Lecture 6: CPA *Secure Encryption and Pseudo Random Functions.*

Notes by Yael Kalai

MIT - 6.5620
Lecture 6 (September 23, 2025)

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

Recap

- Last week we constructed a PRG, which stretches λ bits of random bits to $\text{poly}(\lambda)$ bits of pseudorandom bits, assuming the existence of a OWP.
- Our construction used the Goldreich-Levin (GL) Hardcore predicate, and we proved the GL theorem (which is used a lot in cryptography, complexity theory and coding theory)!
- The week before we showed that PRG can be used to encrypt a long message given a short key, by expanding the key using the PRG and using it as a one-time pad.

It is known that a PRG can be constructed from any OWF but the proof is much more complication.

Today

- Define an encryption scheme that is secure even if arbitrary polynomially many messages are exchanged.
- Define the notion of a *pseudo random function* (PRF) which is the main cryptographic tool used in the construction.
- Construct a PRF from any PRG (and thus from any OWF).

A PRF is a generalization of a PRG.

Next class we will prove security of this PRF and move onto the topic of authentication.

Encryption Scheme with Many-Message Security

We next define the notion of multi-message security of an encryption scheme, which guarantees security even if the adversary sees arbitrarily (polynomially) many ciphertexts. Note that each ciphertext $\text{Enc}(k, m)$ may leak some information about the secret key k, and hence we need to ensure that if the adversary sees many ciphertexts

he does not learn enough information about k that makes the scheme insecure.

Continuing with the philosophy that we should be prepared against a worse-case adversary (never underestimate the power of your adversary!), we would like to ensure security even if the adversary gets to choose the messages that are encrypted.

Definition 1. An encryption scheme is said to be many-message secure if for every polynomial $\ell : \mathbb{N} \to \mathbb{N}$ and every poly-size adversary $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$ and for every two sequence of messages $m_1, \ldots, m_\ell \in \mathcal{M}_\lambda$ and $m'_1, \ldots, m'_\ell \in \mathcal{M}_\lambda$

$$|\Pr[\mathcal{A}(\mathsf{Enc}(k, m_1), \ldots, \mathsf{Enc}(k, m_\ell)) = 1] - \Pr[\mathcal{A}(\mathsf{Enc}(k, m'_1), \ldots, \mathsf{Enc}(k, m'_\ell)) = 1]| \leq \frac{1}{2} + \mu(\lambda). \tag{1}$$

We actually consider an even stronger definition, which allows the adversary to choose the messages *adaptively* based on previously seen ciphertexts. This is called security against *adaptively chosen plaintext attacks*

Definition 2. An encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is said to be secure against *adaptively chosen plaintext attacks* (CPA secure) if for every PPT adversary $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$, $\mathcal{A}$ wins in the following game with probability at most $\frac{1}{2} + \mu(\lambda)$:

- The challenger chooses a key $k \leftarrow \mathcal{K}_\lambda$.
- The adversary $\mathcal{A}$ given 1^λ chooses a message $m_i \in \mathcal{M}_\lambda$ and receives $c_i \leftarrow \mathsf{Enc}_\lambda(k, m_i)$.

  This step can be repeated polynomially many times.
- The adversary $\mathcal{A}$ chooses $m_0, m_1 \in \mathcal{M}_\lambda$.
- The challenger chooses a random bit $b \leftarrow \{0,1\}$, generates $c \leftarrow \mathsf{Enc}(k, m_b)$, and sends the ciphertext c to the adversary.
- The adversary given c outputs a bit b'.

We say that $\mathcal{A}$ wins if $b' = b$.

Remark. Note that it suffices to construct a CPA secure encryption scheme for single bit messages; i.e., for $\mathcal{M} = \{0,1\}$. The reason is that we can encrypt arbitrarily long messages bit-by-bit. Therefore, for simplicity, in what follows we focus on constructing a CPA secure encryption scheme with $\mathcal{M} = \{0,1\}$.

Constructing a CPA*-Secure Encryption Scheme*

We first observe that if we could expand the secret key to an arbitrarily long pad then we could use it to encrypt all our messages, each time using a fresh part of the pad. This scheme would result in a stateful scheme since we need to remember which part of the pad was already used.

It turns out that this is inherent! Specifically, there does not exist a CPA secure encryption scheme where the encryption algorithm is deterministic (unless it is stateful)! The reason is that it is always easy to distinguish between $(\mathsf{Enc}(k,m),\mathsf{Enc}(k,m))$ and $(\mathsf{Enc}(k,m),\mathsf{Enc}(k,m'))$ for randomly chosen $m,m' \in \mathcal{M}_\lambda$. Indeed, in all our CPA secure encryption schemes the encryption algorithm is *randomized*.

Continuing with our intuition above, if we could expand our secret key to an extremely long pad, then each time we encrypt a message, we can use a random part of the pad, and hope that the pad is long enough that we never reuse the same part. The question is how can we expand our secret key into such a long pad *efficiently*?

