CS6222 Cryptography

Topic: Introduction
Lecturer: Wei-Kai Lin (TA: Arup Sarker)

Date: Dec, 2024
Scriber: Chen Gong

---

1 Encryptions

Encryption is widely used in securing communications, protecting sensitive data, and ensuring privacy in digital systems. This course introduced private-key encrytion (PKE) earlier, but PKE is deferred to this section. It is because the hardness assumption: PKE and other primitives can be based on the existence of OWFs, but we need other assumptions to obtain PKE.

1.1 Learning with Errors, LWE

In this section, we redefine the representation of $\mathbb{Z}_q$ as follows:

$$\mathbb{Z}_q \equiv \{-\lfloor\frac{q-1}{2}\rfloor, \ldots, 0, \ldots, \lfloor\frac{q}{2}\rfloor\}.$$

The modular arithmetic remains unchanged under this representation. We also define an element in $\mathbb{Z}_q$ as *small* if its value is close to 0.

Now, consider a matrix $A \in \mathbb{Z}_q^{m\times n}$ and a vector $\vec{s} \in \mathbb{Z}_q^n$, where $m \gg n$. Define $\vec{t}' := A \cdot \vec{s} \in \mathbb{Z}_q^m$. Using standard linear algebra, solving $A \cdot \vec{x} = \vec{t}'$ is efficient when A is full rank.

The Learning with Errors (LWE) problem introduces the following modification:

- A and $\vec{s}$ are chosen as above.
- An additional error vector $\vec{e} \in \mathbb{Z}_q^m$ is sampled such that $\|\vec{e}\|$ is *small* (each coordinate of $\vec{e}$ is small in $\mathbb{Z}_q$).

Given this setup, the goal is to find $\vec{x} \in \mathbb{Z}_q^n$ and $\vec{y} \in \mathbb{Z}_q^m$ such that

$$A \cdot \vec{x} + \vec{y} = \vec{t},$$

where $\vec{t} := A \cdot \vec{s} + \vec{e}$.

The hardness of the LWE problem is parameterized by q, m, n, and the bound on *small*. When parameters are chosen appropriately, this problem is believed to be hard even for quantum algorithms. This belief is supported by reductions from *lattice problems*, which are beyond the scope of this discussion.

Definition, Learning with Errors (LWE) Assumption.

> We define the decisional $\mathrm{LWE}_{m,q,\psi}$ problem to be quantum-hard if, for any quantum polynomial-time distinguisher D, there exists a negligible function ϵ such that for all $n \in \mathbb{N}$,
>
> $$\Pr[A \leftarrow \mathbb{Z}_q^{m\times n}, \vec{s} \leftarrow \psi^n, \vec{e} \leftarrow \psi^m : D(A, A\cdot\vec{s}+\vec{e}) = 1]$$
> $$-\Pr[A \leftarrow \mathbb{Z}_q^{m\times n}, \vec{t} \leftarrow \mathbb{Z}_q^m : D(A, \vec{t}) = 1] \le \epsilon(n),$$
>
> where $m, q \in \mathbb{N}$ are functions of n, and ψ is an efficiently samplable distribution over $\mathbb{Z}_q$.

Definition, Corollary.

Suppose that $\gcd(q, 2) = 1$. If $(A, A \cdot \vec{s} + \vec{e}) \approx (A, \vec{t})$, then $(A, A \cdot \vec{s} + 2 \cdot \vec{e}) \approx (A, \vec{t})$.

Proof: Because $\gcd(q, 2) = 1$, multiplication by 2 is a bijection in $\mathbb{Z}_q$. Therefore, for any matrix $A \in \mathbb{Z}_q^{m \times n}$, the scaled matrix $2A$ is identically distributed to A, as each entry in $2A$ is uniformly distributed over $\mathbb{Z}_q$, just as in A. Similarly, $\vec{t}$, which is independently sampled from $\mathbb{Z}_q^m$, is unaffected by this scaling.

Now consider the two distributions:

$$(A, A \cdot \vec{s} + \vec{e}) \quad \text{and} \quad (A, A \cdot \vec{s} + 2 \cdot \vec{e}).$$

Since $(A, A \cdot \vec{s} + \vec{e}) \approx (A, \vec{t})$ by assumption, the second distribution can be analyzed as follows.

