6.857 Recitation #7

April 9, 2021

Announcements:

- Quiz out; due April 21
- Individual, but please feel free to ask in OH/private Piazza posts
- Project mentors assigned; your group should hear from them soon if not yet

Goal of today is to understand LWE more fully.

Why?

Crypto assumptions are crucial

... the foundation of cryptography's utility are no stronger than the assumptions used.

LWE and its assumed hardness

"learning with errors"

Need four parameters to define instance of LWE

$$(q, n, m, \chi)$$

prime q is the field size of $\mathbb{Z}_q$

n is number of variables to "learn" (i.e. determine)

m is number of equations to solve for the variables

χ is prob. dist. for the "errors" (i.e. noise distribution)

Mildly misleading, χ is not necessarily, nor usually, the chi distribution

Also denoted ϕ, it is common to take noise from the (discrete) normal distribution

w/ $\mathbb{E}[\chi] = 0$

$\mathrm{Var}[\chi] = \sigma^2$ (more on σ later)

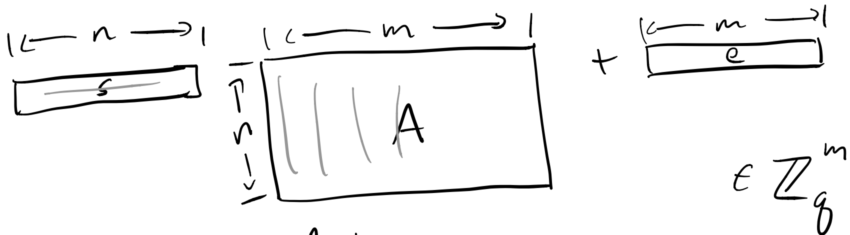
so everything stays an integer and we don't leave $\mathbb{Z}_q$

Two main forms of LWE assumption:

Both take $s \leftarrow \mathbb{Z}_q^n$, $A \leftarrow \mathbb{Z}_q^{n \times m}$, $e \leftarrow \chi^m$ (each element sampled independently)

Arguably the intuitive form:

Search LWE:



Get $(A, sA + e)$.

Find s.

A known
χ known (pdf of e, but not e itself)

Hardness: no PPT algorithm $\mathcal{A}$ can output s with non-negligible probability

What we saw on Wednesday:

Decision LWE:

Get (A, b) where $b \in \mathbb{Z}_q^m$

Decide: did you get

$$(A, sA + e) \quad \text{or} \quad (A, U_{\mathbb{Z}_q^m})?$$

Hardness: no PPT $\mathcal{A}$ can correctly decide with probability greater than $\underbrace{\frac{1}{2}}_{\text{guessing}} + \underbrace{\text{negl}(n)}_{\text{asymptotic advantage is negligible}}$

Big questions:

<u>Why</u> is it hard?

→ No answer. We don't <u>know</u> it is hard, it just seems to be.

<u>When</u> might it be hard?

→ Our next topic.
How to set parameters (q, n, m, χ)

n is the security parameter

What can we say about the other parameters as a function of n?

i.e. $q(n)$
$m(n)$
$\chi(n)$

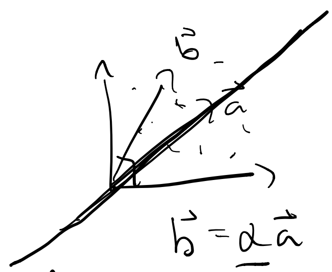
LWE's dependence on m:

If $m < n$: sA will not determine s
$\rightarrow$ s not uniquely defined from system of linear equations sA
$\Rightarrow$ PPT $\mathcal{A}$ has to guess on $n-m$ variables
each variable one of q possibilities
$\Rightarrow$ guessing works with negl prob.
$< \left(\frac{1}{q}\right)^{n-m}$

When $m \geq n$: sA now probably fully determined
$\hookrightarrow$ A is random, so could fail to determine s even here.

s is determined when $\text{rank}(A) = n$

We have $\text{rank}(A) = n$ when the columns of A are orthogonal



Collision argument:

probability of 2 columns of A "colliding", ie being ~~non-orthogonal~~ parallel:

n elements all in $\mathbb{Z}_q$

$|\mathbb{Z}_q| = q$

q^n

$$\Pr[\text{2 columns } \underline{\text{are}} \cancel{\text{not orthogonal}} \text{ parallel}] = \frac{q}{q^n}$$