Suppose we could magically expand the secret key k into a function $F_k : \{0,1\}^\lambda \to \{0,1\}$ that is efficiently computable and at the same time indistinguishable from a truly random function. Then we could encrypt a message $m \in \{0,1\}$ by:

$$\mathsf{Enc}(F,m) = (r, F(r) \oplus m),$$

and decrypt the resulting ciphertext by:

$$\mathsf{Dec}(F,(r,c)) = F(r) \oplus c.$$

As long as we encrypt significantly less than $2^{\lambda/2}$ messages we do not expect to see a collision (i.e., the same r used twice) and hence security follows from the one-time security of the one-time pad.

*Pseudorandom Functions (*PRF*)*

Using the magic of cryptography we construct efficiently computable functions that look like truly random ones! Such functions are called *pseudorandom functions*.

Definition 3 (Pseudorandom function)**.** A pseudorandom function $F = \{F_\lambda\}_{\lambda\in\mathbb{N}}$, where for every $\lambda \in \mathbb{N}$, $F_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$, has the property that for every poly-size $\mathcal{A}$ there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$,

$$|\Pr_{k\leftarrow\mathcal{K}_\lambda}[\mathcal{A}^{F_\lambda(k,\cdot)}(1^\lambda)=1] - \Pr_{R_\lambda}[\mathcal{A}^{R_\lambda(\cdot)}(1^\lambda)=1]| \leq \mu(\lambda)$$

where R_λ is truly random function $R_\lambda : \mathcal{K}_\lambda \times \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. For concreteness, we think of $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$.

CPA-*Secure Encryption Construction from any* PRF *Family*

Let $F = \{F_\lambda\}_{\lambda \in \mathbb{N}}$ be any PRF family where

$$F_\lambda : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}.$$

We use F to construct a symmetric encryption scheme where the keyspace is $\{0,1\}^\lambda$, the message space is $\{0,1\}$, and the ciphertext space is $\{0,1\}^\lambda \times \{0,1\}$. Specifically,

$$\mathsf{Enc}_\lambda(k, m; r) = (r, m \oplus F(k, r))$$

where $r \xleftarrow{R} \{0,1\}^\lambda$.

$$\mathsf{Dec}(k, (r, c)) = F(k, r) \oplus c.$$

Theorem 4. *The encryption scheme defined above is* CPA *secure assuming the underlying function F is a* PRF.

Proof. The CPA security follows immediately from the definition of a PRF. Specifically, suppose there exists a poly-size $\mathcal{A}$ and there exists a non-negligible ϵ such that $\mathcal{A}$ wins in the CPA game with probability at least $\frac{1}{2} + \epsilon(\lambda)$. We construct a poly-size $\mathcal{B}$ that breaks the PRF security of F.

$\mathcal{B}^O(1^\lambda)$ distinguishes between the case that $O = F_\lambda(k, \cdot)$ and the case that $O = R_\lambda$, as follows:

1. Run the adversary $\mathcal{A}$ in the CPA game, while emulating the "challenger" in the security game, as follows:
   (a) Every time $\mathcal{A}$ requests an encryption of a message $m_i \in \{0,1\}$, sample a random $r_i \leftarrow \{0,1\}^\lambda$ and return the ciphertext $(r_i, O(r_i) \oplus m_i)$.
   (b) When $\mathcal{A}$ sends the two challenge ciphertexts $m_0, m_1 \in \{0,1\}$ choose a random $b \leftarrow \{0,1\}$ and send $c = (r, O(r) \oplus m_b)$.
2. Denote the output of $\mathcal{A}$ by b'.
3. If $b' = b$ then guess pseudorandom (say output 1) and otherwise guess random (say output 0).

By definition of $\mathcal{B}$, and by our assumption that $\mathcal{A}$ wins the security game with probability $\frac{1}{2} + \epsilon(\lambda)$, we conclude that

$$\Pr[\mathcal{B}^{F_\lambda(k,\cdot)}(1^\lambda) = 1] \geq \frac{1}{2} + \epsilon(\lambda).$$

On the other hand, we argue that there exists a negligible function μ such that for every $\lambda \in \mathbb{N}$

$$\Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1] = \frac{1}{2} + \mu(\lambda).$$

For the latter, denote by E the event that the challenge ciphertext (r, c_b) encrypting m_b satisfies that r is different from all the random strings $\{r_i\}$ used to encrypt the messages $\{m_i\}$ chosen by $\mathcal{A}$. Then

$$\begin{aligned}
&\Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1] = \\
&\Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1 |\ E] \cdot \Pr[E] + \Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1 |\ \neg E] \cdot \Pr[\neg E] \leq \\
&\Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1 |\ E] \cdot \Pr[E] + \mu(\lambda) \leq \\
&\Pr[\mathcal{B}^{R_\lambda}(1^\lambda) = 1 |\ E] + \mu(\lambda) = \\
&\frac{1}{2} + \mu(\lambda).
\end{aligned}$$

□

PRF *Construction*

We next show how to construct a PRF from any PRG. This is a beautiful construction proposed by Goldreich, Goldwasser and Micali [1].

