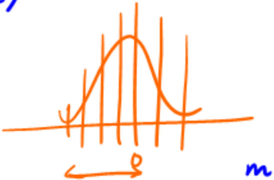
CS6846 – Quantum Algorithms and Cryptography

Learning With Errors and Encryption

Instructor: Shweta Agrawal, IIT Madras
Email: shweta@cse.iitm.ac.in

Learning With Errors: (LWE)

Search LWE $(\mathrm{SLWE}_{n,m,q,\chi})$:



Find $s \in \mathbb{Z}_q^n$ given

$$\{ a_i, \langle a_i, s \rangle + e_i \mid a_i, s \leftarrow \mathbb{Z}_q^n, e_i \leftarrow \chi \}_{i=1}^m$$

Decision Version $(\mathrm{LWE}_{n,m,q,\chi})$:

Simply distinguish above samples from uniform.

Hardness of LWE (Reg05, BLP+13)

It was shown by these (& other) works that for appropriate choices of parameters (n, m, q, χ), $SLWE_{n,m,q,\chi}$ is as hard as solving worst case lattice problems such as SIVP, GapSVP with approx factor $\text{poly}(n) \cdot q/\gamma$ ← smth related to χ.

Note: Best known algorithms for 2^k approximate gapSVP & SIVP run in time $2^{O(n/k)}$.

Hermite Normal Form:

The HNF or short-secret LWE is like LWE but the secret s is also chosen from error distribution χ.

Public Key Encryption:

Keygen (1^λ): Sample $\underline{s} \leftarrow \underline{\chi^n}$, $A \leftarrow \mathbb{Z}_q^{n\times n}$
error $e \leftarrow \chi^n$.

$$PK = \left(A,\ y^T = s^T A + e^T\right) \in \mathbb{Z}_q^{n\times n} \times \mathbb{Z}_q^n$$

$$SK = s.$$

$PK = A, \; s^T A + e^T$

$SK = s.$

Enc (PK, m): ← bit

1). Sample $r, x \leftarrow \chi^n$, $x' \leftarrow \chi$

2). $a = Ar + x$, $b = y^T r + x' + m\lfloor \frac{q}{2} \rfloor$ mod q.

3) output (a, b).

Dec (s, a, b): $s^T a = s^T(Ar + x)$

$$= \cancel{s^T A r} + s^T x$$

$$b = y^T r + x' + m\lfloor \tfrac{q}{2} \rfloor$$

$$= (s^T A + e^T) r + x' + m\lfloor \tfrac{q}{2} \rfloor$$

$$= \cancel{s^T A r} + e^T r + x' + m\lfloor \tfrac{q}{2} \rfloor$$

$$b - s^T a = \overbrace{(s^T x + e^T r + x')}^{\le B} + m\lfloor q/2 \rfloor$$

Correctness:

Let $\mathrm{Supp}(x) \subseteq \left(-\sqrt{q/4(2n+1)},\ \sqrt{q/4\,(2n+1)}\right)$

$$\text{Final error} = b - s^T a = \underset{\overset{\uparrow}{n}}{s^T}\,\underset{\overset{\downarrow}{n}}{x} + \underset{\overset{\downarrow}{n}}{e^T}\,\underset{\overset{\downarrow}{n}}{r} + \underset{\overset{\downarrow}{1}}{x'}$$

Simple Bound $\quad q/4(2n+1) \cdot (2n+1) = \dfrac{q}{4}$

Security:

Recall: Attacker sees the public key,
& ciphertext for random bit.
Must guess bit.

Hyb 0

$PK = A, \; y^T = s^T A + e^T$

Random bit

$$CT = \underbrace{Ar + x}_{a}, \; \underbrace{\left(y^T r + x'\right) + m\lfloor q/2 \rfloor}_{b}$$

≈ Random

Hyb 0.5:

CT = Function of the PK.

PK, f(PK)

Hyb 1: Change the public key

$\widetilde{PK} = A$, random

By ssLWE

Hyb2: $\widetilde{Enc}(\widetilde{pk}, m) = a, b' + m\lfloor q/2 \rfloor \bmod q$

where a, b' are random.

$$CT : \begin{pmatrix} A \\ y^T \end{pmatrix} r + \begin{pmatrix} x \\ x' \end{pmatrix} + \begin{pmatrix} 0 \\ m\lfloor q/2 \rfloor \end{pmatrix} \bmod q.$$

Now CT

$$\underbrace{a, \; b' + m\lfloor \tfrac{q}{2} \rfloor}_{} \rightarrow \text{random.}$$

$$PK \quad \underbrace{A, \; y}_{\text{random}}$$

Hyb 3: Change $b' + m\lfloor \frac{q}{2} \rfloor$ to random

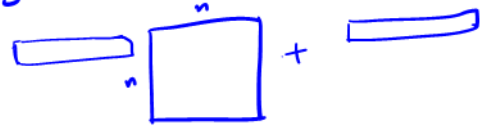
PK = A, random y

CT = random a, random b.

$$\text{Dec}: \underbrace{\underline{b} - \underline{s^T \underline{a}}}_{\text{random.}}$$

$$y = s^T A + e^T$$

$A,$

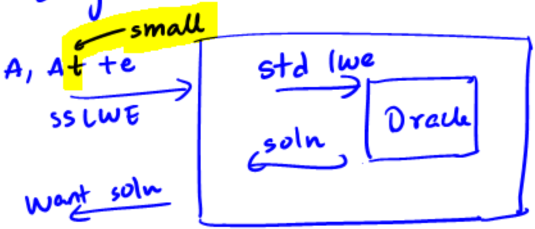


$$A, \underbrace{s^T A}_{\text{recover } s.} + e^T$$

Connection between ssLWE & LWE:

Lemma: There is a polynomial time reduction from ssLWE (n, m, q, χ) to LWE (n, m, q, χ) & one from LWE (n, m, q, χ) to ssLWE $(n, m+n, q, \chi)$

Proof: Say that we have oracle to solve LWE

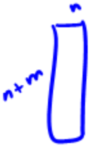


$t' \leftarrow \mathbb{Z}_q^n$

$At + e + At'$

$= A(\underbrace{t + t'}_{\text{random}}) + e$

Now I have solver for ss LWE



A, $At + e$.

Write as
$$\begin{array}{cc} & n \\ \begin{array}{c} n \\ m \end{array} & \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \end{array}, \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} t + \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

W.h.p. A_1 is invertible.

$$b_1 = A_1 t + e_1 \Rightarrow t = A_1^{-1}(b_1 - e_1)$$

$$b_2 = A_2 t + e_2 = A_2 A_1^{-1} b_1 - A_2 A_1^{-1} e_1 + e_2$$

$$\therefore \underbrace{A_2 A_1^{-1}}_{A'} e_1 - e_2 = \underbrace{A_2 A_1^{-1} b_1 - b_2}_{\text{can compute}}.$$

Have $(A', A'e_1 - e_2) \Rightarrow$ solve!