Lecture 2 – Block ciphers, PRFs/PRPs, DES, AES

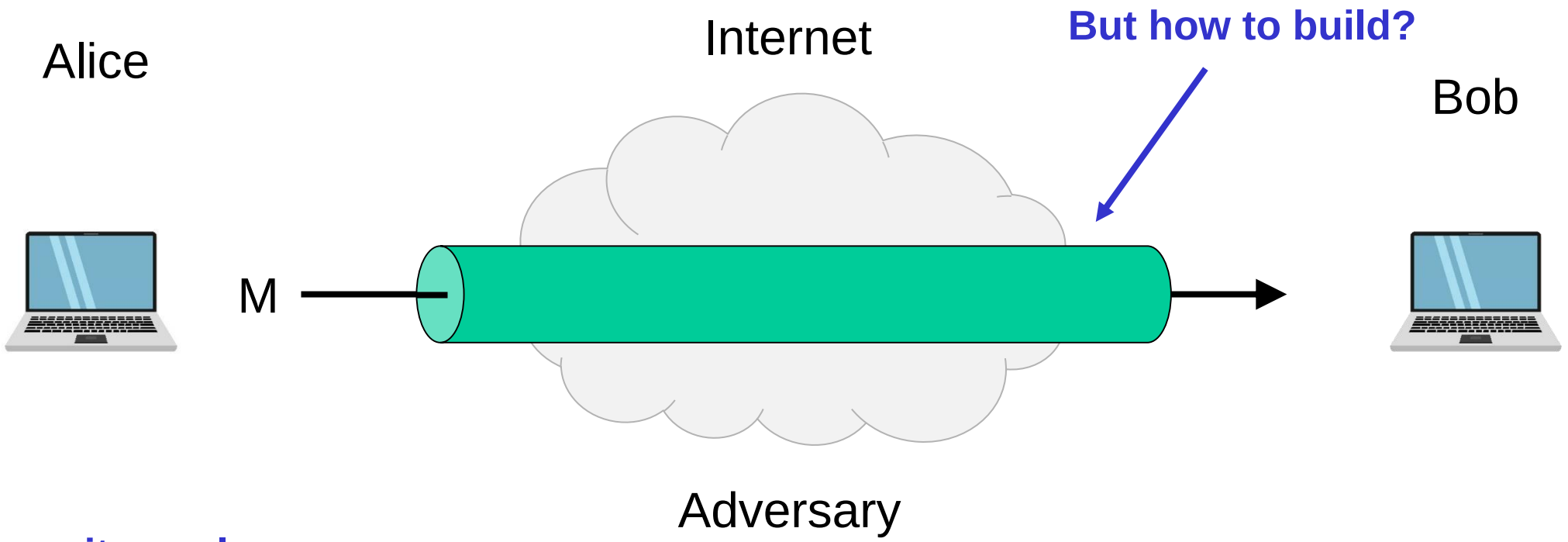
TEK4500

01.09.2021

Håkon Jacobsen

hakon.jacobsen@its.uio.no

Ideal solution: secure channels



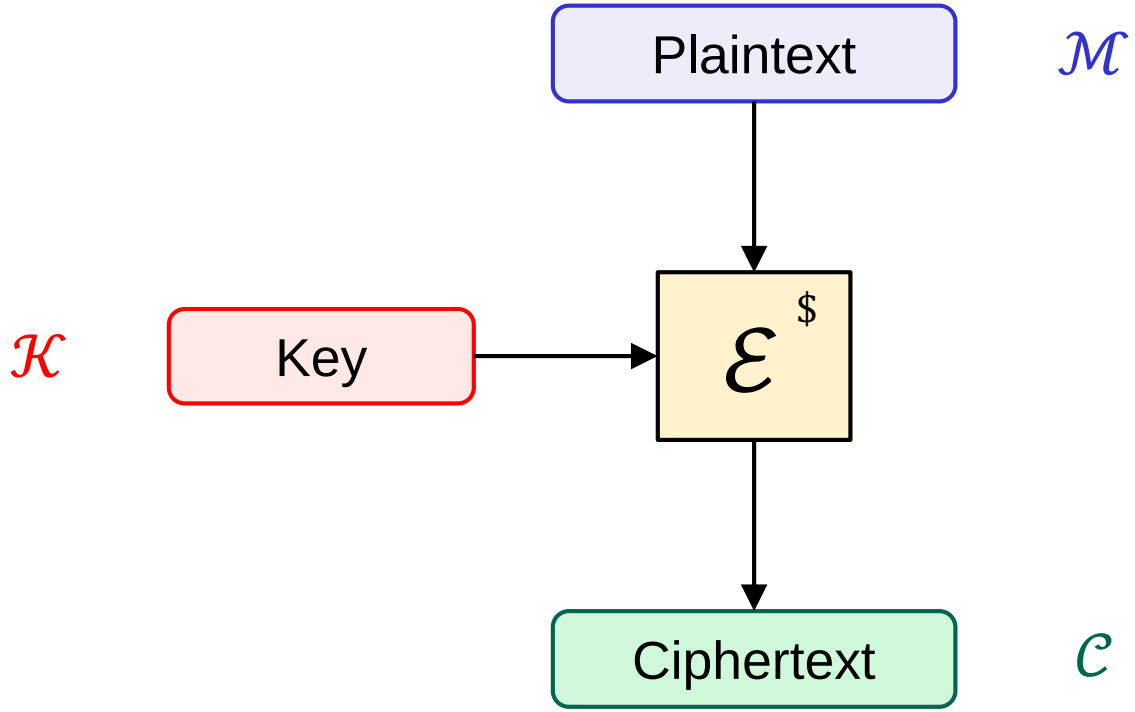
Security goals:

- **Data privacy:** adversary should not be able to read message M ✓
- **Data integrity:** adversary should not be able to modify message M ✓
- **Data authenticity:** message M really originated from Alice ✓

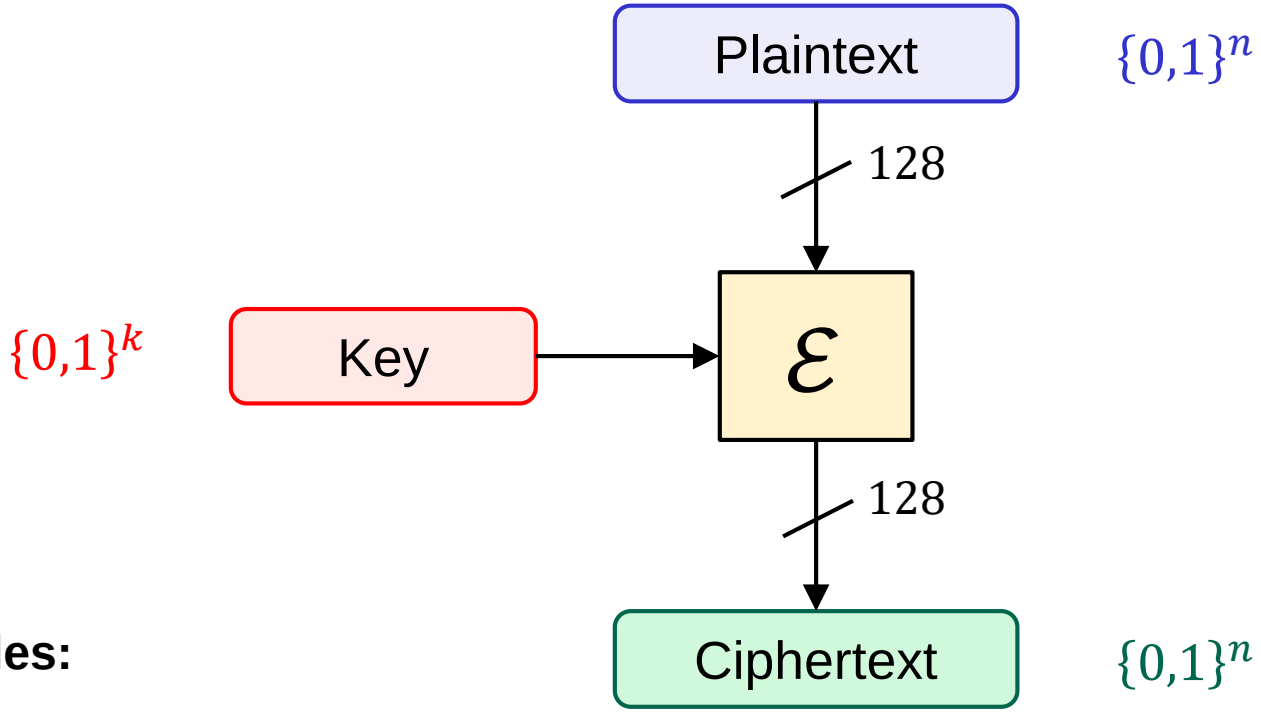
Basic goals of cryptography

	Message privacy	**Message integrity / authentication**
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption)	Digital signatures

Encryption schemes



Block ciphers

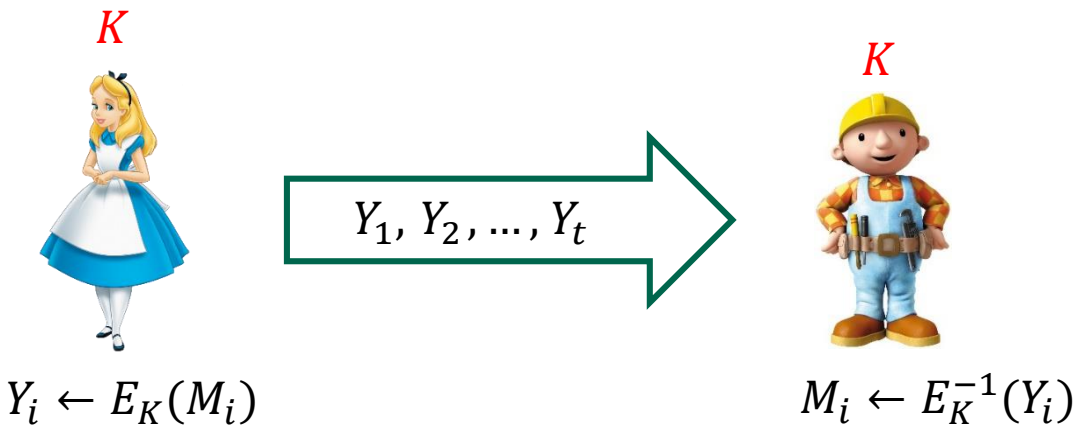


Examples:

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

Block cipher applications (1)

- Encryption of messages of 128 bits (block length)



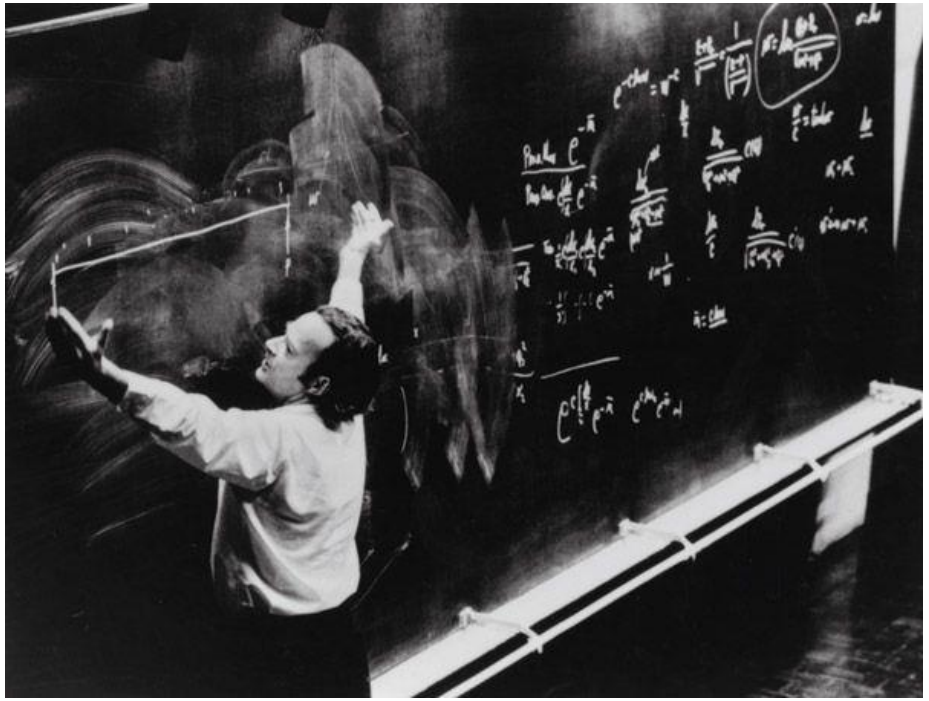
- **However:** we usually want to encrypt messages of *arbitrary* length!
  - Splitting the message into multiple 128 bit blocks (like above) is **not secure!**
  - Need to use them in a proper **mode-of-operation** (covered later in the course)

- Correct viewpoint: block ciphers are **not** encryption schemes!
  - Block ciphers are **primitives** used to construct other things

Block cipher applications (2)

- The “work horse” of crypto

- Can be used to build:
  - Encryption of arbitrary length messages (including stream ciphers)
  - Message authentication codes
  - Authenticated encryption
  - Hash functions
  - (Cryptographically secure) pseudorandom generators
  - Key derivation functions

