Cryptology

Sabyasachi Karati

Assistant Professor
Cryptology and Security Research Unit (C.S.R.U)
R. C. Bose Centre for Cryptology and Security
Indian Statistical Institute (ISI)
Kolkata, India

Lecture 06

Pseudo-Random Function, Pseudo-Random Permutation and Block Cipher

Stream Cipher

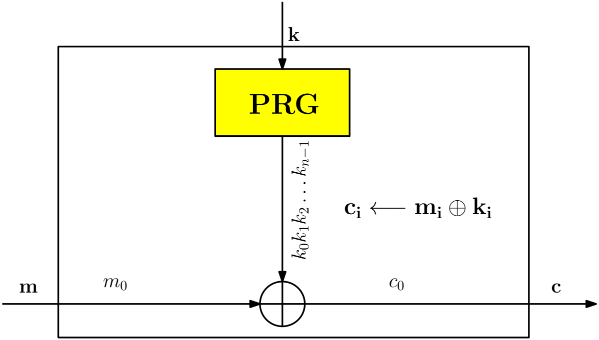
- A stream cipher encryptes bits individually.

Stream Cipher

- A stream cipher encryptes bits individually.
- XORs a bit from a key stream to a plaintext bit.

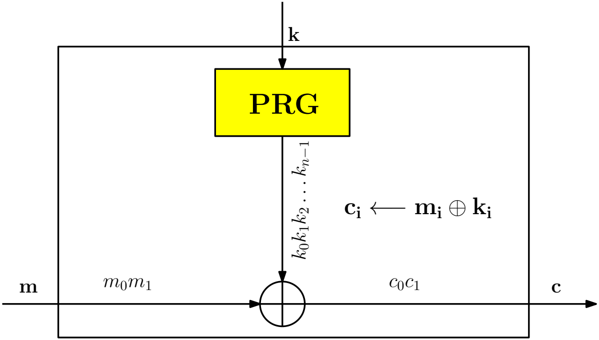
Stream Cipher

- A stream cipher encryptes bits individually.
- XORs a bit from a key stream to a plaintext bit.



At time $t = 1$

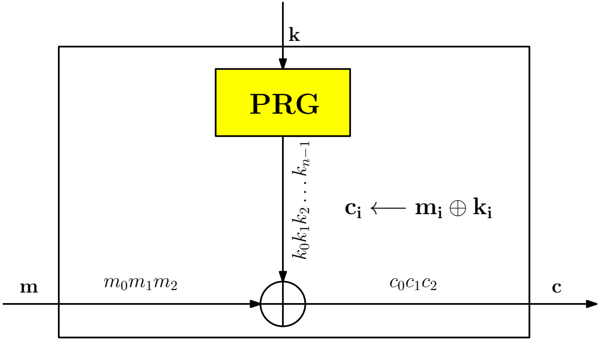
- A stream cipher encryptes bits individually.
- XORs a bit from a key stream to a plaintext bit.



At time $t = 2$

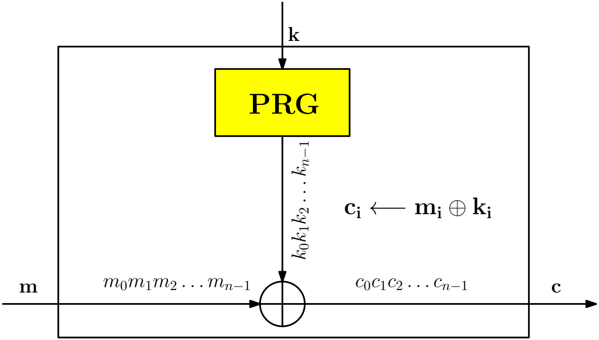
Stream Cipher

- A stream cipher encryptes bits individually.
- XORs a bit from a key stream to a plaintext bit.



At time $t = 3$

- A stream cipher encryptes bits individually.
- XORs a bit from a key stream to a plaintext bit.



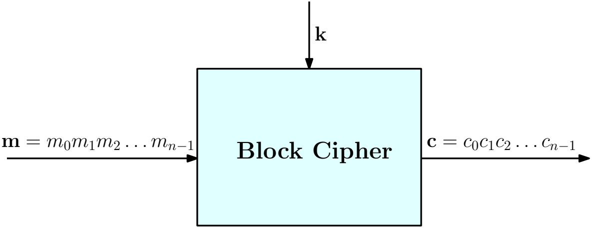
At time $t = n$

Block Cipher

- A block cipher encryptes a block of bits at a time.

Block Cipher

- A block cipher encryptes a block of bits at a time.



Block Cipher

Block Cipher

A deterministic, polynomial-time cipher $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ whose message space and ciphertext space are the same (finite) set $\mathcal{X}$. If the key space of $\mathfrak{E}$ is $\mathcal{K}$, then $\mathfrak{E}$ is defined over $(\mathcal{K}, \mathcal{X})$.

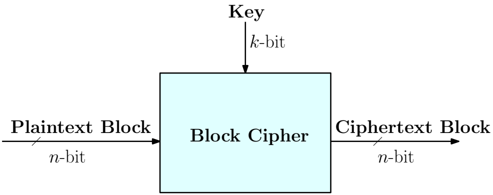
- We call an element $x \in \mathcal{X}$ a data block, and
- We refer to $\mathcal{X}$ as the data block space of $\mathfrak{E}$.

Block Cipher

Block Cipher

A deterministic, polynomial-time cipher $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ whose message space and ciphertext space are the same (finite) set $\mathcal{X}$. If the key space of $\mathfrak{E}$ is $\mathcal{K}$, then $\mathfrak{E}$ is defined over $(\mathcal{K}, \mathcal{X})$.

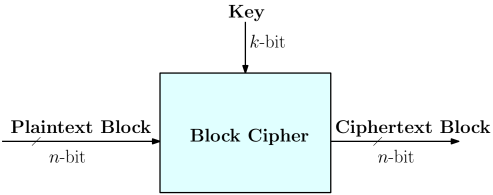
- We call an element $x \in \mathcal{X}$ a data block, and
- We refer to $\mathcal{X}$ as the data block space of $\mathfrak{E}$.



Block Cipher

A deterministic, polynomial-time cipher $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ whose message space and ciphertext space are the same (finite) set $\mathcal{X}$. If the key space of $\mathfrak{E}$ is $\mathcal{K}$, then $\mathfrak{E}$ is defined over $(\mathcal{K}, \mathcal{X})$.

- We call an element $x \in \mathcal{X}$ a data block, and
- We refer to $\mathcal{X}$ as the data block space of $\mathfrak{E}$.



Example

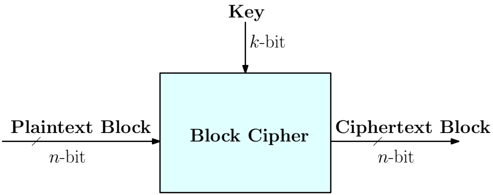
- DES: $n = 64$ and $k = 56$

Block Cipher

Block Cipher

A deterministic, polynomial-time cipher $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ whose message space and ciphertext space are the same (finite) set $\mathcal{X}$. If the key space of $\mathfrak{E}$ is $\mathcal{K}$, then $\mathfrak{E}$ is defined over $(\mathcal{K}, \mathcal{X})$.

- We call an element $x \in \mathcal{X}$ a data block, and
- We refer to $\mathcal{X}$ as the data block space of $\mathfrak{E}$.



Example

- DES: $n = 64$ and $k = 56$
- AES: $n = 128$ and $k = 128, 192, 256$

Performance

Crypto++ (Wei Dai)

	Cipher	Block/Key Size	Speed (mbps)
Steam	RC4		126
	Salsa20/12		643
	Sosemanuk		727
Block	DES	64/56	39
	AES	128/128	109

Block Cipher

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

Block Cipher

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

- Can we create an abstraction of Block cipher?

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

- Can we create an abstraction of Block cipher?
- **Ans:** Yes, as secure and efficient Pseudorandom Permutation (PRP).

Block Cipher

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

- Can we create an abstraction of Block cipher?
- **Ans:** Yes, as secure and efficient Pseudorandom Permutation (PRP).
- Merit:
  - Analysis the block cipher in terms of correct construction and security.

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

- Can we create an abstraction of Block cipher?
- **Ans:** Yes, as secure and efficient Pseudorandom Permutation (PRP).
- Merit:
  - Analysis the block cipher in terms of correct construction and security.
- PRP is a subset of a more generalized class called Pseudorandom Function (PRF).

Block Cipher

- Stream cipher can be abstracted as PRG.

Theorem

If G is a secure PRG, then the stream cipher $\mathfrak{E}$ constructed from G is a semantically secure cipher.

- Can we create an abstraction of Block cipher?
- **Ans:** Yes, as secure and efficient Pseudorandom Permutation (PRP).
- Merit:
  - Analysis the block cipher in terms of correct construction and security.
- PRP is a subset of a more generalized class called Pseudorandom Function (PRF).
- PRF can be used to design
  - CPA-secure encryption,
  - PRG and many more cryptographic primitives.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Random Function

Let $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ be the set of all functions from the domain $\mathcal{X}$ to range $\mathcal{Y}$. We choose a function f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$. We call f a random function.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Random Function

Let $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ be the set of all functions from the domain $\mathcal{X}$ to range $\mathcal{Y}$. We choose a function f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$. We call f a random function.

- Conceptually, it refers to uniform distribution on $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Random Function

Let $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ be the set of all functions from the domain $\mathcal{X}$ to range $\mathcal{Y}$. We choose a function f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$. We call f a random function.

- Conceptually, it refers to uniform distribution on $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Description of a Random function

- Size of $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$, $|\mathsf{Func}[\mathcal{X}, \mathcal{Y}]| = |\mathcal{Y}|^{|\mathcal{X}|}$.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Random Function

Let $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ be the set of all functions from the domain $\mathcal{X}$ to range $\mathcal{Y}$. We choose a function f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$. We call f a random function.

- Conceptually, it refers to uniform distribution on $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Description of a Random function

- Size of $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$, $|\mathsf{Func}[\mathcal{X}, \mathcal{Y}]| = |\mathcal{Y}|^{|\mathcal{X}|}$.
- Each function $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ can be viewed as a look-up table.

Pseudorandom Function (PRF)

- Here we extend the concept of pseudorandom string to pseudorandom function.
- Similarly, random string is analogous to random function.

Random Function

Let $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ be the set of all functions from the domain $\mathcal{X}$ to range $\mathcal{Y}$. We choose a function f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$. We call f a random function.

- Conceptually, it refers to uniform distribution on $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Description of a Random function

- Size of $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$, $|\mathsf{Func}[\mathcal{X}, \mathcal{Y}]| = |\mathcal{Y}|^{|\mathcal{X}|}$.
- Each function $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ can be viewed as a look-up table.
    - Each row of the look-up table stores the value of $f(x_i)$ for some $x_i \in \mathcal{X}$.

Pseudorandom Function (PRF)

Description of a Random function

$$
\begin{array}{c|c|}
 & f \\
\cline{2-2}
x_1 & f(x_1) \\
\cline{2-2}
x_2 & f(x_2) \\
\cline{2-2}
\vdots & \vdots \\
\cline{2-2}
x_{|X|} & f\left(x_{|X|}\right) \\
\cline{2-2}
\end{array}
$$

Pseudorandom Function (PRF)

Description of a Random function

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert\mathcal{X}\rvert}$	$f\left(x_{\lvert\mathcal{X}\rvert}\right)$

- Size of each row = $\log_2(|\mathcal{Y}|)$.

Description of a Random function

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert\mathcal{X}\rvert}$	$f\left(x_{\lvert\mathcal{X}\rvert}\right)$

- Size of each row = $\log_2(|\mathcal{Y}|)$.
- Number of rows = $|\mathcal{X}|$.

Pseudorandom Function (PRF)

Description of a Random function

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert\mathcal{X}\rvert}$	$f\left(x_{\lvert\mathcal{X}\rvert}\right)$

- Size of each row = $\log_2(|\mathcal{Y}|)$.
- Number of rows = $|\mathcal{X}|$.
- Size of the look-up table of $f = |\mathcal{X}|\log_2(|\mathcal{Y}|)$.

Description of a Random function

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert\mathcal{X}\rvert}$	$f\left(x_{\lvert\mathcal{X}\rvert}\right)$

- Size of each row = $\log_2(|\mathcal{Y}|)$.
- Number of rows = $|\mathcal{X}|$.
- Size of the look-up table of $f = |\mathcal{X}| \log_2(|\mathcal{Y}|)$.

Alternative view of Random function

Choosing f uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ is equivalent of choosing each row of look-up table uniformly at random from $\mathcal{Y}$.

Keyed Function

A Keyed Function F is a two-input function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ as

$$F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}, \text{ where}$$

Keyed Function

A Keyed Function F is a two-input function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ as

$$F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}, \text{ where}$$

- the first input is called the key and denoted by k,

Keyed Function

A Keyed Function F is a two-input function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ as

$$F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.

Pseudorandom Function (PRF)

Keyed Function

A Keyed Function F is a two-input function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ as

$$F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $F_k : \mathcal{X} \longrightarrow \mathcal{Y}$ defined as

$$F_k(x) \stackrel{\triangle}{=} F(k, x).$$

Pseudorandom Function (PRF)

Keyed Function

A Keyed Function F is a two-input function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$ as

$$F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $F_k : \mathcal{X} \longrightarrow \mathcal{Y}$ defined as

$$F_k(x) \stackrel{\triangle}{=} F(k, x).$$

- We say F is efficient if there is a deterministic, polynomial-time algorithm that computes $F(k, x)$ given k and x as input.

Pseudorandom Function (PRF)

Intuition on Pseudorandom Function (PRF)

- $S_F = \left\{ F_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Pseudorandom Function (PRF)

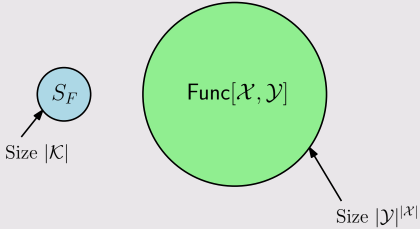
Intuition on Pseudorandom Function (PRF)

- $S_F = \left\{ F_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
- Size of $S_F = |\mathcal{K}|$.

Pseudorandom Function (PRF)

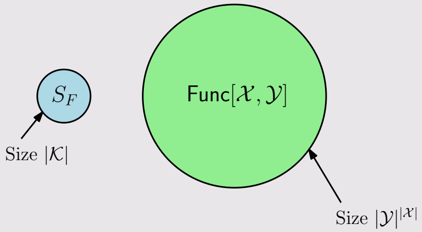
Intuition on Pseudorandom Function (PRF)

- $S_F = \left\{ F_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
- Size of $S_F = |\mathcal{K}|$.



Intuition on Pseudorandom Function (PRF)

- $S_F = \left\{ F_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
- Size of $S_F = |\mathcal{K}|$.



- Choosing F_k uniformly at random from S_F is equivalent of choosing k uniformly at random from $\mathcal{K}$.

Intuition on Pseudorandom Function (PRF)

- A keyed function F induces a natural distribution on S_F given by choosing a random key k.

Intuition on Pseudorandom Function (PRF)

- A keyed function F induces a natural distribution on S_F given by choosing a random key k.
- Intuitively,
  - F is pseudorandom if the function F_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a function f chosen uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

Intuition on Pseudorandom Function (PRF)

- A keyed function F induces a natural distribution on S_F given by choosing a random key k.
- Intuitively,
  - F is pseudorandom if the function F_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a function f chosen uniformly at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
  - Equivalently, F is pseudorandom if no polynomial-time adversary can distinguish whether it is interacting with F_k (for randomly-chosen key k) or f (where f is chosen at random from $\mathsf{Func}[\mathcal{X}, \mathcal{Y}]$).

Pseudorandom Function (PRF)

A Pseudorandom function (PRF) $F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}$ is a keyed function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, for which there exists a deterministic, polynomial-time algorithm to compute $F(k, x)$ given k and x.

Pseudorandom Function (PRF)

A Pseudorandom function (PRF) $F : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}$ is a keyed function defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, for which there exists a deterministic, polynomial-time algorithm to compute $F(k, x)$ given k and x.

- Let $y := F(k, x)$
- x sometimes is referred as input data block, and
- y sometimes is referred as output data block.

PRF Advantage

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$,

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.

PRF Advantage

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0 : k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1 : f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
2. The adversary submits a sequence of queries to the challenger.

PRF Advantage

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.

PRF Advantage

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{Y}$, and gives y_i to the adversary.

PRF Advantage

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

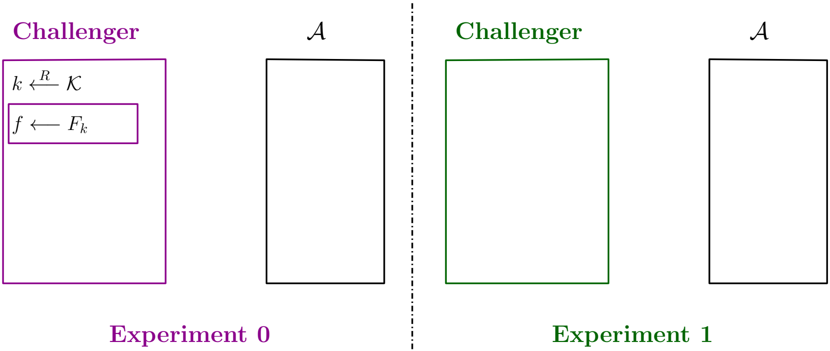
1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0 : k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1 : f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{Y}$, and gives y_i to the adversary.
   - The queries are adaptive.