We can rewrite $A \cdot \vec{s} + 2 \cdot \vec{e}$ as:

$$A \cdot \vec{s} + 2 \cdot \vec{e} = A \cdot \vec{s} + \vec{e} + \vec{e}.$$

Given that $\vec{e}$ is small (with respect to $\mathbb{Z}_q$), the addition of $\vec{e}$ to itself does not significantly alter its statistical properties as long as $\vec{e}$ remains small. Thus, the noise $2 \cdot \vec{e}$ remains statistically similar to $\vec{e}$.

Combining these observations: 1. A and $2A$ are identically distributed. 2. The added noise $2 \cdot \vec{e}$ does not fundamentally change the indistinguishability property.

We conclude that:

$$(A, A \cdot \vec{s} + 2 \cdot \vec{e}) \approx (A, \vec{t}).$$

2 An Encryption Based on LWE

Suppose we are given the decisional $\text{LWE}_{m,q,\psi}$ assumption:

$$\{A, A \cdot \vec{s} + \vec{e}\} \approx \{A, \vec{t}\},$$

where $A \in \mathbb{Z}_q^{m \times n}$, $\vec{s} \in \mathbb{Z}_q^n$, $\vec{e} \in \mathbb{Z}_q^m$ is a small noise vector, and $\vec{t} \in \mathbb{Z}_q^m$ is uniformly random.

Now, let $d \in \mathbb{N}$ such that $\gcd(d, q) = 1$. By a simple reduction, we claim that:

$$\{A, A \cdot \vec{s} + d\vec{e}\} \approx \{A, \vec{t}\}.$$

Reduction: Since $\gcd(d, q) = 1$, multiplication by d is a bijection in $\mathbb{Z}_q$. As a result: 1. The matrix A remains uniform over $\mathbb{Z}_q^{m \times n}$. 2. The vector $\vec{t}$ remains uniform over $\mathbb{Z}_q^m$.

Multiplying the original LWE assumption by d, we observe that the noise $d\vec{e}$ retains its "small" property (scaled appropriately) and does not alter the indistinguishability. Therefore, the modified distributions $\{A, A \cdot \vec{s} + d\vec{e}\}$ and $\{A, \vec{t}\}$ are indistinguishable under the same assumption.

Application to Secret-Key Encryption: This property enables the construction of a *secret-key encryption scheme*. The secret key $\vec{s}$ is derived from the LWE problem, and the ciphertext incorporates the term $A \cdot \vec{s} + d\vec{e}$, leveraging the inherent hardness of LWE to ensure security. The specific details of the encryption scheme can be tailored to exploit this reduction.

2.1 Construction: Secret Key Encryption from LWE

Parameters: Let $m, q \in \mathbb{N}$ be functions of n, and let ψ be a distribution over $\mathbb{Z}_q$.

- $\mathsf{Gen}(1^n)$: Output $k := \vec{s} \leftarrow \psi^n$, where $\vec{s}$ is sampled from ψ^n.
- $\mathsf{Enc}_k(m)$: For a binary message $m \in \{0,1\}$, sample $\vec{a} \leftarrow \mathbb{Z}_q^n$ and $e \leftarrow \phi$, and output the ciphertext $c := (\vec{a}, t = \vec{a} \cdot \vec{s} + 2e + m)$, where all arithmetic is performed in $\mathbb{Z}_q$.
- $\mathsf{Dec}_k(c)$: Given $c = (\vec{a}, t)$, output

$$m' := (t - \vec{a} \cdot \vec{s} \mod 2).$$

Correctness: The correctness follows directly from the construction, as the decryption step correctly retrieves m modulo 2.

CPA Security: The secret-key CPA (Chosen Plaintext Attack) security of the scheme can be proven via a reduction R. Whenever the adversary A queries the encryption oracle for a message m, the reduction R uses the next row from its LWE instance $(\vec{a}, \vec{a} \cdot \vec{s} + e)$ and computes the ciphertext by adding $2e + m$. The indistinguishability of ciphertexts then relies on the decisional LWE assumption.