$\leftarrow$ ways ~~not to be orthogonal~~ to be parallel

$\leftarrow$ total # of vectors in $\mathbb{Z}_q^n$

$$\Rightarrow \Pr[k \text{ "collisions"}] = \left(\frac{1}{q^n}\right)^k$$

k collisions means $\text{rank}(A)$ is $\max(n, \underset{\sim}{m-k})$

n.b. this is at most equal to n

Finally, we need $m-k=n$:

or that $k=m-n$

We can compute the probability that there are too many collisions, which is when $k>m-n$, and therefore we <u>don't</u> have full rank, by summing over $k=m-n+1$ to $m-1$:

$$\Pr[\text{not full rank}] = \sum_{k=m-n+1}^{m-1} \left(\frac{1}{q^{n-1}}\right)^k$$

geometric series
$$\sum_{k=a}^{b-1} x^k = \frac{x^a - x^b}{1-x} \leq \frac{x^a}{1-x}$$

$$x = \frac{1}{y} \Rightarrow \leq \frac{1}{y^{a-1}(y-1)}$$

$$\leq \underbrace{\frac{1}{q^{(n-1)(m-n)}\left(q^{n-1}-1\right)}}_{\text{Easy to make negligible!}}$$

<u>Phew!</u>

So when $m>n$, sA is overwhelmingly likely to determine s uniquely

Can m be too big??

YES!

If $m(n)$ is super-poly(n), for example 2^n, $n^{\text{polylog}(n)}$, then hardness breaks down.

This is because PPT $\mathcal{A}$ can run in poly-time in size of input, and matrix A alone is size $n \cdot m$.

$m = n \log q$

What about the noise χ?

Uniform noise, $e \leftarrow \mathbb{Z}_q^m$, is too much.

Now $sA+e$ really is uniform, so we have information-theoretic hiding ...

but also now A and $sA+e$ are independent so the tuple is useless for crypto.

But no noise, $\chi = \delta(x)$ for $x \in \mathbb{Z}_q$, is not hard at all.

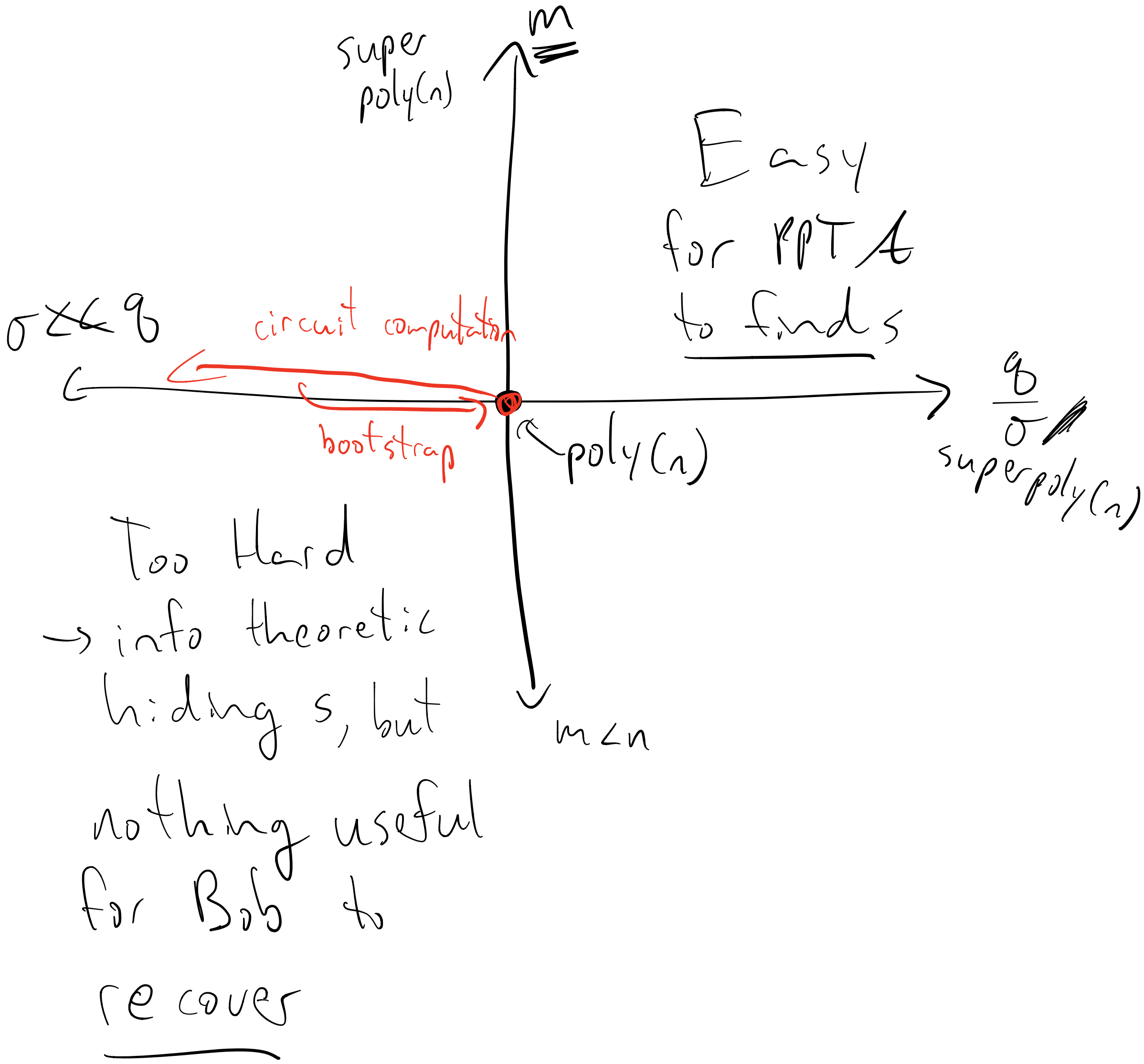
With discrete normal noise, standard deviation σ, it turns out the sweet spot is to have $\sigma \ll q$ such that