PRF Indistinguishability Game

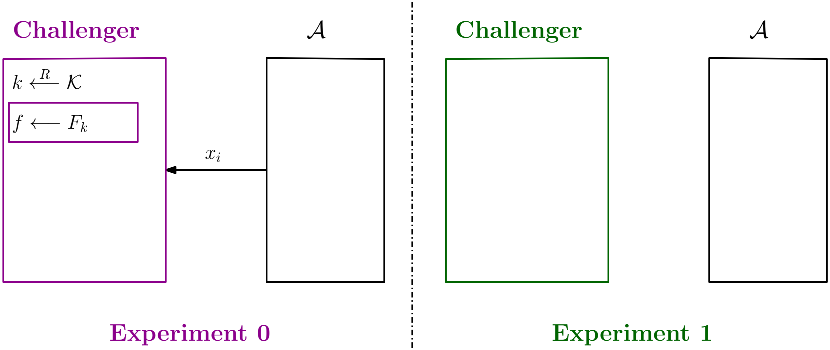
For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{Y}$, and gives y_i to the adversary.
   - The queries are adaptive.
3. The adversary computes and outputs a bit $\hat{b} \in \{0, 1\}$.

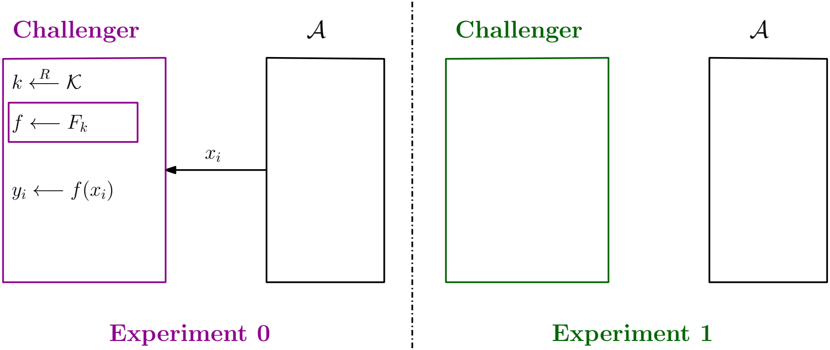
PRF Advantage



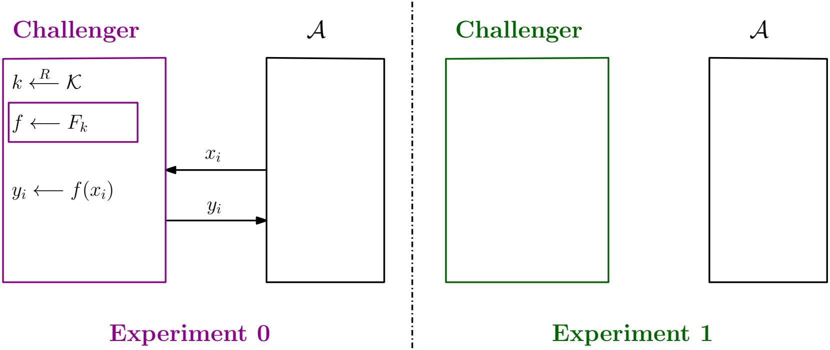
PRF Advantage



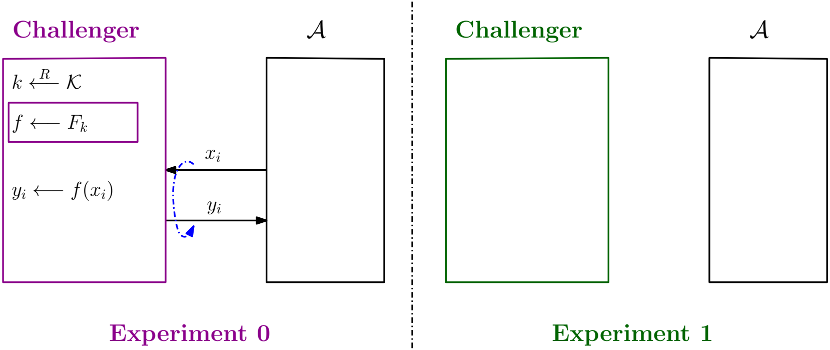
PRF Advantage



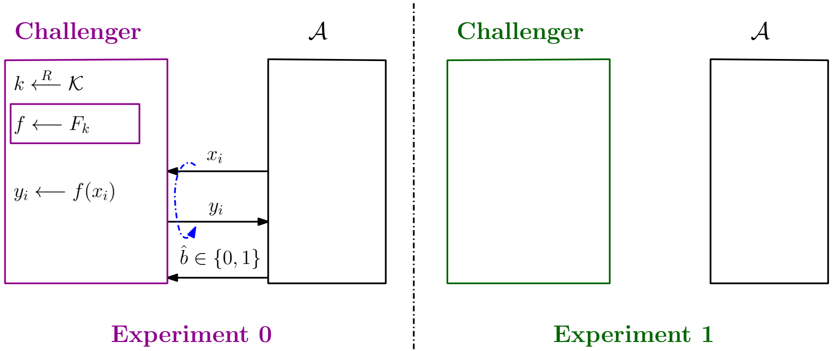
PRF Advantage



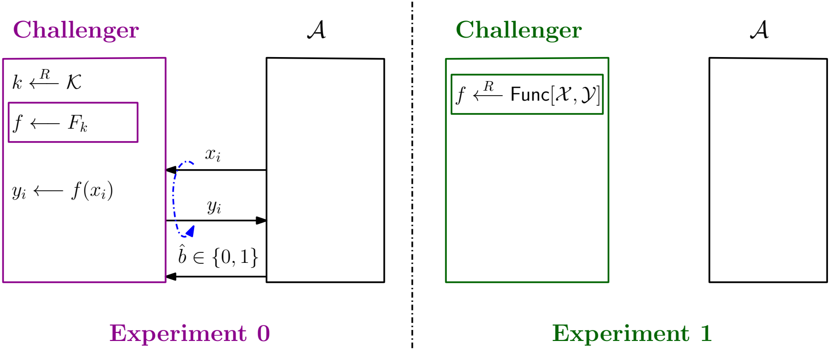
PRF Advantage



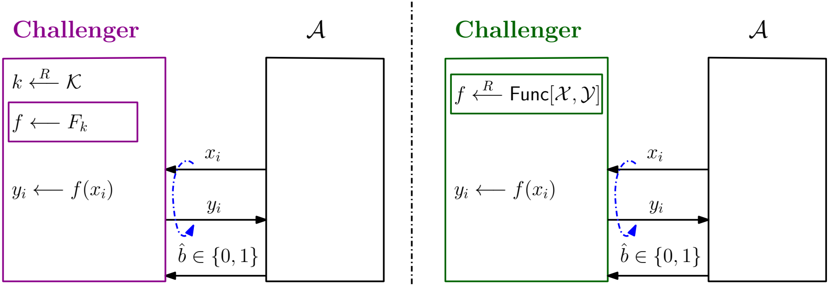
PRF Advantage



PRF Advantage



PRF Advantage



Experiment 0 | **Experiment 1**

PRF Advantage

PRF Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to F as

$$\mathsf{PRFadv}[\mathcal{A}, F] = |\Pr[W_0] - \Pr[W_1]|.$$

PRF Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to F as

$$\mathsf{PRFadv}[\mathcal{A}, F] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

We say that $\mathcal{A}$ is a Q-query PRF adversary if $\mathcal{A}$ issues at most Q queries.

PRF Advantage

PRF Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to F as

$$\mathsf{PRFadv}[\mathcal{A}, F] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

We say that $\mathcal{A}$ is a Q-query PRF adversary if $\mathcal{A}$ issues at most Q queries.

Secure PRF

A PRF F is secure if for all efficient adversaries $\mathcal{A}$, the value $\mathsf{PRFadv}[\mathcal{A}, F]$ is negligible.

PRF Advantage: Bit Guessing Version

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.

PRF Advantage: Bit Guessing Version

PRF Indistinguishability Game

For a given PRF F, defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow F_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$.
3. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{Y}$, and gives y_i to the adversary.
   - The queries are adaptive.
4. The adversary computes and outputs a bit $\hat{b} \in \{0, 1\}$.

PRF Advantage: Bit Guessing Version

Challenger $\qquad\qquad\qquad$ $\mathcal{A}$

$b \xleftarrow{R} \{0,1\}$

if $(b == 0)\{$

$\quad k \xleftarrow{R} \mathcal{K}$

$\quad f \longleftarrow F_k$

$\}$

if $(b == 1)\{$

$\quad f \xleftarrow{R} \mathsf{Func}[\mathcal{X}, \mathcal{Y}]$

$\}$

$y_i \longleftarrow f(x_i)$

$\xleftarrow{\quad x_i \quad}$

$\xrightarrow{\quad y_i \quad}$

$\xleftarrow{\hat{b} \in \{0,1\}}$

Experiment

PRF Advantage: Bit Guessing version

PRF Advantage

Let W be the event that where $\mathcal{A}$ wins if $\mathcal{A}$ outputs $\hat{b} = b$. We define the advantage of $\mathcal{A}$ in the attack game with respect to F as

$$\mathsf{PRFadv}^*[\mathcal{A}, F] = \left| \Pr[\hat{b} = b] - \frac{1}{2} \right|.$$

PRF Advantage: Bit Guessing version

PRF Advantage

Let W be the event that where $\mathcal{A}$ wins if $\mathcal{A}$ outputs $\hat{b} = b$. We define the advantage of $\mathcal{A}$ in the attack game with respect to F as

$$\mathsf{PRFadv}^*[\mathcal{A}, F] = \left| \Pr[\hat{b} = b] - \frac{1}{2} \right|.$$

Theorem

For every PRF F and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{PRFadv}[\mathcal{A}, F] = 2 \cdot \mathsf{PRFadv}^*[\mathcal{A}, F].$$

Weak PRF Advantage

Weak PRF Advantage

- Adversary's queries are severely restricted.

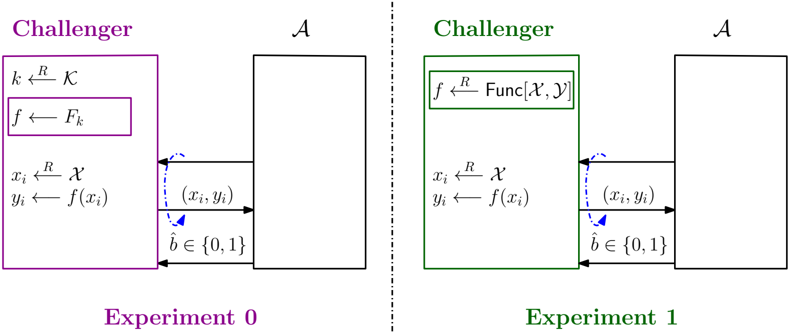
Weak PRF Advantage

- Adversary’s queries are severely restricted.
- It can only query the function at random points in the domain.

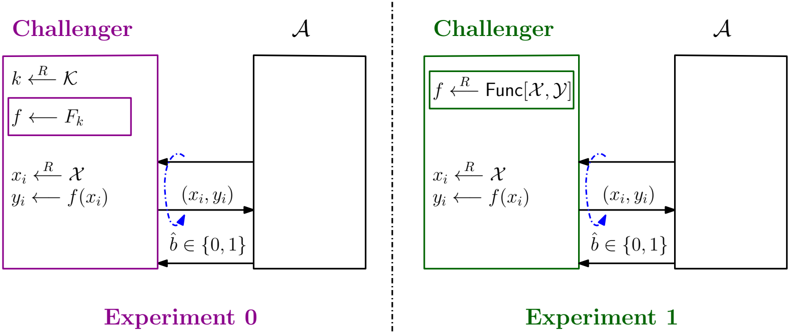
Weak PRF Advantage

- Adversary's queries are severely restricted.
- It can only query the function at random points in the domain.
- Whenever the adversary queries the function, the challenger chooses a random $x_i \in \mathcal{X}$ and sends both x_i and $f(x_i)$ to the adversary.

Weak PRF Advantage



Weak PRF Advantage



PRF Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to F as

$$\mathsf{weakPRFadv}[\mathcal{A}, F] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

- Challenger's protocol in Experiment 1 is not very efficient.

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}| \log_2(|\mathcal{Y}|)$.

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.
- For both aesthetic and technical reasons, it would be nice to have a more efficient implementation.

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}| \log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.
- For both aesthetic and technical reasons, it would be nice to have a more efficient implementation.
- A lazy implementation of f:

---

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.
- For both aesthetic and technical reasons, it would be nice to have a more efficient implementation.
- A lazy implementation of f:

---

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2.   if $x_i = x_j$ for some $j < i$
3.   then $y_i \longleftarrow y_j$

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.
- For both aesthetic and technical reasons, it would be nice to have a more efficient implementation.
- A lazy implementation of f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2.   if $x_i = x_j$ for some $j < i$
3.   then $y_i \longleftarrow y_j$
4.   else {
5. $y_i \xleftarrow{R} \mathcal{Y}$
6.   Store (x_i, y_i)
7.   }

Implementation of Random Function

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{Y}|)$.
- Not a problem from a purely definitional point of view.
- For both aesthetic and technical reasons, it would be nice to have a more efficient implementation.
- A lazy implementation of f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2. if $x_i = x_j$ for some $j < i$
3. then $y_i \longleftarrow y_j$
4. else {
5. $y_i \xleftarrow{R} \mathcal{Y}$
6. Store (x_i, y_i)
7. }
8. send y_i to $\mathcal{A}$.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

- Conceptually, it refers to uniform distribution on $\mathsf{Prem}[\mathcal{X}]$.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

- Conceptually, it refers to uniform distribution on $\mathsf{Prem}[\mathcal{X}]$.

Description of a Random Permutation

- Size of $\mathsf{Prem}[\mathcal{X}]$, $|\mathsf{Prem}[\mathcal{X}]| = |\mathcal{X}|!$

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

- Conceptually, it refers to uniform distribution on $\mathsf{Prem}[\mathcal{X}]$.

Description of a Random Permutation

- Size of $\mathsf{Prem}[\mathcal{X}]$, $|\mathsf{Prem}[\mathcal{X}]| = |\mathcal{X}|!$
- Each permutation $f \in \mathsf{Prem}[\mathcal{X}]$ can be viewed as a look-up table.

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

- Conceptually, it refers to uniform distribution on $\mathsf{Prem}[\mathcal{X}]$.

Description of a Random Permutation

- Size of $\mathsf{Prem}[\mathcal{X}]$, $|\mathsf{Prem}[\mathcal{X}]| = |\mathcal{X}|!$
- Each permutation $f \in \mathsf{Prem}[\mathcal{X}]$ can be viewed as a look-up table.
  - Each row of the look-up table stores the value of $f(x_i)$ for some $x_i \in \mathcal{X}$

Pseudorandom Permutation (PRP)

- Here we restrict the concept of pseudorandom function to pseudorandom permutation.
- Similarly, random function is analogous to random permutation.
- From hereon, we will use Pseudorandom Permutation and Block Cipher interchangeably.

Random Permutation

Let $\mathsf{Prem}[\mathcal{X}]$ be the set of all permutations from the domain $\mathcal{X}$ to range $\mathcal{X}$. We choose a permutation f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$. We call f a random permutation.

- Conceptually, it refers to uniform distribution on $\mathsf{Prem}[\mathcal{X}]$.

Description of a Random Permutation

- Size of $\mathsf{Prem}[\mathcal{X}]$, $|\mathsf{Prem}[\mathcal{X}]| = |\mathcal{X}|!$
- Each permutation $f \in \mathsf{Prem}[\mathcal{X}]$ can be viewed as a look-up table.
  - Each row of the look-up table stores the value of $f(x_i)$ for some $x_i \in \mathcal{X}$
  - No two rows are the same.

Description of a Random Permutation

$$
\begin{array}{c|c|}
 & f \\
\cline{2-2}
x_1 & f(x_1) \\
\cline{2-2}
x_2 & f(x_2) \\
\cline{2-2}
\vdots & \vdots \\
\cline{2-2}
x_{|\mathcal{X}|} & f\left(x_{|\mathcal{X}|}\right) \\
\cline{2-2}
\end{array}
$$

Pseudorandom Permutation (PRP)

Description of a Random Permutation

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert \mathcal{X} \rvert}$	$f\left(x_{\lvert \mathcal{X} \rvert}\right)$

- Size of each row $= \log_2(\lvert \mathcal{X} \rvert)$.

Pseudorandom Permutation (PRP)

Description of a Random Permutation

$$
\begin{array}{c|c|}
 & \color{purple}{f} \\
\cline{2-2}
x_1 & f(x_1) \\
\cline{2-2}
x_2 & f(x_2) \\
\cline{2-2}
\vdots & \vdots \\
\cline{2-2}
x_{|\mathcal{X}|} & f\left(x_{|\mathcal{X}|}\right) \\
\cline{2-2}
\end{array}
$$

- Size of each row = $\log_2(|\mathcal{X}|)$.
- Number of rows = $|\mathcal{X}|$.

Description of a Random Permutation

$$
\begin{array}{c|c|}
 & f \\
\cline{2-2}
x_1 & f(x_1) \\
\cline{2-2}
x_2 & f(x_2) \\
\cline{2-2}
\vdots & \vdots \\
\cline{2-2}
x_{|\mathcal{X}|} & f\left(x_{|\mathcal{X}|}\right) \\
\cline{2-2}
\end{array}
$$

- Size of each row = $\log_2(|\mathcal{X}|)$.
- Number of rows = $|\mathcal{X}|$.
- Size of the look-up table of f = $|\mathcal{X}|\log_2(|\mathcal{X}|)$.