3 Homomorphic Encryption

3.1 Preliminaries

Definition, Homomorphic encryption.

A (public or secret key) encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be *homomorphic* if it provides efficient operations $(\mathsf{Add}, \mathsf{Mul})$ that satisfy the following syntax and correctness guarantees:

- **Addition:** For any messages $m_0, m_1 \in \mathbb{Z}_2$,

$$\Pr_k \left[(c_i \leftarrow \mathsf{Enc}_k(m_i))_{i \in \{0,1\}}; \mathsf{Dec}_k(\mathsf{Add}(c_0, c_1)) = m_0 + m_1 \right] = 1.$$

- **Multiplication:** For any messages $m_0, m_1 \in \mathbb{Z}_2$,

$$\Pr_k \left[(c_i \leftarrow \mathsf{Enc}_k(m_i))_{i \in \{0,1\}}; \mathsf{Dec}_k(\mathsf{Mul}(c_0, c_1)) = m_0 \cdot m_1 \right] = 1.$$

- **Multiplication by a Constant:** For any message $m_0 \in \mathbb{Z}_2$ and constant $m_1 \in \mathbb{Z}_2$,

$$\Pr_k \left[c_0 \leftarrow \mathsf{Enc}_k(m_0); \mathsf{Dec}_k(\mathsf{Mul}(c_0, m_1)) = m_0 \cdot m_1 \right] = 1.$$

Here, the message space and arithmetic operations $(+, \cdot)$ are defined over $\mathbb{Z}_2$. However, this definition can be generalized to other algebraic structures.

3.2 Discussion of Homomorphic Encryption

Correctness vs. Security

The homomorphic operations Add and Mul in a homomorphic encryption scheme are primarily concerned with *correctness*, not *security*. Correctness ensures that the operations performed on ciphertexts yield valid results that can be decrypted back to the correct plaintext. Security guarantees, such as CPA or CCA security, are handled separately by the encryption scheme itself and are orthogonal to the correctness of homomorphic operations.

In practice, correctness is sometimes relaxed to allow for a negligible probability of error. This relaxation is often necessary in technically complex constructions, particularly in noisy encryption schemes such as those based on the Learning with Errors (LWE) problem.

Single-Hop Homomorphic Operations

The definition of homomorphic encryption provided above considers only *single-hop* operations. This means that the operations Add and Mul are defined for ciphertexts that are freshly encrypted by Enc. Specifically, for ciphertexts $c_0 = \mathsf{Enc}_k(m_0)$ and $c_1 = \mathsf{Enc}_k(m_1)$, the scheme guarantees correctness for $\mathsf{Add}(c_0, c_1)$ and $\mathsf{Mul}(c_0, c_1)$. However, the correctness of these operations is not guaranteed for ciphertexts that are outputs of previous homomorphic operations.

Multi-Hop and Unlimited-Hop Operations

To support more complex computations, we extend homomorphic operations to *multi-hop* scenarios:

- **t-Hop Operations:** A scheme supports t-hop homomorphic operations if Add and Mul remain correct for ciphertexts that are outputs of up to $(t-1)$-hop homomorphic operations. For example, if $t = 3$, this means that Add or Mul can be performed on ciphertexts produced by 2-hop operations, such as $\mathsf{Add}(\mathsf{Add}(c_0, c_1), c_2)$.
- **Unlimited-Hop Operations:** Ideally, we want homomorphic operations that work for arbitrarily many hops. This enables the evaluation of arbitrary boolean circuits directly on ciphertexts. Since addition $(+)$ and multiplication $(\cdot)$ over $\mathbb{Z}_2$ correspond to XOR and AND, unlimited-hop homomorphic encryption allows any boolean circuit to be computed on encrypted data. This capability is known as *Fully Homomorphic Encryption (FHE)*.