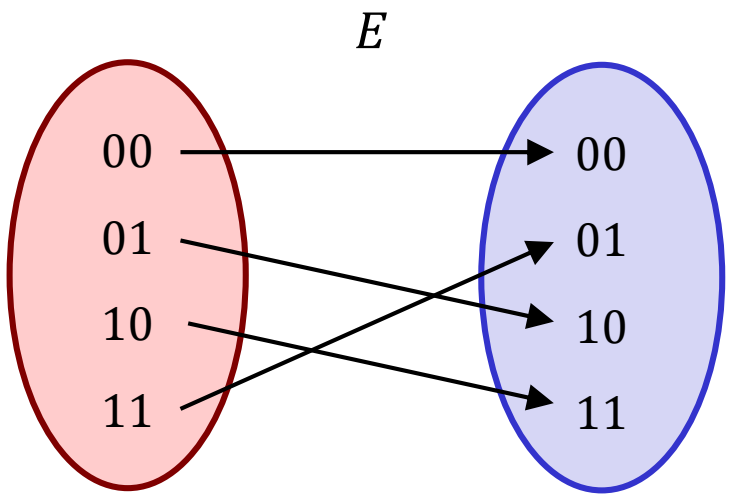


Defining block ciphers

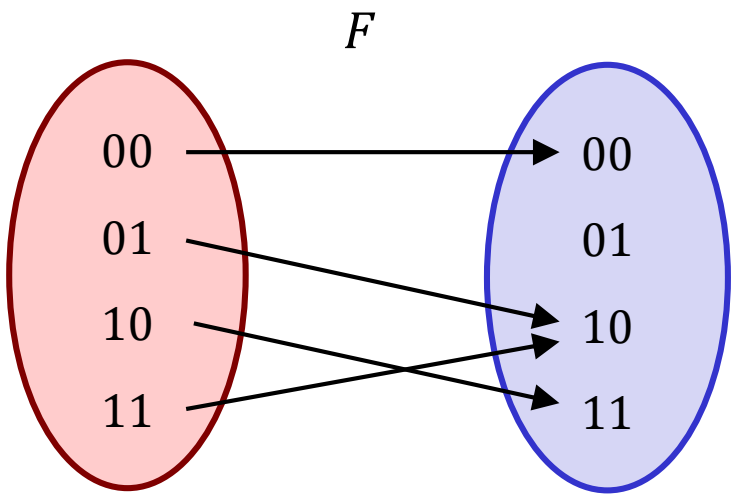
Permutations

Definition: A function $E : \{0,1\}^n \rightarrow \{0,1\}^n$ is a **permutation** if there exists an inverse function E^{-1}:

$$E^{-1}\big(E(X)\big) = X$$



Permutation

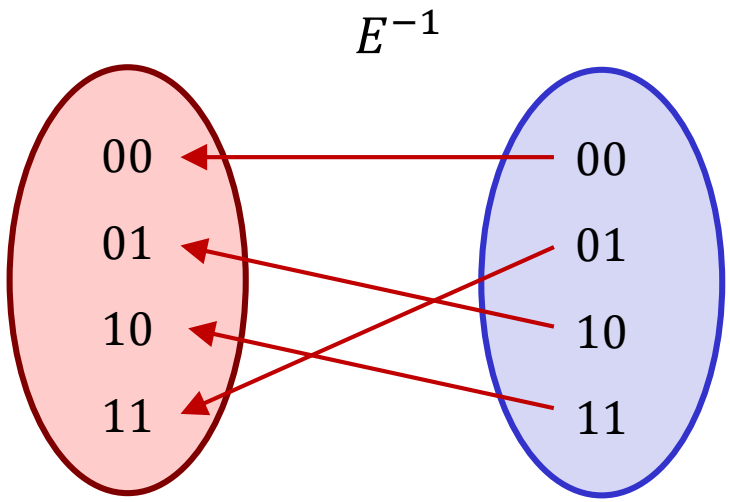


Not a permutation

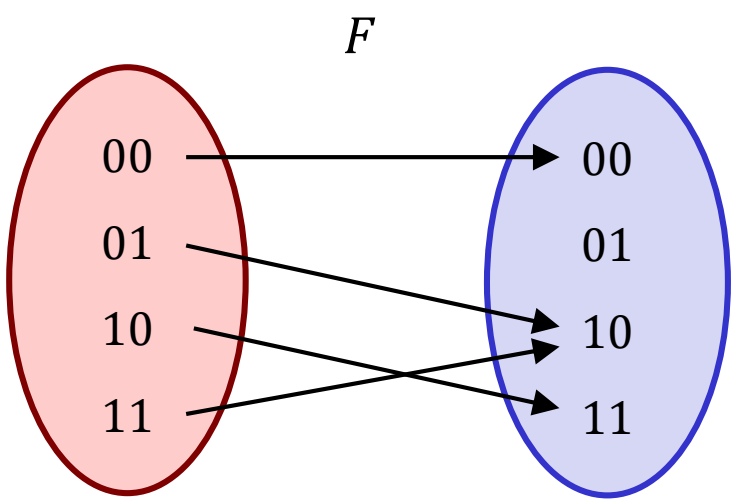
Permutations

Definition: A function $E : \{0,1\}^n \rightarrow \{0,1\}^n$ is a **permutation** if there exists an inverse function E^{-1}:

$$E^{-1}\big(E(X)\big) = X$$



Permutation



Not a permutation

Pseudorandom functions (PRFs) and permutations (PRP)

Definition: A **pseudorandom function (PRF)** is a function

$$F : \{0,1\}^k \times \{0,1\}^{in} \rightarrow \{0,1\}^{out}$$

- k, in, out are called the **key-length**, **input-length**, and **output-length** of F
- Think of a PRF as a *family* of functions:
  - For each $K \in \{0,1\}^k$ we get a function $F_K : \{0,1\}^{in} \rightarrow \{0,1\}^{out}$ de

PRP = block cipher

note: all PRPs are PRFs
(but not all PRFs are PRPs!)

Definition: A **pseudorandom permutation (PRP)** is a function

$$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

such that $E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ is a *permutation* for all $K \in \{0,1\}^k$, where $E_K(X) \stackrel{\text{def}}{=} E(K, X)$

Block cipher security

- Which security properties should a block cipher satisfy?
  - I.e., what should the **security definition** of a block cipher look like?

- Some suggestions:

  - **P1:** Should be hard to obtain K from $E_K(X)$ for secret K
  - **P2:** Should be hard to obtain K from $E_K(X_1), E_K(X_2), E_K(X_3)$ ...
  - **P3:** Should be hard to obtain X from $E_K(X)$
  - **P4:** Should be hard to obtain *any* X_i from $E_K(X_1), E_K(X_2), E_K(X_3)$ ...
  - **P5:** Should be hard to learn any *bit* of X from $E_K(X)$

Not good enough!

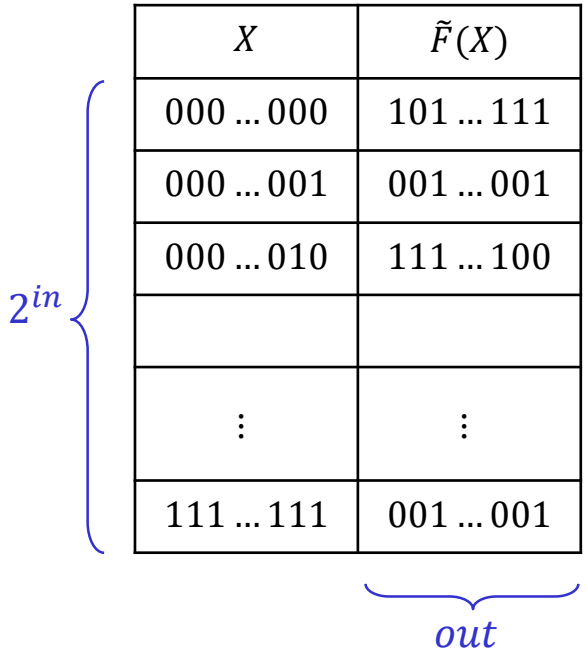
  - ~~**P6:** Should be hard to detect *repetitions* among X_1, X_2, ... from $E_K(X_1), E_K(X_2)$, ...~~

Impossible!

  - **P7:** …

Random functions

$\widetilde{F}: \{0,1\}^{in} \rightarrow \{0,1\}^{out}$



Random functions

$\widetilde{F}: \{0,1\}^{in} \rightarrow \{0,1\}^{out}$

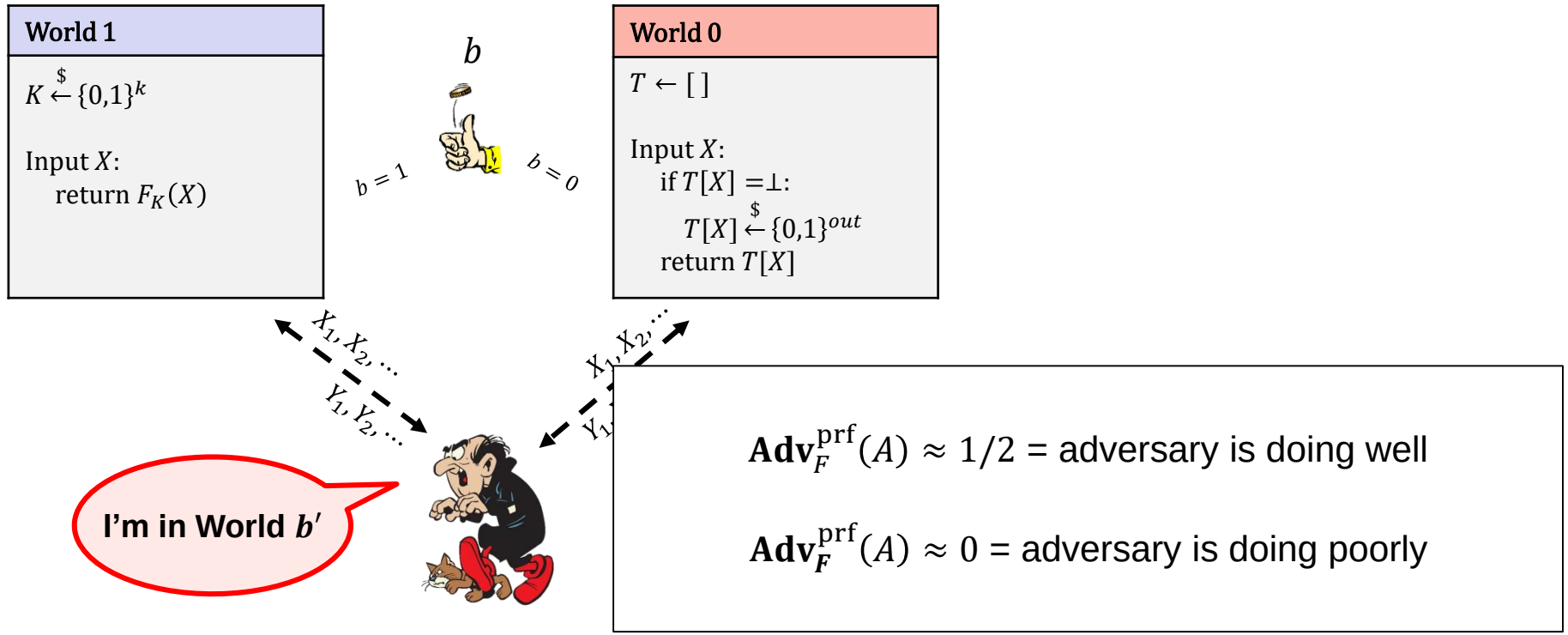
X	$\tilde{F}(X)$
⋮	⋮

1. $T \leftarrow [\,]$

$\underline{\tilde{F}(X):}$

1. **if** $T[X] = \perp$**:**
2. $\quad T[X] \xleftarrow{\$} \{0,1\}^{out}$
3. **return** $T[X]$

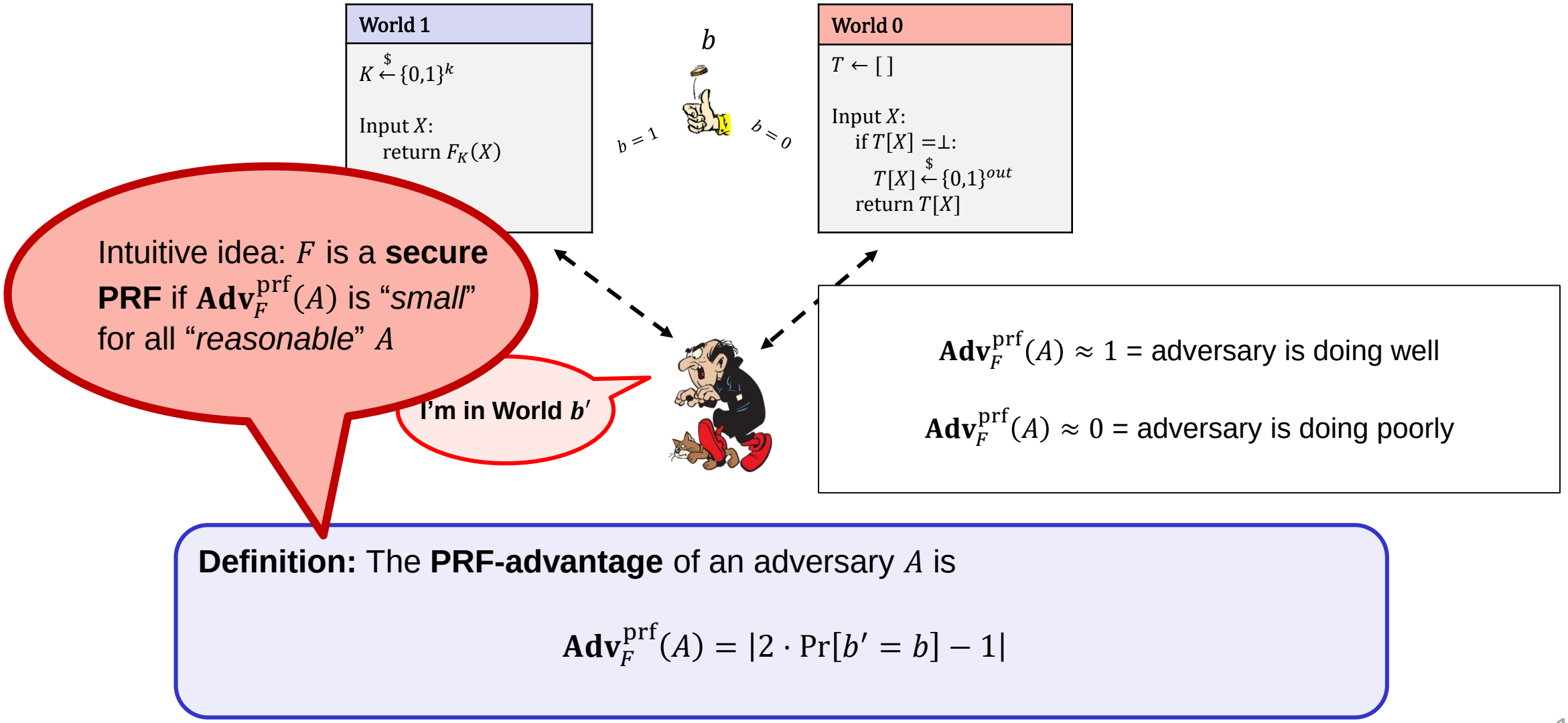
PRF – security; formal definition



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |\Pr[b' = b] - 1/2|$$

PRF – security; formal definition



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

Understanding "advantage"

- F is a **secure PRF** if $\mathbf{Adv}_F^{\mathrm{prf}}(A)$ is "*small*" for *all* adversaries A that use a "*reasonable*" amount of resources

- Advantage depends on the adversary's:
  - strategy
  - available resources: running time, number of oracle calls (calls to $\tilde{F}$ / F), memory…

- What does *small* and *reasonable* mean?
  - **Example:** **128-bit** security:

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) \leq \frac{ct}{2^{128}}$$

for all A runs in time $\leq t$ for some constant c

  - **Example:** a PRF is *insecure* if we can come up with an adversary having good advantage and not using too many resources

Example

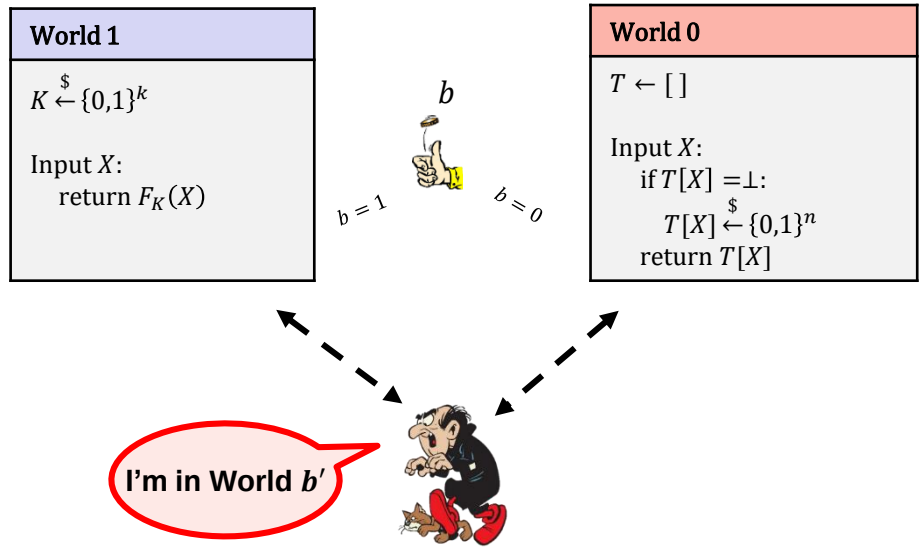
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ // $b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ // $b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \quad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 \; + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \quad 1 \cdot 1/2 \qquad\qquad\qquad\qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2
\end{aligned}$$