Pseudorandom Permutation (PRP)

Description of a Random Permutation

	f
x_1	$f(x_1)$
x_2	$f(x_2)$
$\vdots$	$\vdots$
$x_{\lvert\mathcal{X}\rvert}$	$f\left(x_{\lvert\mathcal{X}\rvert}\right)$

- Size of each row = $\log_2(|\mathcal{X}|)$.
- Number of rows = $|\mathcal{X}|$.
- Size of the look-up table of f = $|\mathcal{X}|\log_2(|\mathcal{X}|)$.

Alternative view of Random Permutation

Choosing f uniformly at random from $\mathsf{Prem}[\mathcal{X}]$ is equivalent of choosing each row of look-up table uniformly at random from $\mathcal{X}$ without replacement.

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $E_k : \mathcal{X} \longrightarrow \mathcal{X}$ is one-to-one and defined as
$$E_k(x) \stackrel{\triangle}{=} E(k, x).$$
    - The domain and range of $E_k(\cdot)$ are the same, and $E_k(\cdot)$ is one-to-one,

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $E_k : \mathcal{X} \longrightarrow \mathcal{X}$ is one-to-one and defined as
$$E_k(x) \stackrel{\triangle}{=} E(k, x).$$
    - The domain and range of $E_k(\cdot)$ are the same, and $E_k(\cdot)$ is one-to-one,
    - Then $E_k(\cdot)$ is onto.

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $E_k : \mathcal{X} \longrightarrow \mathcal{X}$ is one-to-one and defined as
$$E_k(x) \stackrel{\triangle}{=} E(k, x).$$
  - The domain and range of $E_k(\cdot)$ are the same, and $E_k(\cdot)$ is one-to-one,
  - Then $E_k(\cdot)$ is onto.
  - Therefore, $E_k(\cdot)$ is a bijection.

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $E_k : \mathcal{X} \longrightarrow \mathcal{X}$ is one-to-one and defined as
$$E_k(x) \triangleq E(k, x).$$
  - The domain and range of $E_k(\cdot)$ are the same, and $E_k(\cdot)$ is one-to-one,
  - Then $E_k(\cdot)$ is onto.
  - Therefore, $E_k(\cdot)$ is a bijection.
- We say E is efficient if there is
  - a deterministic, polynomial-time algorithm to computes $E(k, x)$,

Pseudorandom Permutation (PRP)

Keyed Permutation

A Keyed Permutation E is a two-input function defined over $(\mathcal{K}, \mathcal{X})$ as

$$E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}, \text{ where}$$

- the first input is called the key and denoted by k,
- the second input is just called the input.
- Choose k and fix it, we have a single-input function $E_k : \mathcal{X} \longrightarrow \mathcal{X}$ is one-to-one and defined as
$$E_k(x) \stackrel{\triangle}{=} E(k, x).$$
    - The domain and range of $E_k(\cdot)$ are the same, and $E_k(\cdot)$ is one-to-one,
    - Then $E_k(\cdot)$ is onto.
    - Therefore, $E_k(\cdot)$ is a bijection.
- We say E is efficient if there is
    - a deterministic, polynomial-time algorithm to computes $E(k, x)$, and
    - a deterministic, polynomial-time algorithm to compute $E^{-1}(k, x)$,

  given k and x as input.

Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

- $S_E = \left\{ E_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Prem}[\mathcal{X}]$.

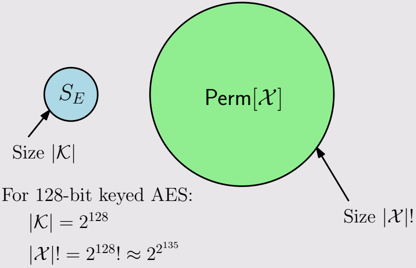
Intuition on Pseudorandom Permutation (PRP)

- $S_E = \left\{ E_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Prem}[\mathcal{X}]$.
- Size of $S_E = |\mathcal{K}|$.

Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

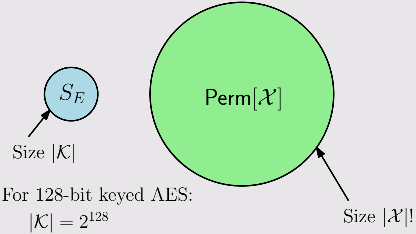
- $S_E = \left\{ E_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Prem}[\mathcal{X}]$.
- Size of $S_E = |\mathcal{K}|$.



Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

- $S_E = \left\{ E_k(\cdot) \mid k \xleftarrow{R} \mathcal{K} \right\} \subseteq \mathsf{Prem}[\mathcal{X}]$.
- Size of $S_E = |\mathcal{K}|$.



For 128-bit keyed AES:

$|\mathcal{K}| = 2^{128}$

$|\mathcal{X}|! = 2^{128}! \approx 2^{2^{135}}$

- Choosing E_k uniformly at random from S_E is equivalent of choosing k uniformly at random from $\mathcal{K}$.

Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

- Intuitively,
  - E is pseudorandom if the permutation E_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a permutation f chosen uniformly at random from $\mathsf{Prem}[\mathcal{X}]$.

Intuition on Pseudorandom Permutation (PRP)

- Intuitively,
  - E is pseudorandom if the permutation E_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a permutation f chosen uniformly at random from $\mathsf{Prem}[\mathcal{X}]$.
  - Equivalently, E is pseudorandom if no polynomial-time adversary can distinguish whether it is interacting with E_k (for randomly-chosen key k) or f (where f is chosen at random from $\mathsf{Prem}[\mathcal{X}]$).

Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

- Intuitively,
  - E is pseudorandom if the permutation E_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a permutation f chosen uniformly at random from $\mathsf{Prem}[\mathcal{X}]$.
  - Equivalently, E is pseudorandom if no polynomial-time adversary can distinguish whether it is interacting with E_k (for randomly-chosen key k) or f (where f is chosen at random from $\mathsf{Prem}[\mathcal{X}]$).

Pseudorandom Permutation (PRP)

A Pseudorandom Permutation (PRP) $E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}$ is a keyed permutation defined over $(\mathcal{K}, \mathcal{X})$, for which there exist deterministic, polynomial-time algorithms to compute $E(k,x)$ and $E^{-1}(k,x)$ given k and x.

Pseudorandom Permutation (PRP)

Intuition on Pseudorandom Permutation (PRP)

- Intuitively,
  - E is pseudorandom if the permutation E_k (for a randomly-chosen key k) is indistinguishable (for all practical purposes) from a permutation f chosen uniformly at random from $\mathsf{Prem}[\mathcal{X}]$.
  - Equivalently, E is pseudorandom if no polynomial-time adversary can distinguish whether it is interacting with E_k (for randomly-chosen key k) or f (where f is chosen at random from $\mathsf{Prem}[\mathcal{X}]$).

Pseudorandom Permutation (PRP)

A Pseudorandom Permutation (PRP) $E : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{X}$ is a keyed permutation defined over $(\mathcal{K}, \mathcal{X})$, for which there exist deterministic, polynomial-time algorithms to compute $E(k,x)$ and $E^{-1}(k,x)$ given k and x.

- Let $y := E(k,x)$
- x sometimes is referred as input data block, and
- y sometimes is referred as output data block.

PRP or Block Cipher Advantage

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0 : k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$,

PRP or Block Cipher Advantage

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.

PRP or Block Cipher Advantage

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.

PRP or Block Cipher Advantage

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
   - The queries are adaptive.

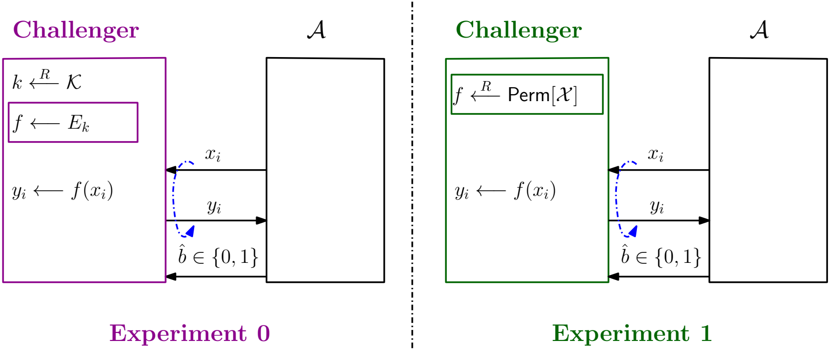
PRP or Block Cipher Advantage

PRP or Block Cipher Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
   - The queries are adaptive.
3. The adversary computes and outputs a bit $\hat{b} \in \{0, 1\}$.

PRP or Block Cipher Advantage



PRP or Block Cipher Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to E as

$$\mathsf{BCadv}[\mathcal{A}, E] = | \Pr[W_0] - \Pr[W_1] | .$$

We say that $\mathcal{A}$ is a Q-query PRP adversary if $\mathcal{A}$ issues at most Q queries.

PRP or Block Cipher Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to E as

$$\mathsf{BCadv}[\mathcal{A}, E] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

We say that $\mathcal{A}$ is a Q-query PRP adversary if $\mathcal{A}$ issues at most Q queries.

Secure PRP or Block Cipher

A PRP or Block Cipher E is secure if for all efficient adversaries $\mathcal{A}$, the value $\mathsf{BCadv}[\mathcal{A}, E]$ is negligible.

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0,1\}$.

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$,

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
    - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
    - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
3. The adversary submits a sequence of queries to the challenger.

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
    - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
    - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
3. The adversary submits a sequence of queries to the challenger.
    - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \textsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \textsf{Prem}[\mathcal{X}]$.
3. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.

PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
3. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
   - The queries are adaptive.

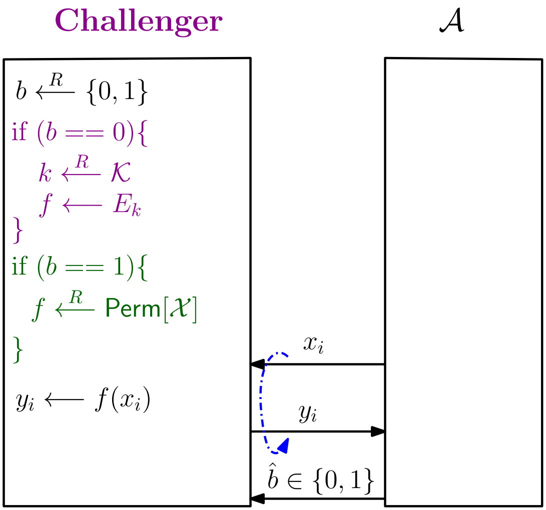
PRP or Block Cipher Advantage: Bit Guessing Version

PRP Indistinguishability Game

For a given PRP E, defined over $(\mathcal{K}, \mathcal{X})$, and for a given adversary $\mathcal{A}$, we define Experiment as:

1. Challenger first computes $b \xleftarrow{R} \{0, 1\}$.
2. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $k \xleftarrow{R} \mathcal{K}, f \longleftarrow E_k(\cdot)$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$.
3. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
   - The queries are adaptive.
4. The adversary computes and outputs a bit $\hat{b} \in \{0, 1\}$.

PRP or Block Cipher Advantage: Bit Guessing Version



Experiment

PRP or Block Cipher Advantage: Bit Guessing version

PRP or Block Cipher Advantage

Let W be the event that where $\mathcal{A}$ wins if $\mathcal{A}$ outputs $\hat{b} = b$. We define the advantage of $\mathcal{A}$ in the attack game with respect to E as

$$\mathsf{BCadv}^*[\mathcal{A}, E] = \left| \Pr[\hat{b} = b] - \frac{1}{2} \right|.$$

PRP or Block Cipher Advantage: Bit Guessing version

PRP or Block Cipher Advantage

Let W be the event that where $\mathcal{A}$ wins if $\mathcal{A}$ outputs $\hat{b} = b$. We define the advantage of $\mathcal{A}$ in the attack game with respect to E as

$$\mathsf{BCadv}^*[\mathcal{A}, E] = \left| \Pr[\hat{b} = b] - \frac{1}{2} \right|.$$

Theorem

For every PRP E and every PPT adversary $\mathcal{A}$, we have

$$\mathsf{BCadv}[\mathcal{A}, E] = 2 \cdot \mathsf{BCadv}^*[\mathcal{A}, E].$$

- Challenger's protocol in Experiment 1 is not very efficient.

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{X}|)$.

Implementation of Random Permutation

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}| \log_2(|\mathcal{X}|)$.
- Similar to random function, a lazy implementation of random permutation f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{X}|)$.
- Similar to random function, a lazy implementation of random permutation f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2. if $x_i = x_j$ for some $j < i$
3. then $y_i \longleftarrow y_j$

Implementation of Random Permutation

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}| \log_2(|\mathcal{X}|)$.
- Similar to random function, a lazy implementation of random permutation f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2. if $x_i = x_j$ for some $j < i$
3. then $y_i \longleftarrow y_j$
4. else {
5. $y_i \xleftarrow{R} \mathcal{X} \setminus \{y_1, \ldots, y_{i-1}\}$
6. Store (x_i, y_i)
7. }

Implementation of Random Permutation

- Challenger's protocol in Experiment 1 is not very efficient.
- Supposed to choose a very large random objec of size $|\mathcal{X}|\log_2(|\mathcal{X}|)$.
- Similar to random function, a lazy implementation of random permutation f:

1. upon receiving the i-th query $x_i \in \mathcal{X}$ from $\mathcal{A}$ do:
2. if $x_i = x_j$ for some $j < i$
3. then $y_i \longleftarrow y_j$
4. else {
5. $y_i \xleftarrow{R} \mathcal{X} \setminus \{y_1, \ldots, y_{i-1}\}$
6. Store (x_i, y_i)
7. }
8. send y_i to $\mathcal{A}$.

Strong PRP or Block Cipher Advantage

- The analogue for the case of strong pseudorandom permutations in practice is a block cipher.

Strong PRP or Block Cipher Advantage

- The analogue for the case of strong pseudorandom permutations in practice is a block cipher.
- It is often not stated in the literature that a block cipher is actually assumed to be a strong pseudorandom permutation.

Strong PRP or Block Cipher Advantage

- The analogue for the case of strong pseudorandom permutations in practice is a block cipher.
- It is often not stated in the literature that a block cipher is actually assumed to be a strong pseudorandom permutation.
- When proving security of a construction, it is important to specify whether the block cipher is being modeled as a pseudorandom permutation or a strong pseudorandom permutation.

Strong PRP or Block Cipher Advantage

- The analogue for the case of strong pseudorandom permutations in practice is a block cipher.
- It is often not stated in the literature that a block cipher is actually assumed to be a strong pseudorandom permutation.
- When proving security of a construction, it is important to specify whether the block cipher is being modeled as a pseudorandom permutation or a strong pseudorandom permutation.
- Although most block ciphers in use today are designed to satisfy the second, stronger requirement, a scheme that can be proven secure based on the former, weaker assumption may be preferable (since the requirements on the block cipher are potentially easier to satisfy).

Strong PRP or Block Cipher Advantage

- The analogue for the case of strong pseudorandom permutations in practice is a block cipher.
- It is often not stated in the literature that a block cipher is actually assumed to be a strong pseudorandom permutation.
- When proving security of a construction, it is important to specify whether the block cipher is being modeled as a pseudorandom permutation or a strong pseudorandom permutation.
- Although most block ciphers in use today are designed to satisfy the second, stronger requirement, a scheme that can be proven secure based on the former, weaker assumption may be preferable (since the requirements on the block cipher are potentially easier to satisfy).
- Strong pseudorandom permutations are useful in the design and analysis of efficient cryptographic schemes, we will only use pseudorandom permutations(that are not necessarily strong) in the rest of this lecture.

Strong PRP or Block Cipher Advantage

- We allow adversary to do two type of queries:

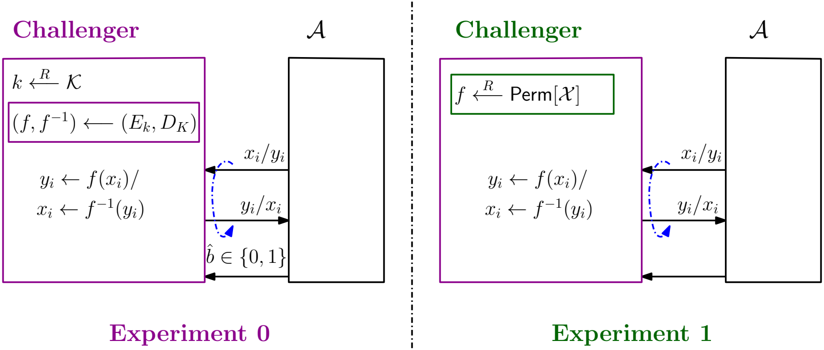
Strong PRP or Block Cipher Advantage

- We allow adversary to do two type of queries:
  - Forward queries: the adversary sends a value $x_i \in \mathcal{X}$ to the challenger, who sends $y_i := f(x_i)$ to the adversary;

Strong PRP or Block Cipher Advantage

- We allow adversary to do two type of queries:
  - Forward queries: the adversary sends a value $x_i \in \mathcal{X}$ to the challenger, who sends $y_i := f(x_i)$ to the adversary;
  - Inverse queries: the adversary sends a value $y_i \in \mathcal{X}$ to the challenger, who sends $x_i := f^{-1}(y_i)$ to the adversary.

Strong PRP or Block Cipher Advantage



PRP or Block Cipher Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to E as

$$\mathsf{strongBCadv}[\mathcal{A}, E] = \mid \Pr[W_0] - \Pr[W_1] \mid .$$

Strong PRP or Block Cipher Advantage

PRP or Block Cipher Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to E as

$$\mathsf{strongBCadv}[\mathcal{A}, E] = |\Pr[W_0] - \Pr[W_1]|.$$

Strongly Secure PRP or Block Cipher

A PRP or Block Cipher E is strongly secure if for all efficient adversaries $\mathcal{A}$, the value $\mathsf{strongBCadv}[\mathcal{A}, E]$ is negligible.

