CS555 Cryptography Notes - Lecture 12

Public-Key Encryption from LWE

October 2, 2025

1 Regev's Public-Key Encryption Scheme

$c = (a, b = \langle a, s\rangle + e + \mu\lfloor q/2\rfloor) \leftarrow \mathrm{Enc}_s(\mu)$

$c' = (a', b' = \langle a', s\rangle + e' + \mu'\lfloor q/2\rfloor) \leftarrow \mathrm{Enc}_s(\mu')$

This scheme is additively homomorphic.

μ and μ' are the 2 bits. How to get ciphertext of $\mu + \mu'$ from c and c'?

$$c + c' = (a + a', b + b') = (\langle a + a', s\rangle + (e + e') + (\mu + \mu')\lfloor q/2\rfloor)$$

$c + c'$ is an encryption of $\mu + \mu' \bmod 2$ if $e + e'$ is not too large.

Multiplication of μ and μ' is also possible but the noise grows faster.

How do we clean up the noise periodically to perform operations?

For now, note that the error increases when you add two ciphertexts. That is, $|e_{\mathrm{add}}| \approx |e_1 + e_2| \leq 2B$.

Setting $q = n^{\log n}$ and $B = \sqrt{n}$ (for example) lets us support any polynomial number of additions.

2 Public-Key Encryption

2.1 Initial Idea

Let c_0 be the encryption of 0 and c_1 be the encryption of 1.
Public key and c_1 respectively for 1.
If you want to encrypt 0, output c_0.
If you want to encrypt 1, output c_1.

This is not a valid encryption. But the GM construction makes a similar idea

possible. If the bit is 0, a quadratic residue is the output. If the bit is 1, a non-quadratic residue is the output. This works because the outputs are random elements of equal-size sets.
Therefore, we need fresh encryptions of 0 and 1.

2.2 Better Idea

Public key has many encryptions of 0 and an encryption of 1 (c_1).

- **Secret key** sk = Uniformly random vector $s \in \mathbb{Z}_q^n$
- **Public key** pk: for i from 1 to $m = \text{poly}(n)$

$$\mathbf{c}_i = (\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$$

- **Encrypting a bit μ:** pick m random bits $r_1, \ldots, r_m$

$$\sum_{i=1}^{m} r_i \mathbf{c}_i + (0, 0, \ldots, 0, \mu) \cdot \left\lfloor \frac{q}{2} \right\rfloor$$

Correctness: As long as $|\sum r_i e_i| < q/4$ is small enough.

2.3 Correctness Proof

is distributed when A, b are random (and public). If r is truly random, so is $r[A|b]$. But r is neither 0 or 1, but uniform over $\text{mod}\frac{q}{2}$. Nevertheless r has entropy. Leftover hash lemma tells us that matrix multiplication turns (sufficient) entropy into true randomness. We need $m \gg (n+1)\log q$. This means in LWE, the sampling used not be random or Gaussian or be in any fixed distribution; it just needs sufficient (and nearly large) entropy being the compared to the entropy that the output needs. Here, we need $m > (n+1)\log q + 2\log(1/\epsilon)$ where ϵ is the distinguisher's advantage $\approx e^{-n}$. We use this lemma to prove security.

2.4 Security Proof

Theorem: Under decisional LWE, the scheme is IND-secure. In fact, a ciphertext together with the public key is pseudorandom.

Hybrid 1
Change the public key to random (from LWE):

$$\widetilde{pk} = (A, b), \quad \widetilde{c} = \text{Enc}_{\widetilde{pk}}(\mu) = (rA, rb + \mu\lfloor q/2 \rfloor)$$

Hybrids 0 and 1 are comp. indist. by decisional LWE.

Hybrid 2
Change rA, rb to random (using Leftover Hash Lemma):

$$\widetilde{pk} = (A, b), \quad \widetilde{c} = \mathrm{Enc}_{\widetilde{pk}}(\mu) = (u, u' + \mu\lfloor q/2 \rfloor)$$

Hybrids 1 and 2 are statistically indistinguishable by LHL.

Hybrid 3
Change $u' + \mu\lfloor q/2 \rfloor$ to a random bit:

$$\widetilde{pk} = (A, b), \quad \widetilde{c} = \mathrm{Enc}_{\widetilde{pk}}(\mu) = (u, u')$$

Hybrids 2 and 3 are perfectly indistinguishable.

3 LWE with Small Secrets

Theorem: LWE with small secrets is as hard as LWE.
Given:

$$A, \quad As + e$$

where $A \in \mathbb{Z}_q^{m \times n}$, s from χ^n, e from χ^m.
When we add with high probability it has full rank.
Known:

$$A_1 s + e_1 = b_1, \quad A_2 s + e_2 = b_2$$

approximately.
Proof:

$$s : A_1^{-1}(b_1 - e_1) \Rightarrow A_2 A_1^{-1}(b_1 - e_1) + e_2 = b_2$$

Still a random matrix, but $n \times n$ now, then new noise:

$$= A_2 A_1^{-1} e_1 + b_2 \ \Rightarrow A_2 A_1^{-1} b_1 \Rightarrow \text{Another LWE},$$

If we think of e_1 as a secret, then $-A_2 A_1^{-1} e_1$ is just another LWE term since $A_1 A_1^{-1}$ is random. Since solving the first equation with s is hard, solving the equation with s eliminated should still be hard. Therefore, LWE with small secrets is still just as hard.

4 New Encryption Scheme

- **Secret key** sk = Small secret s from χ^n

- **Public key** pk: for i from 1 to n

$$(A, b = As + e)$$

- **Encrypting a message bit** μ**:** pick a random vector r from χ^n

$$(rA + e', rb + e'' + \mu\lfloor q/2 \rfloor)$$

- **Decryption:** compute

$$(rb + e'' + \mu\lfloor q/2 \rfloor) - (rA + e')s$$

  and round to nearest multiple of $q/2$.

Hybrid 0
Original public key, ciphertext pair.

$$pk = (A, b = As + e), \quad c = \mathrm{Enc}(pk, \mu) = (rA + e', rb + e'' + \mu\lfloor q/2 \rfloor)$$

Hybrid 1
Change the public key to random (from LWE):

$$\widetilde{pk} = (A, b), \quad \widetilde{c} = \mathrm{Enc}_{\widetilde{pk}}(\mu) = (rA + e', rb + e'' + \mu\lfloor q/2 \rfloor)$$

Hybrids 0 and 1 are comp. indist. by decisional LWE

Hybrid 2
Change rA, rb to random (using Leftover Hash Lemma):

$$\widetilde{pk} = (A, b), \quad \widetilde{c} = \mathrm{Enc}_{\widetilde{pk}}(\mu) = (a', b' + \mu\lfloor q/2 \rfloor)$$

Hybrids 1 and 2 are comp. indist. by LWE.

5 Big Open Question

Public-key Encryption from one-way functions?
Negative Result in one case: Roughly, any construction of a public-key encryption scheme in an "OWF-oracle-model" can be broken with $O(Q^2)$ queries if the honest parties make at most Q queries.
This is tight w.r.t. Merkle puzzles.

6 Practical Considerations

1. Public-key infrastructure: a trustworthy directory of identities and public keys.
2. Public-key encryption is orders of magnitude worse slower than secret-key encryption:
   - Except El-Gamal, the encryptions are bit-by-bit! Not truly secure
   - Exponentiation takes $O(n^2)$ as opposed to linear time for secret-key (AES).
   - n itself is large for PKE: RSA $n \geq 2048$, AES: $n = 128$.

   Can solve problem 1 and 2, 3 using hybrid encryption.