Lattice-based Crypto (Lec # 17, May 27'26)

Course Progress:

OWF / PRG / RO, dLog / Pairings (RSA), SNARKs, MPC, PIR, LWE, FHE →

Foundations + ECC, Cryptanalysis, ZKP, Secure Comput^n., Lattices

Outline:

- Why Lattices?
- The Learning with Errors assumption (LWE)
- Regev Encryption (+ Informal Proof)
- worst-case Lattice Problems.

(I) Why introduce new assumptions?

1.) Plausibly post-Quantum:

* Factoring, DLog are easy (poly-time) on a quantum computer. ⇒ SHOR's ALGO '94

⇒ All constructions based on DLog / Factoring (e.g. RSA, ElGamal, BLS Sigs,

DH Key Exchange) can be broken with a quantum computer. 😳

*But, there's hope with lattices:
There are many lattice-based problems, for which there is NO known efficient quantum algorithm ...

"Plausibly Post Quantum"

* Ongoing standardization effort by NIST.
  - Post Quantum Signatures
  - " " Encryption

2.) Diversify cryptographic assumptions:

* If tmrw, someone makes an efficient Dlog algo, it'd be good to have options...

(skipped)

* Opens avenues to base crypto on worst-case hardness.

(Crypto usually based on average-case hardness)

e.g. Existence of OWF; Factoring°

° Problem is Hard for most instances from some distribut"

But Lattice-based crypto can be based

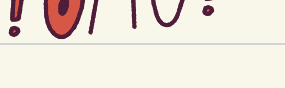
on worst-case hardness.

i.e. There is no efficient algorithm that solves all instances.

3) New functionalities!

Fully Homomorphic Encryption: (FHE)

Given ONLY Encryption of x, one can Efficiently compute Encryption of $f(x)$, for ANY function f!!!



e.g. Query ChatGPT on secret

[It is NOT known HOW to build FHE from other assumptions!]

II Warm-up: Solving systems of linear Equations over $\mathbb{Z}_q$:

e.g.

$$3x_1 + 4x_2 + 1x_3 = 0 \pmod 7$$

$$4x_1 + 2x_2 + 6x_3 = 1 \pmod 7$$

$$1x_1 + 1x_2 + 1x_3 = 1 \pmod 7$$

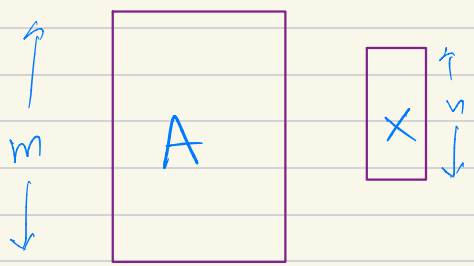
Let n = # Variables.

m = # Equations.

$(n = m = 3$ in our example$)$ (A, b)

* We can solve such a system using Gaussian Elimination efficiently, in time poly(m, n). (works for any field $\mathbb{F}$)

Matrix notation:



$\in \mathbb{Z}_q^{m \times n}$ $\in \mathbb{Z}_q^{n}$ $\in \mathbb{Z}_q^{m}$

eg. $A = \begin{bmatrix} 3 & 4 & 1 \\ 4 & 2 & 6 \end{bmatrix}$ $b = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$

The Learning With Errors problem (LWE)

What if the system is noisy?

⇒ Given A and $Ax + e$ (instead of b), can you recover x?

Noise, $e \in \mathbb{Z}_q^m$ (FROM SOME DISTRIBUTION)

i.e. $\begin{bmatrix} A \end{bmatrix} \begin{bmatrix} x \end{bmatrix} \approx \begin{bmatrix} b \end{bmatrix} + \begin{bmatrix} e \end{bmatrix}$ $\quad e \in \mathbb{Z}_q^m$

NOISY !!

even for quantum computers.

This problem is conjectured to be HARD for some choices of parameters & noise.

