University of Illinois, Urbana Champaign
CS/ECE 498AC3/4 Applied Cryptography

Instructor: Dakshita Khurana
Scribe: Dawei Wang
Date: November 19, 2020

LECTURE
25

Learning with Errors 2

In the previous lecture we discussed the Learning With Errors (LWE) problem which is defined as: given $b_i = (< \vec{a_i}, \vec{s} > + e_i) \bmod \mathbb{Z}_q$, try to find $\vec{s} = (s_1, ..., s_n)$ where each $s_i \in \mathbb{Z}_q$. In fact, people cannot break the LWE problem with quantum resources and this problem is considered to be quantum secure. In today's lecture, we continue our discussion for the two versions of the LWE problem: search LWE assumption, decision LWE assumption. Then we also discussed constructing encryption from decision LWE assumption.

25.1 Learning with errors

There are two versions of the LWE problem: search LWE assumption, decision LWE assumption and they are considered as one assumption because they imply each other. In our lecture, we only care about decision LWE since it is much easier to build cryptography algorithms assuming the decision LWE.

The LWE assumption is defined as follows:

- Sample n bit long prime q
- Sample secrete vector $\vec{s} = (s_1, ..., s_n)$ where each s_i is randomly sampled from $\mathbb{Z}_q$
- Set $m = n^2$ where m can be any polynomial in n
- For every $i \in [m]$:
  - Sample $\vec{a_i} = (a_{i1}, a_{i1}, ..., a_{in})$ where each a_{ij} is sampled randomly from $\mathbb{Z}_q$
  - Sample a small error $e_i \leftarrow \mathcal{X}$
  - Compute scalar $b_i = (< \vec{a_i}, \vec{s} > + e_i) \bmod \mathbb{Z}_q$
- Output $(\vec{a_1}, ..., \vec{a_m})$, $(b_1, ..., b_m)$

The search LWE assumption states that it is computationally hard to find the secrete $\vec{s}$ without knowing the noise e_i.
The decision LWE assumption says that the adversary cannot distinguish $(b_1, ..., b_m)$ from

uniformly random values. That is, define adversary A to be what we described above and adversary $A^{\mathcal{R}}$ having randomly sampled values $(b_1, ..., b_m)$, then:

$$Pr[A(\vec{a_1}, ..., \vec{a_m}, b_1, ..., b_m) = 1] \; - \; Pr[A^{\mathcal{R}}(\vec{a_1}, ..., \vec{a_m}, b_1, ..., b_m) = 1] = negl(n)$$

25.2 Symmetric key encryption from LWE

Once we assume that solving the decision LWE problem is computationally hard, we can construct an encryption scheme that relies on this hardness.
The encryption scheme works as follows:

- KeyGen(r): for some randomness r, sample secrete vector $\vec{s} = (s_1, ..., s_n)$
- Enc(s, m, r$'$): given a secrete vector s, a message m, and randomness r$'$. First sample a, e use the randomness r$'$. Then compute $(a, (< a, s > +e) + m \left\lfloor \frac{q}{2} \right\rfloor)$
- Dec(s, ct): given secrete vector s and cipher text ct, parse $ct = (a, b)$.
  - Output 0 if $b - < a, s > mod\ q \in [-\frac{q}{4}, \frac{q}{4}]$
  - Output 1 if $b - < a, s > mod\ q \in [\frac{q}{4}, \frac{3q}{4}]$

25.3 Proof of single message security

To prove single message security, we only need to prove that Enc(s,0,r) $\approx_c$ Enc(s,1,r), which means that the 2 encryption are computationally indistinguishable.
We define:

$$Enc(s, 0, r) = (a, < a, s > +e)$$
$$Enc(s, 1, r) = (a, < a, s > +e + \left\lfloor \frac{q}{2} \right\rfloor)$$

By decision LWE assumption:

$$(a, < a, s > +e) \approx_c (a, b)$$
$$(a, < a, s > +e + \left\lfloor \frac{q}{2} \right\rfloor) \approx_c (a, b + \left\lfloor \frac{q}{2} \right\rfloor)$$

Since a, b are uniformly random, constant offset of uniform distribution is identical to uniform, which means that:

$$(a, b) = (a, b + \left\lfloor \frac{q}{2} \right\rfloor)$$

This proved that Enc(s,0,r) $\approx_c$ Enc(s,1,r) and the encryption scheme at 25.2 is secure for single message.

