

CPA Security and Pseudorandom Functions (CPA 安全 与伪随机函数)

Sheng Zhong Yuan Zhang

Computer Science and Technology Department
Nanjing University

- 1 Need for Stronger Security
 - The indistinguishable multiple encryptions
 - CPA-security
- 2 Pseudorandom Functions
 - Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
 - Constructing CPA-secure encryptions with PRFs
- 4 The existence of PRFs
 - Pseudorandom permutations
 - Pseudorandom permutations and PRFs
 - PRFs and block ciphers
 - PRFs and PRGs

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

- 1 Need for Stronger Security
 - The indistinguishable multiple encryptions
 - CPA-security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

Single encryption v.s. multiple encryptions

- In $PrivK_{\mathcal{A},\Pi}^{eav}$, the adversary is only allowed to observe **one** ciphertext. What if the adversary can observe **multiple** ciphertexts (encrypted with the same key)?
- We use a new experiment $PrivK_{\mathcal{A},\Pi}^{mult}$ to model this case.

The multiple-message eavesdropping experiment

The multiple-message eavesdropping experiment $\text{PrivK}_{\mathcal{A},\Pi}^{\text{mult}}$:

- 1 Given the security parameter n , \mathcal{A} outputs a pair of equal-length lists of messages $\vec{M}_0 = (m_{0,1}, \dots, m_{0,t})$ and $\vec{M}_1 = (m_{1,1}, \dots, m_{1,t})$, with $|m_{0,i}| = |m_{1,i}|$ for all i , and sends them to the challenger \mathcal{C} .
- 2 \mathcal{C} computes a key k by running $\text{Gen}(1^n)$, a uniform bit $b \in \{0, 1\}$, $c_i \leftarrow \text{Enc}_k(m_{b,i})$ for all i , and sends $\vec{C} = (c_1, \dots, c_t)$ to \mathcal{A} .
- 3 \mathcal{A} outputs a bit b' .
- 4 The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

The indistinguishable multiple encryptions

DEFINITION 3.19 Indistinguishable multiple encryptions in the presence of an eavesdropper

A private-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has **indistinguishable multiple encryptions in the presence of an eavesdropper** if for all PPT adversaries \mathcal{A} there is a negligible function $negl$ such that

$$Pr[PrivK_{\mathcal{A}, \Pi}^{mult}(n) = 1] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the randomness used by \mathcal{A} and the randomness used in the experiment.

Indistinguishable encryption \neq indistinguishable multiple encryptions

Consider the indistinguishable encryption scheme constructed using a PRG:

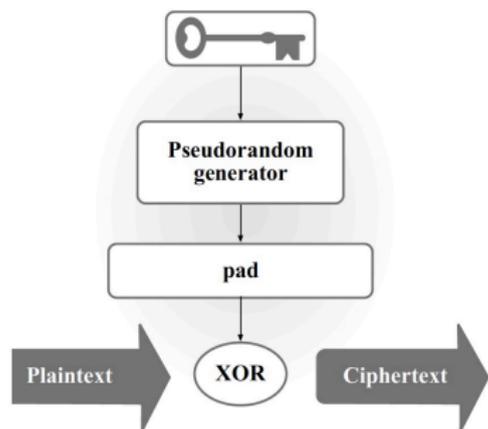


图 1: An indistinguishable encryption scheme

- Assuming the PRG generates 3-bit pseudo-random pads, can an adversary differentiate the ciphertext of "cat,cat" and the ciphertext of "cat,dog"?
- Yes.
- Lessons learnt: **Probabilistic encryption is needed.**

- 1 Need for Stronger Security
 - The indistinguishable multiple encryptions
 - CPA-security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

What are chosen-plaintext attacks?

- When an adversary performs **chosen-plaintext attacks**, it can exercise (partial) **control over what the honest parties encrypt**.
- Chosen-plaintext attacks encompass **known-plaintext attacks**.

