Encryption Schemes

Notes by Yael Kalai

MIT - 6.5610
Lecture 4 (February 14, 2024)

> ***Warning:*** This document is a rough draft, so it may contain bugs. Please feel free to email me with corrections.

Outline

- public-key encryption
- Learning with Error assumption

Last time we defined the notion of a CPA-secure symmetric key encryption and showed how to construct it from any PRF family.

There is a stronger definition of security, which is the golden standard for security, known as security against chosen ciphertext attacks (or CCA-security). Here the attacker can also get the decryption of ciphertexts of its choice.

Public-Key Encryption

So far we focused on symmetric encryption, which assumes that both parties share a secret key k. What if the parties do not have a shared secret key? In the first lecture Henry showed how they can generate such a secret key by communicating over a public network.

Recall that during the first lecture Henry covered Merkle's key-exchange scheme, which only offers mild security guarantees, though he mentioned the secure Diffie-Hellman key-exchange which is secure under the Discrete-Log assumption (which is quantumly broken).

Definition 1. A public key encryption scheme consists of three PPT algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ where

- Gen is a PPT algorithm that takes as input the security parameter 1^λ and outputs a key pair $(\mathsf{pk}, \mathsf{sk})$.
- Enc is a PPT algorithm that takes as input a public key pk and a message $m \in \mathcal{M}_\lambda$ and outputs a ciphertext $\mathsf{Enc}(\mathsf{pk}, m)$.
- Dec is a polynomial time algorithm that takes as input a secret key sk and a ciphertext ct and outputs a message m.

The correctness guarantee is that for every $\lambda \in \mathbb{N}$ and for every $m \in \mathcal{M}_\lambda$,

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m] = 1$$

where the probability is over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and over the randomness used by Enc.

Sometimes we relax the definition of perfect completeness and allow a negligible probability of error over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and over the randomness used by Enc.

Definition 2. A public key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is said to be secure if for every $\lambda \in \mathbb{N}$ and every $m_0, m_1 \in \mathcal{M}_\lambda$ it holds that

$$(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_0)) \approx (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, m_1))$$

where the distributions are over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$ and over the randomness of Enc.

Are we assuming here that the adversary is given only a single ciphertext? No! Notice that in the public key setting ciphertexts can be computed efficiently given pk, and pk is given in both distributions above.

As mentioned above one can consider a stronger definition, called CCA security (security against adaptive chosen message attacks) where the adversary is given a decryption oracle.

Typically, in cryptography classes the first public key encryption scheme taught is El-Gamal, which is based on the Diffie-Hellman key-exchange protocol (and thus on the Discrete Log assumption), or the RSA encryption scheme which was developed here at MIT by Rivest, Shamir and Adleman (Ron Rivest will give a guest lecture in our class later this semester!). The RSA encryption scheme is also known to be broken given a quantum computer since it relies on an assumptions stronger than factoring.

We depart from this tradition, and focus on constructions that are believed to be post-quantum secure. Typically, such constructions are based on lattices, which is a mathematical construct that is quite different from the factoring bases or the discrete-log based ones. In particular, we will focus on the Learning with Errors (LWE) assumption.

*Learning with Errors (*LWE*) Assumption*

The LWE assumption was introduced by Regev [1] in a breakthrough work for which he won the Godel prize. Loosely speaking, the LWE assumption asserts that it is hard to solve noisy linear equations over finite fields. We will not define the notion of a finite field, rather we will focus on the specific class of finite fields which contain all the elements $\{0, 1, \ldots, q-1\}$, where q is a prime, and where addition and multiplication is done modulo q.

Such a finite field is often denoted by $\mathsf{GF}[q]$, which is short for Galois Field of order q.

We note that if q is not a prime then this would not be a field. One of the requirements of a field is that every non-zero element has a multiplicative inverse, which holds if and only if q is a prime (unless multiplication is defined in a more complicated way in which case we can handle power of primes as well).