Example

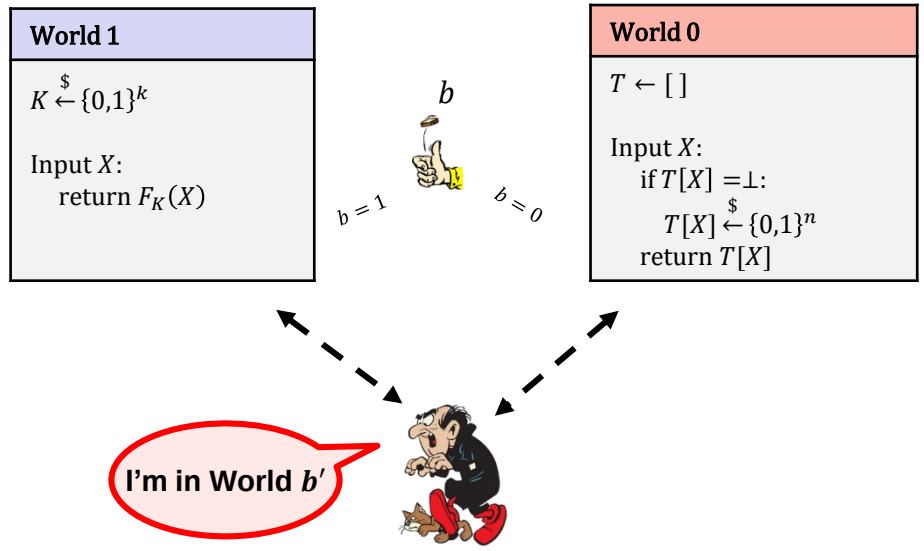
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ $\quad // b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ $\quad // b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= 1 \cdot 1/2 \qquad + (1 - \Pr[\, b' = 1 \mid b = 0\,]) \cdot 1/2
\end{aligned}$$

Example

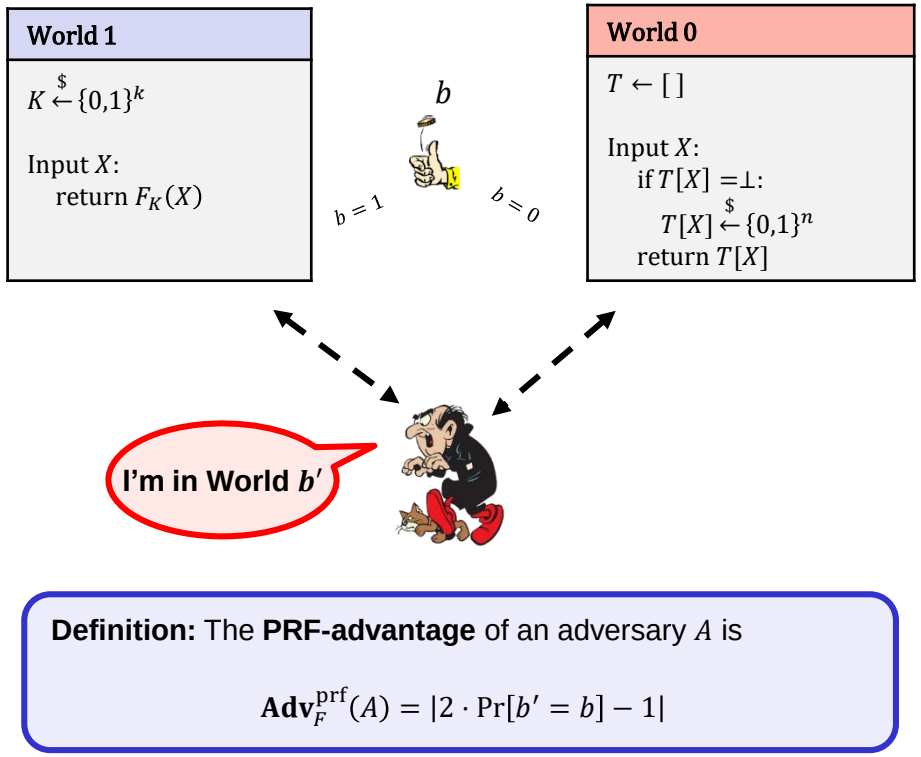
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ // $b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ // $b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \quad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= 1 \cdot 1/2 \quad + (1 - \Pr[\, Y \oplus Y' = X \oplus X' \mid b = 0\,]) \cdot 1/2
\end{aligned}$$

Example

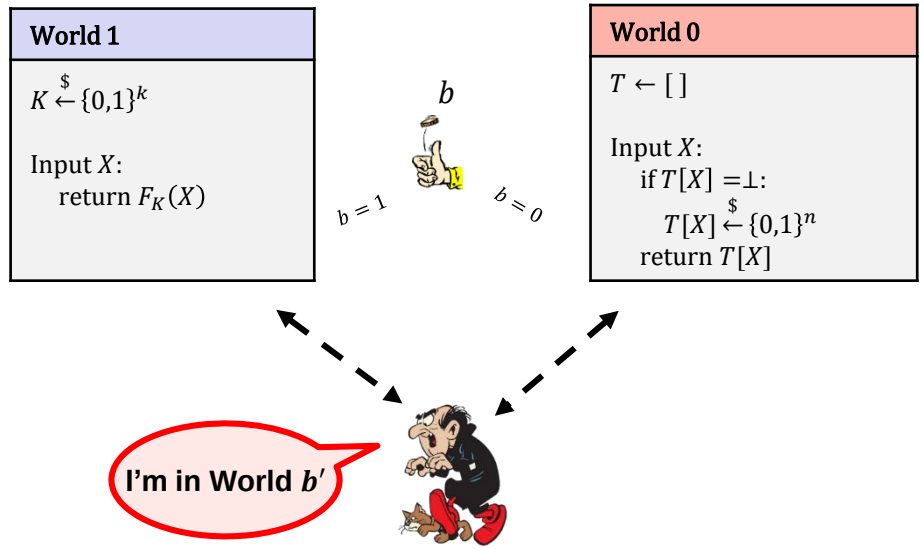
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ $\quad // b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ $\quad // b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= 1 \cdot 1/2 \qquad + (1 - \Pr[\, Y = X \oplus X' \oplus Y' \mid b = 0\,]) \cdot 1/2
\end{aligned}$$

Example

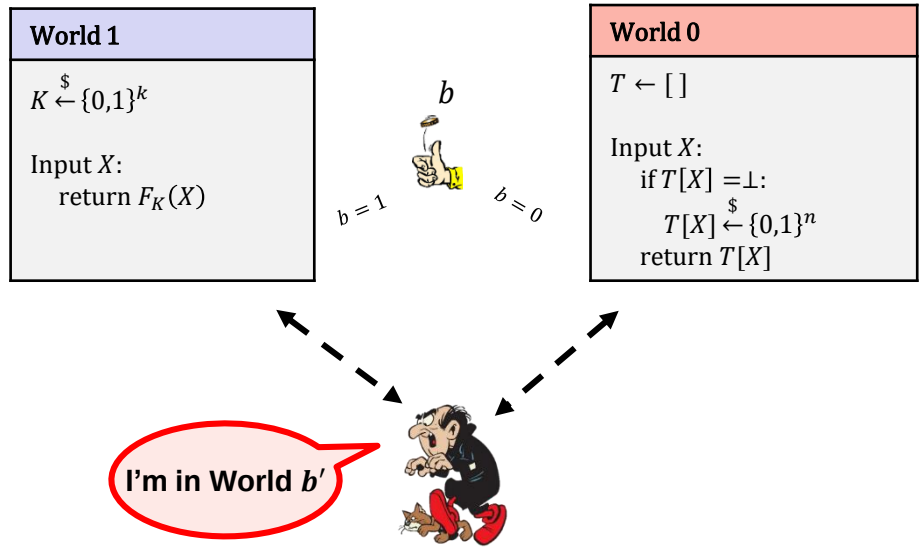
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ $\quad // b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ $\quad // b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= 1 \cdot 1/2 \qquad + (1 - \Pr[\, Y = Z \mid b = 0\,]) \cdot 1/2
\end{aligned}$$

Example

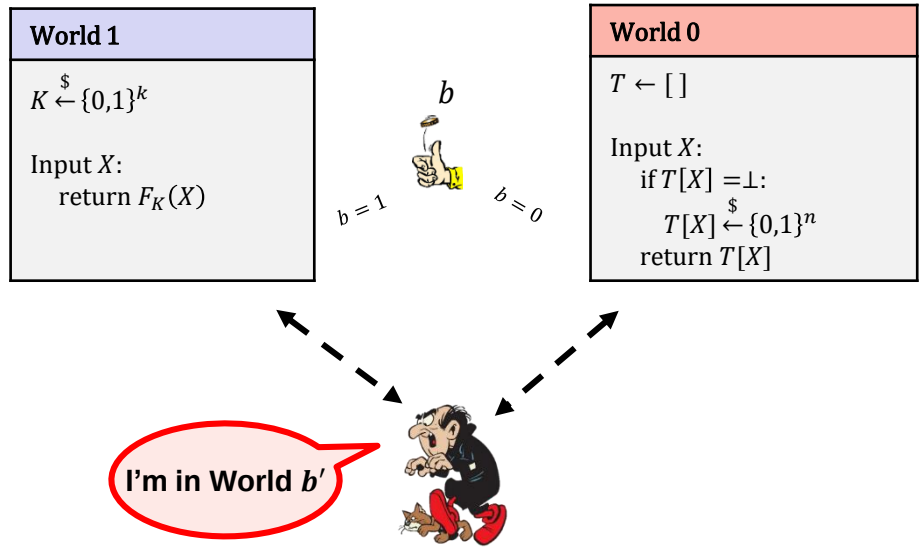
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ // $b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ // $b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 \; + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \quad 1 \cdot 1/2 \qquad\qquad\qquad\qquad + (1 - 2^{-n}) \cdot 1/2
\end{aligned}$$

Example

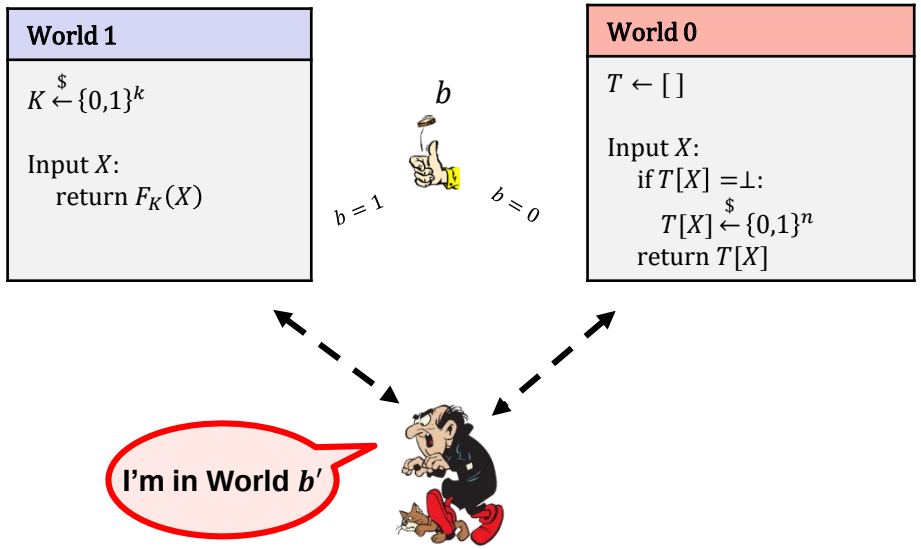
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ // $b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ // $b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] + \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] \\
&= \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \qquad + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \Pr[Y \oplus Y' = X \oplus X' \mid b = 1] \cdot 1/2 \; + \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 \\
&= \; 1 - 2^{-n} \cdot 1/2
\end{aligned}$$

Example

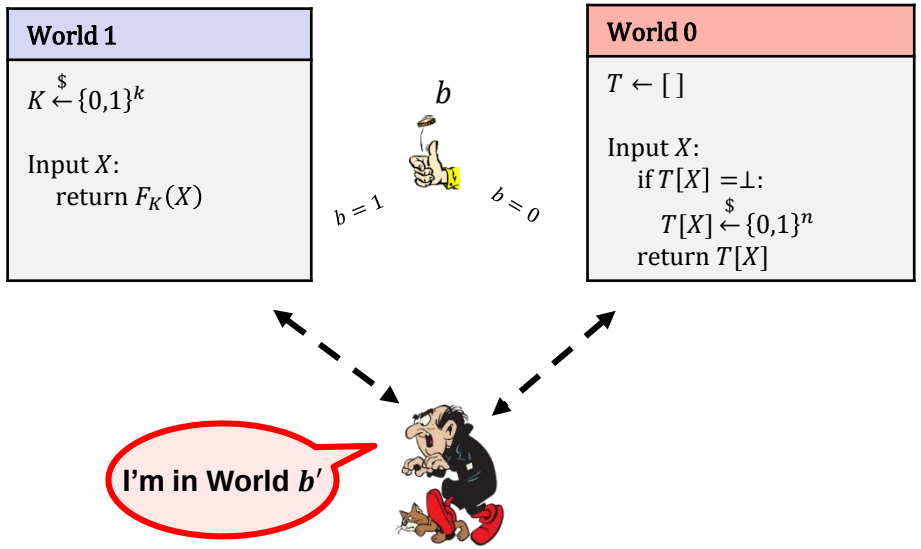
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$
- **Claim:** F is not a secure PRF

A

1. Choose $X \neq X' \in \{0,1\}^n$
2. Query X and X' to challenger
3. Receive back *either* $Y = F_K(X)$ and $Y' = F_K(X')$ // $b = 1$

   or $Y \xleftarrow{\$} \{0,1\}^n$ and $Y' \xleftarrow{\$} \{0,1\}^n$ // $b = 0$
4. **if** $Y \oplus Y' = X \oplus X'$: output $b' = 1$

   else: output $b' = 0$



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

$$\Pr[b' = b] \quad = \quad 1 - 2^{-n} \cdot 1/2$$

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}_F^{\mathrm{prf}}(A) \Rightarrow \mathrm{true}\right] - 1\right| \quad = |2 \cdot (1 - 2^{-n} \cdot 1/2) - 1| = 1 - 2^{-n} \approx 1$$

Why is this definition good?