Is secure PRP (Block Cipher) is a secure PRF?

Question

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$, and let $N := |\mathcal{X}|$. Now suppose that $\mathfrak{E}$ is a secure block cipher; that is, no efficient adversary can effectively distinguish $\mathfrak{E}$ from a random permutation. **Does this imply that $\mathfrak{E}$ is also a secure PRF?**

Is secure PRP (Block Cipher) is a secure PRF?

Question

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$, and let $N := |\mathcal{X}|$. Now suppose that $\mathfrak{E}$ is a secure block cipher; that is, no efficient adversary can effectively distinguish $\mathfrak{E}$ from a random permutation. **Does this imply that $\mathfrak{E}$ is also a secure PRF?**

Answer

- Let E be a PRP defined over $(\mathcal{K}, \mathcal{X})$.

Is secure PRP (Block Cipher) is a secure PRF?

Question

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$, and let $N := |\mathcal{X}|$. Now suppose that $\mathfrak{E}$ is a secure block cipher; that is, no efficient adversary can effectively distinguish $\mathfrak{E}$ from a random permutation. **Does this imply that $\mathfrak{E}$ is also a secure PRF?**

Answer

- Let E be a PRP defined over $(\mathcal{K}, \mathcal{X})$.
    - Can be viewed as a PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{X})$.

Is secure PRP (Block Cipher) is a secure PRF?

Question

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$, and let $N := |\mathcal{X}|$. Now suppose that $\mathfrak{E}$ is a secure block cipher; that is, no efficient adversary can effectively distinguish $\mathfrak{E}$ from a random permutation. **Does this imply that $\mathfrak{E}$ is also a secure PRF?**

Answer

- Let E be a PRP defined over $(\mathcal{K}, \mathcal{X})$.
  - Can be viewed as a PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{X})$.

1. Case 1: N is small: No

Is secure PRP (Block Cipher) is a secure PRF?

Question

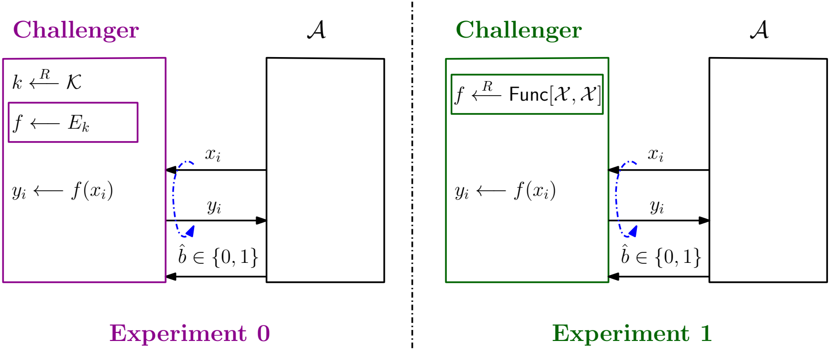
Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$, and let $N := |\mathcal{X}|$. Now suppose that $\mathfrak{E}$ is a secure block cipher; that is, no efficient adversary can effectively distinguish $\mathfrak{E}$ from a random permutation. **Does this imply that $\mathfrak{E}$ is also a secure PRF?**

Answer

- Let E be a PRP defined over $(\mathcal{K}, \mathcal{X})$.
  - Can be viewed as a PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{X})$.

1. Case 1: N is small: No
2. Case 2: N is Super-poly: Yes

PRF Advantage



Is secure PRP (Block Cipher) is a secure PRF?

Strategy of $\mathcal{A}$

- Make query on Q distinct values $x_i \in \mathcal{X}$.

Is secure PRP (Block Cipher) is a secure PRF?

Strategy of $\mathcal{A}$

- Make query on Q distinct values $x_i \in \mathcal{X}$.
- Checks whether $f(x_i) \stackrel{?}{=} f(x_j)$ for some $i \neq j$.

Is secure PRP (Block Cipher) is a secure PRF?

Strategy of $\mathcal{A}$

- Make query on Q distinct values $x_i \in \mathcal{X}$.
- Checks whether $f(x_i) \stackrel{?}{=} f(x_j)$ for some $i \neq j$.
- If Yes, Return 1, else Return 0.

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.
- $\Pr[W_0] = 0$

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.
- $\Pr[W_0] = 0$
- Total number of functions = N^N
- Total number of Permutations = $N!$
- Total number of functions that are not onto = $N^N - N!$

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.
- $\Pr[W_0] = 0$
- Total number of functions = N^N
- Total number of Permutations = $N!$
- Total number of functions that are not onto = $N^N - N!$
- $\frac{N!}{N^N} \leqslant \frac{1}{2}$, if $N \geqslant 2$

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.
- $\Pr[W_0] = 0$
- Total number of functions = N^N
- Total number of Permutations = $N!$
- Total number of functions that are not onto = $N^N - N!$
- $\frac{N!}{N^N} \leqslant \frac{1}{2}$, if $N \geqslant 2$
- $\Pr[W_1] = \frac{N^N - N!}{N^N} = 1 - \frac{N!}{N^N} \geqslant \frac{1}{2}$

Is secure PRP (Block Cipher) is a secure PRF?

Case 1: N is small

- Take $Q = N$.
- $\Pr[W_0] = 0$
- Total number of functions = N^N
- Total number of Permutations = $N!$
- Total number of functions that are not onto = $N^N - N!$
- $\frac{N!}{N^N} \leqslant \frac{1}{2}$, if $N \geqslant 2$
- $\Pr[W_1] = \frac{N^N - N!}{N^N} = 1 - \frac{N!}{N^N} \geqslant \frac{1}{2}$
- PRFadv $= \mid \Pr[W_0] - \Pr[W_1] \mid \geqslant \frac{1}{2}$, not negligible.

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

- Take $Q = 2N^{1/2}$, we will have a collision with probability almost 1.

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

- Take $Q = 2N^{1/2}$, we will have a collision with probability almost 1.
- The birthday attack is about the best that any adversary can do.

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

- Take $Q = 2N^{1/2}$, we will have a collision with probability almost 1.
- The birthday attack is about the best that any adversary can do.
- Make query on $Q = 2N^{1/2}$ distinct values $x_i \in \mathcal{X}$.

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

- Take $Q = 2N^{1/2}$, we will have a collision with probability almost 1.
- The birthday attack is about the best that any adversary can do.
- Make query on $Q = 2N^{1/2}$ distinct values $x_i \in \mathcal{X}$.
- Checks whether $f(x_i) \stackrel{?}{=} f(x_j)$ for some $i \neq j$.

Is secure PRP (Block Cipher) is a secure PRF?

Refined Strategy of $\mathcal{A}$

- By Birthday Paradox, if f is not a permutation, then $\mathcal{A}$ finds a collision, that is $f(x_i) = f(x_j)$ for some $i \neq j$, after Q queries with probability

$$\geqslant \frac{Q(Q-1)}{4N}.$$

- Take $Q = 2N^{1/2}$, we will have a collision with probability almost 1.
- The birthday attack is about the best that any adversary can do.
- Make query on $Q = 2N^{1/2}$ distinct values $x_i \in \mathcal{X}$.
- Checks whether $f(x_i) \stackrel{?}{=} f(x_j)$ for some $i \neq j$.
- If Yes, Return 1, else Return 0.

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$
-

$$\begin{aligned}
\Pr[W_1] &= \Pr[\text{Collision}] \\
&= \Pr[f \in \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] + \Pr[f \notin \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] \\
&= 0 + \Pr[\text{Collision} \mid f \notin \mathsf{Prem}[\mathcal{X}]] \cdot \Pr[f \notin \mathsf{Prem}[\mathcal{X}]] \\
&= \frac{Q(Q-1)}{4N}\left(1 - \frac{N!}{N^N}\right)
\end{aligned}$$

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$
- $$\begin{aligned}\Pr[W_1] &= \Pr[\text{Collision}] \\ &= \Pr[f \in \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] + \Pr[f \notin \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] \\ &= 0 + \Pr[\text{Collision} \mid f \notin \mathsf{Prem}[\mathcal{X}]] \cdot \Pr[f \notin \mathsf{Prem}[\mathcal{X}]] \\ &= \frac{Q(Q-1)}{4N}\left(1 - \frac{N!}{N^N}\right)\end{aligned}$$
- Q is poly-bounded and N is superpoly, then $\frac{Q(Q-1)}{4N}$ is negligible.

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$
- $$\begin{aligned} \Pr[W_1] &= \Pr[\text{Collision}] \\ &= \Pr[f \in \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] + \Pr[f \notin \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] \\ &= 0 + \Pr[\text{Collision} \mid f \notin \mathsf{Prem}[\mathcal{X}]] \cdot \Pr[f \notin \mathsf{Prem}[\mathcal{X}]] \\ &= \frac{Q(Q-1)}{4N}\left(1 - \frac{N!}{N^N}\right) \end{aligned}$$
- Q is poly-bounded and N is superpoly, then $\frac{Q(Q-1)}{4N}$ is negligible.
- $\left(1 - \frac{N!}{N^N}\right) < 1$.

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$
- $$\begin{aligned} \Pr[W_1] &= \Pr[\text{Collision}] \\ &= \Pr[f \in \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] + \Pr[f \notin \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] \\ &= 0 + \Pr[\text{Collision} \mid f \notin \mathsf{Prem}[\mathcal{X}]] \cdot \Pr[f \notin \mathsf{Prem}[\mathcal{X}]] \\ &= \frac{Q(Q-1)}{4N}\left(1 - \frac{N!}{N^N}\right) \end{aligned}$$
- Q is poly-bounded and N is superpoly, then $\frac{Q(Q-1)}{4N}$ is negligible.
- $\left(1 - \frac{N!}{N^N}\right) < 1$.
- $\Pr[W_1] \leqslant$ negligible

Is secure PRP (Block Cipher) is a secure PRF?

Case 2: N is Super-Poly

- As $\mathcal{A}$ is efficient PPT adversary, Q must be poly-bounded. Therefore, we can not take $Q = N$.
- $\Pr[W_0] = 0$
-

$$\begin{aligned}
\Pr[W_1] &= \Pr[\text{Collision}] \\
&= \Pr[f \in \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] + \Pr[f \notin \mathsf{Prem}[\mathcal{X}] \wedge \text{Collision}] \\
&= 0 + \Pr[\text{Collision} \mid f \notin \mathsf{Prem}[\mathcal{X}]] \cdot \Pr[f \notin \mathsf{Prem}[\mathcal{X}]] \\
&= \frac{Q(Q-1)}{4N}\left(1 - \frac{N!}{N^N}\right)
\end{aligned}$$

- Q is poly-bounded and N is superpoly, then $\frac{Q(Q-1)}{4N}$ is negligible.
- $\left(1 - \frac{N!}{N^N}\right) < 1$.
- $\Pr[W_1] \leqslant$ negligible
- $\mathsf{PRFadv} = |\Pr[W_0] - \Pr[W_1]| \leqslant$ negligible.

PF Indistinguishability Game

For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$,

Permutations Vs. Functions

PF Indistinguishability Game

For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0 : f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$, and
   - if $b = 1 : f \xleftarrow{R} \mathsf{Func}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.

PF Indistinguishability Game

For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0 : f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$, and
   - if $b = 1 : f \xleftarrow{R} \mathsf{Func}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.

Permutations Vs. Functions

PF Indistinguishability Game

For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0 : f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$, and
   - if $b = 1 : f \xleftarrow{R} \mathsf{Func}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.

Permutations Vs. Functions

PF Indistinguishability Game

For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
   - if $b = 0$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$, and
   - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
   - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
   - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
   - The queries are adaptive.

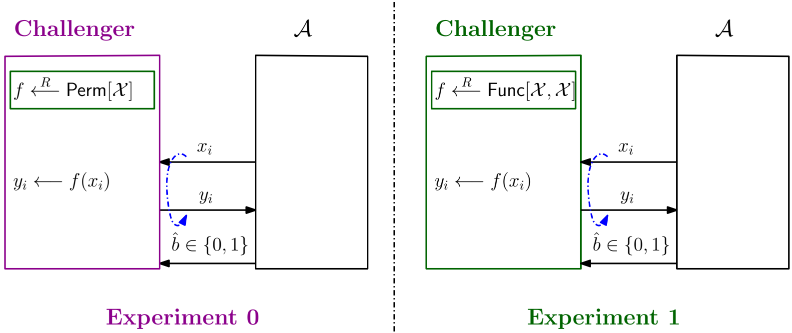
Permutations Vs. Functions

PF Indistinguishability Game

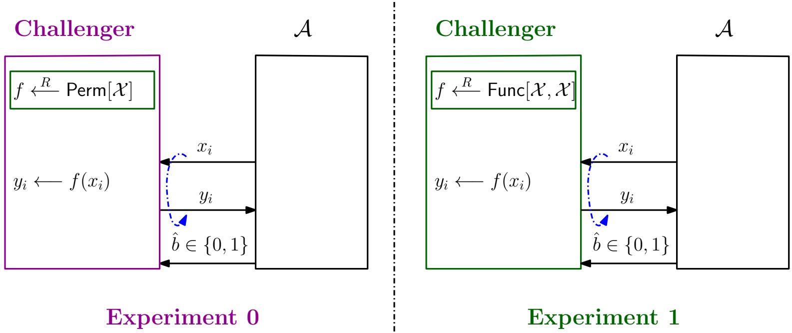
For a given finite set $\mathcal{X}$, and for a given adversary $\mathcal{A}$, we define two experiments, Experiment 0 and Experiment 1. For $b = 0, 1$, we define Experiment b as:

1. The challenger selects $f \in \mathsf{Prem}[\mathcal{X}]$ as follows:
    - if $b = 0$: $f \xleftarrow{R} \mathsf{Prem}[\mathcal{X}]$, and
    - if $b = 1$: $f \xleftarrow{R} \mathsf{Func}[\mathcal{X}]$.
2. The adversary submits a sequence of queries to the challenger.
    - For $i = 1, 2, \ldots$ the i-th query is an input data block $x_i \in \mathcal{X}$.
    - The challenger computes the output data block $y_i \longleftarrow f(x_i) \in \mathcal{X}$, and gives y_i to the adversary.
    - The queries are adaptive.
3. The adversary computes and outputs a bit $\hat{b} \in \{0, 1\}$.

Permutations Vs. Functions



Permutations Vs. Functions



PF Advantage

For $b = 0, 1$, let W_b be the event that $\mathcal{A}$ outputs 1 in Experiment b. We define $\mathcal{A}$'s advantage with respect to $\mathcal{X}$ as

$$\mathsf{PFadv}[\mathcal{A},\mathcal{X}] = |\Pr[W_0] - \Pr[W_1]|.$$

We say that $\mathcal{A}$ is a Q-query PF adversary if $\mathcal{A}$ issues at most Q queries.

Theorem

Let $\mathcal{X}$ be a finite set of size N. Let $\mathcal{A}$ be an adversary that makes at most Q queries to its challenger. Then

$$\mathsf{PFadv}[\mathcal{A}, \mathcal{X}] \leqslant \frac{Q^2}{2N}.$$

Theorem

Let $\mathcal{X}$ be a finite set of size N. Let $\mathcal{A}$ be an adversary that makes at most Q queries to its challenger. Then

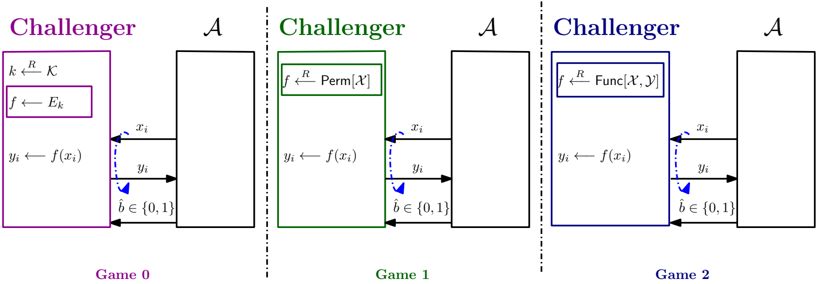
$$\mathsf{PFadv}[\mathcal{A},\mathcal{X}] \leqslant \frac{Q^2}{2N}.$$

PRF Switching Lemma

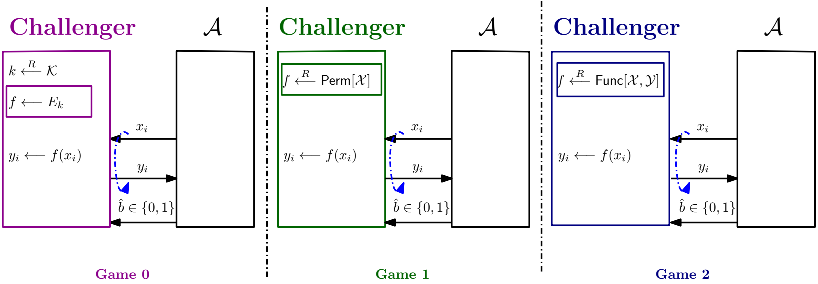
Let $\mathfrak{E} = (\mathcal{E},\mathcal{D})$ be a block cipher defined over $(\mathcal{K},\mathcal{X})$, and let $N := |\mathcal{X}|$. Let $\mathcal{A}$ be an adversary that makes at most Q queries to its challenger. Then

$$|\mathsf{BCadv}[\mathcal{A},\mathfrak{E}] - \mathsf{PRFadv}[\mathcal{A},\mathcal{E}]| \leqslant \frac{Q^2}{2N}.$$

PRF Switching Lemma



PRF Switching Lemma



PF AdvantagePRF Switching Lemma

- $p_0 = \Pr[\mathcal{A}$ outputs 1 in Game 0$]$.
- $p_1 = \Pr[\mathcal{A}$ outputs 1 in Game 1$]$.
- $p_2 = \Pr[\mathcal{A}$ outputs 1 in Game 2$]$.