The LWE assumption is a family of assumptions associated with paramaters $n = n(\lambda)$, $m = m(\lambda)$, $q = q(\lambda)$ and an error distribution χ over $\mathsf{GF}[q]$.

Definition 3. The $\mathsf{LWE}_{n,m,q,\chi}$ assumption asserts that

$$(A, sA + e) \approx (A, U)$$

where $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$ $s \xleftarrow{R} \mathbb{Z}_q^n$, $e \leftarrow \chi^m$.

Note that there are some parameters in which the $\mathsf{LWE}_{n,m,q,\chi}$ assumption is clearly true. For example, if χ is the uniform distribution over $\mathsf{GF}[q]$ or if $m \leq n$. However, in these regimes the assumption is not useful. There are also regimes in which this assumption is clearly false. For example if χ is the distribution that always outputs 0. In this case there is no error and one can solve s by Gaussian elimination. Luckily, there are regimes in which the assumption is believed to be true and is extremely useful. For example: let $n(\lambda) = \lambda$, $m = \text{poly}(\lambda)$, $q = \text{poly}(\lambda)$ or even larger, and χ is a small norm distribution. Often we let χ be the discrete Gaussian distribution where we restrict the outputs to be in $[-B, B]$, where $-B = q - B$. This distribution χ has the special property that if there exists a PPT adversary that breaks $\mathsf{LWE}_{n,m,q,\chi}$ then one can use this adversary to break worse-case lattice problems (such as finding approximating the shortest vector in a lattice). We will not elaborate on this, and for those who are interested, Vinod Vaikuntanathan has fantastic lecture notes on this topic here.

For us, all we will use is that χ takes values in $[-B, B]$ where B is a small error parameter (significantly smaller than q).

Symmetric Encryption Scheme from LWE

The encryption scheme is quite straightforward: The message space is $\{0,1\}$. The secret key is $s \xleftarrow{R} \mathbb{Z}_q^n$. To encrypt a bit $b \in \{0,1\}$:

$$\mathsf{Enc}(s, b) = (a, s \cdot a + e + b\lfloor q/2 \rfloor)$$

where $a \xleftarrow{R} \mathbb{Z}_q^n$. To decrypt a ciphertext $(a, c) \in \mathbb{Z}_q^{n+1}$:

$$\mathsf{Dec}(s, (a, c)) : \text{ output 0 iff } |c - s \cdot a| \leq q/4.$$

One nice property about this scheme (which the PRF based construction does not have) is that it is linearly homomorphic. Namely, for every $b_1, b_2 \in \{01\}$ it holds that

$$\mathsf{Dec}(s, \mathsf{Enc}(s, b_1) + \mathsf{Enc}(s, b_2)) = b_1 \oplus b_2.$$

Note that there is an error growth. Namely, the error in the ciphertext $\mathsf{Enc}(s, b_1) + \mathsf{Enc}(s, b_2)$ is $e_1 + e_2$, which is twice as large as the noise in a fresh ciphertext $\mathsf{Enc}(s, b_1 \oplus b_2)$. But as long as q is large enough compared to the error we can do many additive homomorphisms.

There is question about this in Pset 2.

Why do we care that the scheme is linearly homomorphic? It turns out that this is a very useful property. As we will see next week, this can be used to build a Private Information Retrieval (PIR) scheme.

Is this scheme CPA*-secure?*

Recall that that to prove CPA security we need to argue that a PPT adversary $\mathcal{A}$ that is given an encryption oracle, cannot distinguish between $\mathsf{Enc}(s, 0)$ and $\mathsf{Enc}(s, 1)$. Note that the encryption oracle can be simulated given many samples of the form $\{a_i, s \cdot a_i + e_i\}_{i=1}^{t}$, where $a_1, \ldots, a_t \xleftarrow{R} \mathbb{Z}_q$ and $e_1, \ldots, e_t \leftarrow \chi$. The LWE assumption asserts that even given these samples, $(a, s \cdot a + e) \approx (a, u)$ where $a, u \xleftarrow{R} \mathbb{Z}_q$ and $e \leftarrow \chi$. and thus $(a, s \cdot a + e + b\lfloor q/2 \rfloor) \approx (a, u)$, as desired.