- **P1:** Should be hard to obtain K from $F_K(X)$ for secret K
- **P2:**
- **P3:**
- **P4:**
- **P5:**
- **P7:**

A

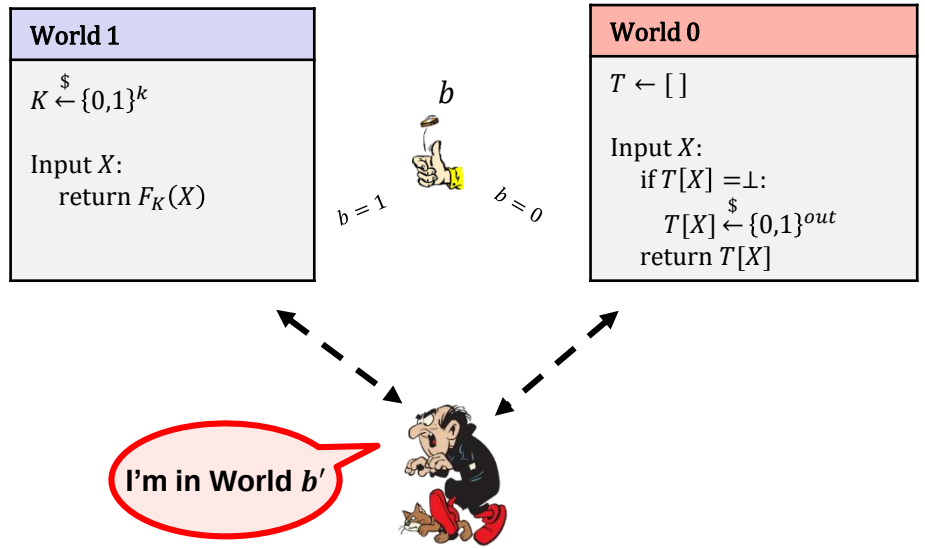
1. Query 0^{in} to challenger

101

$\Rightarrow B$

valent to:

$\Leftarrow$ not B



Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

F is PRF secure $\Rightarrow$ F has properties P1 – P5, P7, …

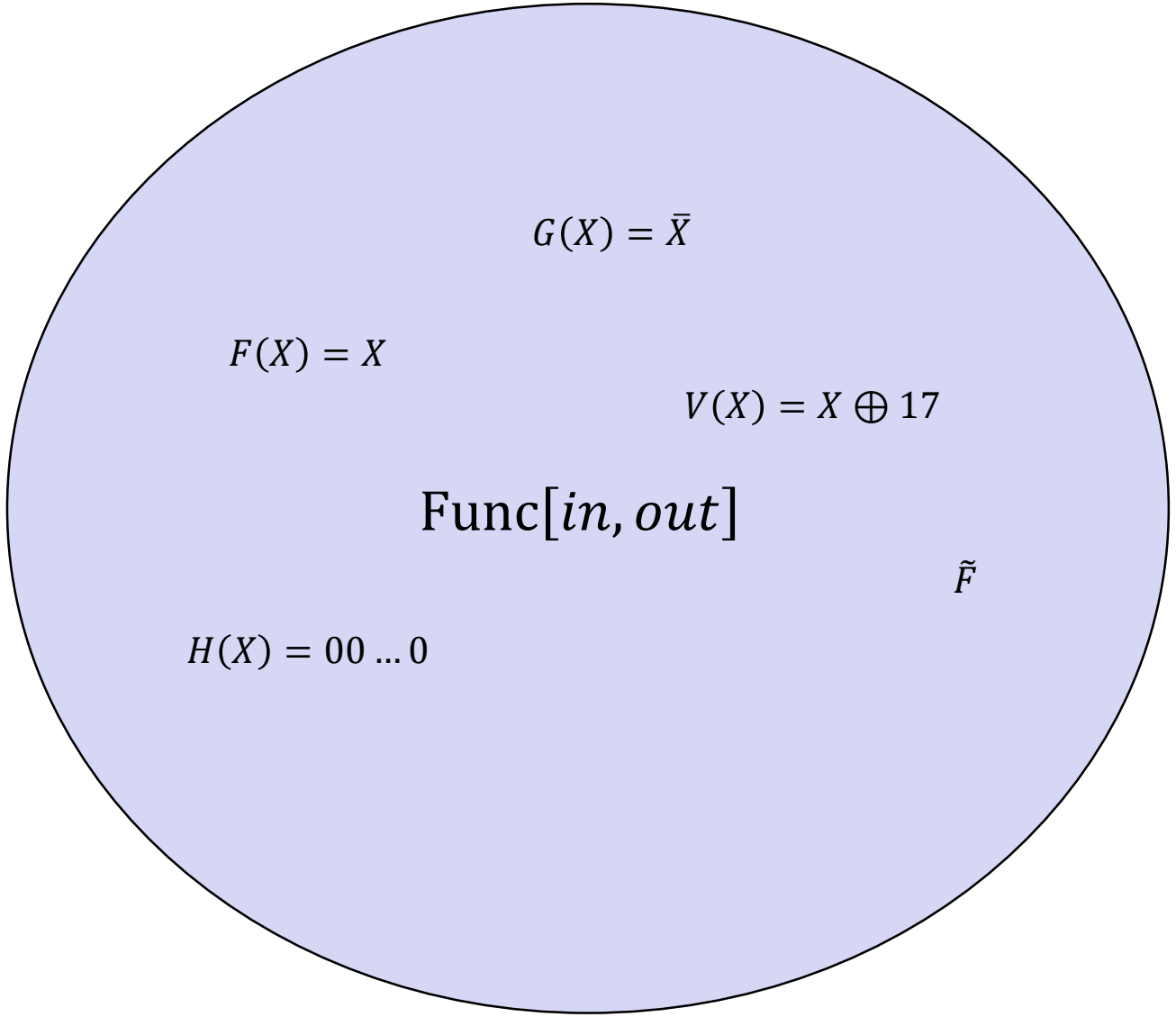
F is ***not*** PRF secure $\Leftarrow$ F does ***not*** have properties P1 – P5, P7, …

$$\Pr[\, b' = 1 \mid b = 1 \,] = 1$$

$$\Pr[\, b' = 0 \mid b = 0 \,] = 1 - \frac{1}{2^{out}}$$

PRF – security; equivalent view

$|\text{Func}[in, out]| =$



X	$\tilde{F}(X)$
000 … 000	101 … 111
000 … 001	001 … 001
000 … 010	111 … 100
000 … 011	101 … 000
⋮	⋮
111 … 111	001 … 001

2^{in}

out

PRF – security; equivalent view

$|\text{Func}[in, out]| =$ # bitstrings of length $2^{in} \cdot out$

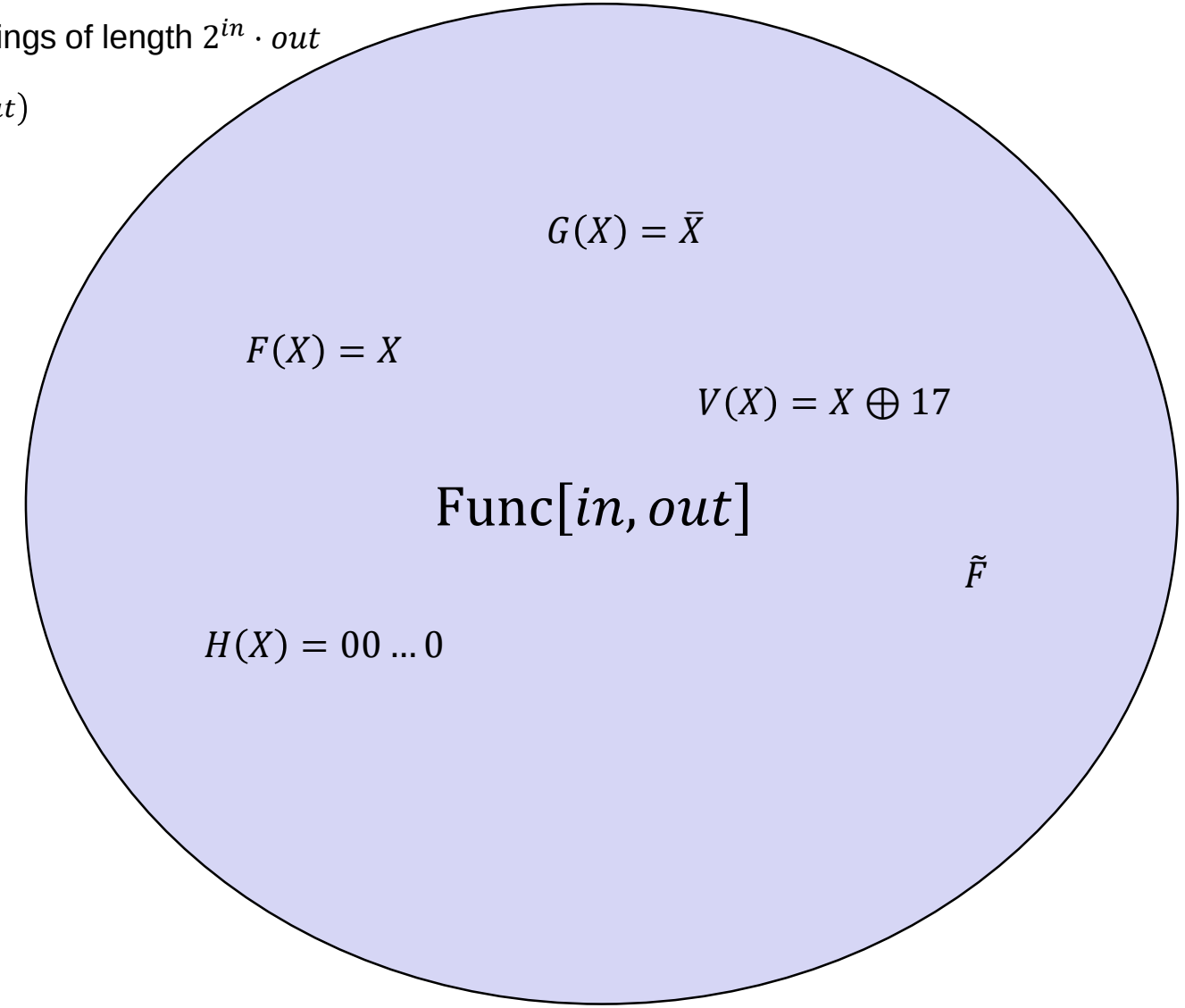
$= 2^{(2^{in} \cdot out)}$

Example:

$|\text{Func}[3,2]| = 2^{2^3 \cdot 2}$

$= 2^{16}$

$= 65536$



$\tilde{F}(X)$
101 … 111
001 … 001
111 … 100
101 … 000
⋮
001 … 001

2^{in}

out

- Bits needed to specify *one* function: $2^{in} \cdot out$

- Each bitstring of length $2^{in} \cdot out$ represents a *distinct* function

PRF – security; equivalent view

$|\text{Func}[in, out]| =$ # bitstrings of length $2^{in} \cdot out$

$= 2^{(2^{in} \cdot out)}$

Example:

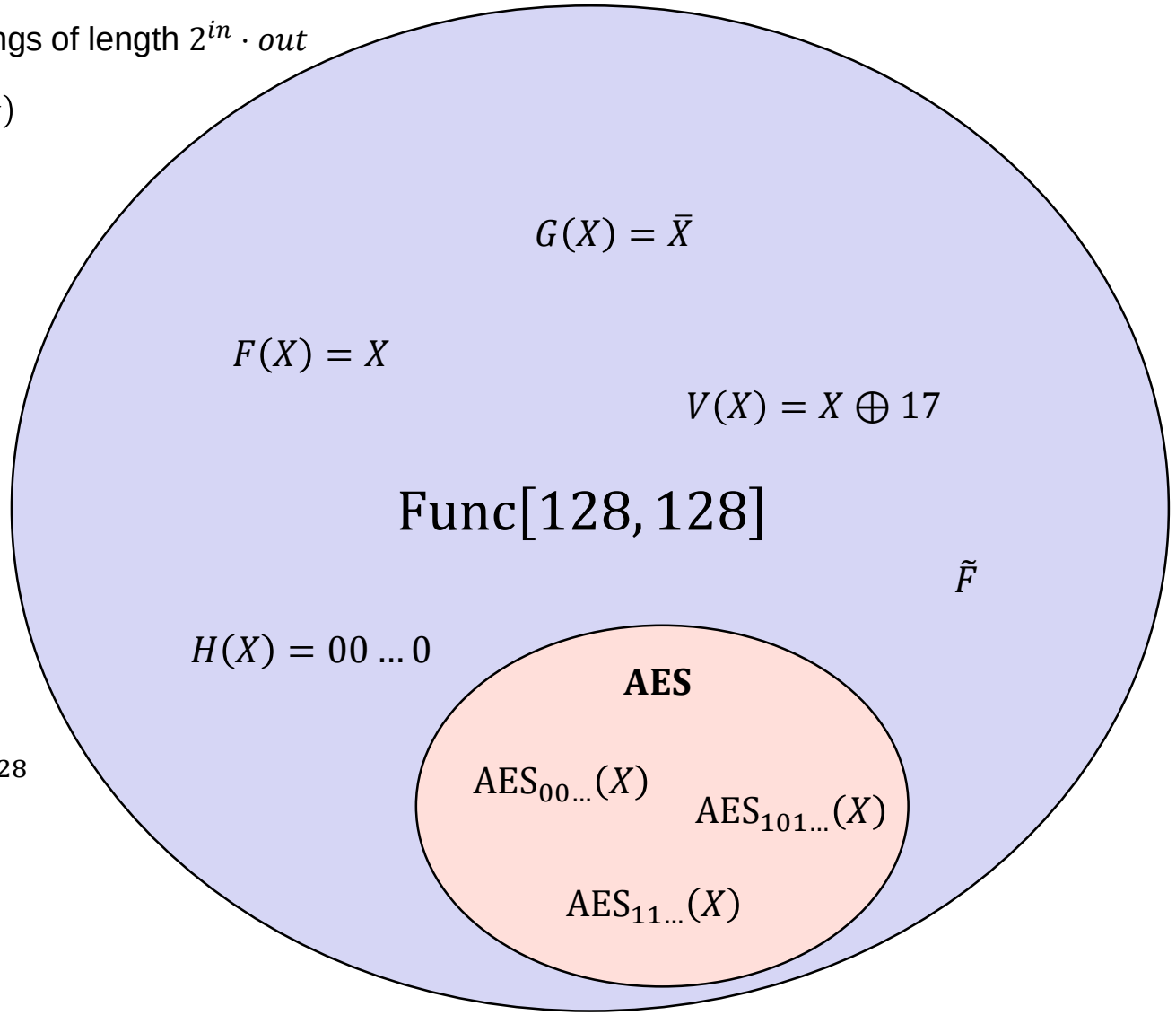
$|\text{Func}[3,2]| = 2^{2^3 \cdot 2}$

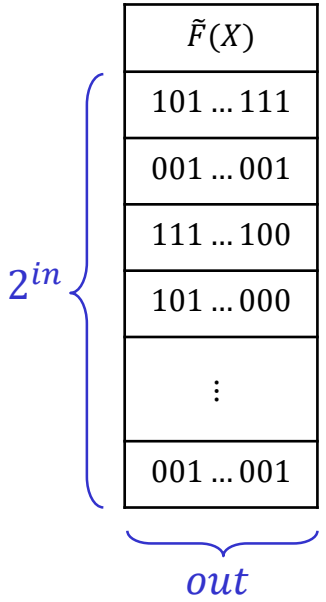
$= 2^{16}$

$= 65536$

$|\text{Func}[128,128]| = 2^{2^{128} \cdot 128}$

$|\text{AES}| = 2^{128}$





- Bits needed to specify *one* function: $2^{in} \cdot out$

- Each bitstring of length $2^{in} \cdot out$ represents a *distinct* function

PRF – security; equivalent view

$|\text{Func}[in, out]| =$ # bitstrings of length $2^{in} \cdot out$

$= 2^{(2^{in} \cdot out)}$

Example:

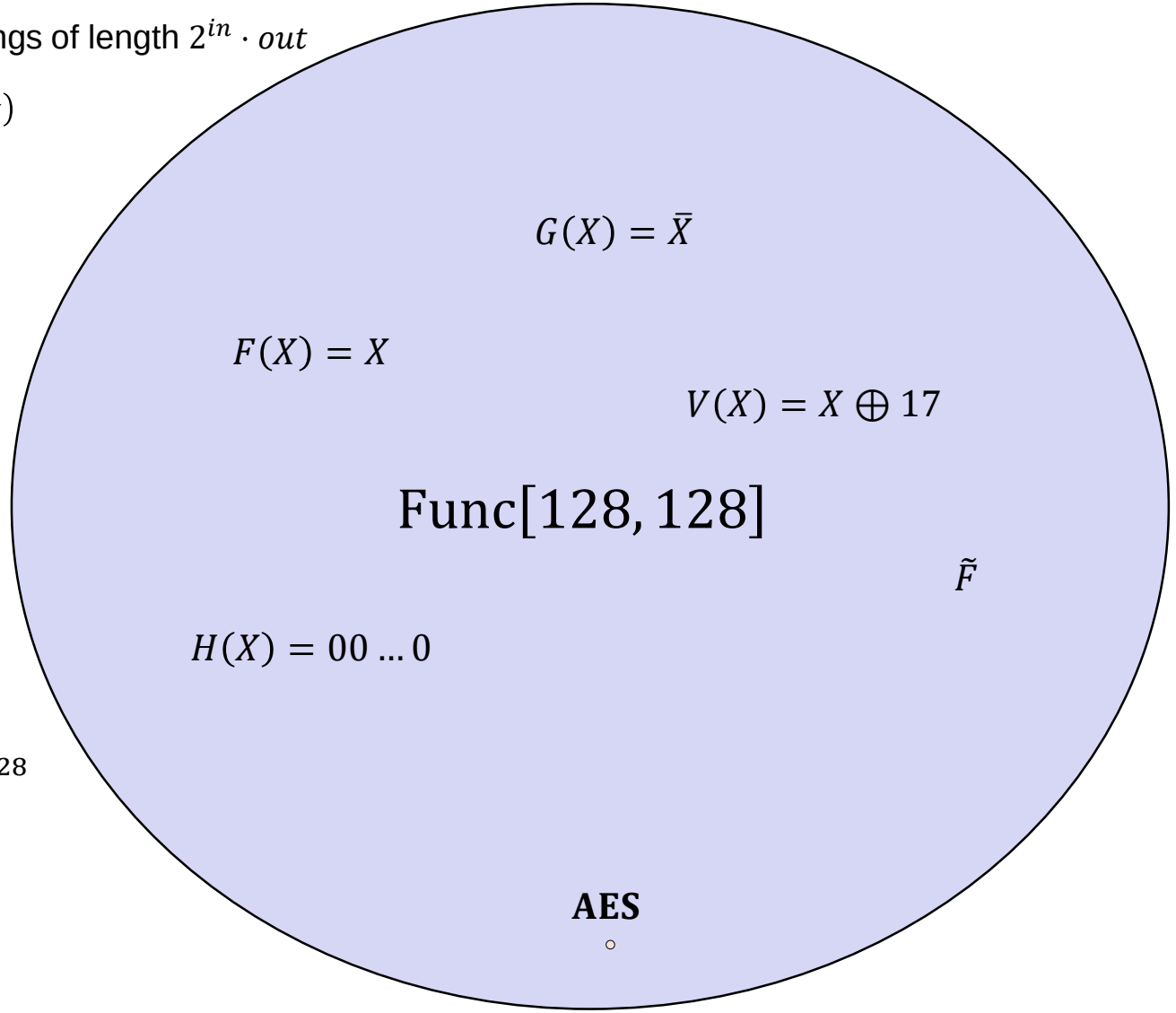
$|\text{Func}[3,2]| = 2^{2^3 \cdot 2}$

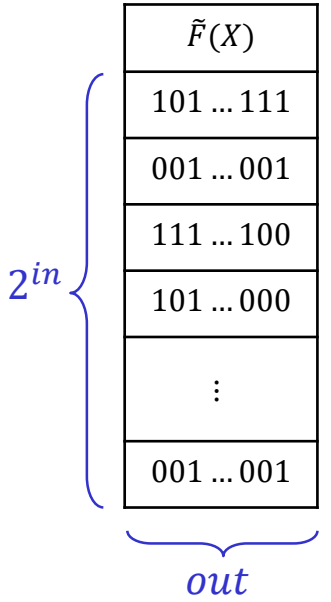
$= 2^{16}$

$= 65536$

$|\text{Func}[128,128]| = 2^{2^{128} \cdot 128}$

$|\text{AES}| = 2^{128}$

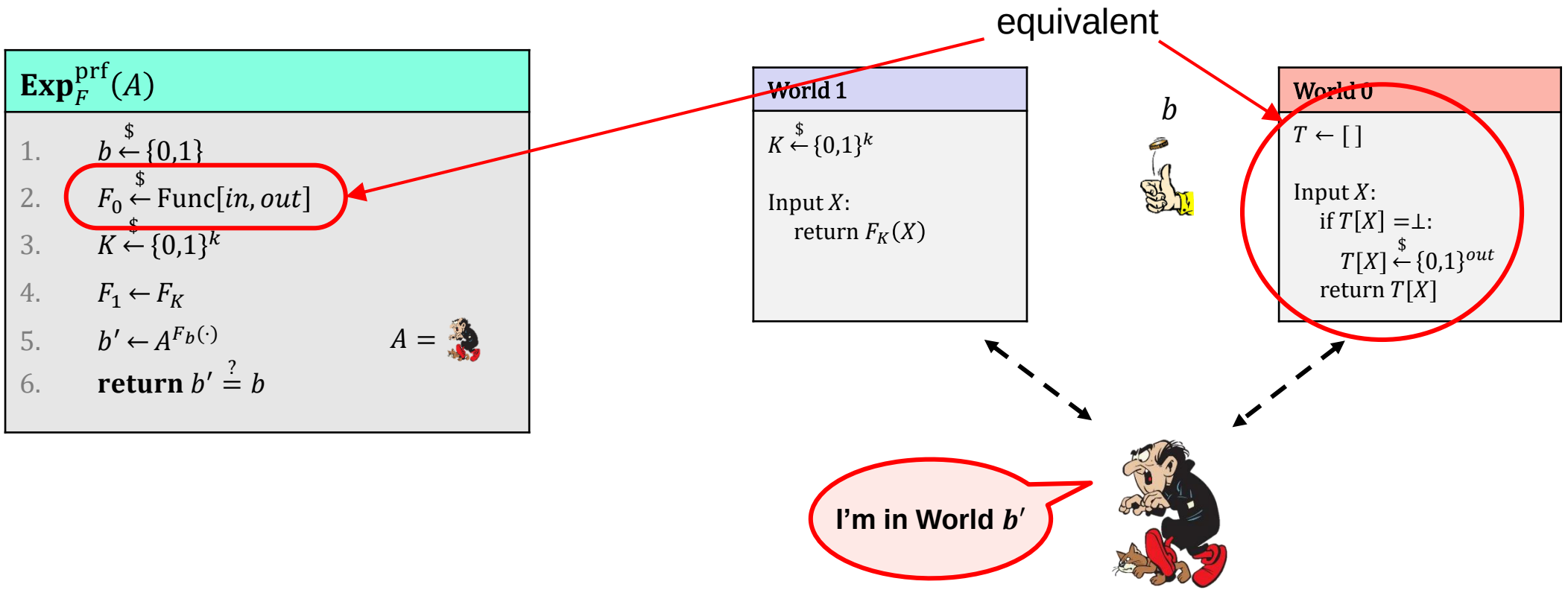




- Bits needed to specify *one* function: $2^{in} \cdot out$

- Each bitstring of length $2^{in} \cdot out$ represents a *distinct* function

PRF – security; formal definition

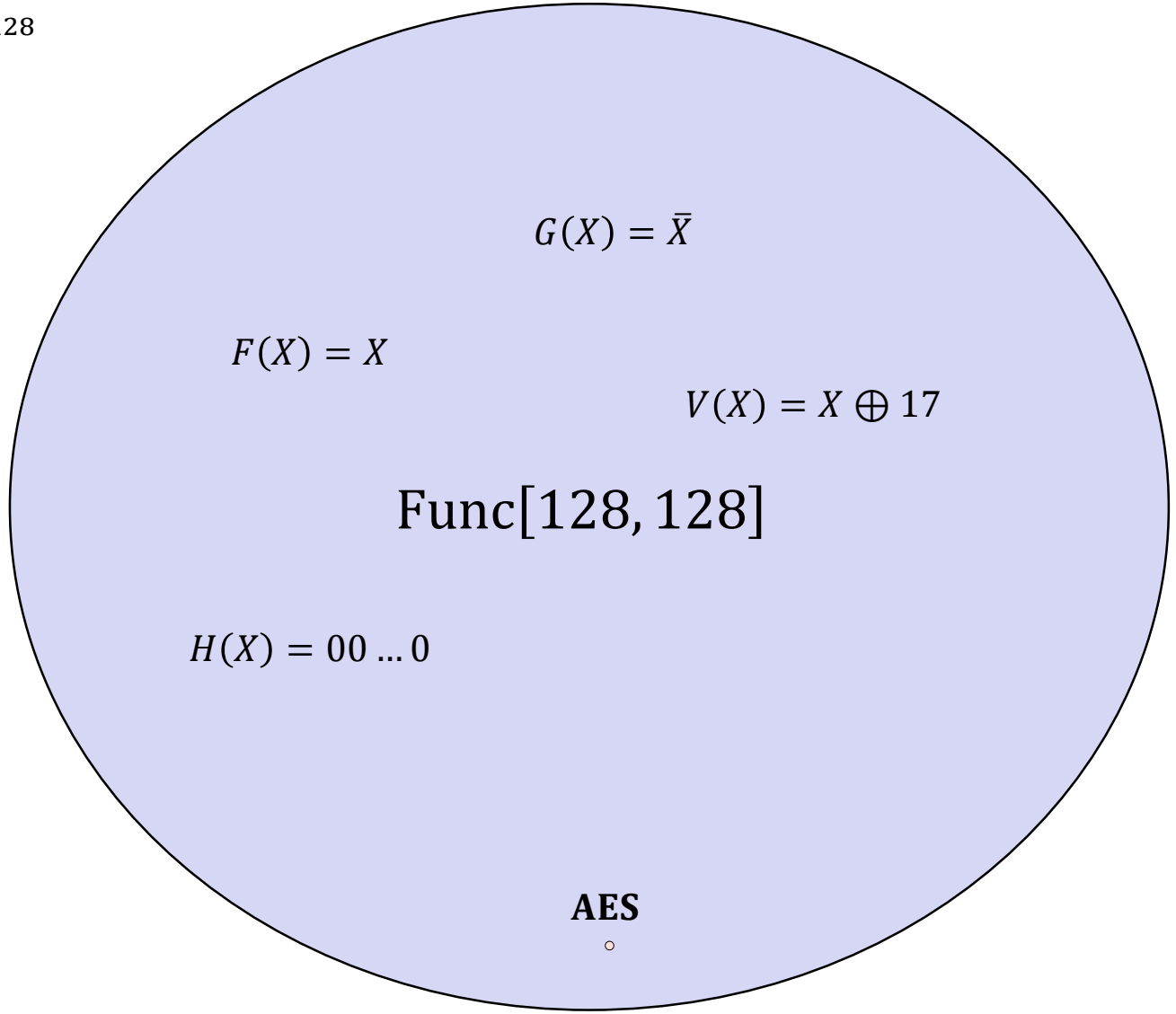


Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}_F^{\mathrm{prf}}(A) \Rightarrow \mathrm{true}\right] - 1\right|$$

Block ciphers – security

$|\text{Func}[128,128]| = 2^{2^{128} \cdot 128}$



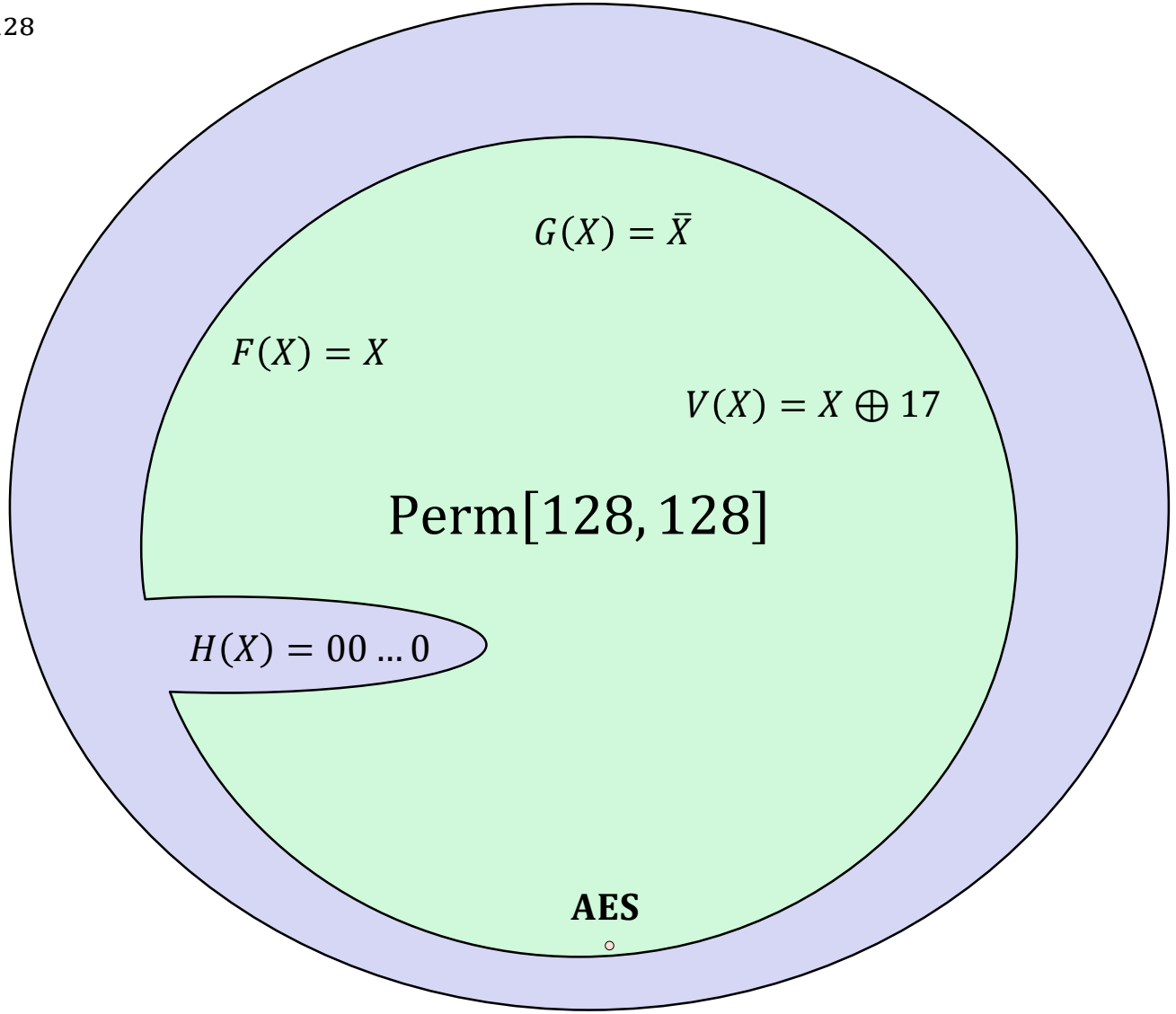
$|\text{AES}| = 2^{128}$

Block ciphers – security

$|\text{Func}[128,128]| = 2^{2^{128} \cdot 128}$

$|\text{Perm}[128,128]| = 2^{128}!$

$|\text{AES}| = 2^{128}$



~~PRF~~ PRP – security; formal definition

$\text{Exp}_F^{\text{~~prf~~ prp}}(A)$ $A =$ 

1. $b \xleftarrow{\$} \{0,1\}$
2. $F_0 \xleftarrow{\$}$ ~~$\text{Func}[in, out]$~~ $\text{Perm}[128,128]$
3. $K \xleftarrow{\$} \{0,1\}^k$
4. $F_1 \leftarrow F_K$
5. $b' \leftarrow A^{F_b(\cdot)}$
6. **return** $b' \stackrel{?}{=} b$