PRF Switching Lemma

- $\mathsf{BCadv}[\mathcal{A},\mathfrak{E}] = \mid p_1 - p_0 \mid$
- $\mathsf{PRFadv}[\mathcal{A},\mathcal{E}] = \mid p_2 - p_0 \mid$

$$
\begin{aligned}
|\mathsf{BCadv}[\mathcal{A},\mathfrak{E}] - \mathsf{PRFadv}[\mathcal{A},\mathcal{E}]| &= || p_1 - p_0 | - | p_2 - p_0 || \\
&\leqslant |p_1 - p_0 - p_2 - p_0| \\
&= |p_2 - p_1| \\
&= \mathsf{PFadv}[\mathcal{A},\mathcal{X}] \\
&\leqslant \frac{Q^2}{2N}.
\end{aligned}
$$

Modes of Operation

- Essentially, a way of encrypting arbitrary-length messages using a block cipher or PRP.
- Arbitrary-length messages can be unambiguously padded to a total length that is a multiple of any desired block size by appending a 1 followed by sufficiently-many 0s.
- Assume that the length of the plaintext message is an exact multiple of the block size.
- Let data block size of pseudorandom permutation/block cipher = n
- Let $\mathcal{X} = \{0,1\}^n$
- Consider messages consisting of ℓ blocks each of length n.

Modes of Operation

Modes of Operation

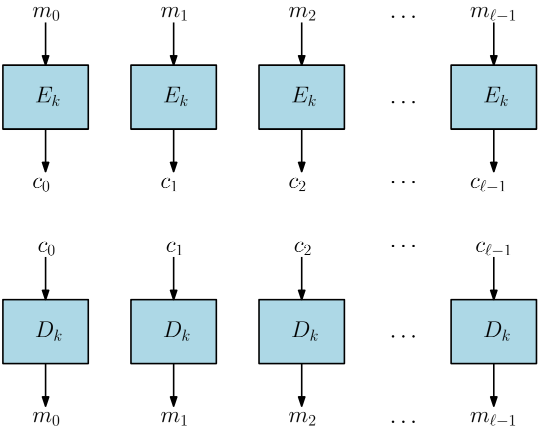
Five most popular modes of operations:

- Electronic CodeBook mode (ECB mode),
- Cipher Block Chaining mode (CBC mode),
- Output FeedBack mode (OFB mode),
- Cipher FeedBack mode (CFB mode), and
- Counter mode (CTR mode).

Electronic codebook mode (ECB mode)

$$\begin{array}{ccccc}
m_0 & m_1 & m_2 & \cdots & m_{\ell-1} \\
\downarrow & \downarrow & \downarrow & & \downarrow \\
\boxed{E_k} & \boxed{E_k} & \boxed{E_k} & \cdots & \boxed{E_k} \\
\downarrow & \downarrow & \downarrow & & \downarrow \\
c_0 & c_1 & c_2 & \cdots & c_{\ell-1}
\end{array}$$

Electronic codebook mode (ECB mode)



Electronic codebook mode (ECB mode)

Encryption(m, k)

1. For $i = 0, 1, \ldots, \ell - 1$ do
2. Compute $c_i := E_k(m_i) = E(k, m_i)$
3. End For;
4. Return $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Electronic codebook mode (ECB mode)

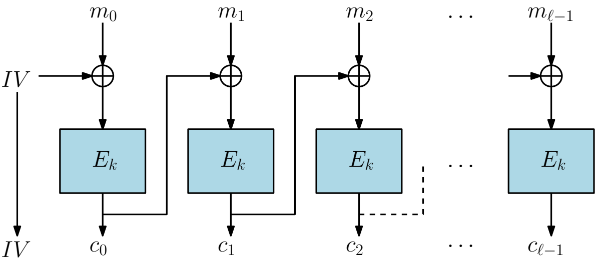
Encryption(m, k)

1. For $i = 0, 1, \ldots, \ell - 1$ do
2. Compute $c_i := E_k(m_i) = E(k, m_i)$
3. End For;
4. Return $c = (c_0, c_1, \ldots, c_{\ell-1})$.

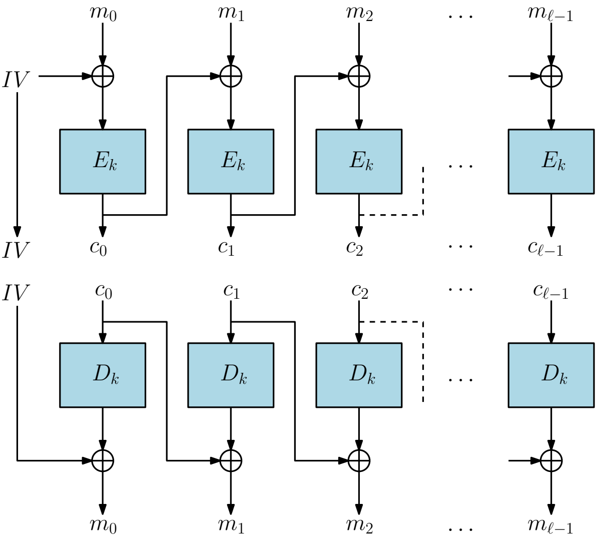
Decryption(c, k)

1. For $i = 0, 1, \ldots, \ell - 1$ do
2. Compute $m_i := E_k^{-1}(m_i) = D(k, c_i)$
3. End For;
4. Return $m = (m_0, m_1, \ldots, m_{\ell-1})$.

Cipher Block Chaining mode (CBC mode)



Cipher Block Chaining mode (CBC mode)



Cipher Block Chaining mode (CBC mode)

Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. Compute $c_0 := E_k(IV \oplus m_0) = E(k, IV \oplus m_0)$
3. For $i = 1, \ldots, \ell - 1$ do
4.   Compute $c_i := E_k(m_i \oplus c_{i-1}) = E(k, m_i \oplus c_{i-1})$
5. End For;
6. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Cipher Block Chaining mode (CBC mode)

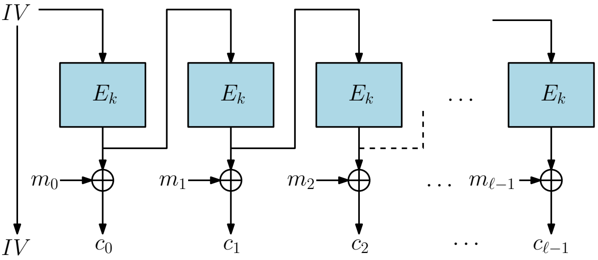
Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. Compute $c_0 := E_k(IV \oplus m_0) = E(k, IV \oplus m_0)$
3. For $i = 1, \ldots, \ell - 1$ do
4. Compute $c_i := E_k(m_i \oplus c_{i-1}) = E(k, m_i \oplus c_{i-1})$
5. End For;
6. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

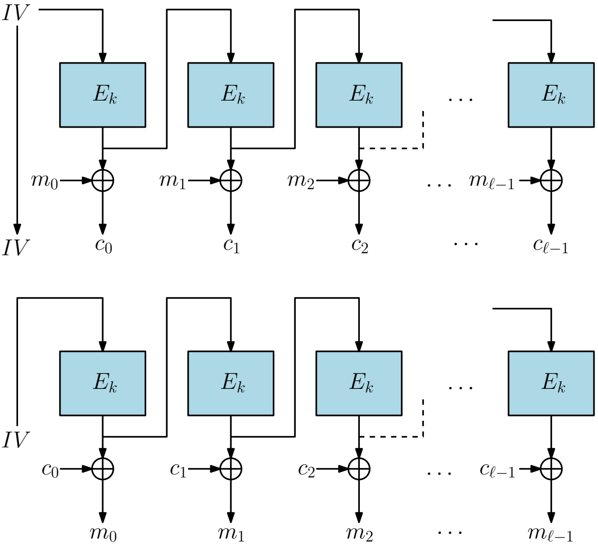
Decryption$((IV, c), k)$

1. Compute $m_0 := D_k(c_0) \oplus IV = D(k, c_0) \oplus IV$
2. For $i = 1, \ldots, \ell - 1$ do
3. Compute $m_i := D_k(c_i) \oplus c_{i-1} = D(k, c_i) \oplus c_{i-1}$
4. End For;
5. Return $m = (m_0, m_1, \ldots, m_{\ell-1})$.

Output FeedBack mode (OFB mode)



Output FeedBack mode (OFB mode)



Output FeedBack mode (OFB mode)

Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. $y_0 := E_k(IV) = E(k, IV)$; $c_0 := y_0 \oplus m_0$
3. For $i = 1, \ldots, \ell - 1$ do
4. Compute $y_i := E_k(y_{i-1}) = E(k, y_{i-1})$
5. Compute $c_i := y_i \oplus m_i$
6. End For;
7. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Output FeedBack mode (OFB mode)

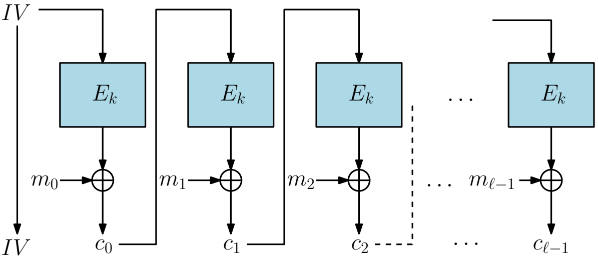
Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. $y_0 := E_k(IV) = E(k, IV)$; $c_0 := y_0 \oplus m_0$
3. For $i = 1, \ldots, \ell - 1$ do
4. Compute $y_i := E_k(y_{i-1}) = E(k, y_{i-1})$
5. Compute $c_i := y_i \oplus m_i$
6. End For;
7. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

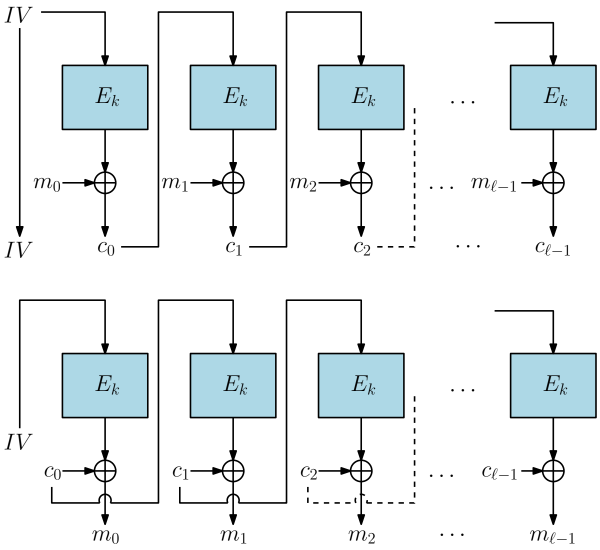
Decryption$((IV, c), k)$

1. $y_0 := E_k(IV) = E(k, IV)$; $m_0 := y_0 \oplus c_0$
2. For $i = 1, \ldots, \ell - 1$ do
3. Compute $y_i := E_k(y_{i-1}) = E(k, y_{i-1})$
4. Compute $m_i := y_i \oplus c_i$
5. End For;
6. Return $m = (m_0, m_1, \ldots, m_{\ell-1})$.

Cipher FeedBack mode (CFB mode)



Cipher FeedBack mode (CFB mode)



Cipher FeedBack mode (CFB mode)

Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. $c_0 := E_k(IV) \oplus m_0 := E(k, IV) \oplus m_0$
3. For $i = 1, \ldots, \ell - 1$ do
4. Compute $c_i := E_k(c_{i-1}) \oplus m_i = E(k, c_{i-1}) \oplus m_i$
5. End For;
6. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Cipher FeedBack mode (CFB mode)

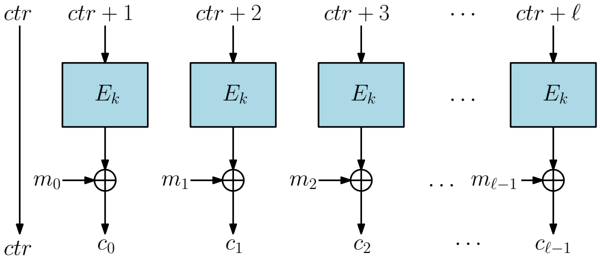
Encryption(m, k)

1. Choose a random $IV \xleftarrow{R} \mathcal{X}$
2. $c_0 := E_k(IV) \oplus m_0 := E(k, IV) \oplus m_0$
3. For $i = 1, \ldots, \ell - 1$ do
4. Compute $c_i := E_k(c_{i-1}) \oplus m_i = E(k, c_{i-1}) \oplus m_i$
5. End For;
6. Return (IV, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

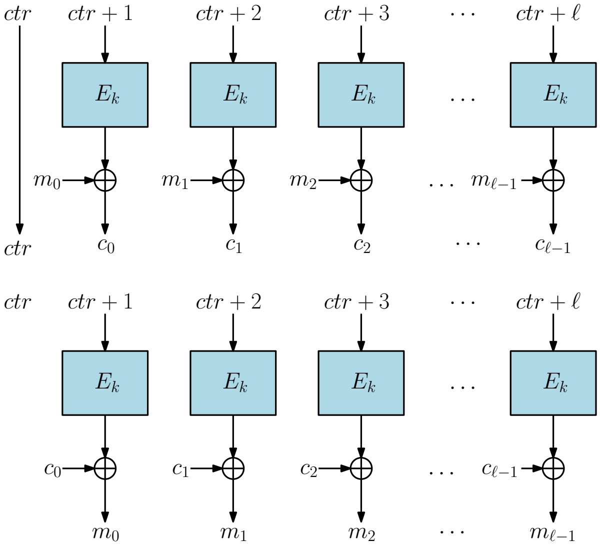
Decryption$((IV, c), k)$

1. $m_0 := E_k(IV) \oplus c_0 := E(k, IV) \oplus c_0$
2. For $i = 1, \ldots, \ell - 1$ do
3. Compute $m_i := E_k(c_{i-1}) \oplus c_i = E(k, c_{i-1}) \oplus c_i$
4. End For;
5. Return $m = (m_0, m_1, \ldots, m_{\ell-1})$.

Counter mode (CTR mode)



Counter mode (CTR mode)



Counter mode (CTR mode)

Encryption(m, k)

1. Choose a random $ctr \xleftarrow{R} \mathcal{X}$
3. For $i = 0, 1, \ldots, \ell - 1$ do
4. Compute $ctr_i := ctr + i + 1 \pmod{2^n}$
5. Compute $c_i := E_k(ctr_i) \oplus m_i = E(k, ctr_i) \oplus m_i$
6. End For;
7. Return (ctr, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Counter mode (CTR mode)

Encryption(m, k)

1. Choose a random $ctr \xleftarrow{R} \mathcal{X}$
3. For $i = 0, 1, \ldots, \ell - 1$ do
4. Compute $ctr_i := ctr + i + 1 \pmod{2^n}$
5. Compute $c_i := E_k(ctr_i) \oplus m_i = E(k, ctr_i) \oplus m_i$
6. End For;
7. Return (ctr, c), where $c = (c_0, c_1, \ldots, c_{\ell-1})$.

Decryption$((ctr, c), k)$

1. For $i = 0, 1, \ldots, \ell - 1$ do
2. Compute $ctr_i := ctr + i + 1 \pmod{2^n}$
3. Compute $m_i := E_k(ctr_i) \oplus c_i = E(k, ctr_i) \oplus c_i$
4. End For;
5. Return $m = (m_0, m_1, \ldots, m_{\ell-1})$.

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Proof

- Choose m_0, m_1 in such a way that:
  - For all $i \neq j$, $m_{0,i} \neq m_{0,j}$,

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Proof

- Choose m_0, m_1 in such a way that:
    - For all $i \neq j$, $m_{0,i} \neq m_{0,j}$, and
    - $m_{1,0} = m_{1,1}$

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Proof

- Choose m_0, m_1 in such a way that:
  - For all $i \neq j$, $m_{0,i} \neq m_{0,j}$, and
  - $m_{1,0} = m_{1,1}$
- Output 1 if $c_{1,0} = c_{1,1}$, else 0.

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Proof

- Choose m_0, m_1 in such a way that:
  - For all $i \neq j$, $m_{0,i} \neq m_{0,j}$, and
  - $m_{1,0} = m_{1,1}$
- Output 1 if $c_{1,0} = c_{1,1}$, else 0.
- $\mathsf{INDadv} = \mid \Pr[W_1] - \Pr[W_0] \mid = \mid 1 - 0 \mid = 1.$

Electronic codebook mode (ECB mode)

Theorem

ECB-mode encryption does not have indistinguishable encryptions in the presence of an eavesdropper.

Proof

- Choose m_0, m_1 in such a way that:
  - For all $i \neq j$, $m_{0,i} \neq m_{0,j}$, and
  - $m_{1,0} = m_{1,1}$
- Output 1 if $c_{1,0} = c_{1,1}$, else 0.
- $\mathsf{INDadv} = \mid \Pr[W_1] - \Pr[W_0] \mid = \mid 1 - 0 \mid = 1.$
- Not secure.

Theorem

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher. Let $\ell \geqslant 1$ be any poly-bounded value, and let $\mathfrak{E}' = (\mathcal{E}', \mathcal{D}')$ be the ℓ-wise ECB cipher derived from $\mathfrak{E}$, but with the message space restricted to all sequences of at most ℓ distinct data blocks. If $\mathfrak{E}$ is a secure block cipher, then $\mathfrak{E}'$ is a semantically secure cipher.