Some notation:

- $\mathbb{Z}_q$ = integers in the range $\left(-\frac{q}{2}, \frac{q}{2}\right)$

Specifically, $\left\{-\frac{(q-1)}{2}, \dots, 0, 1, \dots, \frac{q-1}{2}\right\}$

(assuming $q > 2$ is prime)

This is ISOMORPHIC to

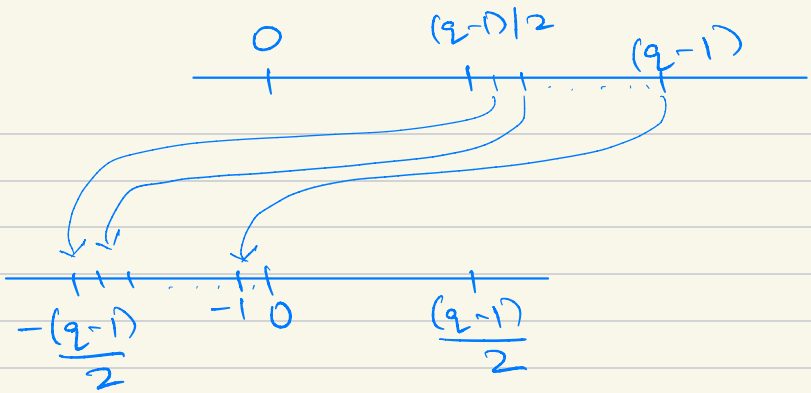
$\{0, 1, 2, \dots, \frac{(q-1)}{2}, \frac{(q-1)}{2}+1, \dots, q-1\}$

How? ↓ ↓ ↓ ↓

$\{0, 1, \dots, \frac{(q-1)}{2}, \quad = -1 \bmod q.$

$\frac{(q-1)}{2} + 1 = \frac{(q-1)}{2} + 1 - q \bmod q.$

$= -\frac{(q-1)}{2}$



e.g. $\mathbb{Z}_7 = \{-3, -2, -1, 0, 1, 2, 3\}$.

- For any vector $v \in \mathbb{Z}_q^m$,

$$\|v\|_\infty = \max_i |v_i|$$ (infinity norm)

- Errors are usually sampled from a bounded norm distribution. $(B \ll \frac{q}{2})$

χ_B = B~ bounded distribution

i.e. $\Pr_{e \leftarrow\$ \chi_B^m}[\|e\|_\infty > B] \leq \text{negl}(\lambda)$

$\Leftrightarrow$ every entry $< B$ w.h.p.

e.g. Sample e_i uniformly from $\{-B, \cdots, 0, \cdots, B\}$ $\forall\ i \in [m]$.

Usually, we use a discrete Gaussian distribution over $\mathbb{Z}_q$, with mean = 0.

Formal defn: parametrized by (q, m, n, χ_B)

The LWE assumption states that

for random $A \leftarrow\$ \mathbb{Z}_q^{m \times n}$, $s \leftarrow\$ \mathbb{Z}_q^n$

and $e \leftarrow\$ \chi_B^m$,

given $(A, As+e)$, no efficient adversary can find s' s.t. $As' \approx (As+e)$,

i.e. $\| As' - (As+e) \|_\infty \leq B$.

(s is a possible solution, but there may be other solutions...)

(Basically, it is hard to (efficiently) solve a noisy system of linear equations.)

A lot of parameters... (common in lattice-based crypto...)

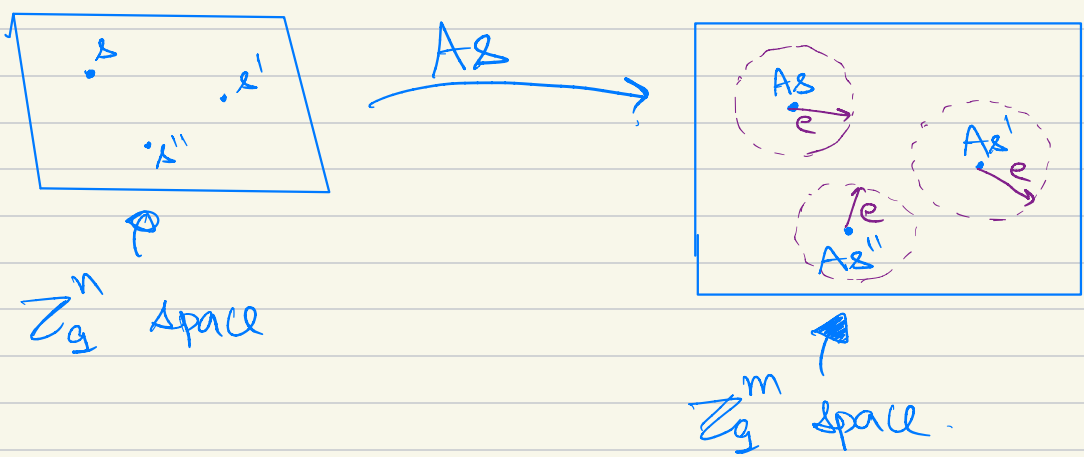
* n: # Variables.
  ~ security parameter (More Unknowns = Harder problem)