25.4 Proof of multi message security

For multi message security, we want to prove that for $\forall\ c_0, b_0, c_1, b_1,\ ...$:

$$Enc(s, c_0, r_1), Enc(s, b_0, r_2)\ ... \approx_c Enc(s, c_1, r_1), Enc(s, b_1, r_2)\ ...$$

First, by definition:

$$Enc(s, c_0, r_1) = a_1, < s, a_1 > + e_1 + c_0 \left\lfloor \frac{q}{2} \right\rfloor$$
$$Enc(s, b_0, r_2) = a_2, < s, a_2 > + e_2 + b_0 \left\lfloor \frac{q}{2} \right\rfloor$$

By decision LWE assumption, with $b_1, b_2 \in \mathbb{Z}_q$:

$$a_1, < s, a_1 > + e_1 \approx_c a_1, b_1$$
$$a_2, < s, a_2 > + e_2 \approx_c a_2, b_2$$

Since shifting a uniform distribution by a constant does not change the distribution, we have:

$$a_1, b_1 + c_0 \left\lfloor \frac{q}{2} \right\rfloor = a_1, b_1 + c_1 \left\lfloor \frac{q}{2} \right\rfloor \approx_c Enc(s, c_1, r_1)$$
$$a_2, b_2 + b_0 \left\lfloor \frac{q}{2} \right\rfloor = a_2, b_2 + b_1 \left\lfloor \frac{q}{2} \right\rfloor \approx_c Enc(s, b_1, r_2)$$

Thus, we proved that the encryption scheme at 25.2 is also secure over multiple number of messages.

25.5 LWE matrix representation

As we successfully build the LWE symmetric key encryption, our next goal is to build the LWE public key encryption. We need to look into another form of the LWE.
Suppose we have n equations where each $s \in \mathbb{Z}_q$ and we sampled m error terms from $\mathcal{X}$. Then we calculate each $b_i =< a_i, s > + e_i,\ i \in m$ and represent it as a vector $\vec{b}$. Thus, we obtain the matrix representation of LWE as:

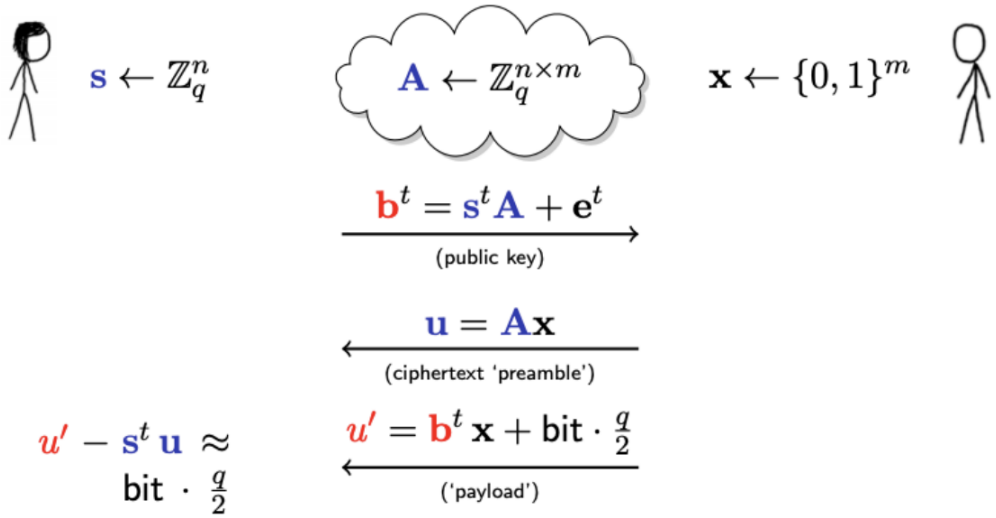
$$\vec{b}^T = \vec{S}^T A + e^T$$

Then the decision LWE can also be represent as, $\forall$ PPT adversary $\mathcal{A}$:

$$|Pr[\mathcal{A}(A, \vec{b}) = 1] - Pr[\mathcal{A}(A, b) = 1]| = negl(n)$$

where $A \leftarrow \mathbb{Z}_q^{n \times m}$, $S \leftarrow \mathbb{Z}_q^{n \times 1}$, $e \leftarrow \mathcal{X}^{m \times 1}$, $\vec{b} = (s^T A + e^T)^T$, $b \leftarrow \mathbb{Z}_q^{m \times 1}$.