Electronic codebook mode (ECB mode)

Theorem

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher. Let $\ell \geqslant 1$ be any poly-bounded value, and let $\mathfrak{E}' = (\mathcal{E}', \mathcal{D}')$ be the ℓ-wise ECB cipher derived from $\mathfrak{E}$, but with the message space restricted to all sequences of at most ℓ distinct data blocks. If $\mathfrak{E}$ is a secure block cipher, then $\mathfrak{E}'$ is a semantically secure cipher.

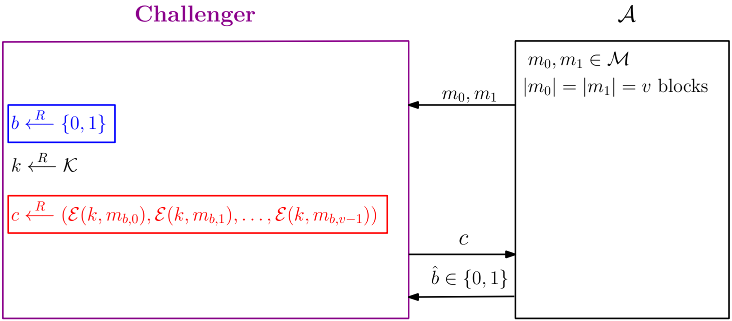
In particular, for every indistinguishability adversary $\mathcal{A}$ that plays symmetric-encryption indistinguishability with respect to $\mathfrak{E}'$, there exists a BC adversary $\mathcal{B}$ that plays PRP indistinguishability with respect to $\mathfrak{E}$, where $\mathcal{B}$ calls $\mathcal{A}$ as subroutine, such that

$$\mathsf{INDadv}[\mathcal{A}, \mathfrak{E}'] = 2 \cdot \mathsf{BCadv}[\mathcal{B}, \mathfrak{E}].$$

Electronic codebook mode (ECB mode)

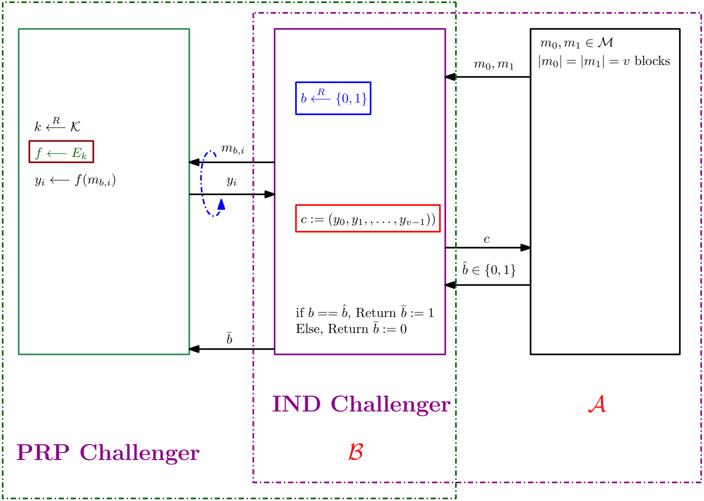
Proof

- If $\mathfrak{E}$ is defined over $(\mathcal{K}, \mathcal{X})$, let $\mathcal{X}_*^{\leqslant \ell}$ denote the set of all sequences of at most ℓ distinct elements of $\mathcal{X}$.



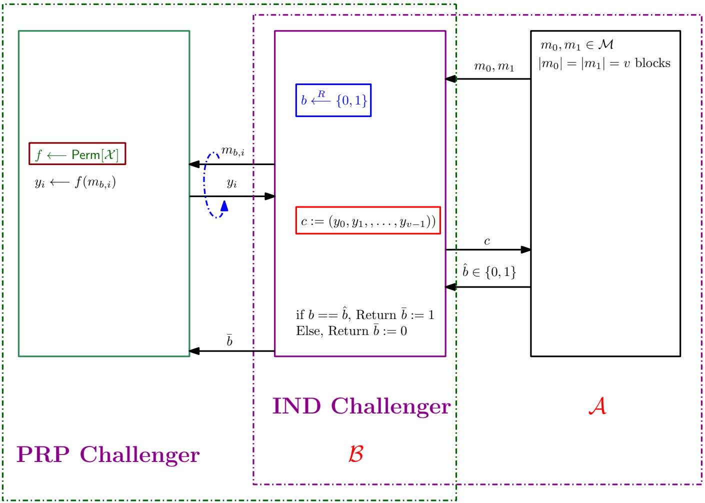
IND Bit-Guessing Experiment

Electronic codebook mode (ECB mode)



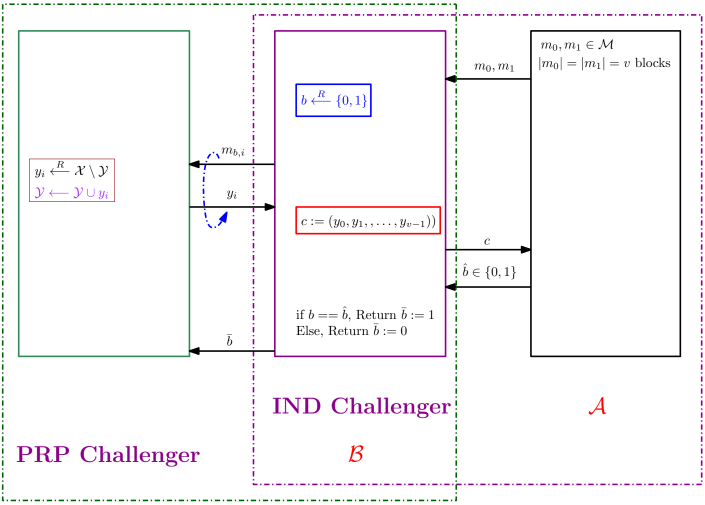
Game 0

Electronic codebook mode (ECB mode)



Game 1

Electronic codebook mode (ECB mode)



Game 2

Electronic codebook mode (ECB mode)

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.

Electronic codebook mode (ECB mode)

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.
- Notice that, $\mathcal{B}$ outputs when 1 if and only if $\mathcal{A}$ outputs $\hat{b} = b$.

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.
- Notice that, $\mathcal{B}$ outputs when 1 if and only if $\mathcal{A}$ outputs $\hat{b} = b$.
- That is, $\Pr[W_b] = \Pr[\bar{b} = 1] = \Pr[\hat{b} = b]$.

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.
- Notice that, $\mathcal{B}$ outputs when 1 if and only if $\mathcal{A}$ outputs $\hat{b} = b$.
- That is, $\Pr[W_b] = \Pr[\bar{b} = 1] = \Pr[\hat{b} = b]$.
- In Game 0:
  - The view of $\mathcal{A}$ is exactly the view of $\mathcal{A}$ in bit-guessing version of symmetric-encryption indistinguishability game.

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.
- Notice that, $\mathcal{B}$ outputs when 1 if and only if $\mathcal{A}$ outputs $\hat{b} = b$.
- That is, $\Pr[W_b] = \Pr[\bar{b} = 1] = \Pr[\hat{b} = b]$.
- In Game 0:
  - The view of $\mathcal{A}$ is exactly the view of $\mathcal{A}$ in bit-guessing version of symmetric-encryption indistinguishability game.

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}'] = \left| \Pr[W_0] - \frac{1}{2} \right|.$$

Electronic codebook mode (ECB mode)

Proof

- For $b = 0, 1, 2$, W_b be the event where $\mathcal{B}$ outputs when 1.
- Notice that, $\mathcal{B}$ outputs when 1 if and only if $\mathcal{A}$ outputs $\hat{b} = b$.
- That is, $\Pr[W_b] = \Pr[\bar{b} = 1] = \Pr[\hat{b} = b]$.
- In Game 0:
  - The view of $\mathcal{A}$ is exactly the view of $\mathcal{A}$ in bit-guessing version of symmetric-encryption indistinguishability game.

$$\mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}'] = \left|\Pr[W_0] - \frac{1}{2}\right|.$$

  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 0 of PRP indistinguishability game.

Electronic codebook mode (ECB mode)

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.

Electronic codebook mode (ECB mode)

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.
  - Therefore,
$$\mathsf{BCadv}[\mathcal{B},\mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|\,.$$

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.
  - Therefore,
$$\mathsf{BCadv}[\mathcal{B},\mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|\,.$$
- In Game 2:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Game 1.

Electronic codebook mode (ECB mode)

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.
  - Therefore,

$$\mathsf{BCadv}[\mathcal{B}, \mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|\,.$$

- In Game 2:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Game 1.
  - In Game 1, $\mathcal{B}$ interact with a PRP given from definitional point of view.

Electronic codebook mode (ECB mode)

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.
  - Therefore,
  $$\mathsf{BCadv}[\mathcal{B},\mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|.$$
- In Game 2:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Game 1.
  - In Game 1, $\mathcal{B}$ interact with a PRP given from definitional point of view.
  - In Game 2, $\mathcal{B}$ interact with a PRP given from implementation point of view.

Electronic codebook mode (ECB mode)

Proof

- In Game 1:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Experiment 1 of PRP indistinguishability game.
  - Therefore,
$$\mathsf{BCadv}[\mathcal{B},\mathfrak{E}] = |\Pr[W_0] - \Pr[W_1]|\,.$$
- In Game 2:
  - The view of $\mathcal{B}$ is exactly the view of $\mathcal{B}$ in Game 1.
  - In Game 1, $\mathcal{B}$ interact with a PRP given from definitional point of view.
  - In Game 2, $\mathcal{B}$ interact with a PRP given from implementation point of view.
  - Therefore,
$$\Pr[W_1] = \Pr[W_2].$$

Electronic codebook mode (ECB mode)

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.

Electronic codebook mode (ECB mode)

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b,

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b, and that implies, $\bar{b}$ is independent of b.

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b, and that implies, $\bar{b}$ is independent of b.

$$\Pr[W_2] = \frac{1}{2}.$$

Then,

$$\mathsf{BCadv}[\mathcal{B}, \mathfrak{E}] \quad = \quad |\Pr[W_0] - \Pr[W_1]| = |\Pr[W_0] - \Pr[W_2]|$$

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b, and that implies, $\bar{b}$ is independent of b.

$$\Pr[W_2] = \frac{1}{2}.$$

Then,

$$\begin{aligned}\mathsf{BCadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0] - \Pr[W_1]| = |\Pr[W_0] - \Pr[W_2]| \\ &= \left|\Pr[W_0] - \frac{1}{2}\right|\end{aligned}$$

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b, and that implies, $\bar{b}$ is independent of b.

$$\Pr[W_2] = \frac{1}{2}.$$

Then,

$$\begin{aligned}
\mathsf{BCadv}[\mathcal{B}, \mathfrak{E}] &= |\Pr[W_0] - \Pr[W_1]| = |\Pr[W_0] - \Pr[W_2]| \\
&= \left|\Pr[W_0] - \frac{1}{2}\right| \\
&= \mathsf{INDadv}^*[\mathcal{A}, \mathfrak{E}'].
\end{aligned}$$

Proof

- In Game 2:
  - The y_is are chosen uniformly at random from $\mathcal{X} \setminus \mathcal{Y}$ and are independent of $m_{b,i}$s.
  - $\hat{b}$ is independent of b, and that implies, $\bar{b}$ is independent of b.

$$\Pr[W_2] = \frac{1}{2}.$$

Then,

$$\begin{aligned} \mathsf{BCadv}[\mathcal{B},\mathfrak{E}] &= |\Pr[W_0] - \Pr[W_1]| = |\Pr[W_0] - \Pr[W_2]| \\ &= \left|\Pr[W_0] - \frac{1}{2}\right| \\ &= \mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}']. \end{aligned}$$

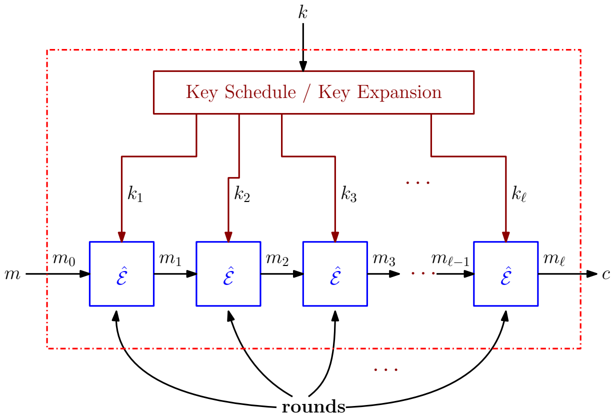
- As $\mathsf{INDadv}[\mathcal{A},\mathfrak{E}'] = 2 \cdot \mathsf{INDadv}^*[\mathcal{A},\mathfrak{E}']$,

$$\mathsf{INDadv}[\mathcal{A},\mathfrak{E}'] = 2 \cdot \mathsf{BCadv}[\mathcal{B},\mathfrak{E}].$$

Design Paradigm

- Commonly designed as iterated cipher.
- Has a Round Function, say $(\hat{\mathcal{E}}, \hat{\mathcal{D}})$.
- Has a Key Schedule algorithm.
  - $k_1, k_2, \ldots, k_\ell$ are called Key.
- Round function is applied multiple times, say ℓ times.

Block Cipher Design Paradigm

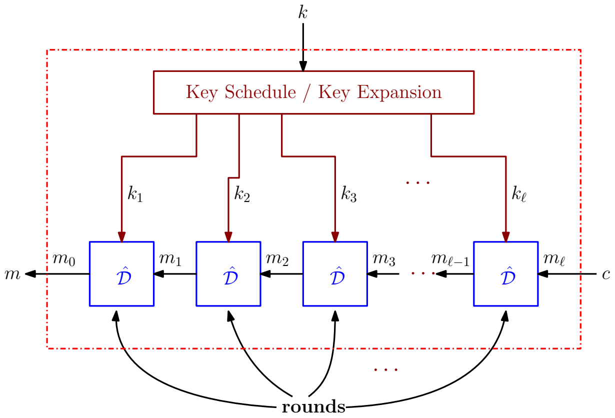


Block Cipher Design Paradigm

$c := \mathcal{E}(k, m)$

$$
\begin{array}{rcl}
m_0 & \longleftarrow & m; \\
m_1 & \longleftarrow & \hat{\mathcal{E}}(k_1, m_0); \\
m_2 & \longleftarrow & \hat{\mathcal{E}}(k_2, m_1); \\
m_3 & \longleftarrow & \hat{\mathcal{E}}(k_3, m_2); \\
 & \vdots & \\
m_\ell & \longleftarrow & \hat{\mathcal{E}}(k_\ell, m_{\ell-1}); \\
c & \longleftarrow & m_\ell;
\end{array}
$$

Block Cipher Design Paradigm



Block Cipher Design Paradigm

$m := \mathcal{D}(k, c)$

$$
\begin{aligned}
c &\longleftarrow m_\ell; \\
m_{\ell-1} &\longleftarrow \hat{\mathcal{D}}(k_\ell, m_\ell); \\
&\vdots \\
m_2 &\longleftarrow \hat{\mathcal{D}}(k_3, m_3); \\
m_1 &\longleftarrow \hat{\mathcal{D}}(k_2, m_2); \\
m_0 &\longleftarrow \hat{\mathcal{D}}(k_1, m_1); \\
m &\longleftarrow m_0;
\end{aligned}
$$

Round Function

- Each round function must be a permutation.

Round Function

- Each round function must be a permutation.
- Each round function performs two types of operations:

Round Function

- Each round function must be a permutation.
- Each round function performs two types of operations:
    - Confusion,

Round Function

- Each round function must be a permutation.
- Each round function performs two types of operations:
  - Confusion, and
  - Diffusion.

Confusion

Confusion is an encryption operation where the relationship between key and ciphertext is obscured.

Confusion

Confusion is an encryption operation where the relationship between key and ciphertext is obscured.

- $\hat{\mathcal{E}}_k : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}$

Confusion

Confusion is an encryption operation where the relationship between key and ciphertext is obscured.

- $\hat{\mathcal{E}}_k : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}$
- $\hat{\mathcal{E}}_k$ is designed as $\hat{\mathcal{E}}_k = (f_1, f_2, \ldots, f_m)$, where
  - f_is are chosen based on k,
  - Each f_i is a permutation defined as $f_i : \{0,1\}^l \longrightarrow \{0,1\}^l, \forall i = 1, \ldots, m$.

Confusion

Confusion is an encryption operation where the relationship between key and ciphertext is obscured.

- $\hat{\mathcal{E}}_k : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}$
- $\hat{\mathcal{E}}_k$ is designed as $\hat{\mathcal{E}}_k = (f_1, f_2, \ldots, f_m)$, where
  - f_is are chosen based on k,
  - Each f_i is a permutation defined as $f_i : \{0,1\}^l \longrightarrow \{0,1\}^l, \forall i = 1, \ldots, m$.
- Let $x \in \{0,1\}^{lm}$, then we can write x as

$$x = (x_{<1>}, x_{<2>}, \ldots, x_{<m>}), \text{ where } x_{<i>} = x_{(i-1)l+1} x_{(i-1)l+2} \cdots x_{(i-1)l+l}.$$

Confusion

Confusion is an encryption operation where the relationship between key and ciphertext is obscured.

- $\hat{\mathcal{E}}_k : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}$
- $\hat{\mathcal{E}}_k$ is designed as $\hat{\mathcal{E}}_k = (f_1, f_2, \ldots, f_m)$, where
  - f_is are chosen based on k,
  - Each f_i is a permutation defined as $f_i : \{0,1\}^l \longrightarrow \{0,1\}^l, \forall i = 1, \ldots, m$.
- Let $x \in \{0,1\}^{lm}$, then we can write x as

  $x = (x_{<1>}, x_{<2>}, \ldots, x_{<m>})$, where $x_{<i>} = x_{(i-1)l+1} x_{(i-1)l+2} \cdots x_{(i-1)l+l}$.

- $\hat{\mathcal{E}}_k(x) = f_1(x_{<1>}) \| f_2(x_{<2>}) \| \cdots f_m(x_{<m>})$.