Note that PRFs and PRGs are very similar objects. They both take a short random seed $k \leftarrow \{0,1\}^\lambda$ and generate many pseudorandom bits out of it. The difference is that a PRG expands λ bits to $n(\lambda) \leq \text{poly}(\lambda)$ bits whereas a PRF can in principle expand by 2^λ bits:

$$(F(k,(0,\ldots,0),\ldots,F(k,(1,\ldots,1)).$$

One can think of a PRF as a PRG with random access.

The GGM Construction Recall that last week we showed how to increase the stretch of a PRG

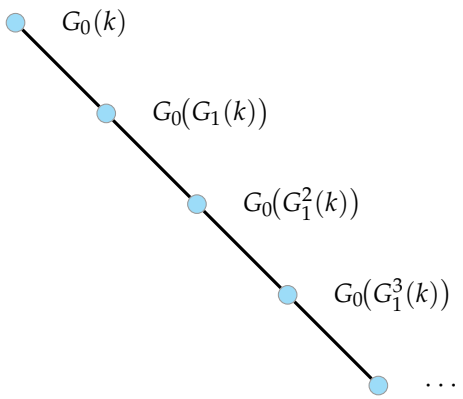
$$G : \{0,1\}^\lambda \to \{0,1\}^{\lambda+1}$$

into a PRG with arbitrary stretch

$$G^k : \{0,1\}^\lambda \to \{0,1\}^{\lambda+k}$$

The problem is that accessing the $\lambda + i$th bit of $G^k(x)$ takes time $\geq i$. The reason is that this construction increases expansion via a line. .

Let $G(s) = G_0(s) \,||\, G_1(s)$ where $G_0(s)$ is 1 bit and $G_1(s)$ is λ bits.



The idea of GGM is to use a tree structure as opposed to a line structure to have a more efficient random access. Suppose we are given a length doubling PRG

$$G : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}.$$

Denote by

$$G(x) = (G_0(x)), G_1(x))$$

where for every $b \in \{0,1\}$

$$G_b : \{0,1\}^\lambda \to \{0,1\}^\lambda.$$

Similarly, for every $b_1, b_2 \in \{0,1\}$ let

$$G_{b_1,b_2} : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

defined by

$$G_{b_1,b_2}(x) = G_{b_2}(G_{b_1}(x)).$$

More generally, for every $b_1, \ldots, b_i \in \{0,1\}$ let

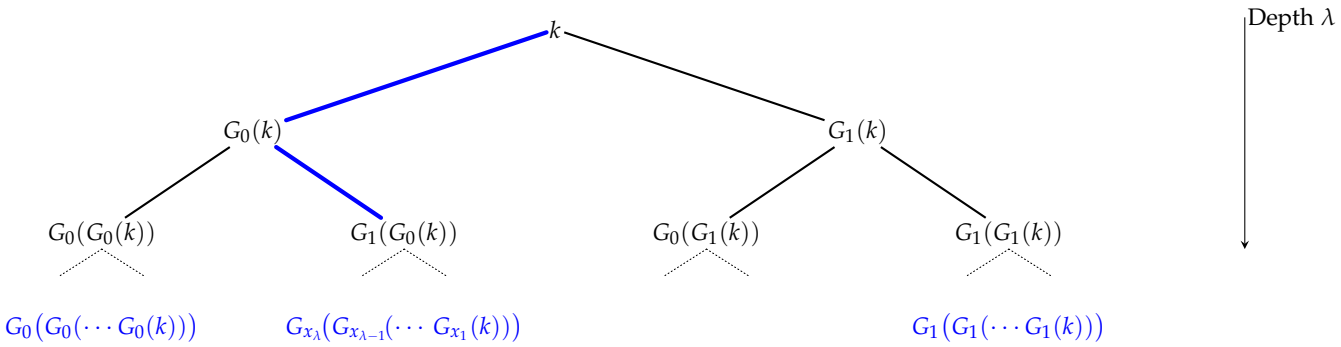
$$G_{b_1,\ldots,b_i} : \{0,1\}^\lambda \to \{0,1\}^\lambda$$

defined by

$$G_{b_1,\ldots,b_i}(x) = G_{b_i}(\ldots(G_{b_1}(x))\ldots).$$

Goldreich–Goldwasser–Micali PRF

Construction: Let $G(s) = G_0(s) \,\|\, G_1(s)$ where $G_0(s)$ and $G_1(s)$ are both λ bits.



Each path/leaf labeled by $x \in \{0,1\}^\lambda$ corresponds to $F(k,x)$.

Theorem 5. *The GGM construction is a* PRF.

We will see the proof next class.

References

[1] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.