25.6 Public key encryption from LWE



We can model the public key encryption scheme using Alice and Bob and the LWE matrix representation. Alice chooses a secret key s and a random n × m matrix A. And the public key is (A, b^T) where $A \leftarrow \mathbb{Z}_q^{n \times m}$, $e \leftarrow \mathcal{X}^m$, $b = (s^T A + e^T)^T$.
In order to encrypt a message to Alice, Bob chooses an m-dimensional binary vector r and calculate $u = Ar$. Then the main cipher text is $u' = br + m \left\lfloor \frac{q}{2} \right\rfloor$ where m is Bob's message. In order to decrypt the message, Alice subtracts information dependent on her secret key which is $u' - s^T u \approx_c m \left\lfloor \frac{q}{2} \right\rfloor$.

The whole public key encryption algorithm works as follows:

- KeyGen(r):
  - sk = $\vec{s} \leftarrow \mathbb{Z}_q^{n \times 1}$
  - pk = $(A, b^T = s^T A + e^T)$ where $A \leftarrow \mathbb{Z}_q^{n \times m}$, $e \leftarrow \mathcal{X}^{m \times 1}$. Note that we require $\mathcal{X}$ is set s.t. samples from $\mathcal{X} \in [-\frac{q}{4m}, \frac{q}{4m}]$. This is important for the correctness.
- Enc(m, pk, r):
  - Parse $pk = (A, \vec{b}^T)$
  - Sample vector $\vec{r} \leftarrow \{0,1\}^T$
  - Compute $\vec{C}_{n \times 1} = A_{n \times m} \vec{r}_{m \times 1}$
  - Compute $d = (\vec{b}^T_{1 \times m} \vec{r}_{m \times 1} + m \left\lfloor \frac{q}{2} \right\rfloor) \bmod q$
  - Enc(m, pk, r) = $(\vec{C}$, d)
- Dec(ct, pk):
  - Parse ct = $(\vec{C}$, d)
  - Compute $\alpha = \vec{s}^T_{1 \times n} \vec{C}_{n \times 1} = \vec{s}^T A \vec{r}$
  - Compute $\beta = (d - \alpha) = m \left\lfloor \frac{q}{2} \right\rfloor + \vec{b}^T \vec{r} - s^T A \vec{r} = \vec{e}^T \vec{r}$
  - Output m = 0 if $\beta \in [-\frac{q}{4}, \frac{q}{4}]$, else output m = 1

25.7 Proof of public key encryption security

Previously in the public-key setting, we already showed that single message CPA is the same as multi message CPA. So all we need to approve is single message CPA security, that is, $(pk, Enc(pk, 0, r)) \approx_c (pk, Enc(pk, 1, r))$.
By definition:

$$(pk, Enc(pk, 0, r)) = (A, \vec{s}^T A + e^T), (A\vec{r}, \vec{b}^T\vec{r}) \approx_c (A, \vec{b}^T), (A\vec{r}, \vec{b}^T\vec{r}) \text{ where } b \leftarrow \mathbb{Z}_q^m$$

$$(pk, Enc(pk, 1, r)) = (A, \vec{s}^T A + e^T), (A\vec{r}, \vec{b}^T\vec{r} + \left\lfloor \frac{q}{2} \right\rfloor) \approx_c (A, \vec{b}^T), (A\vec{r}, \vec{b}^T\vec{r} + \left\lfloor \frac{q}{2} \right\rfloor) \text{ where } b \leftarrow \mathbb{Z}_q^m$$

Define matrix $E = (A, \vec{b}^T)$, then we get:

$$(pk, Enc(pk, 0, r)) = (E, Er)$$
$$(pk, Enc(pk, 1, r)) = (E, Er + \left\lfloor \frac{q}{2} \right\rfloor)$$

Apply the Leftover Hash Lemma, we have:

$$Er \approx_c \text{uniform random vector} = \text{uniform random vector} + \left\lfloor \frac{q}{2} \right\rfloor \approx_c Er + \left\lfloor \frac{q}{2} \right\rfloor$$

Thus, we proved the public key encryption security.

Acknowledgement

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