* $m = \text{poly}(n)$, $m \gg n$. (More Equations ~ Easier problem)
  ~ $n \log q$.

* $q = poly(n)$ (Note, q is $poly(\lambda)$, unlike DLog based schemes.)

* $B << q$ (smaller noise bound ≈ easier problem.)

n, B, q are chosen so that "search" LWE has a unique solution with high probability.



Spheres: space of all possible values of As+e.

Diameter of Sphere $\propto$ B.

* If the spheres don't intersect: then unique solution of LWE problem

The above LWE problem is the "search" version: A needs to find s.

We can also define a "decision" version of LWE:

(e.g. DDH vs. CDH)
decision ≤ search

LWE (decision) assumption:

The following distributions are computationally indistinguishable:

$$\left\{ (A, As+e) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ s \leftarrow\$ \mathbb{Z}_q^{n} \\ e \leftarrow\$ \chi_B^{m} \end{array} \right\} \approx_c \left\{ (A, v) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ v \leftarrow\$ \mathbb{Z}_q^{m} \end{array} \right\}$$

i.e. $(A, As+e)$ looks random...

This is called the $\mathrm{LWE}_{n,m,q,\chi_B}$ assumption.

* Search and decision versions of LWE are actually equally hard!!

(this is not true for DDH vs. CDH
e.g. DDH is easy in pairing groups,
but CDH is not known to be easy...)

II. Regev Encryption:

we'll build a Public Key Encryption scheme...

i.e. Alice runs KeyGen to get pk, sk, and publishes pk.

Then, anyone can encrypt msg x to Alice's pk, by calling $c = Enc(pk, x)$.

Alice can decrypt c using her sk, by running $x = Dec(sk, c)$.

A PKE scheme must be:
- correct: Dec. decrypts to correct plaintext.
- Semantically Secure: Adv. can't distinguish b/w Enc. to m_0, m_1, without sk.

Regev's PKE:

A simple, El-Gamal style public key encryption from LWE:

Similar to El Gamal, pk: Instance, sk: Solution.

KeyGen(1^λ):

$A \leftarrow\!\$\ \mathbb{Z}_q^{m \times n}$, $s \leftarrow\!\$\ \mathbb{Z}_q^{n}$, $e \leftarrow\!\$\ \chi_B^{m}$.

Compute $b = As + e \in \mathbb{Z}_q^m$

Set $sk = s$, $pk = (A, b)$.

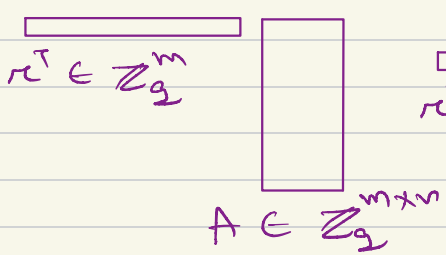
sk: "LWE secret"

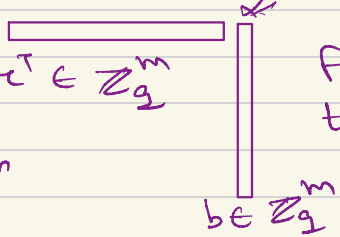
pk: an LWE instance.

Encrypt$(pk, x \in \{0,1\})$: Encrypting 1 bit. (Not so efficient...)