World 1

$K \xleftarrow{\$} \{0,1\}^{128}$

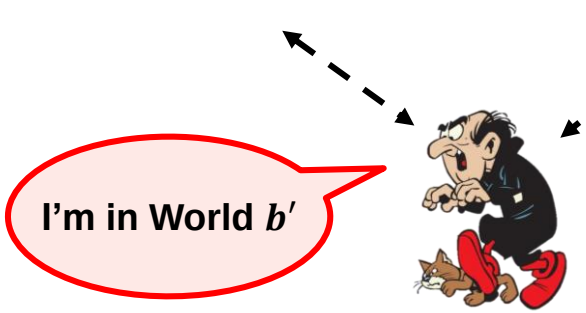
Input X:
  return $F_K(X)$

b

World 0

$T \leftarrow [\,]$

Input X:
  if $T[X] = \perp$:
    $T[X] \xleftarrow{\$} \{0,1\}^{out}$ \ T.values
  return $T[X]$



Definition: The ~~**PRF**~~ **PRP-advantage** of an adversary A is

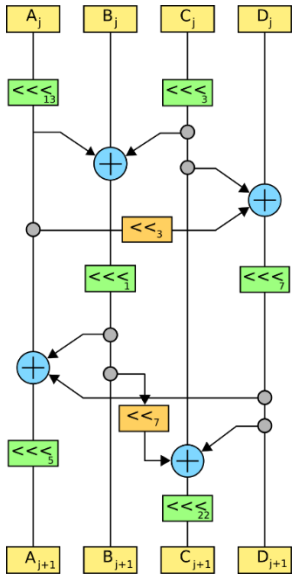
$$\mathbf{Adv}_F^{\text{prp}}(A) = \left|2 \cdot \Pr\left[A \text{ wins in PRP experiment}\right] - 1\right|$$

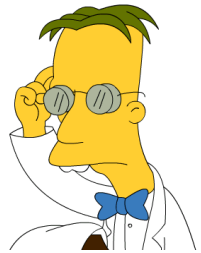
PRP security $\Longrightarrow$ PRF security

Theorem: a secure PRP $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is also a secure PRF.

Detailed: for all A making at most q oracle queries:

$$\mathbf{Adv}_E^{\mathrm{prf}}(A) \leq \mathbf{Adv}_E^{\mathrm{prp}}(A) + \frac{2q^2}{2^n}$$



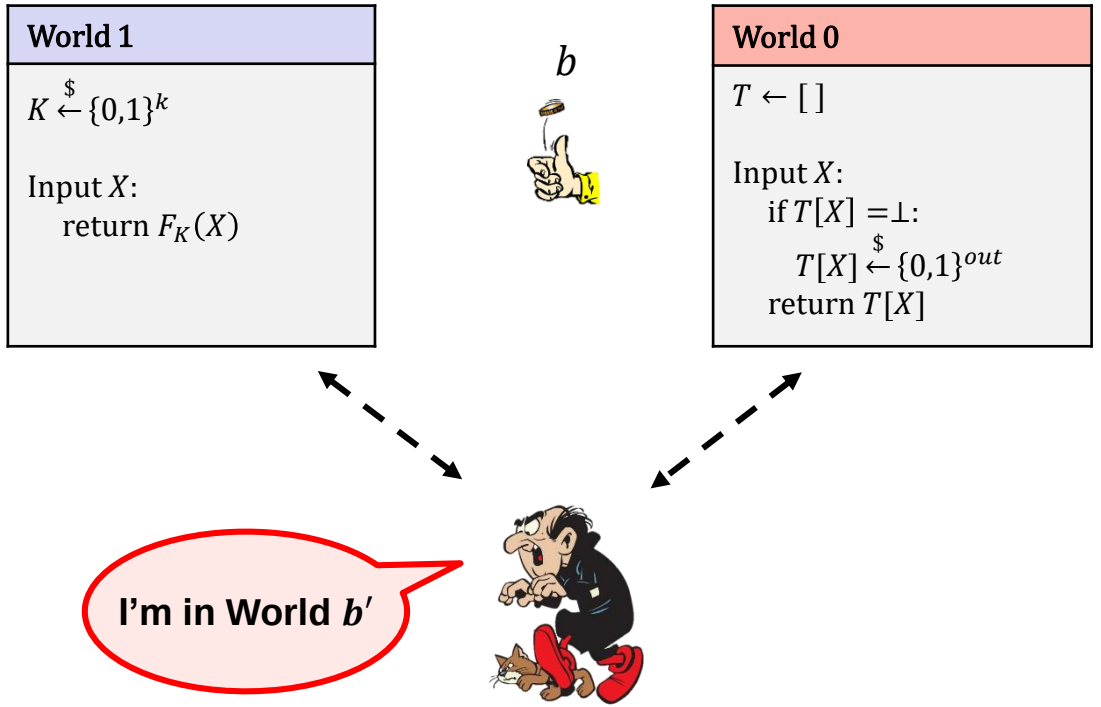


Constructing block ciphers

PRF – security; formal definition

$\mathbf{Exp}_F^{\mathrm{prf}}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $F_0 \xleftarrow{\$} \mathrm{Func}[in, out]$
3. $K \xleftarrow{\$} \{0,1\}^k$
4. $F_1 \leftarrow F_K$
5. $b' \leftarrow A^{F_b(\cdot)}$ $\qquad A =$ 
6. **return** $b' \stackrel{?}{=} b$



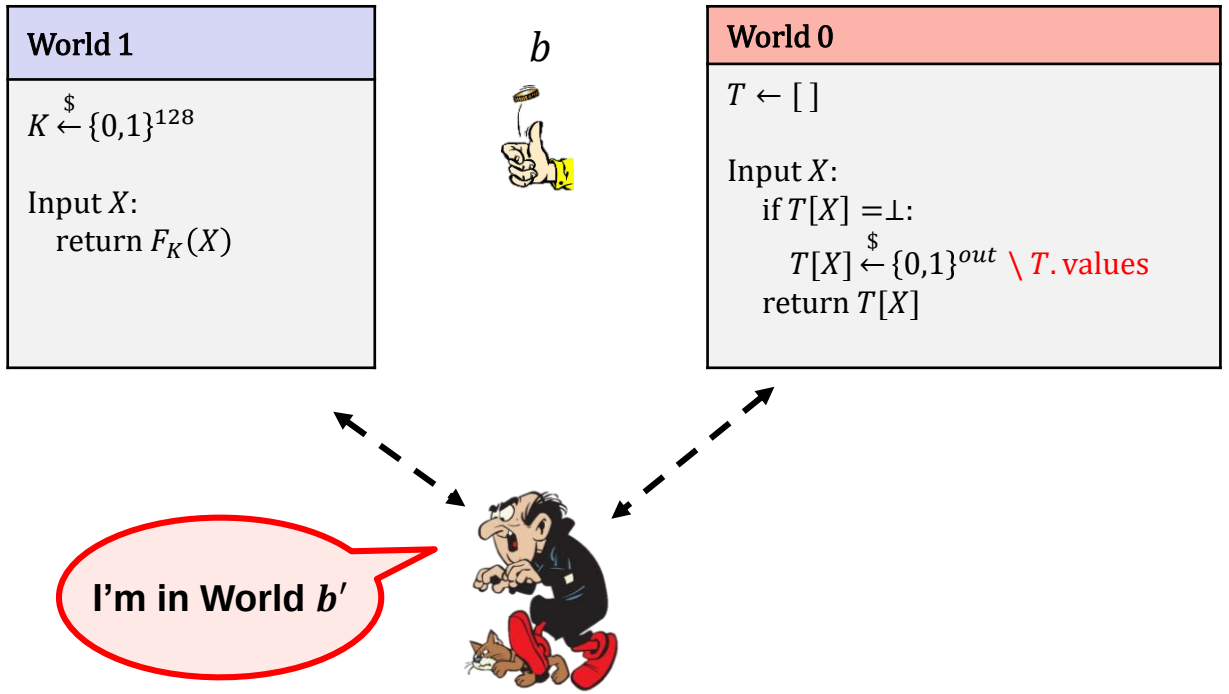
Definition: The **PRF-advantage** of an adversary A is

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \left|2 \cdot \Pr\left[\mathbf{Exp}_F^{\mathrm{prf}}(A) \Rightarrow \mathrm{true}\right] - 1\right|$$

~~PRF~~ PRP – security; formal definition

$\text{Exp}_F^{\text{~~prf~~ prp}}(A)$ $A =$ 

1. $b \xleftarrow{\$} \{0,1\}$
2. $F_0 \xleftarrow{\$}$ ~~$\text{Func}[in, out]$~~ $\text{Perm}[128,128]$
3. $K \xleftarrow{\$} \{0,1\}^k$
4. $F_1 \leftarrow F_K$
5. $b' \leftarrow A^{F_b(\cdot)}$
6. **return** $b' \stackrel{?}{=} b$



Definition: The ~~**PRF**~~ **PRP-advantage** of an adversary A is

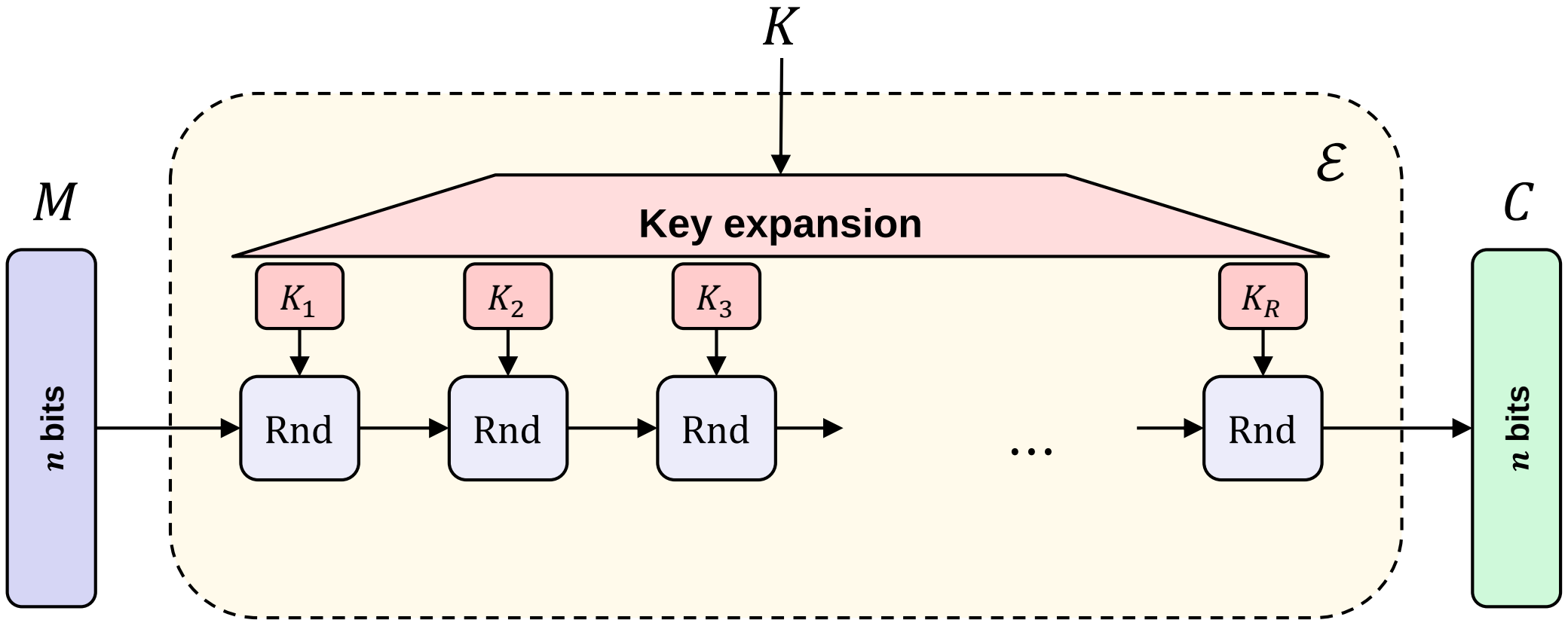
$$\mathbf{Adv}_F^{\text{prp}}(A) = \left|2 \cdot \Pr\left[A \text{ wins in PRP experiment}\right] - 1\right|$$

Principles for designing block ciphers

Claude Shannon, “Communication Theory of Secrecy Systems”(1949):

- **Diffusion**: plaintext spread over large parts of the ciphertext
- **Confusion**: a complex relation between plaintext, key and ciphertext

Block ciphers



$\mathrm{Rnd}(K_i, M)$ is called a **round function**

DES	$R = 16$
AES-128/192/256	$R = 10/12/14$

PRPs from PRFs – the Feistel construction

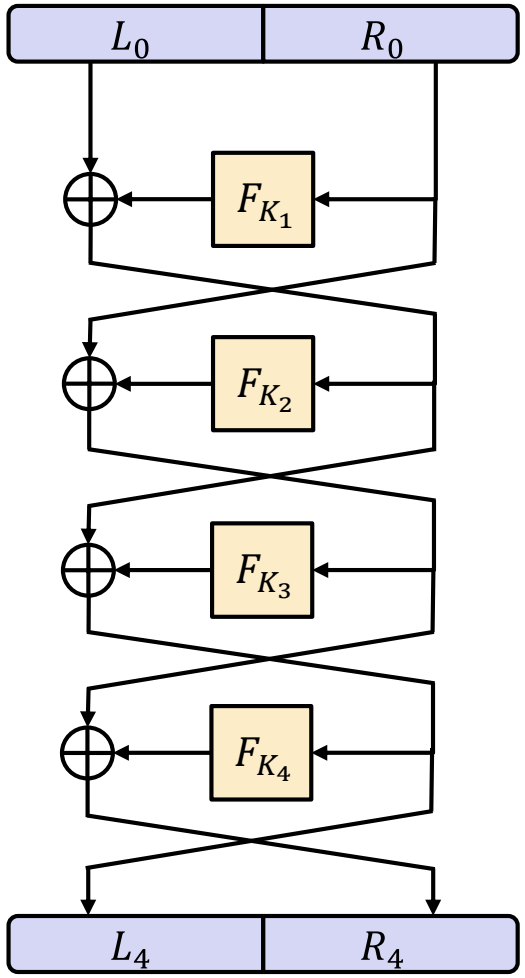
- Let $F : \{0,1\}^k \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ be a **PRF**
  - not a *permutation!*

