, from $(\mathbf{A}, \mathbf{b})$, where $\mathbf{b}$ has simply been sampled uniformly at random. So if Alice's public key were to be replaced by the latter version, the attacker wouldn't be able to tell the difference between the two types of $\mathbf{b}$s.

Next, we note that if Bob were to encrypt his message using a uniformly sampled $\mathbf{b}$, given that $\mathbf{A}$ is already uniformly random, the encryption is equivalent to a One Time Pad. This can be shown using the Leftover Hash Lemma.

Stated more formally, we say that the view of the adversary if they intercepted the encryption of the message $\mu = 0$ is: $\mathsf{View}_0 = (\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{A}\mathbf{r}, \mathbf{b}^T\mathbf{r}))$

By the appropriate LWE assumption, this is indistinguishable from:
$\mathsf{View}_0' = (\mathbf{A}, \mathbf{b}^T \xleftarrow{uniform} Z_q^m, \mathbf{A}\mathbf{r}, \mathbf{b}^T\mathbf{r}))$

Construct a matrix C of size $n+1, m$ by placing the row vector $\mathbf{b}^T$ below the matrix A. Then, the distribution View_0' is exactly $(\mathbf{C}, \mathbf{C}\mathbf{r})$ by construction.

We now use our requirement that $m > n \log q$. The Leftover Hash Lemma states that the distributions $(\mathbf{C}, \mathbf{C}\mathbf{r}) \approx (\mathbf{A}, \mathbf{u} \xleftarrow{uniform} Z_q^{n+1})$ under these conditions ($\mathbf{C}$ is composed of uniformly random elements drawn from $\mathbb{Z}_q$, and $\mathbf{r}$ is a uniformly random binary vector). Denote by View_0'', the distribution $(\mathbf{A}, \mathbf{u} \xleftarrow{uniform} Z_q^{n+1})$.

Since A and u consist of uniformly random elements drawn from $\mathbb{Z}_q$, View_0'' is indistinguishable from another pair $\mathsf{View}_1'' = (\mathbf{A}, \mathbf{u} \xleftarrow{uniform} Z_q^{n+1} + \lfloor \frac{q}{2} \rfloor \mathbf{1}^{n+1})$ in which every element in the second vector has been shifted by $\lfloor \dfrac{q}{2} \rfloor$, because shifting a uniformly random variable by a constant does not change the distribution. [3].

Next, we note that View_1'' is also indistinguishable from another pair $\mathsf{View}_1' = (\mathbf{C}, \mathbf{Cr} + \lfloor \frac{q}{2} \rfloor \mathbf{1}^{n+1})$ by another application of the Leftover Hash Lemma to the distribution $(\mathbf{C}, \mathbf{Cr})$.

This view is then exactly the same as the view $\mathsf{View}_1' = (\mathbf{A}, \mathbf{b}^T \xleftarrow{uniform} Z_q^m, \mathbf{Ar}, \mathbf{b}^T\mathbf{r} + \lfloor \dfrac{q}{2} \rfloor))$, which, by the LWE assumption, is computationally indistinguishable from $\mathsf{View}_1 = (\mathbf{A}, \mathbf{b}^T = \mathbf{s}^T\mathbf{A} + \mathbf{e}^T, \mathbf{Ar}, \mathbf{b}^T\mathbf{r} + \lfloor \dfrac{q}{2} \rfloor)$ - which is exactly the adversary's view when $\mu = 1$. Thus, the adversary's view when $\mu = 0$ (which is the distribution View_0) is computationally indistinguishable from their view when $\mu = 1$ (which is the distribution View_1). □

This is why an adversary reading the transcript cannot tell what the message bit is, and this encryption scheme is semantically secure.

Acknowledgement

These scribe notes were prepared by editing a light modification of the template designed by Alexander Sherstov.

References

[1] The learning with errors problem: Introduction and basic cryptography.

[2] S. T. Elisa Baumer, Jan-Grimo Sobez. Shor's algorithm, 2015.

[3] S. Park. Advanced topics in cryptography: Lattices, 2015.

[4] C. Peikert. Winter school on cryptography: Learning with errors, 2012.

[5] O. Regev. Winter school on cryptography: Introduction to lattices, 2012.

[6] Wikipedia. Lattice problem, 2019.

[7] D. J. Wu. Fully homomorphic encryption: Cryptography's holy grail, 2015.