Sample $r \leftarrow\!\$\ \{0,1\}^m$: Low-norm!! $\|r\|_b \le 1$.

$c_0 = r^T A$, $c_1 = r^T b + \lfloor \frac{q}{2} \rfloor \cdot x$.





Floor: Round down to nearest integer.

Output $ct = (c_0, c_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

we know, $b = As + e$.

Decrypt $(sk = s, ct = (c_0, c_1))$:

Observe that,

$$c_1 = r^T(As + e) + \lfloor \frac{q}{2} \rfloor x$$

$$= r^T As + r^T e + \lfloor \frac{q}{2} \rfloor x.$$

we want to get rid of these terms...

Also, $c_0 = r^T A$ and we have $sk = s$.

consider :

$$c_1 - c_0 \cdot s = r^T As + r^T e + \lfloor \frac{q}{2} \rfloor x - r^T As.$$

$$= r^T e + \lfloor \frac{q}{2} \rfloor x.$$

Observe that, e : Low norm : $\|e\|_\infty \leq B \ll q$ w.h.p.

also $r \in \{0,1\}^m$: $\|r\|_\infty \leq 1$.

So, $r^T e$ will be a small number!!!
(Bounded by $m \cdot B$)

$\because \sum r_i e_i \leq \sum^{[m]} |r_i e_i| \leq m \cdot 1 \cdot B$ w.h.p.

Specifically, lets assume that $m \cdot B < \lfloor \frac{q}{4} \rfloor$

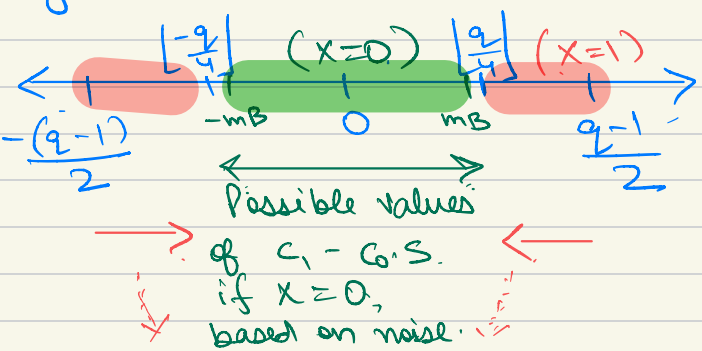
Then, $c_1 - c_0 \cdot s$

$= \underbrace{r^T e}_{\text{norm} \le mB < \lfloor \frac{q}{4} \rfloor} + \underbrace{\lfloor \frac{q}{2} \rfloor \cdot x}_{\begin{array}{l} 0 \text{ if } x = 0 \\ \lfloor \frac{q}{2} \rfloor \text{ if } x = 1. \end{array}}$

So, we can decide if $x = 0$ or 1 based on $\| c_1 - c_0 \cdot s \|$.

$\left(\lfloor \frac{q}{2} \rfloor = \frac{q-1}{2} \right.$ assuming $\left. q > 2 : \text{prime} \right)$

Visually:



link: Possible values of $c_1 - c_0 s$ if $x = 1$.

if $x = 0$, $\| c_1 - c_0 \cdot s \| \le mB < \lfloor \frac{q}{4} \rfloor$

if $x = 1$, $\| c_1 - c_0 \cdot s \| > \lfloor \frac{q}{4} \rfloor$

formally,

if $\| c_1 - c_0 \cdot s \| < \left\lfloor \frac{q}{4} \right\rfloor$: O/P $x = 0$

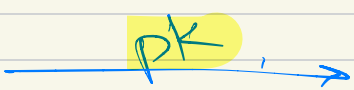
otherwise : O/P $x = 1$.

Correctness: Follows from above analysis.

Security: The semantic Security game:

Chal. A

KGen().

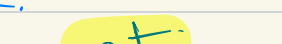


w.l.o.g, $m_0 = 0$, $m_1 = 1$, because we're only encrypting one bit

$\hat{b} \leftarrow_s \{0,1\}$

$ct = Enc(pk, \hat{b})$

$m_{\hat{b}} = \hat{b}$.



we'll make a series of Hybrids to prove that View of A is computationally indistinguishable from random.