Public-Key Encryption from LWE

- $\mathsf{Gen}(1^\lambda)$:
  1. Let $n = \lambda$, $q = \mathrm{poly}(\lambda)$ and $m = \theta(n \cdot \log q)$. Let χ be the discrete Gaussian distribution with error bound B such that $B \cdot m << q/4$.
  2. Generate $s \xleftarrow{R} \mathbb{Z}_q^n$, $A \xleftarrow{R} \mathbb{Z}_q^{n \times m}$, and $e \leftarrow \chi^m$.
  3. Output $\mathsf{pk} = (A, sA + e)$ and $\mathsf{sk} = s$.

  Denote by $B \in \mathbb{Z}^{(n+1) \times m}$ the matrix whose first n rows is the matrix A and the last row is the vector $sA + e$. Thus, we can think of $\mathsf{pk} = B$. Note that

$$(-s, 1) \cdot B = e.$$

- $\mathsf{Enc}(\mathsf{pk}, b)$ chooses a random $r \xleftarrow{R} \{0,1\}^m$ and outputs

$$B \cdot r + b(0, \ldots, 0, \lfloor q/2 \rfloor).$$

- $\mathsf{Dec}(\mathsf{sk}, c)$ outputs o iff $|(-s, 1) \cdot c| \leq q/4$.

Note that

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, b)) = 0 \text{ iff } |(-s, 1) \cdot (B \cdot r + b(0, \ldots, 0, \lfloor q/2 \rfloor))| \leq q/4$$

Moreover, note that

$$(-s, 1) \cdot (B \cdot r) = ((-s, 1) \cdot B) \cdot r = e \cdot r \leq m \cdot B,$$

where B is a bound on the error. Correctness follows from the fact that $m \cdot B < q/4$.

Is this scheme secure?

Security follows from the LWE assumption, but proving this is trickier than in the symmetric key setting. To prove security we argue that

if $\mathsf{pk} = B$ was truly random in $\mathbb{Z}_1^{(n+1)\times m}$ then for every $b \in \{0,1\}$

$$(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, b) \equiv (\mathsf{pk}, U), \tag{1}$$

where U is uniformly distributed in $\mathbb{Z}_q^m$, and where $\equiv$ means that the two distributions are equivalent. Namely, if B was truly random then $\mathsf{Enc}(\mathsf{pk}, b)$ information theoretically loses all information about b and thus the scheme is information theoretically secure (but decryption is lost). Then we can rely on the LWE assumption to argue that pk is computationally indistinguishable from being uniform in $\mathbb{Z}_q^m$.

To prove Equation (1), we need to prove that

$$(B, B \cdot r) \equiv (B, U)$$

for $B \xleftarrow{R} \mathbb{Z}^{(n+1)\times m}$, $r \xleftarrow{R} \{0,1\}^m$ and $U \xleftarrow{R} \mathbb{Z}_q^{n+1}$. This follows from the fact that $m \geq n \log q$ together with the Leftover Hash Lemma (see wikipedia for an explanation of this lemma). Let me offer an intuitive explanation: Denoting the i'th row of B by B_i, note that each $B_i \cdot r \in \mathbb{Z}_q$ leaks at most $\log q$ bits of information about r. Since $n \log q << m$ even after leaking $(B_1 \cdot r, \ldots, B_{i-1} \cdot r)$ for a given random matrix B, r still has high min-entropy (it has min-entropy at least $m - (i-1) \cdot \log q$. It is known that if r hash high min-entropy ad is independent of B_i, then $(B_i, B_i \cdot r) \equiv (B_i, U)$ where $U \xleftarrow{R} \mathbb{Z}_q$.

References

[1] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005.