3.3 Homomorphic Encryption for a Circuit Class

Definition: Homomorphic Encryption for a Circuit Class

> Let $\mathcal{C}$ be a class of circuits. We say that an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is *homomorphic* for $\mathcal{C}$ if the following holds:
>
> For any $C \in \mathcal{C}$, for any inputs $m_1, \ldots, m_\ell \in \{0,1\}$, where ℓ is the input size of C, let $k \leftarrow \mathsf{Gen}(1^n)$ be the encryption key. Then:
>
> $$\Pr_k \left[(c_i \leftarrow \mathsf{Enc}_k(m_i))_{i \in [\ell]}; \mathsf{Dec}_k\left(\mathsf{Eval}(C, c_1, \ldots, c_\ell)\right) = C(m_1, \ldots, m_\ell) \right] = 1.$$

The above secret-key encryption scheme based on LWE supports direct homomorphic addition. To see why, consider two ciphertexts encrypted using the same key $\vec{s}$:

$$c_0 := (\vec{a}_0, \vec{a}_0 \cdot \vec{s} + 2e_0 + m_0), \quad c_1 := (\vec{a}_1, \vec{a}_1 \cdot \vec{s} + 2e_1 + m_1).$$

The homomorphic addition $\mathsf{Add}(c_0, c_1)$ is defined as the coordinate-wise addition of the ciphertexts:

$$\begin{aligned}\mathsf{Add}(c_0, c_1) &:= (\vec{a}_0 + \vec{a}_1, (\vec{a}_0 \cdot \vec{s} + 2e_0 + m_0) + (\vec{a}_1 \cdot \vec{s} + 2e_1 + m_1)) \\ &= (\vec{a}', \vec{a}' \cdot \vec{s} + 2e' + (m_0 + m_1)),\end{aligned}$$

where $\vec{a}' = \vec{a}_0 + \vec{a}_1$ and $e' = e_0 + e_1$.

The operation "multiplication by a constant" is similarly defined as a coordinate-wise multiplication by the plaintext constant m_1.

Correctness of Addition and Multiplication by a Constant

Correctness holds as long as the accumulated error e' satisfies $e' \leq q/4$ before reduction modulo q. This constraint ensures that decryption still recovers the correct plaintext. Consequently, we can perform up to $O(q/B)$ operations (additions or multiplications by constants) while preserving correctness, where B and q are the LWE parameters that bound the error and modulus, respectively.

Multiplicative Homomorphism

Homomorphic multiplication is more involved and requires additional techniques to manage error growth. Before delving into its details, we first demonstrate an application of additive homomorphic encryption to illustrate its utility.

3.4 Public-key encryption

Definition: Public-key encryption

A $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be a *public-key encryption scheme* if the following syntax, correctness, and security hold:

1. Gen is a PPT algorithm, $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$.
2. Enc is a PPT algorithm, for all pk and all $m \in \{0,1\}$, $c \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$.
3. Dec is a deterministic algorithm, for all sk and c, $m \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c)$ such that $m \in \{0,1\} \cup \{\perp\}$.

Correctness: For all $n \in \mathbb{N}, m \in \{0,1\}$,

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) : \mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m)) = m\right] = 1.$$

Security: For all NUPPT D, there exists a negligible function $\epsilon(\cdot)$ such that for all $n \in \mathbb{N}$, $m_0, m_1 \in \{0,1\}$, D distinguishes between the following distributions with probability at most $\epsilon(n)$:

$$\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) : (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_0))\}_n,$$
$$\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) : (\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_1))\}_n.$$

With the above definitions, we can immediately derive some fundamental impossibility results:

- **Perfect secrecy is impossible:** Given the public key pk, the adversary can try to encrypt all possible messages using all possible randomness, effectively gaining access to the entire message space.
- **Deterministic encryption is also impossible:** The adversary can repeatedly encrypt the same message and obtain the same ciphertext, which reveals information about the message.

However, **IND-CPA security** (indistinguishability under chosen plaintext attack) follows directly from the definitions. Another key difference from secret-key encryption is that security for *long* messages is directly implied.