The CPA indistinguishability experiment

The CPA indistinguishability experiment $\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}}(n)$:

- 1 A key k is generated by running $\text{Gen}(1^n)$.
- 2 The adversary \mathcal{A} is given input 1^n and **oracle access to $\text{Enc}_k(\cdot)$** , and outputs a pair of messages m_0, m_1 of the same length.
- 3 A uniform bit $b \in \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \text{Enc}_k(m_b)$ is computed and given to \mathcal{A} .
- 4 The adversary \mathcal{A} **continues to have oracle access to $\text{Enc}_k(\cdot)$** , and outputs a bit b' .
- 5 The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. In the former case, we say that \mathcal{A} **succeeds**.

What is CPA-security?

DEFINITION 3.22

A private-key encryption scheme $\Pi = (Gen, Enc, Dec)$ has **indistinguishable encryptions under a chosen-plaintext attack**, or is **CPA-secure**, if for all PPT adversaries \mathcal{A} there is a negligible function $negl$ such that

$$Pr[PrivK_{\mathcal{A}, \Pi}^{cpa} = 1] \leq \frac{1}{2} + negl(n),$$

where the probability is taken over the randomness used by \mathcal{A} , as well as the randomness used in the experiment.

- We can use **pseudorandom functions** to construct a CPA-secure encryption scheme.

- 1 Need for Stronger Security
- 2 Pseudorandom Functions**
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
 - Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

What are pseudorandom functions used for?

- If we want “random-looking” strings, we resort to **pseudorandom generators**.
- If we want “random-looking” functions, we resort to **pseudorandom functions**.

What is a function? a random function?

We use functions on $\{0, 1\}$ to explain:

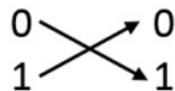


图 2: A function on $\{0, 1\}$.

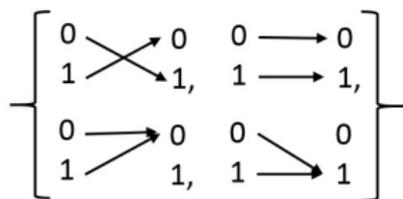


图 3: $Func_1$: the set of all functions on $\{0, 1\}$

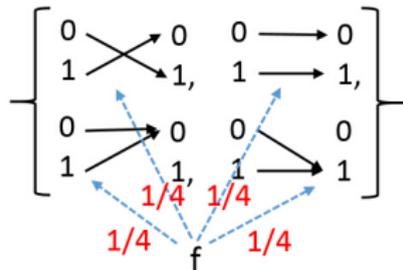


图 4: A random function from $\{0, 1\}$ to $\{0, 1\}$

What is a function? a random function?

- A **function** is a mapping from $\{0, 1\}^{l_{in}}$ to $\{0, 1\}^{l_{out}}$. If $l_{in} = l_{out}$, we say the function is **length-preserving**.
- Let $Func_n$ denote the set of all functions from $\{0, 1\}^n$ to $\{0, 1\}^n$.
- The size of $Func_n$ equals $2^{n2^n} = 2^n \cdot 2^n \cdot \dots \cdot 2^n$.
- A **(uniformly) random function** is a function that is chosen **uniformly at random** from $Func_n$.

What is a keyed function? pseudorandom function?

- A **keyed function** $F : \{0, 1\}^{l_{key}} \times \{0, 1\}^{l_{in}} \rightarrow \{0, 1\}^{l_{out}}$ is a two-input function, where the first input is called the a **key** and denoted k . We only consider F is **length-preserving**, meaning $l_{key}(n) = l_{in}(n) = l_{out}(n) = n$.
- If there is a polynomial-time algorithm that computes $F(k, x)$ given k and x , we say F is **efficient**.
- In typical usage, a key k is chosen and fixed, and we are interested in the single-input function $F_k : \{0, 1\}^* \rightarrow \{0, 1\}^*$ denoted by

$$F_k(x) = F(k, x).$$

- If we choose k uniformly at random, **the keyed function F induces a natural distribution on $Func_n$** .
- If the function F_k (for a uniformly random key k) is **indistinguishable from a (uniformly) random function**, we say F_k is **pseudorandom**.