Confusion

- Normally designed as S-box.
    - S stands for substitution.

Confusion

- Normally designed as S-box.
    - S stands for substitution.
- Implemented as look-up table.

Block Cipher Design Paradigm

Confusion

- Normally designed as S-box.
    - S stands for substitution.
- Implemented as look-up table.
- Non-linear component of the design.

Confusion

- Normally designed as S-box.
  - S stands for substitution.
- Implemented as look-up table.
- Non-linear component of the design.
- By linearity, we imply

$$S(x \oplus y) = S(x) \oplus S(y), \forall x, y.$$

Confusion

- Notice that $\hat{\mathcal{E}}$ is not Pseudorandom.

Block Cipher Design Paradigm

Confusion

- Notice that $\hat{\mathcal{E}}$ is not Pseudorandom.
- Let x and x' differs only in first l bits.

Confusion

- Notice that $\hat{\mathcal{E}}$ is not Pseudorandom.
- Let x and x' differs only in first l bits.
    - $\hat{\mathcal{E}}_k(x)$ and $\hat{\mathcal{E}}_k(x')$ will only differ in first l bits.

Confusion

- Notice that $\hat{\mathcal{E}}$ is not Pseudorandom.
- Let x and x' differs only in first l bits.
  - $\hat{\mathcal{E}}_k(x)$ and $\hat{\mathcal{E}}_k(x')$ will only differ in first l bits.
  - If $\hat{\mathcal{E}}$ is truly random, it is expected to that the change in one bit of input will affect all the output bits.

Diffusion

Diffusion is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.

Diffusion

Diffusion is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.

- A simple diffusion element is the bit or mixing permutation.

Diffusion

Diffusion is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.

- A simple diffusion element is the bit or mixing permutation.
- Independent of round key.

Diffusion

Diffusion is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.

- A simple diffusion element is the bit or mixing permutation.
- Independent of round key.
- The output bits of any given S-box are spread into different S-boxes in the next round.

Diffusion

Diffusion is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext.

- A simple diffusion element is the bit or mixing permutation.
- Independent of round key.
- The output bits of any given S-box are spread into different S-boxes in the next round.
- Goal is to achieve avalanche effect.

Does iteration give a secure block cipher?

Iteration

- Nobody knows.

Does iteration give a secure block cipher?

Iteration

- Nobody knows.
- Heuristic evidence suggests that security of a block cipher comes from iterating Confusion-Diffusion many times.

Does iteration give a secure block cipher?

Iteration

- Nobody knows.
- Heuristic evidence suggests that security of a block cipher comes from iterating Confusion-Diffusion many times.
- Ensures that any small change in the input will be mixed throughout and propagated to all the bits of the output.

Does iteration give a secure block cipher?

Iteration

- Nobody knows.
- Heuristic evidence suggests that security of a block cipher comes from iterating Confusion-Diffusion many times.
- Ensures that any small change in the input will be mixed throughout and propagated to all the bits of the output.
- Small changes to the input have a significant effect on the output.

Does iteration give a secure block cipher?

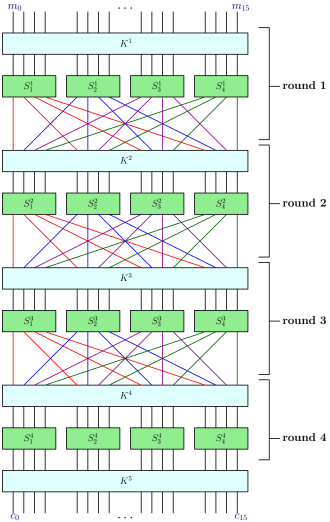
Iteration

- Nobody knows.
- Heuristic evidence suggests that security of a block cipher comes from iterating Confusion-Diffusion many times.
- Ensures that any small change in the input will be mixed throughout and propagated to all the bits of the output.
- Small changes to the input have a significant effect on the output.
- Expected result is a pseudorandom permutation.

SPN

- Introduced by Feistel in 1973.
- Let l and m be two positive integers.
- Block length = lm
- Has three operations per round:
  - Substitution by S-box,
  - Mixing permutation, and
  - Key Mixing.

Substitution-Permutation Network (SPN)



$\ell = 4$, $l = 4$, and $m = 4$

S-Box

$$\pi_S : \{0,1\}^l \longrightarrow \{0,1\}^l.$$

S-Box

$$\pi_S : \{0,1\}^l \longrightarrow \{0,1\}^l.$$

input	0	1	2	3	4	5	6	7
output	E	4	D	1	2	F	B	8

input	8	9	A	B	C	D	E	F
output	3	A	6	C	5	9	0	7

Mixing Permutation

$$\pi_P : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}.$$

Mixing Permutation

$$\pi_P : \{0,1\}^{lm} \longrightarrow \{0,1\}^{lm}.$$

input	1	2	3	4	5	6	7	8
output	1	5	9	13	2	6	10	14

input	9	10	11	12	13	14	15	16
output	3	7	11	15	4	8	12	16

Design Principle 1

- Invertibility of the S- boxes

Design Principle 1

- Invertibility of the S- boxes
  - S-boxes must be invertible, otherwise SPN block cipher will not be a permutation.

Design Principle 1

- Invertibility of the S- boxes
  - S-boxes must be invertible, otherwise SPN block cipher will not be a permutation.
  - One-to-one and onto suffices.

Substitution-Permutation Network (SPN)

Design Principle 1

- Invertibility of the S- boxes
  - S-boxes must be invertible, otherwise SPN block cipher will not be a permutation.
  - One-to-one and onto suffices.

Design Principle 2

- The Avalanche Effect

Design Principle 1

- Invertibility of the S- boxes
  - S-boxes must be invertible, otherwise SPN block cipher will not be a permutation.
  - One-to-one and onto suffices.

Design Principle 2

- The Avalanche Effect
  - The S-boxes are designed so that changing a single bit of the input to an S-box changes at least two bits in the output of the S-box.

Design Principle 1

- Invertibility of the S- boxes
  - S-boxes must be invertible, otherwise SPN block cipher will not be a permutation.
  - One-to-one and onto suffices.

Design Principle 2

- The Avalanche Effect
  - The S-boxes are designed so that changing a single bit of the input to an S-box changes at least two bits in the output of the S-box.
  - The mixing permutations are designed so that the output bits of any given S-box are spread into different S-boxes in the next round.

SPN: example one round

- $x = 0010\ \ 0110\ \ 1011\ \ 0111$
- $K^1 = 0011\ \ 1010\ \ 1001\ \ 0100.$

$$\begin{aligned} w0 &= 0010\ \ 0110\ \ 1011\ \ 0111 \\ K^1 &= 0011\ \ 1010\ \ 1001\ \ 0100 \end{aligned}$$

SPN: example one round

- $x = 0010\ \ 0110\ \ 1011\ \ 0111$
- $K^1 = 0011\ \ 1010\ \ 1001\ \ 0100.$

$$
\begin{array}{rcl}
w0 & = & 0010\ \ 0110\ \ 1011\ \ 0111 \\
K^1 & = & 0011\ \ 1010\ \ 1001\ \ 0100 \\
u^1 & = & 0001\ \ 1100\ \ 0010\ \ 0011
\end{array}
$$

Substitution-Permutation Network (SPN)

SPN: example one round

- $x = 0010\ \ 0110\ \ 1011\ \ 0111$
- $K^1 = 0011\ \ 1010\ \ 1001\ \ 0100.$

$$\begin{array}{rcl} w0 & = & 0010\ \ 0110\ \ 1011\ \ 0111 \\ K^1 & = & 0011\ \ 1010\ \ 1001\ \ 0100 \\ u^1 & = & 0001\ \ 1100\ \ 0010\ \ 0011 \\ v^1 & = & 0100\ \ 0101\ \ 1101\ \ 0001 \end{array}$$

Substitution-Permutation Network (SPN)

SPN: example one round

- $x = 0010\ \ 0110\ \ 1011\ \ 0111$
- $K^1 = 0011\ \ 1010\ \ 1001\ \ 0100.$

$$\begin{array}{rcl} w0 & = & 0010\ \ 0110\ \ 1011\ \ 0111 \\ K^1 & = & 0011\ \ 1010\ \ 1001\ \ 0100 \\ u^1 & = & 0001\ \ 1100\ \ 0010\ \ 0011 \\ v^1 & = & 0100\ \ 0101\ \ 1101\ \ 0001 \\ w^1 & = & 0010\ \ 1110\ \ 0000\ \ 0111 \end{array}$$

Substitution-Permutation Network (SPN)

SPN

- Let the input be $x \in \{0,1\}^{lm}$.
- We can write x as $x = x_{<1>} \| x_{<2>} \| \cdots \| x_{<m>}$, where

$$x_{<i>} = x_{(i-1)l+1} x_{(i-1)l+2} \cdots x_{(i-1)l+l}.$$

1. $w^0 \longleftarrow x$
2. for $r \longleftarrow 1$ to $\ell - 1$ do
3. $\quad u^r \longleftarrow w^{r-1} \oplus K^r$
4. $\quad$ for $i \longleftarrow 1$ to m do
5. $\quad\quad v^r_{<i>} \longleftarrow \pi_S \left(u^r_{<i>} \right)$
6. $\quad v^r := v^r_{<1>} \| v^r_{<2>} \| \cdots \| v^r_{<m>}$
7. $\quad w^r \longleftarrow \pi_P \left(v^r \right)$
8. $u^\ell \longleftarrow w^{\ell-1} \oplus K^\ell$
9. for $i \longleftarrow 1$ to m do
10. $\quad v^\ell_{<i>} \longleftarrow \pi_S \left(u^\ell_{<i>} \right)$
11. $v^\ell := v^\ell_{<1>} \| v^\ell_{<2>} \| \cdots \| v^\ell_{<m>}$
12. $y \longleftarrow v^\ell \oplus K^{\ell+1}$
12. Return y

DES

- In 1972, US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST), initiated a request for proposals for a standardized cipher in the USA.

DES

- In 1972, US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST), initiated a request for proposals for a standardized cipher in the USA.
- The NBS received the proposal of DES in 1974 from a team of cryptographers working at IBM.

DES

- In 1972, US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST), initiated a request for proposals for a standardized cipher in the USA.
- The NBS received the proposal of DES in 1974 from a team of cryptographers working at IBM.
    - Submitted algorithm is based on the cipher Lucifer of 128 data-block with 128-bit key.

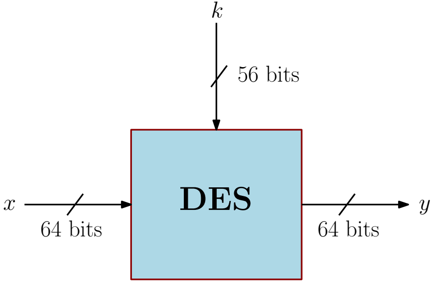
DES

- In 1972, US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST), initiated a request for proposals for a standardized cipher in the USA.
- The NBS received the proposal of DES in 1974 from a team of cryptographers working at IBM.
  - Submitted algorithm is based on the cipher Lucifer of 128 data-block with 128-bit key.
  - Lucifer was a family of ciphers developed by Horst Feistel in the late 1960s.

DES

- In 1972, US National Bureau of Standards (NBS), which is now called National Institute of Standards and Technology (NIST), initiated a request for proposals for a standardized cipher in the USA.
- The NBS received the proposal of DES in 1974 from a team of cryptographers working at IBM.
  - Submitted algorithm is based on the cipher Lucifer of 128 data-block with 128-bit key.
  - Lucifer was a family of ciphers developed by Horst Feistel in the late 1960s.
  - DES is a special type of iterated cipher called Feistel Cipher.

Data Encryption Standard (DES)



DES

- Let $f : \{0,1\}^{48} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$ be a keyed-function.

DES

- Let $f : \{0,1\}^{48} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$ be a keyed-function.
- Using f, we construct Feistel Permutation, $\pi : \{0,1\}^{48} \times \{0,1\}^{64} \longrightarrow \{0,1\}^{64}$.

DES

- Let $f : \{0,1\}^{48} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$ be a keyed-function.
- Using f, we construct Feistel Permutation, $\pi : \{0,1\}^{48} \times \{0,1\}^{64} \longrightarrow \{0,1\}^{64}$.

$$
\begin{aligned}
(L_i, R_i) \quad = \quad & \pi_{K_i}(L_{i-1}, R_{i-1}) \\
& L_i = R_{i-1} \\
& R_i = L_{i-1} \oplus f(K_i, R_{i-1})
\end{aligned}
$$

DES

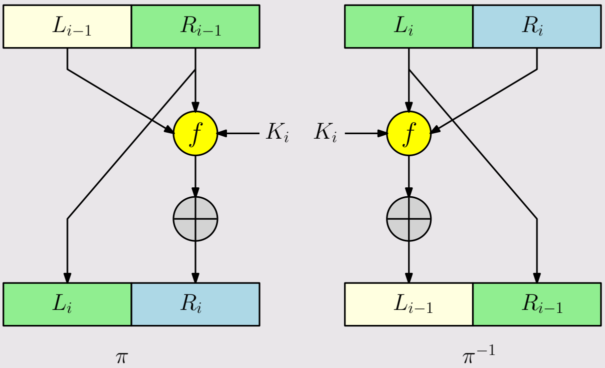
- Let $f : \{0,1\}^{48} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32}$ be a keyed-function.
- Using f, we construct Feistel Permutation, $\pi : \{0,1\}^{48} \times \{0,1\}^{64} \longrightarrow \{0,1\}^{64}$.

$$\begin{aligned}(L_i, R_i) \quad = \quad & \pi_{K_i}(L_{i-1}, R_{i-1}) \\ & L_i = R_{i-1} \\ & R_i = L_{i-1} \oplus f(K_i, R_{i-1})\end{aligned}$$

$$\begin{aligned}(L_{i-1}, R_{i-1}) \quad = \quad & \pi^{-1}_{K_i}(L_i, R_i) \\ & R_{i-1} = L_i \\ & L_{i-1} = R_i \oplus f(K_i, L_i)\end{aligned}$$

Data Encryption Standard (DES)

Feistel Permutation



Data Encryption Standard (DES)

1. Apply initial permutation as $(L^0, R^0) = \mathbf{IP}(x)$.

Data Encryption Standard (DES)

1. Apply initial permutation as $(L^0, R^0) = \mathbf{IP}(x)$.
2. Apply Feistel permutation for 16 rounds.

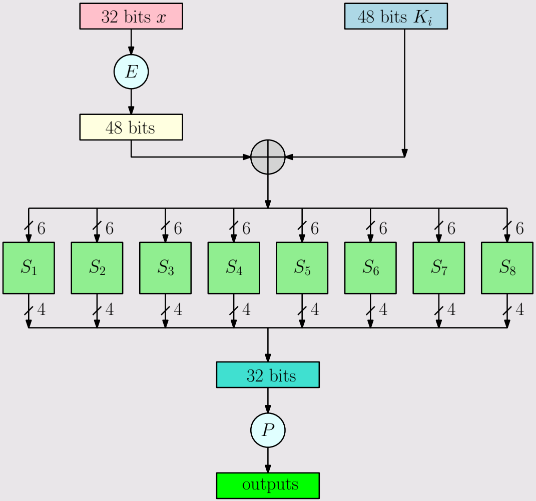
Data Encryption Standard (DES)

1. Apply initial permutation as $(L^0, R^0) = \mathbf{IP}(x)$.
2. Apply Feistel permutation for 16 rounds.
3. Let the output after 16 rounds be (L^{16}, R^{16}).

Data Encryption Standard (DES)

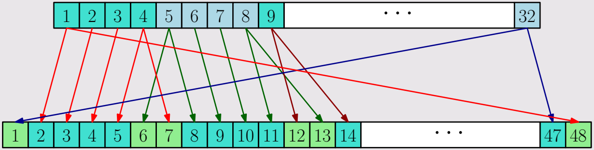
1. Apply initial permutation as $(L^0, R^0) = \mathbf{IP}(x)$.
2. Apply Feistel permutation for 16 rounds.
3. Let the output after 16 rounds be (L^{16}, R^{16}).
4. Apply initial permutation inverse as $y = \mathbf{IP}^{-1}(R^{16} \| L^{16})$.

Structure of f



Data Encryption Standard (DES)

Expansion function E



S-Box

- DES uses 8 different S-boxes.

S-Box

- DES uses 8 different S-boxes.
- Let $b = (b_5 b_4 b_3 b_2 b_1 b_0)_2$ be the input to a S-box.

S-Box

- DES uses 8 different S-boxes.
- Let $b = (b_5 b_4 b_3 b_2 b_1 b_0)_2$ be the input to a S-box.
  - $b_r = (b_5 b_0)_2$ and $b_c = (b_4 b_3 b_2 b_1)_2$.

S-Box

- DES uses 8 different S-boxes.
- Let $b = (b_5 b_4 b_3 b_2 b_1 b_0)_2$ be the input to a S-box.
  - $b_r = (b_5 b_0)_2$ and $b_c = (b_4 b_3 b_2 b_1)_2$.
  - Output is the entry of the b_r-th row and b_c-th column.

Data Encryption Standard (DES)

S-Box

- DES uses 8 different S-boxes.
- Let $b = (b_5 b_4 b_3 b_2 b_1 b_0)_2$ be the input to a S-box.
  - $b_r = (b_5 b_0)_2$ and $b_c = (b_4 b_3 b_2 b_1)_2$.
  - Output is the entry of the b_r-th row and b_c-th column.
  - For S-box S_7, if $b = 110010$, then output is 1111.