Lemma: Multi-Message Security

If $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a secure public-key encryption scheme, then for any polynomial function* $\ell(n)$*, and for all messages* $m_0, m_1 \in \{0,1\}^{\ell(n)}$*, the following two distributions are indistinguishable:*

$$\left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) : \left(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_{0,1}), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_{0,\ell(n)})\right)\right\}_n$$

and

$$\left\{(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n) : \left(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m_{1,1}), \ldots, \mathsf{Enc}_{\mathsf{pk}}(m_{1,\ell(n)})\right)\right\}_n$$

where $m_{b,i}$ *denotes the* i*-th bit of* m_b*.*

Moreover, the adversary may *adaptively* choose $(m_{0,i}, m_{1,i})$ depending on the previous ciphertexts $\mathsf{Enc}_{\mathsf{pk}}(m_{b,i'})$ for $i' < i$. This is referred to as **multi-message security**, which is also implied by single-message security.

3.5 Public-Key Encryption from Additive Homomorphic Encryption

Let (G, E, D, Eval) be a secret-key homomorphic encryption scheme for the class of functions $\mathcal{C}_\ell := \{f_z : z \in \{0,1\}^\ell\}$, where $\ell := \ell(n)$ is a polynomial function of the security parameter n. Each function $f_z(x)$ computes the inner product $z \odot x$ of two ℓ-bit vectors. Then, the following $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ defines a secure public-key encryption scheme:

- $\mathsf{Gen}(1^n)$: Let $\mathsf{sk} = k \leftarrow G(1^n)$. Sample a random ℓ-bit string $r = (r_1, \ldots, r_\ell) \leftarrow \{0,1\}^\ell$ such that $r \neq 0^\ell$. Compute $R_1 \leftarrow E_k(r_1), \ldots, R_\ell \leftarrow E_k(r_\ell)$. Output sk and the public key:

$$\mathsf{pk} = (r_1, \ldots, r_\ell, R_1, \ldots, R_\ell).$$

- $\mathsf{Enc}_{\mathsf{pk}}(m)$: Sample a random ℓ-bit string $u \in \{0,1\}^\ell$ such that $r \odot u = m$. Output the ciphertext:

$$c := \mathsf{Eval}(f_u, R_1, \ldots, R_\ell).$$

- $\mathsf{Dec}_{\mathsf{sk}}(c)$: Simply output $D_{\mathsf{sk}}(c)$.

The correctness follows because $r \neq 0^\ell$ implies the existence of u, and by the correctness of Eval. The efficiency is also direct (it only requires careful sampling of r and u). The security is outlined below.

Firstly, consider a hybrid scheme where in Gen, we encrypt $R_i \leftarrow \mathsf{Enc}_k(0)$ instead of $R_i \leftarrow \mathsf{Enc}_k(r_i)$. By the CPA security of (G, E, D), this hybrid scheme is indistinguishable from the real $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ scheme.

Secondly, in the hybrid scheme, the ciphertext c is an encryption of 0 in the (G, E, D) scheme. While this may seem sufficient, we must recognize that c depends on u, and u in turn depends on m. In fact, different encryptions of 0 could potentially reveal something about m.

The crucial observation is that $|c|$ is short. Let $t := |c|$ be the output size of Eval, and let $\ell := 4t$. Informally, a t-bit ciphertext c is too short to provide significant information about the $4t$-bit string u, and thus it is information-theoretically hiding m.

To formalize this, we apply the Leftover Hash Lemma. Let $F_R(\cdot) := \mathsf{Eval}(f_\odot, R_1, \ldots, R_n)$. The adversary is given $(r, F_R(u))$ and aims to compute $m = r \odot u$, where R is independent of r. We can rewrite the computation as:

$$\mathrm{Ext}(r, u) := (r, m = r \odot u).$$

Note that r is uniform, u depends on m, and $F_R(u)$ is t-bit long. Observe that, for any fixed m (with a length of 1 bit) and any fixed $F_R(u)$ (with $t = \ell/4$ bits), the min-entropy of u is at least $\ell/2$.

Since $h_r(u) = r \odot u$ is a universal hash family, by the Leftover Hash Lemma, $r \odot u$ is statistically close to uniform, with a statistical difference of at most $2^{-\Omega(\ell/2)}$.

4 Conclusions

Encryption is the process of converting plain data into a coded format to protect it from unauthorized access. It uses algorithms and keys to ensure that only authorized parties can decrypt and access the original data. Common encryption methods include symmetric encryption and asymmetric encryption.