View of Adversary in the Semantic Security expt:

Consider

H_0 : View of Adv in above game.

$= (pk, ct)$.

Formally,

$$H_0 = \left\{ (A, b, C_0, C_1) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ s \leftarrow\$ \mathbb{Z}_q^{n} \\ e \leftarrow\$ \chi_B^{m} \\ r \leftarrow\$ \{0,1\}^m \\ \hat{b} \leftarrow\$ \{0,1\}. \end{array} \right\}$$

$$\rightarrow \; A s + e \quad r^T A \quad r^T b + \lfloor \tfrac{q}{2} \rfloor \hat{b}$$

By LWE assumption, $(A, As+e) \approx_c (A, v)$ where v is uniform in $\mathbb{Z}_q^m$.

So, by LWE,

$$H_0 \approx_c \left\{ (A, v, r^T A, r^T v + \lfloor \tfrac{q}{2} \rfloor \hat{b}) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ v \leftarrow\$ \mathbb{Z}_q^{m} \\ r \leftarrow\$ \{0,1\}^m \\ \hat{b} \leftarrow\$ \{0,1\} \end{array} \right\}$$

$= H_1$.

{ we need to somehow prove that distribution of $r^T v + \lfloor \frac{q}{2} \rfloor \hat{b}$ does NOT depend on $\hat{b}$ ... }

We will now use the Leftover Hash Lemma

(very useful in Lattice-crypto)

For $m \geq 2n \log q$,

the following distributions are statistically indistinguishable:

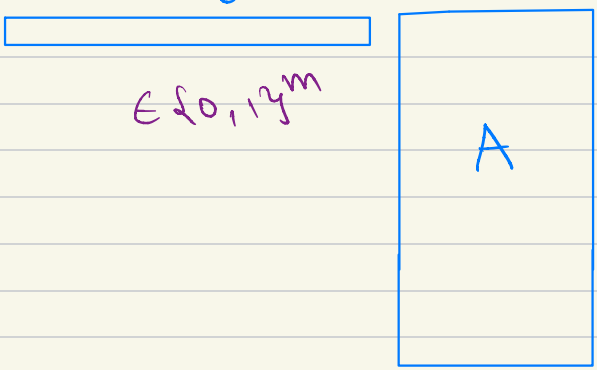
(low statistical distance)

Stronger than computational notion.

$$\left\{ (A, v, r^T A, r^T v) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ r \leftarrow\$ \{0,1\}^m \\ v \leftarrow\$ \mathbb{Z}_q^m \end{array} \right\}$$

$$\approx_s \left\{ (A, v, u, w) : \begin{array}{l} A \leftarrow\$ \mathbb{Z}_q^{m \times n} \\ v \leftarrow\$ \{0,1\}^m \\ u \leftarrow\$ \mathbb{Z}_q^n \\ w \leftarrow\$ \mathbb{Z}_q \end{array} \right\}$$

Basically, Left multiplying by a random binary vector gives a random-looking



vector.

$\in \mathbb{Z}_q^m$

$\approx$ u.r. vector

* Important for m to be $> 2n \log q$ to have enough "randomness" in r.

Back to H_1:

$$\left(\underbrace{A,\ v \xleftarrow{\$} \mathbb{Z}_q^m}_{\text{u.r., so we can apply LHL.}},\ \underbrace{r^T A}_{\approx_s},\ \underbrace{r^T v + \lfloor \tfrac{q}{2} \rfloor \cdot b}_{\approx_s}\right)$$

These are $\approx_s$ uniform, by LHL. So, $r^T v + \lfloor \frac{q}{2} \rfloor \cdot b$ is also $\approx_s$ uniform.

So,

$$H_1 \approx_s \left\{ (A, v, u, w) : \begin{array}{l} A \xleftarrow{\$} \mathbb{Z}_q^{m \times n} \\ v \xleftarrow{\$} \mathbb{Z}_q^m \quad w \xleftarrow{\$} \mathbb{Z}_q \\ u \xleftarrow{\$} \mathbb{Z}_q^n \end{array} \right\}$$

In H_2, the ciphertext is independent of the encrypted bit b.

