LECTURE 24

Learning with Errors

$$14 s_1 + 5 s_2 + 10 s_3 + 2 s_4 = 8 \pmod{17}$$

$$13 s_1 + 14 s_2 + 14 s_3 + 6 s_4 = 16 \pmod{17}$$

$$6 s_1 + 10 s_2 + 13 s_3 + 15 s_4 = 3 \pmod{17}$$

$$6 s_1 + 7 s_2 + 16 s_3 + 25 s_4 = 3 \pmod{17}$$

Use Gaussian elimination to find s_1, s_2, s_3, s_4

Approximate / Noisy equations

$\vec{a_1} = (14, 5, 10, 2)$

$$14 s_1 + 5 s_2 + 10 s_3 + 2 s_4 + \overset{-1}{e_1} = 8 \pmod{17}$$

$$13 s_1 + 14 s_2 + 14 s_3 + 6 s_4 + \overset{0}{e_2} = 16 \pmod{17}$$

$$6 s_1 + 10 s_2 + 13 s_3 + 15 s_4 + \overset{-1}{e_3} = 3 \pmod{17}$$

$$6 s_1 + 7 s_2 + 16 s_3 + 5 s_4 + \overset{1}{e_4} = 3 \pmod{17}$$

$$5 s_1 + 3 s_2 + 17 s_3 + 2 s_4 + \overset{1}{e_5} = 11 \pmod{17}$$

$$10 s_1 + 4 s_2 + 10 s_3 + 7 s_4 + \overset{0}{e_6} = 8 \pmod{17}$$

where $e_i \in \{-1, 0, 1\}$ for $i \in [1, 4]$.

Guess: $e_1 = -1$, $e_2 = 0$, $e_3 = -1$, $e_4 = 1$,
$e_5 = 1$, $e_6 = 0$.

Find s_1, s_2, s_3, s_4 by Gaussian elimination

Suppose $n = 256$ equations, ≈ 256 variables.

How many possible assignments to $(e_1, e_2 \ldots e_{256})$?

$\boxed{3^{256}.}$

Search Learning-with-Errors

Informally: For the "right parameters", given $(\vec{a}_1, \vec{a}_2 \ldots \vec{a}_n)$ and a set of equations of the form $\langle \vec{a}_i, \vec{s} \rangle + e_i$, it is hard for PPT machines to find $\vec{s}$.

* First sample prime q. (n bits long).

* Sample $\vec{s} = (s_1 \ldots\ldots s_n)$ where

  $\hookrightarrow$ # variables

  each $s_i \xleftarrow{\$} \mathbb{Z}_q$.

* Set $m = n^2$.

  $\hookrightarrow$ # equations

* For every $i \in [m]$,

  sample $\vec{a_i} = (a_{i1}, a_{i2} \ldots\ldots a_{in})$ where each $a_{ij} \xleftarrow{\$} \mathbb{Z}_q$

  sample $e_i \leftarrow \chi$. [T.B.D.].

  Compute $b_i = \langle \vec{a_i}, s \rangle + e_i \pmod{\mathbb{Z}_q}$

  $$(a_{i1} s_1 + a_{i2} s_2 + \ldots\ldots a_{in} s_n) + e_i$$

* Output $(\vec{a}_1 \ldots\ldots \vec{a}_m), (b_1 \ldots\ldots b_m)$

Search LWE assumption #1.