- Function $E(K, X) = \text{Feistel}_F^{(4)}(K, X)$ *is* a **PRP**
  - Called a **Feistel network/construction**
  - $E : \{0,1\}^{4k} \times \{0,1\}^n \to \{0,1\}^n$

- More or less DES:

  $\text{DES} \approx \text{Feistel}_F^{(16)} : \{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$

  (56-bit key is expanded to 16 48-bit roundkeys)

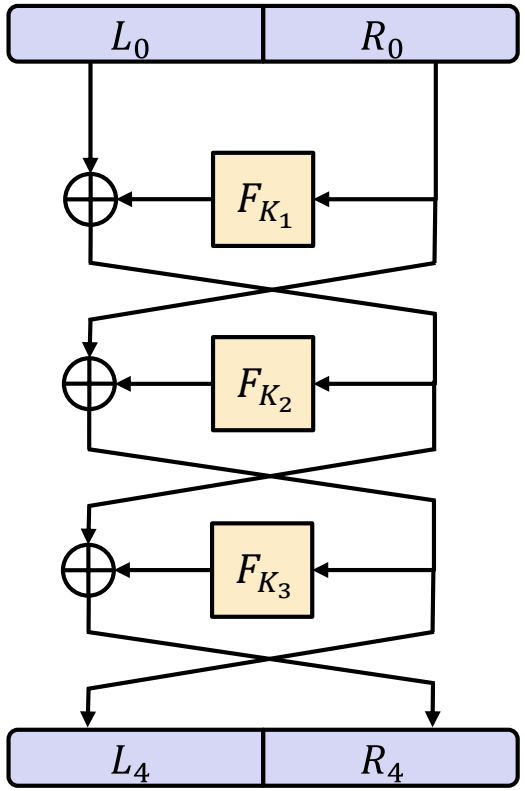


Feistel network security – theory

Theorem: (Luby & Rackoff '86)

$F : \{0,1\}^k \times \{0,1\}^{n/2} \rightarrow \{0,1\}^{n/2}$ is a **secure** PRF

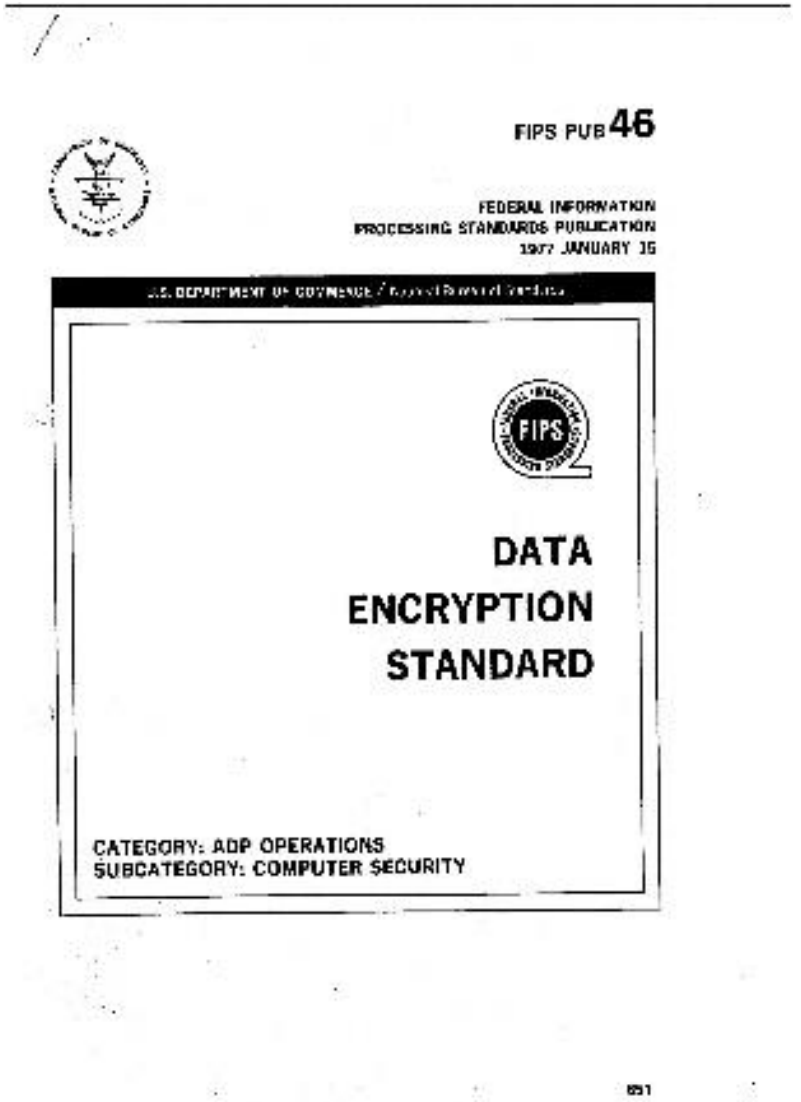
$\Longrightarrow$ 3-round Feistel $E : \{0,1\}^{3k} \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a **secure** PRP



Data Encryption Standard (DES)

- 1972 – NIST calls for a block cipher standard
- 1974 – Horst Feistel at IBM designs *Lucifer*
  - Key-length: 128 bits; block-length: 128 bits
- Lucifer evolves into *DES*
  - Input from the NSA
  - Key-length: 56 bits; block-length: 64 bits
  - #Rounds: 16
- 1976 – Lucifer (now DES) is standardized
- Widely implemented

- 1997 – Broken by exhaustive search
- 2001 – Replaced by AES


FIPS PUB 46

FEDERAL INFORMATION
PROCESSING STANDARDS PUBLICATION
1977 JANUARY 15

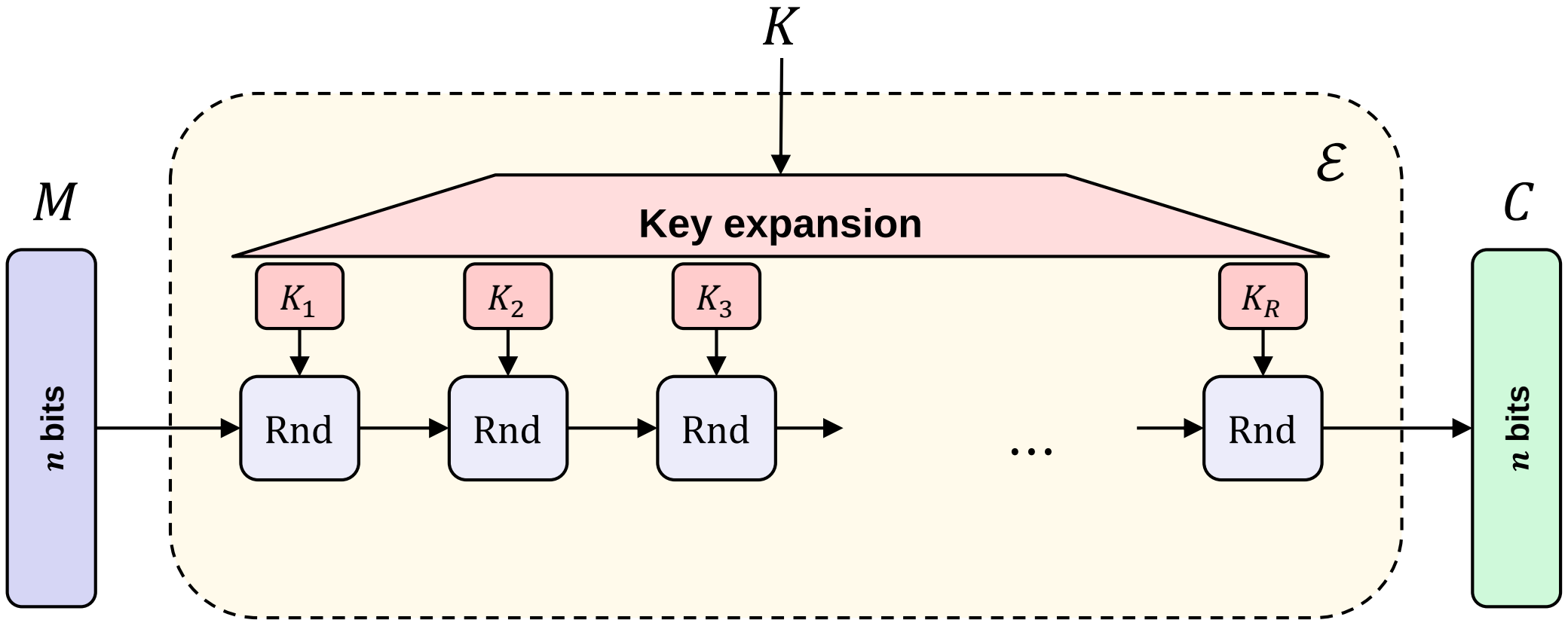
U.S. DEPARTMENT OF COMMERCE / National Bureau of Standards

FIPS

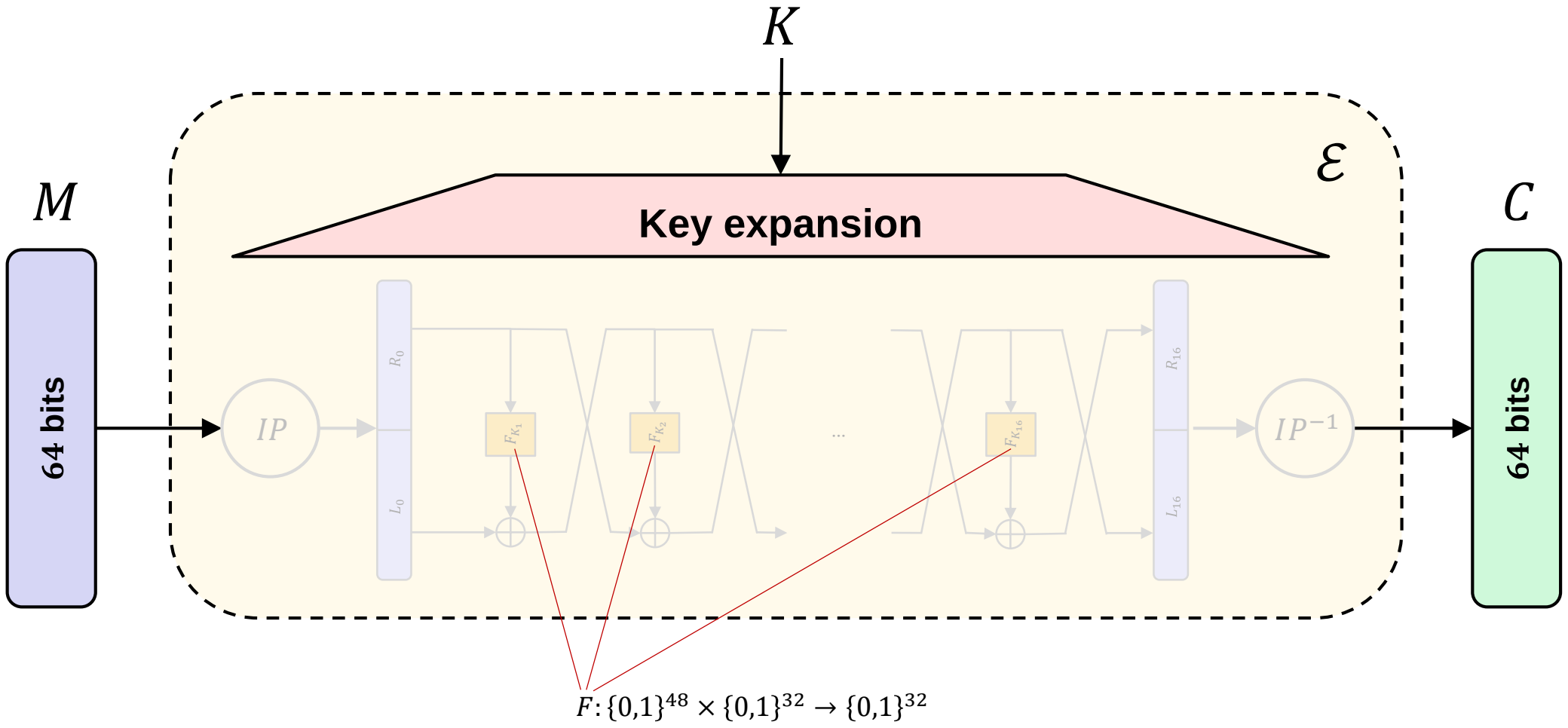
DATA
ENCRYPTION
STANDARD

CATEGORY: ADP OPERATIONS
SUBCATEGORY: COMPUTER SECURITY

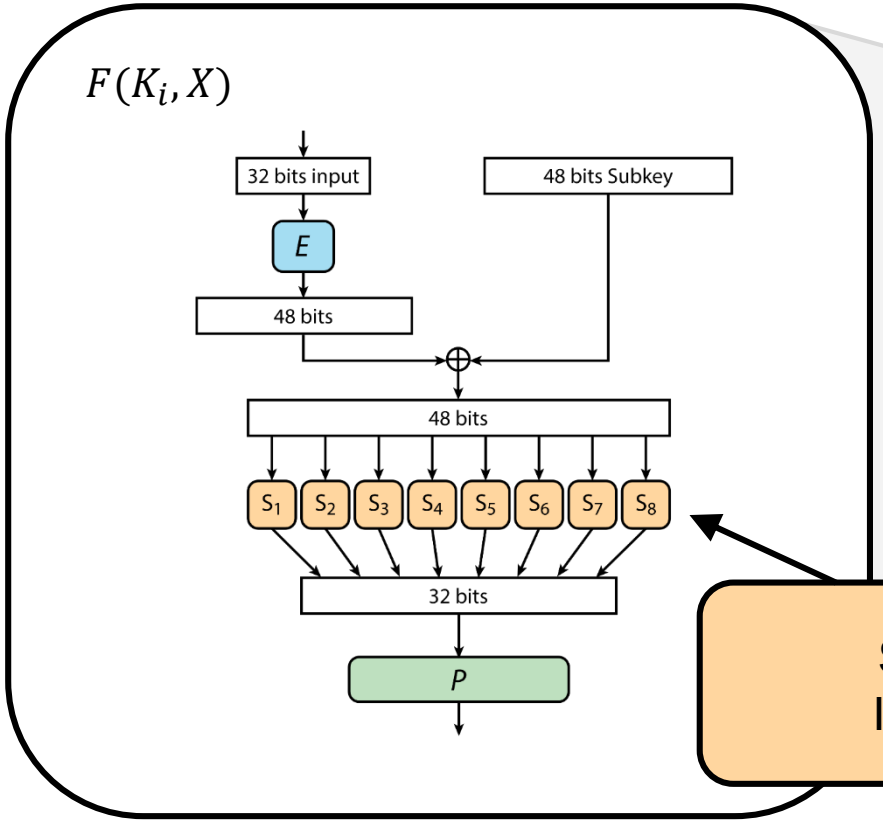
Block ciphers



DES


K
$\mathcal{E}$
Key expansion
M
64 bits
IP
R_0
L_0
F_{K_1}
F_{K_2}
...
$F_{K_{16}}$
R_{16}
L_{16}
IP^{-1}
C
64 bits
$F: \{0,1\}^{48} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$