Formal definition of pseudorandom function

DEFINITION 3.25

Consider a **length-preserving keyed function** $F: \{0, 1\}^{l_{key}(n)} \times \{0, 1\}^{l_{in}(n)} \rightarrow \{0, 1\}^{l_{out}(n)}$ (i.e. $l_{key}(n)=l_{in}(n)=l_{out}(n)=n$), and f is a random function that is uniformly chosen from $Func_n$. F is a **pseudorandom function** if for all PPT distinguishers D , there is a negligible function $negl$ such that,

$$|Pr[D^{F_k(\cdot)}(1^n) = 1] - Pr[D^{f(\cdot)}(1^n) = 1]| \leq negl(n),$$

where $Func_n$ is the set of all functions mapping n -bit string to n -bit string, the first probability is taken over uniform choice of $k \in \{0, 1\}^n$ and the randomness of D , and the second probability is taken over uniform choice of $f \in Func_n$ and the randomness of D .

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs**
- 4 The existence of PRFs

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
 - Constructing CPA-secure encryptions with PRFs
- 4 The existence of PRFs

Constructing CPA-secure encryption with PRFs

We can construct a CPA-secure encryption scheme with PRFs as follows.

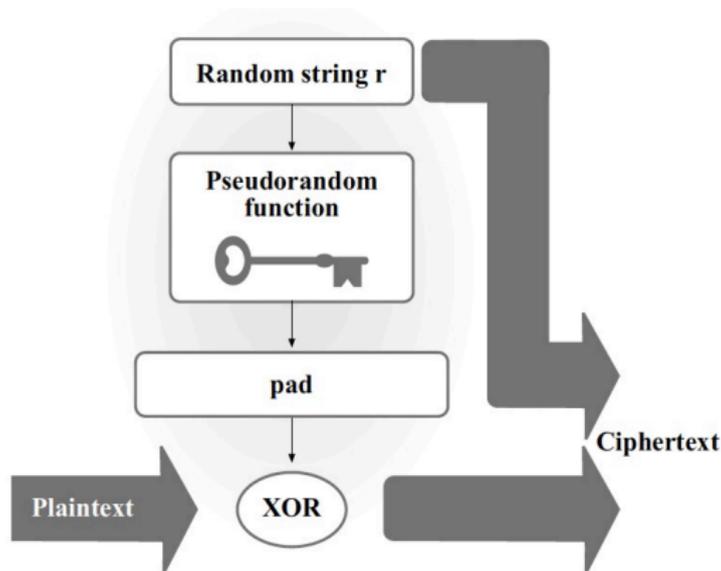


图 5: Constructing CPA-secure encryption with a PRF

Construction 3.30: A CPA-secure scheme from any pseudo-random function

Let F be a pseudorandom function. Define a private-key encryption scheme for messages of length n as follows:

- *Gen*: on input 1^n , choose uniform $k \in \{0, 1\}^n$ and output it.
- *Enc*: on input a key $k \in \{0, 1\}^n$ and a message $m \in \{0, 1\}^n$, choose uniform $r \in \{0, 1\}^n$ and outputs the ciphertext

$$c := \langle r, F_k(r) \oplus m \rangle .$$

- *Dec*: on input a key $k \in \{0, 1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message

$$m := F_k(r) \oplus s$$

Theorem 3.31

If F is a pseudorandom function, then Construction 3.30 is a CPA-secure private-key encryption scheme for messages of length n .

Proof sketch: Construct a similar encryption scheme $\tilde{\Pi} = (\tilde{Gen}, \tilde{Enc}, \tilde{Dec})$ that differs the scheme Π in Construction 3.30 only by replacing F_k with a truly random function f .