$\forall$ PPT adversaries $\mathcal{A}$,

$$\Pr\left[\mathcal{A}(\vec{a}_1 \ldots \vec{a}_m, b_1 \ldots b_m) \longrightarrow \vec{s^*}\right] = negl(n)$$

$\vec{a}_i \xleftarrow{\$} \mathbb{Z}_q^n$, $\vec{s} \xleftarrow{\$} \mathbb{Z}_q$, $\vec{e} \leftarrow \chi$

$(\vec{a}_1, \ldots \vec{a}_m)$ $(s_1 \ldots s_n)$ $(e_1 \ldots e_m)$

where $\vec{s^*}$ is s.t. there exist $\vec{e}$ (small)

s.t. $\langle \vec{a}_i, \vec{s^*} \rangle + e_i = b_i \quad \forall i \in [m]$

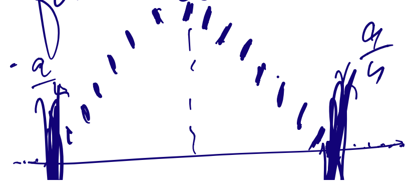
Decision LWE assumption

$\forall$ PPT Adv. $\mathcal{A}$,

$$\left| \Pr_{\substack{\vec{a}, \vec{s}, \vec{e} \text{ as above} \\ b = \vec{a}\vec{s} + e}} \left[\mathcal{A}(\vec{a}_1, \ldots, \vec{a}_m, b_1, \ldots, b_m) = 1 \right] - \Pr_{\substack{\vec{a} \text{ as above} \\ \forall i \in [m]\, b_i \xleftarrow{\$} \mathbb{Z}_q}} \left[\mathcal{A}(\vec{a}_1, \ldots, \vec{a}_m, b_1, \ldots, b_m) = 1 \right] \right| = \text{negl}(n)$$

$\boxed{\text{FACT. 1}}$ Search LWE $\Longleftrightarrow$ Decision LWE

$\boxed{\text{FACT 2.}}$ DLWE (and Search LWE) are NOT hard for all choices of q, n, m, $\underbrace{\chi}_{\downarrow}$.



[Discrete Gaussian distribution]

Let's Build crypto.

Symmetric / Private Key Encryption from DLWE assumption

Recall DLWE: $(\vec{a}_i, \vec{a}_i \vec{s} + e_i) \approx (\vec{a}_i, b_i)$ where $b_i \xleftarrow{\$} \mathbb{Z}_q$.

Hint 1: Finding $\vec{s}$ is hard given $(\vec{a}_i, \vec{a}_i \vec{s} + e_i)$

Hint 2: $(\vec{a}_i, \underline{\vec{a}_i \vec{s} + e_i}) \approx (a_i, b_i \xleftarrow{\$} \mathbb{Z}_q)$

use this to mask message.

SKE : (KeyGen, Enc, Dec).

KeyGen $\rightarrow \vec{s} \xleftarrow{\$} \mathbb{Z}_q^n$.

Enc $(\overset{\text{msg}}{m}, \overset{\text{key}}{\vec{s}} ; r)$: Use r to sample $\vec{a}$

sample e

$\text{Enc}(m, \vec{s}; r)$: (1) sample $\vec{a}, e$ using r

(2) Output $ct = (\vec{a}, (m+b) \bmod q)$

where $b = \vec{a}\vec{s} + e$

$\vec{a} \to (a_1 \ldots a_n)$, $\vec{s} \to (s_1, \ldots s_n)$

$\text{Dec}(ct, \underline{\vec{s}})$: (1) Parse $ct = (\vec{a}, \underline{y})$

(2) Compute $\langle \vec{a}, \vec{s} \rangle$

(3) To decrypt, need to recover

$m = (\underline{y} - b) \bmod q$.

which means, need to compute $\underline{b}$.

$b = \vec{a}\vec{s} + e$

Suppose $e \in \left[-\left\lfloor \frac{q}{4} \right\rfloor, \left\lfloor \frac{q}{4} \right\rfloor \right]$

~~Attempt~~ Final !