So, the adv. cannot tell whether (C_0, C_1) is an encryption of 0 or 1.

$\Rightarrow$ Regev Encryption is Semantically Secure, under LWE assumption.

IV. Lattices:

Why is LWE a "lattice" problem?

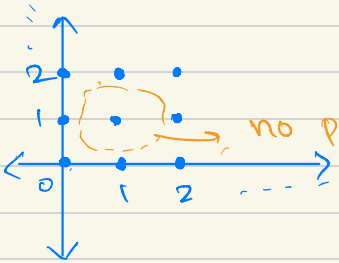
What's a lattice?

An n-dimensional lattice $\mathcal{L}$ is a discrete, additive subspace of $\mathbb{R}^n$.

- Discrete: Every $x \in \mathcal{L}$ has a neighborhood in $\mathbb{R}^n$ where it is the only point.
- Additive subspace: $0^n \in \mathcal{L}$. and

$\forall x, y \in \mathcal{L}$, $x+y \in \mathcal{L}$, $-x \in \mathcal{L}$.

e.g. $\mathbb{Z}^n$: the integer lattice.



for $n=2$:

no point $\in \mathbb{Z}^2$ in this neighborhood $\Rightarrow$ Discrete

Additive ✓

Lattices are defined by a set of Basis vectors:

$$B = \{b_1, \dots, b_k\} \in \mathbb{R}^{n \times k}$$

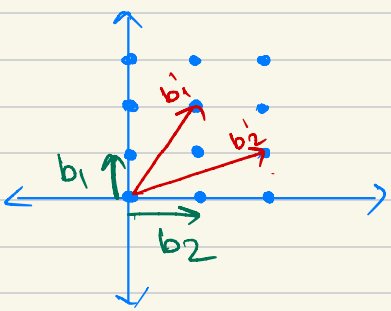
* Basis vectors are linearly independent.

$\mathcal{L}(B)$ = set of points that are integer, linear combinations of basis vectors.

$$\mathcal{L}(B) = \left\{ \sum_{i \in [k]} a_i b_i \;:\; a_i \in \mathbb{Z} \right\}.$$

Here, k is the rank of $\mathcal{L}$.

e.g. $\mathbb{Z}^2$:



(0,1)
(1,0)

b_1, b_2 : Standard basis for $\mathbb{Z}^2$

Any point $(x, y) = x \cdot b_2 + y \cdot b_1$.

(b_1', b_2') is also a basis for $\mathbb{Z}^2$.

Hard problems in Lattices :

(Conjectured to be hard even for Quantum computers!)

1.) Shortest Vector Problem : (SVP)

Given a basis B for a lattice $\mathcal{L}(B)$, find the shortest non-zero vector $v \in \mathcal{L}(B)$. ← lowest norm.

. This is NP-hard (Randomized Reduction)

2.) Closest vector Problem: (CVP)

Given a point $t \in \mathbb{R}^n$, find the closest point in $\mathcal{L}(B)$, i.e.
Find $v \in \mathcal{L}(B)$ that minimizes $\|t - v\|$.

* Similar to LWE:
we can define a lattice w.r.t. A:

$$\mathcal{L}(A) = \{As : s \in \mathbb{Z}_q^n\} + q\mathbb{Z}^n$$

Given $As + e$ (a point t), find the closest $v \in \mathcal{L}(A)$ $\approx$ Find s.

3.) Approximate SVP / CVP:

γ-SVP / γ-CVP : Solve approximately upto a factor of $\gamma > 1$.

eg. γ-SVP : if shortest vector in $\mathcal{L}(B)$ has norm N, find a vector with norm $\leq \gamma N$.

★ For $\gamma = \text{poly}(n)$, γ-SVP/CVP is conjectured to be hard.

★ If γ-SVP is Hard for SOME lattice in $\mathbb{R}^n$, then, LWE(n, m, q, B) is also

← "WORST CASE HARDNESS"

hard !!

i.e. we can base crypto on LWE, which is based on (conjectured) WORST-CASE hardness of a lattice problem !!