First, we can show no PPT adversary can differentiate the two scheme with a non-negligible probability **due to indistinguishability between a PRF and a random function**, i.e.

$$|\Pr[\text{PrivK}_{\mathcal{A}, \Pi}^{cpa} = 1] - \Pr[\text{PrivK}_{\mathcal{A}, \tilde{\Pi}}^{cpa} = 1]| \leq \text{negl}(n).$$

Then, we can show no PPT adversary can win the CPA experiment on $\tilde{\Pi}$ with a non-negligible probability:

$$\Pr[\text{PrivK}_{\mathcal{A}, \tilde{\Pi}}^{\text{cpa}} = 1] \leq \frac{1}{2} + \frac{q(n)}{2^n},$$

where $q(n)$ is a bound on the number of encryption queries made by \mathcal{A} .

CPA-security proof (Contd.)

Specifically,

$$\begin{aligned} & Pr[\text{PrivK}_{\mathcal{A}, \tilde{\Pi}}^{\text{cpa}} = 1] \\ &= Pr[\text{PrivK}_{\mathcal{A}, \tilde{\Pi}}^{\text{cpa}} = 1 | \text{no queries match}] \cdot Pr[\text{no queries match}] + \\ & \quad Pr[\text{PrivK}_{\mathcal{A}, \tilde{\Pi}}^{\text{cpa}} = 1 | \geq 1 \text{ query matches}] \cdot Pr[\geq 1 \text{ query matches}] \\ &\leq \frac{1}{2} \cdot 1 + 1 \cdot \frac{q(n)}{2^n} \\ &\leq \frac{1}{2} + \frac{q(n)}{2^n} \end{aligned}$$

where $q(n)$ is a bound on the number of encryption queries made by \mathcal{A} .

Finally, combining the two inequations, we know no PPT adversary can win the CPA experiment on Π with a non-negligible advantage over $1/2$ since

$$\Pr[\text{PrivK}_{\mathcal{A}, \Pi}^{\text{cpa}} = 1] \leq \frac{1}{2} + \frac{q(n)}{2^n} + \text{negl}(n).$$

Theorem proved! □

THEOREM 3.24

Any private-key encryption scheme that is CPA-secure is also CPA-secure for **multiple** encryptions.

- A significant advantage of CPA-security: **It suffices to prove that a scheme is CPA-secure (for a single encryption), and we then obtain “for free” that it is CPA-secure for multiple encryptions as well.**
- Why is the theorem true? Basically, we can see:
 - CPA-secure encryption is NOT deterministic.
 - What an adversary can see in the CPA indistinguishability experiment covers what the adversary sees in the CPA multiple-encryption distinguishability experiment.

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 **The existence of PRFs**
 - **Pseudorandom permutations**
 - Pseudorandom permutations and PRFs
 - PRFs and block ciphers
 - PRFs and PRGs

What is a permutation? a random permutation?

We use permutations on $\{0, 1\}$ to explain:

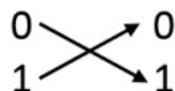
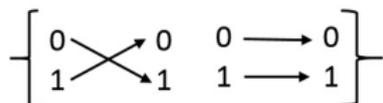


图 6: A permutation (function) on $\{0, 1\}$.



Perm_1

图 7: Perm_1 : the set of all permutations on $\{0, 1\}$

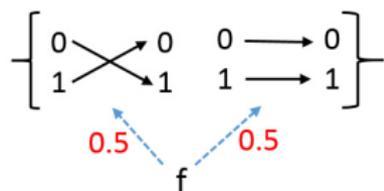


图 8: A random permutation on $\{0, 1\}$ is a function that is chosen uniformly at random from Perm_1

What is a permutation? a random permutation?

- A **permutation** (function) is a **bijection** or a **one-to-one** mapping from $\{0, 1\}^n$ to $\{0, 1\}^n$.
- Let $Perm_n$ be the set of all permutations on $\{0, 1\}^n$.
- The size of $Perm_n$ equals $(2^n)! = 2^n \cdot (2^n - 1) \cdot \dots \cdot 1$.
- A **random permutation** on $\{0, 1\}^n$ f is a permutation that is chosen **uniformly at random** from $Perm_n$.