$$\frac{q}{\sigma} = \text{poly}(n)$$

So q and σ are relatively determined for hardness of LWE to hold.

When $\frac{q}{\sigma} = \text{poly}(n)$, best known algorithm can find s in $O(2^n)$ [Blum-Kalai-Wasserman 2003] not Yael, Adam

(If $\frac{q}{\sigma} = \text{superpoly}(n)$, not enough noise and $\exists \mathcal{A}$...)



super poly(n)
m
Easy for PPT $\mathcal{A}$ to find s
$\sigma \ll q$
circuit computation
bootstrap
poly(n)
$\frac{q}{\sigma}$
superpoly(n)
Too Hard
→ info theoretic hiding s, but nothing useful for Bob to recover
$m < n$

PK crypto from LWE:

$Gen(1^n)$: $s \leftarrow \mathbb{Z}_q^n$ $\quad A \leftarrow \mathbb{Z}_q^{n \times n}$ $\quad$ m=n here!

$e \leftarrow \chi^n$

$pk = (A, b)$ $\quad b = sA + e$

$sk = s$

$Enc(pk, \mu)$: $\mu \in \{0,1\}$ is a bit

$t \leftarrow \chi^n$ $\quad \tilde{\mu} = \mu \cdot \lfloor \frac{q}{3} \rfloor$

$$CT = t \cdot [A \| b] + [0, 0, \ldots, 0, \tilde{\mu}]$$

$(n \times n+1)$ matrix

$Dec(sk, CT)$: compute: $CT \cdot \begin{bmatrix} -s \\ 1 \end{bmatrix} = \left(t \cdot [A \| b] + [0^n \| \tilde{\mu}]\right) \cdot \begin{bmatrix} -s \\ 1 \end{bmatrix}$

$$= t \cdot [A \| b] \cdot \begin{bmatrix} -s \\ 1 \end{bmatrix} + \tilde{\mu}$$

$$= t \cdot (-sA + b) + \tilde{\mu}$$

$$= t \cdot (\cancel{-sA} + \cancel{sA} + e) + \tilde{\mu}$$

$$= t \cdot e + \tilde{\mu}$$

$\mathbb{E}[|t \cdot e|] = n\sigma^2 \ll q$, so if $t \cdot e + \tilde{\mu} \sim \frac{q}{3}$,

then μ was 1

otherwise, μ was 0