KeyGen $\longrightarrow \vec{s}$

Enc$(m, \vec{s}\,; r)$:

$m \in \{0,1\}$

* Sample $\vec{a}$, e using r.
* $ct = \left(\vec{a}, \; b + m\left\lfloor \frac{q}{2} \right\rfloor\right)$

where $b = \langle \vec{a} \cdot \vec{s} \rangle + \underline{e}$

Dec$(\underline{ct}, \vec{s})$:

Parse $ct = (\underline{\vec{a}}, \; y)$

We know $\left(m\left\lfloor \frac{q}{2} \right\rfloor\right) = \underline{y - b}$ $\{$ off by $e \Rightarrow$ off by $\underline{\left[-\left\lfloor \frac{q}{4} \right\rfloor, \left\lfloor \frac{q}{4} \right\rfloor\right]}$

Compute $y - \langle a \cdot s \rangle$

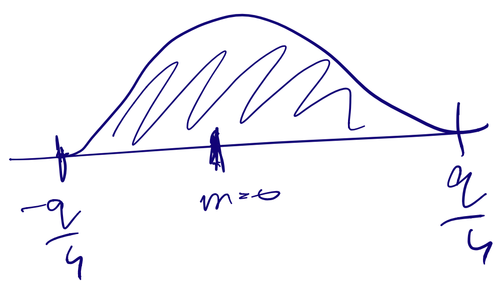
If $m = 0$, then $y - \langle a, s \rangle \notin$

will lie between $\left[\left\lfloor -\frac{q}{4} \right\rceil, \left\lfloor \frac{q}{4} \right\rceil \right]$

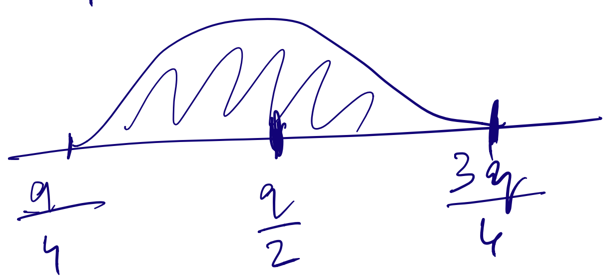
If $m = 1$, then $y - \langle a, s \rangle$

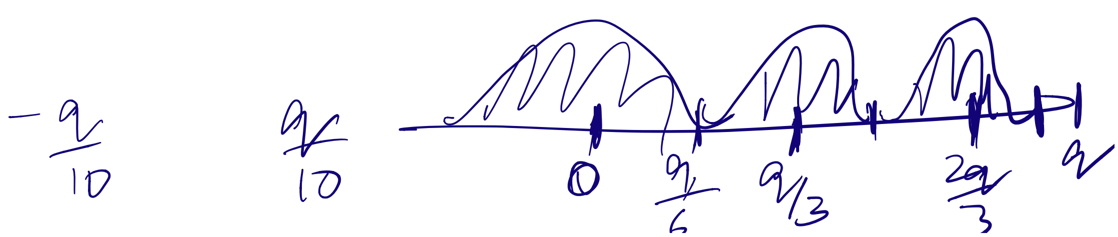
will lie between $\left[\left\lfloor \frac{q}{4} \right\rceil, \left\lfloor \frac{3q}{4} \right\rceil \right]$

$m = 0$



$m = 1$





$$e \in \left[-\left\lfloor \frac{q}{n} \right\rfloor, \left\lfloor \frac{q}{n} \right\rfloor \right] \approx n \text{ buckets}$$

Can encrypt $m \in \{1, \ldots n\}$

$$e \in \left[\left\lfloor \frac{-q}{2^{\sqrt{n}}} \right\rfloor, \left\lfloor \frac{q}{2^{\sqrt{n}}} \right\rfloor \right] \approx 2^{\sqrt{n}} \text{ buckets}$$

$m \in \{1 \ldots 2^{\sqrt{n}}\}$

$\sqrt{n}$ - bit messages.

Each $\vec{a_i} = (a_{i1} \ldots\ldots a_{in})$

where $\forall j \in [n]$, $a_{ij} \xleftarrow{\$} \mathbb{Z}_q^n$

Use r to sample each a_{ij} as random number mod q.

sample each $e_i \leftarrow \chi$.