Pseudorandom permutations

- Let F be a keyed function. If $l_{in} = l_{out}$, and for all k the function $F_k : \{0, 1\}^{l_{in}} \rightarrow \{0, 1\}^{l_{out}}$ is **one-to-one**, we call F a **keyed permutation**.
- We call l_{in} the **block length** of F .
- If both $F_k(x)$ and its inverse function $F_k^{-1}(y)$ can be computed within polynomial time given k, x and k, y resp., we say F is **efficient**.
- If NO efficient algorithm can distinguish between a F_k (for uniform key k) and a random permutation, i.e. a function that is chosen uniformly at random from $Perm_n$, we say F_k is a **pseudorandom permutation**.

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs
 - Pseudorandom permutations
 - Pseudorandom permutations and PRFs
 - PRFs and block ciphers
 - PRFs and PRGs

Pseudorandom permutations are PRFs when the block size is long

In fact, when a pseudorandom permutation's block size is sufficiently long, it is indistinguishable from a random function or a PRF.

PROPOSITION 3.27

If F is a pseudorandom permutation, and additionally $l_{in} \geq n$, then F is also a pseudorandom function.

- Intuitively, this is due to the fact that a uniform function f looks identical to a uniform permutation unless distinct values x and y are witnessed for which $f(x) = f(y)$. However, the probability of finding such values using a polynomial number of queries is negligible when the block size is large.

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs
 - Pseudorandom permutations
 - Pseudorandom permutations and PRFs
 - PRFs and block ciphers
 - PRFs and PRGs

Strong pseudorandom permutation

Often, a honest party may be required to compute the inverse function F_k^{-1} in addition to F_k itself, therefore we may assume the adversary is able to perform such computations also, and require F_k is indistinguishable from a uniform permutation EVEN IF **the distinguisher is additionally given oracle access to the inverse of the permutation**. If F has such probability, we call it a **strong pseudorandom permutation**.

DEFINITION 3.28

Let $F : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^*$ be an efficient, length-preserving, keyed permutation. F is a **strong pseudorandom permutation** if for all PPT distinguishers D , there exists a negligible function $negl$ such that:

$$|\Pr[D^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot), f^{-1}(\cdot)}(1^n) = 1]| \leq negl(n),$$

where the first probability is taken over uniform choice of $k \in \{0, 1\}^n$ and the randomness of D , and the second probability is taken over uniform choice of $f \in Perm_n$ and the randomness of D .

Strong pseudorandom permutation and block ciphers

- In practice, **block ciphers** are designed to be secure instantiations of (strong) pseudorandom permutations/PRFs with some fixed key length and block length.

- 1 Need for Stronger Security
- 2 Pseudorandom Functions
- 3 Constructing CPA-secure encryption with PRFs
- 4 The existence of PRFs
 - Pseudorandom permutations
 - Pseudorandom permutations and PRFs
 - PRFs and block ciphers
 - PRFs and PRGs

- One can easily construct a PRG G from a PRF F for any desired l as follows:

$$G(s) \stackrel{\text{def}}{=} F_s(1) || F_s(2) || \dots || F_s(l).$$

- Also, a PRG G with expansion factor $n \cdot 2^{t(n)}$ can be used to construct a PRF with *small input length* $F: \{0, 1\}^n \times \{0, 1\}^{t(n)} \rightarrow \{0, 1\}^n$ by setting

$$F_k(x) = y_{x \cdot n} y_{x \cdot n + 1} \dots y_{x \cdot n + n - 1},$$

where y_0, y_1, \dots , are the bits generated by $G(k)$, and $t(n) = \mathcal{O}(\log n)$.



Katz, J. and Lindell, Y..

Chapter 3 of “Introduction to modern cryptography” (2nd ed).

Chapman & Hall/CRC, 2014