16 Column

4 Rows

S_i	0	$\cdots$	8	9	10	$\cdots$	15
0							
1							
2				15			
3							

Exhaustive search on DES

- The adversary is given a small number of plaintext-ciphertext pairs $(x_i, y_i) \in \mathcal{X}^2, 1 \leqslant i \leqslant Q$ using a block cipher key $k \in \mathcal{K}$.

Data Encryption Standard (DES)

Exhaustive search on DES

- The adversary is given a small number of plaintext-ciphertext pairs $(x_i, y_i) \in \mathcal{X}^2, 1 \leqslant i \leqslant Q$ using a block cipher key $k \in \mathcal{K}$.
- The adversary finds k by trying all possible keys $k \in \mathcal{K}$ until it finds a key that maps all the given plaintext blocks to the given ciphertext blocks.

Exhaustive search on DES

- The adversary is given a small number of plaintext-ciphertext pairs $(x_i, y_i) \in \mathcal{X}^2, 1 \leqslant i \leqslant Q$ using a block cipher key $k \in \mathcal{K}$.
- The adversary finds k by trying all possible keys $k \in \mathcal{K}$ until it finds a key that maps all the given plaintext blocks to the given ciphertext blocks.
- For block ciphers like DES and AES-128 three blocks are enough to ensure that with high probability there is a unique key mapping the given plaintext blocks to the given ciphertext blocks.

DES challenges

The DES challenges were set up by RSA data security.

- Rules:
  - n DES outputs $y_1, y_2, \ldots, y_n$ where the first three outputs, y_1, y_2, y_3, were the result of applying DES to the 24-byte plaintext message: (x_1, x_2, x_3)=The unknown message is:
  - The first group to find the corresponding key wins ten thousand US dollars.

DES challenges

- Challenge 1 was posted on January 1997.
  - Was solved in 96 days by DESCHALL project by distributed Internet search.

DES challenges

- Challenge 1 was posted on January 1997.
  - Was solved in 96 days by DESCHALL project by distributed Internet search.
- Challenge 2 was posted on January 1998.
  - Was solved in 41 days by distributed.net project by distributed Internet search on a more larger scale.

DES challenges

- Challenge 1 was posted on January 1997.
  - Was solved in 96 days by DESCHALL project by distributed Internet search.
- Challenge 2 was posted on January 1998.
  - Was solved in 41 days by distributed.net project by distributed Internet search on a more larger scale.
- Challenge 3 was posted on July 1998.
  - Was solved in 56 hours by DeepCrack machine created by Paul Kocher for Electronic Frontiers Foundation (EFF).

Data Encryption Standard (DES)

DES challenges

- Challenge 1 was posted on January 1997.
  - Was solved in 96 days by DESCHALL project by distributed Internet search.
- Challenge 2 was posted on January 1998.
  - Was solved in 41 days by distributed.net project by distributed Internet search on a more larger scale.
- Challenge 3 was posted on July 1998.
  - Was solved in 56 hours by DeepCrack machine created by Paul Kocher for Electronic Frontiers Foundation (EFF).
- Challenge 4 (last) was posted on January 1999.
  - Was solved in 22 hours by DeepCrack and distributed.net.

Triple DES

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.

Triple DES

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.
- $3\mathfrak{E} = (\mathcal{E}_3, \mathcal{D}_3)$ is defined over $(\mathcal{K}^3, \mathcal{X})$ as

$$\mathcal{E}_3((k_1, k_2, k_3), m) := \mathcal{E}(k_3, \mathcal{E}(k_2, \mathcal{E}(k_1, m))).$$

Triple DES

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.
- $3\mathfrak{E} = (\mathcal{E}_3, \mathcal{D}_3)$ is defined over $(\mathcal{K}^3, \mathcal{X})$ as

$$\mathcal{E}_3((k_1, k_2, k_3), m) := \mathcal{E}(k_3, \mathcal{E}(k_2, \mathcal{E}(k_1, m))).$$

- $3\mathfrak{E}$ designed with DES is called Triple DES.

Double DES is insecure

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.

Double DES is insecure

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.
- $2\mathfrak{E} = (\mathcal{E}_2, \mathcal{D}_2)$ is defined over $(\mathcal{K}^2, \mathcal{X})$ as

$$\mathcal{E}_2((k_1, k_2), m) := \mathcal{E}(k_2, \mathcal{E}(k_1, m)).$$

Double DES is insecure

- Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$.
- $2\mathfrak{E} = (\mathcal{E}_2, \mathcal{D}_2)$ is defined over $(\mathcal{K}^2, \mathcal{X})$ as

$$\mathcal{E}_2((k_1, k_2), m) := \mathcal{E}(k_2, \mathcal{E}(k_1, m)).$$

- $2\mathfrak{E}$ designed with DES is called Double DES.

Theorem

Let $\mathfrak{E} = (\mathcal{E}, \mathcal{D})$ be a block cipher defined over $(\mathcal{K}, \mathcal{X})$. There is an algorithm $\mathcal{A}_{EX}$ that takes as input Q plaintext/ciphertext pairs $(x_i, y_i) \in \mathcal{X}$ for $i = 1, \ldots, Q$ and outputs a key pair $(k_1, k_2) \in \mathcal{K}^2$ such that

$$\mathcal{E}_2((k_1, k_2), m) := \mathcal{E}(k_2, \mathcal{E}(k_1, m)), \forall i = 1, \ldots, Q.$$

Its running time is dominated by a total of $2Q \cdot |\mathcal{K}|$ evaluations of algorithms $\mathcal{E}$ and $\mathcal{D}$.

Proof

Let $\hat{x} := (x_1, x_2, \ldots, x_Q)$ and $\hat{y} := (y_1, y_2, \ldots, y_Q)$.

Proof

Let $\hat{x} := (x_1, x_2, \ldots, x_Q)$ and $\hat{y} := (y_1, y_2, \ldots, y_Q)$.

$$\hat{y} = \mathcal{E}_2((k_1, k_2), \hat{x}) = \mathcal{E}(k_2, \mathcal{E}(k_1, \hat{x}))$$

Proof

Let $\hat{x} := (x_1, x_2, \ldots, x_Q)$ and $\hat{y} := (y_1, y_2, \ldots, y_Q)$.

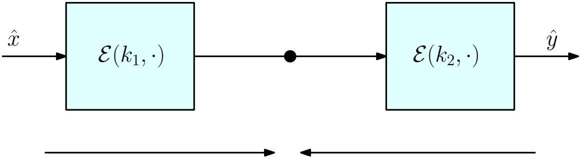
$$\begin{aligned} & \hat{y} = \mathcal{E}_2((k_1, k_2), \hat{x}) = \mathcal{E}(k_2, \mathcal{E}(k_1, \hat{x})) \\ \Leftrightarrow \quad & \mathcal{D}(k_2, \hat{y}) = \mathcal{E}(k_1, \hat{x}). \end{aligned}$$

Data Encryption Standard (DES)

Proof

Let $\hat{x} := (x_1, x_2, \ldots, x_Q)$ and $\hat{y} := (y_1, y_2, \ldots, y_Q)$.

$$\begin{aligned} & \hat{y} = \mathcal{E}_2((k_1, k_2), \hat{x}) = \mathcal{E}(k_2, \mathcal{E}(k_1, \hat{x})) \\ \Leftrightarrow \quad & \mathcal{D}(k_2, \hat{y}) = \mathcal{E}(k_1, \hat{x}). \end{aligned}$$



$$\mathcal{E}(k_1, \hat{x}) = \mathcal{D}(k_2, \hat{y})$$

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x}) \forall k_1 \in \mathcal{K}$

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x}) \forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x}) \forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:
    2.1 $\hat{z} \longleftarrow \mathcal{D}(k_2, k_2)$.

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x})\forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:
   2.1 $\hat{z} \longleftarrow \mathcal{D}(k_2, k_2)$.
   2.2 Table Lookup: if T contains a pair $(\cdot, \hat{x})$ then let $(k_1, \hat{z})$ be that pair, output (k_1, k_2) and halt.

Data Encryption Standard (DES)

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x})\forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:
   2.1 $\hat{z} \longleftarrow \mathcal{D}(k_2, k_2)$.
   2.2 Table Lookup: if T contains a pair $(\cdot, \hat{x})$ then let $(k_1, \hat{z})$ be that pair, output (k_1, k_2) and halt.

Running time

- Step 1 requires $Q \cdot |\mathcal{K}|$ evaluations of $\mathcal{E}$.

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x})\forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:
   2.1 $\hat{z} \longleftarrow \mathcal{D}(k_2, k_2)$.
   2.2 Table Lookup: if T contains a pair $(\cdot, \hat{x})$ then let $(k_1, \hat{z})$ be that pair, output (k_1, k_2) and halt.

Running time

- Step 1 requires $Q \cdot |\mathcal{K}|$ evaluations of $\mathcal{E}$.
- Step 2 requires $Q \cdot |\mathcal{K}|$ evaluations of $\mathcal{D}$.

Data Encryption Standard (DES)

$\mathcal{A}_{EX}$

1. Construct a table T containing all pairs $(k_1, \mathcal{E}(k_1, \hat{x})\forall k_1 \in \mathcal{K}$
2. For all $k_2 \in \mathcal{K}$ do:

   2.1 $\hat{z} \longleftarrow \mathcal{D}(k_2, k_2)$.

   2.2 Table Lookup: if T contains a pair $(\cdot, \hat{x})$ then let $(k_1, \hat{z})$ be that pair, output (k_1, k_2) and halt.

Running time

- Step 1 requires $Q \cdot |\mathcal{K}|$ evaluations of $\mathcal{E}$.
- Step 2 requires $Q \cdot |\mathcal{K}|$ evaluations of $\mathcal{D}$.
- Assumption: Insertion in to table T and lookup takes negligible time.

Meet in the Middle attack on Triple-DES

- Similar meet in the middle attack applies to the $3\mathfrak{E}$ construction.

Meet in the Middle attack on Triple-DES

- Similar meet in the middle attack applies to the $3\mathfrak{E}$ construction.
- $3\mathfrak{E}$ has key space $\mathcal{K}^3$.

Meet in the Middle attack on Triple-DES

- Similar meet in the middle attack applies to the $3\mathfrak{E}$ construction.
- $3\mathfrak{E}$ has key space $\mathcal{K}^3$.
- The meet in the middle attack takes time about $|\mathcal{K}|^2$ and takes space $|\mathcal{K}|$.

Meet in the Middle attack on Triple-DES

- Similar meet in the middle attack applies to the $3\mathfrak{E}$ construction.
- $3\mathfrak{E}$ has key space $\mathcal{K}^3$.
- The meet in the middle attack takes time about $|\mathcal{K}|^2$ and takes space $|\mathcal{K}|$.
- In the case of Triple-DES,
  - $|\mathcal{K}|^2 = 2^{112}$

Meet in the Middle attack on Triple-DES

- Similar meet in the middle attack applies to the $3\mathfrak{E}$ construction.
- $3\mathfrak{E}$ has key space $\mathcal{K}^3$.
- The meet in the middle attack takes time about $|\mathcal{K}|^2$ and takes space $|\mathcal{K}|$.
- In the case of Triple-DES,
  - $|\mathcal{K}|^2 = 2^{112}$
  - too long to run in practice.

Advanced Encryption Standard (AES)

The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.

Advanced Encryption Standard (AES)

The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.
- In September of 1997, NIST received 15 submissions.

Advanced Encryption Standard (AES)

The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.
- In September of 1997, NIST received 15 submissions.
- After two open conferences, in 1999 NIST narrowed down the list to five candidates.

Advanced Encryption Standard (AES)

The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.
- In September of 1997, NIST received 15 submissions.
- After two open conferences, in 1999 NIST narrowed down the list to five candidates.
- A further round of intense cryptanalysis followed,
- AES3 conference was held in April of 2000.

Advanced Encryption Standard (AES)

The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.
- In September of 1997, NIST received 15 submissions.
- After two open conferences, in 1999 NIST narrowed down the list to five candidates.
- A further round of intense cryptanalysis followed,
- AES3 conference was held in April of 2000.
- In October of 2000, NIST announced that Rijndael, a Belgian block cipher, had been selected as the AES cipher.
- AES became an official standard in November of 2001 as FIPS 197.

Advanced Encryption Standard (AES)

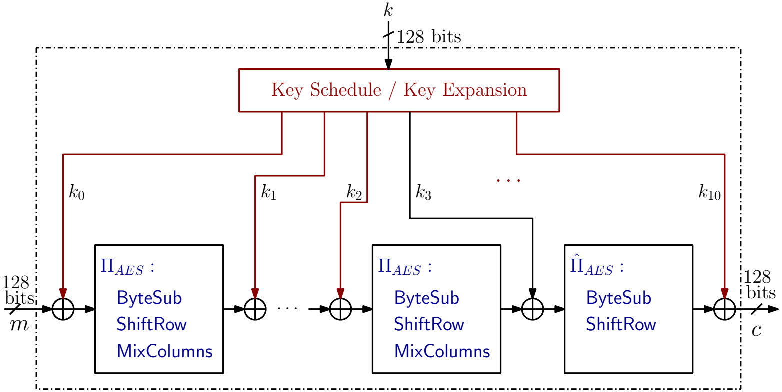
The AES process

- In 1997, NIST put out a request for proposals for a new block cipher standard.
- It is to be called the Advanced Encryption Standard or AES.
- Had to operate on 128-bit blocks and support three key sizes: 128, 192, and 256 bits.
- In September of 1997, NIST received 15 submissions.
- After two open conferences, in 1999 NIST narrowed down the list to five candidates.
- A further round of intense cryptanalysis followed,
- AES3 conference was held in April of 2000.
- In October of 2000, NIST announced that Rijndael, a Belgian block cipher, had been selected as the AES cipher.
- AES became an official standard in November of 2001 as FIPS 197.
- Rijndael was designed by Belgian cryptographers Joan Daemen and Vincent Rijmen.

Advanced Encryption Standard (AES)

Cipher Name	Key-size (bits)	Block Size (bits)	Number of Rounds
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

Advanced Encryption Standard (AES)



AES 128

AES

- Ciphers that follow the structure shown in Figure are called alternating key ciphers.
- They are also known as iterated Even-Mansour ciphers.

AES round permutation

- The permutation Π_{AES} is made up of a sequence of three invertible operations
  - SubBytes
  - ShiftRows
  - MixColumns

AES round Input

- The 128 bits are organized as a blue 4×4 array of cells, where each cell is made up of eight bits.

$$m = m_0 \| m_1 \| m_2 \| m_3 \| m_4 \| m_5 \| m_6 \| m_7 \| m_8 \| m_9 \| m_{10} \| m_{11} \| m_{12} \| m_{13} \| m_{14} \| m_{15},$$

where each m_i = 8-bit

AES round Input

- The 128 bits are organized as a blue 4×4 array of cells, where each cell is made up of eight bits.

$$m = m_0 \| m_1 \| m_2 \| m_3 \| m_4 \| m_5 \| m_6 \| m_7 \| m_8 \| m_9 \| m_{10} \| m_{11} \| m_{12} \| m_{13} \| m_{14} \| m_{15},$$

where each m_i = 8-bit

$$m = \begin{pmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{pmatrix}$$

AES round operation: SubBytes

- Let $S : \{0,1\}^8 \longrightarrow \{0,1\}^8$ be a fixed permutation (a one-to-one function).
- Applied to each of the 16 cells, one cell at a time.
- The permutation S is specified in the AES standard as a hard-coded table of 256 entries.
- It is designed to have
  - No fixed points, namely $S(x) \neq x$ for all $x \in \{0,1\}^8$.
  - No inverse fixed points, namely $S(x) \neq \bar{x}$ where $\bar{x}$ is the bit-wise complement of x.

AES round operation: SubBytes

- Let $S : \{0,1\}^8 \longrightarrow \{0,1\}^8$ be a fixed permutation (a one-to-one function).
- Applied to each of the 16 cells, one cell at a time.
- The permutation S is specified in the AES standard as a hard-coded table of 256 entries.
- It is designed to have
  - No fixed points, namely $S(x) \neq x$ for all $x \in \{0,1\}^8$.
  - No inverse fixed points, namely $S(x) \neq \bar{x}$ where $\bar{x}$ is the bit-wise complement of x.

$$\begin{pmatrix} S(m_0) & S(m_1) & S(m_2) & S(m_3) \\ S(m_4) & S(m_5) & S(m_6) & S(m_7) \\ S(m_8) & S(m_9) & S(m_{10}) & S(m_{11}) \\ S(m_{12}) & S(m_{13}) & S(m_{14}) & S(m_{15}) \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix}$$

AES round operation: ShiftRows

- The First row is cyclically shifted zero byte to the left,
- The Second row is cyclically shifted one byte to the left,
- The Third row is cyclically shifted two bytes to the left,
- The Fourth row is cyclically shifted three bytes to the left,

AES round operation: ShiftRows

- The First row is cyclically shifted zero byte to the left,
- The Second row is cyclically shifted one byte to the left,
- The Third row is cyclically shifted two bytes to the left,
- The Fourth row is cyclically shifted three bytes to the left,

$$\begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix} \longrightarrow \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 & a_4 \\ a_{10} & a_{11} & a_8 & a_9 \\ a_{15} & a_{12} & a_{13} & a_{14} \end{pmatrix}$$

AES round operation: MixColumns

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix} \longrightarrow \begin{pmatrix} a'_0 & a'_1 & a'_2 & a'_3 \\ a'_5 & a'_6 & a'_7 & a'_4 \\ a'_{10} & a'_{11} & a'_8 & a'_9 \\ a'_{15} & a'_{12} & a'_{13} & a'_{14} \end{pmatrix}$$

AES round operation: MixColumns

- Multiplications are done over the field $GF(2^8)$.
- Irreducible Polynomial: $x^8 + x^4 + x^3 + x + 1$.

End