DES round function



S1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
1	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
2	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
3	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D

S2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	F	1	8	E	6	B	3	4	9	7	2	D	C	0	5	A
1	3	D	4	7	F	2	8	E	C	0	1	A	6	9	B	5
2	0	E	7	B	A	4	D	1	5	8	C	6	9	3	2	F
3	D	8	A	1	3	F	4	2	B	6	7	C	0	5	E	9

S3

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	A	0	9	E	6	3	F	5	1	D	C	7	B	4	2	8
1	D	7	0	9	3	4	6	A	2	8	5	E	C	B	F	1
2	D	6	4	9	8	F	3	0	B	1	2	C	5	A	E	7
3	1	A	D	0	6	9	8	7	4	F	E	3	B	5	2	C

S4

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	7	D	E	3	0	6	9	A	1	2	8	5	B	C	4	F
1	D	8	B	5	6	F	0	3	4	7	2	C	1	A	E	9
2	A	6	9	0	C	B	7	D	F	1	3	E	5	2	8	4
3	3	F	0	6	A	1	D	8	9	4	5	B	C	7	2	E

S5

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	2	C	4	1	7	A	B	6	8	5	3	F	D	0	E	9
1	E	B	2	C	4	7	D	1	5	0	F	A	3	9	8	6
2	4	2	1	B	A	D	7	8	F	9	C	5	6	3	0	E
3	B	8	C	7	1	E	2	D	6	F	0	9	A	4	5	3

S6

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C	1	A	F	9	2	6	8	0	D	3	4	E	7	5	B
1	A	F	4	2	7	C	9	5	6	1	D	E	0	B	3	8
2	9	E	F	5	2	8	C	3	7	0	4	A	1	D	B	6
3	4	3	2	C	9	5	F	A	B	E	1	7	6	0	8	D

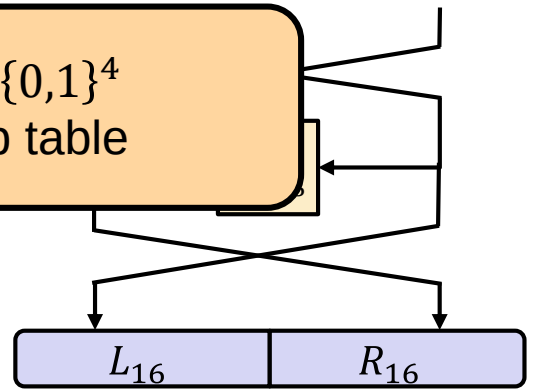
S7

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	4	B	2	E	F	0	8	D	3	C	9	7	5	A	6	1
1	D	0	B	7	4	9	1	A	E	3	5	C	2	F	8	6
2	1	4	B	D	C	3	7	E	A	F	6	8	0	5	9	2
3	6	B	D	8	1	4	A	7	9	5	0	F	E	2	3	C

S8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	D	2	8	4	6	F	B	1	A	9	3	E	5	0	C	7
1	1	F	D	8	A	3	7	4	C	5	6	B	0	E	9	2
2	7	B	4	1	9	C	E	2	0	6	A	D	F	3	5	8
3	2	1	E	7	4	A	8	D	F	C	9	0	3	5	6	B

S-box: function $\{0,1\}^6 \rightarrow \{0,1\}^4$
Implemented as a look-up table

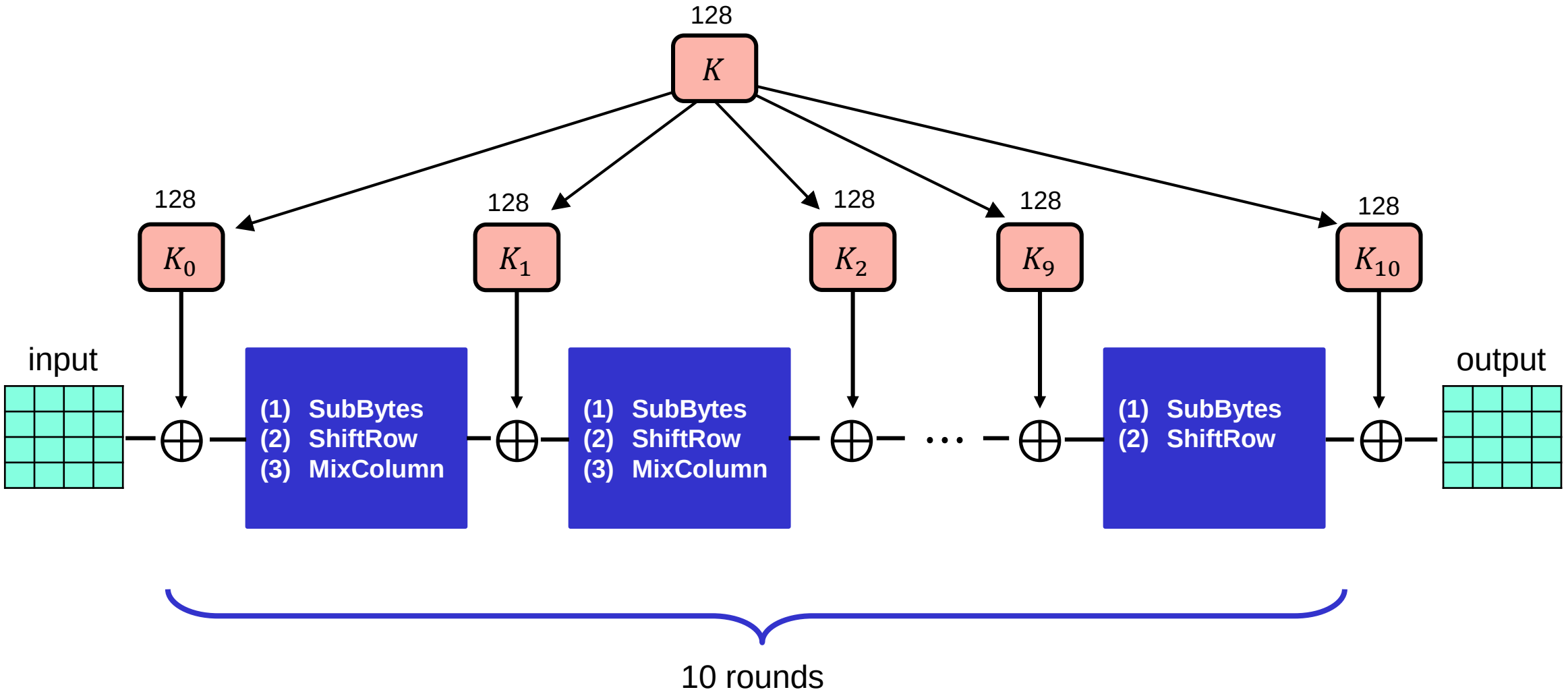


DES properties

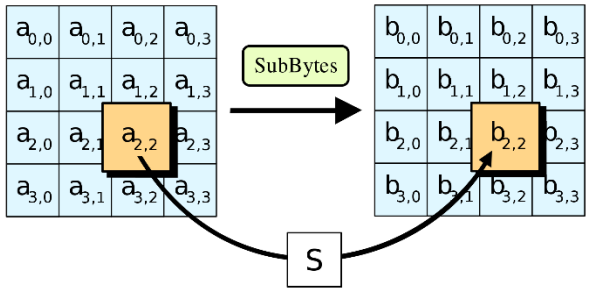
- Easy to implement in hardware
- Not as efficient in software
- Many design decisions still unclear
  - Design criteria classified for many years
  - Controversy around NSA influence
  - Initial S-boxes were changed
  - Switching to 56-bit keys (from 128 bits) probably to allow NSA to decrypt
- **Not secure** since key space and block length too small $\Longrightarrow$ replacement needed

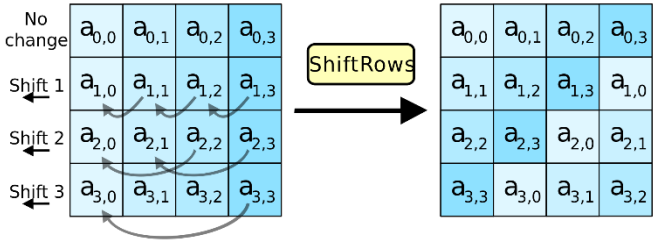
Advanced Encryption Standard

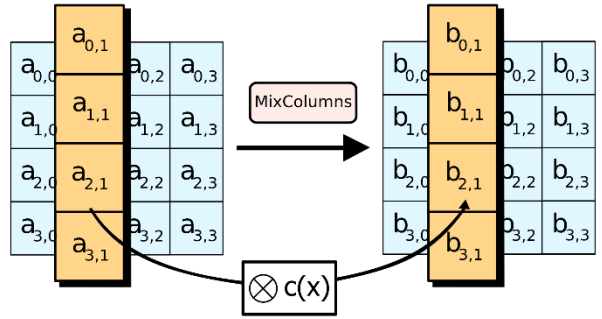
AES-128


128
K
128
K_0
128
K_1
128
K_2
128
K_9
128
K_{10}
input
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
output
10 rounds

AES round function

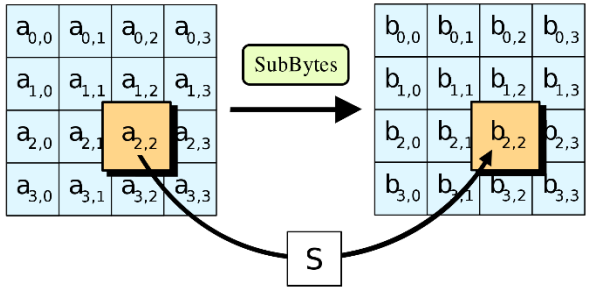


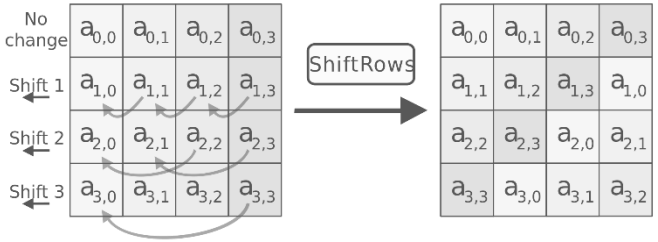


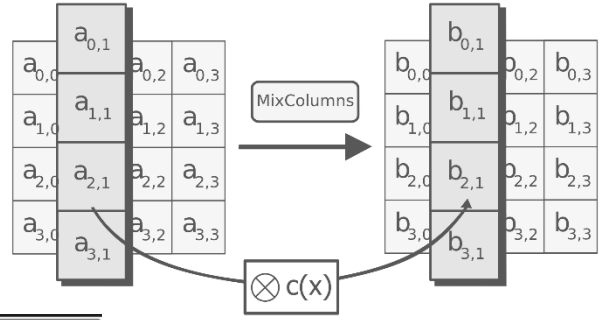


(1) SubBytes
(2) ShiftRow
(3) MixColumn

AES round function - SubBytes





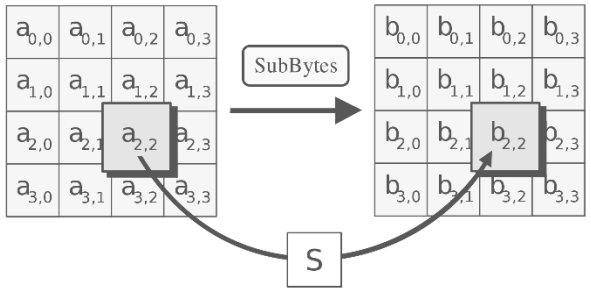


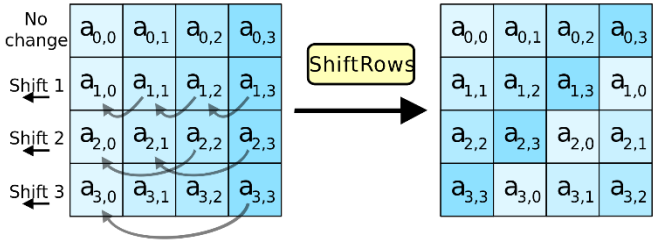
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

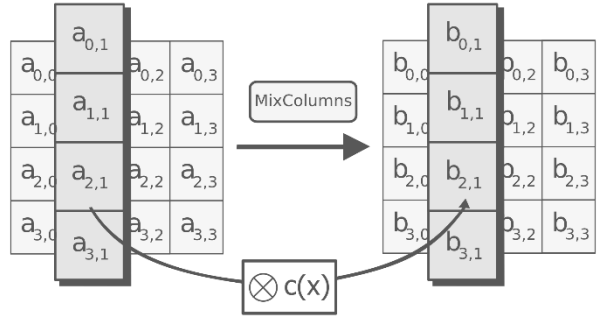
$a_{2,2} = 8A$

$b_{2,2} = 7E$

AES round function - ShiftRows

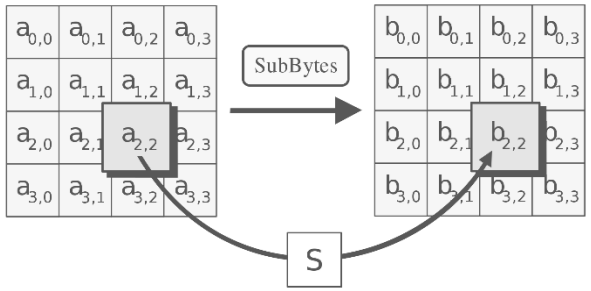


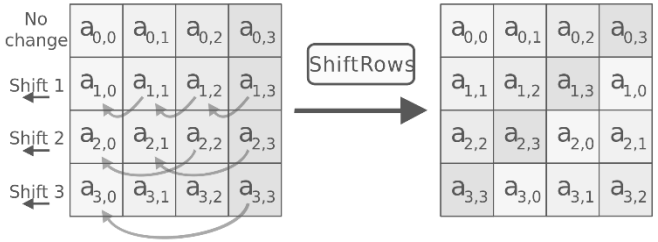


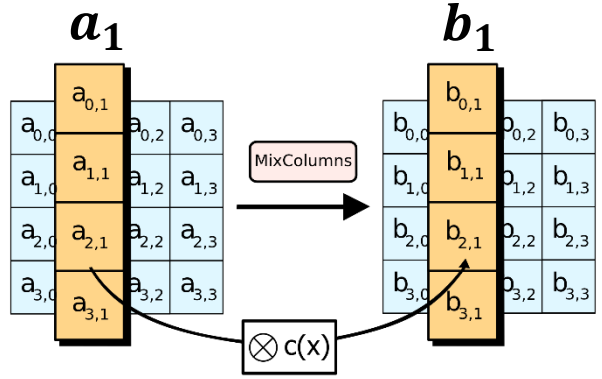


(1) SubBytes
(2) ShiftRow
(3) MixColumn

AES round function - MixColumns







(1) SubBytes
(2) ShiftRow
(3) MixColumn

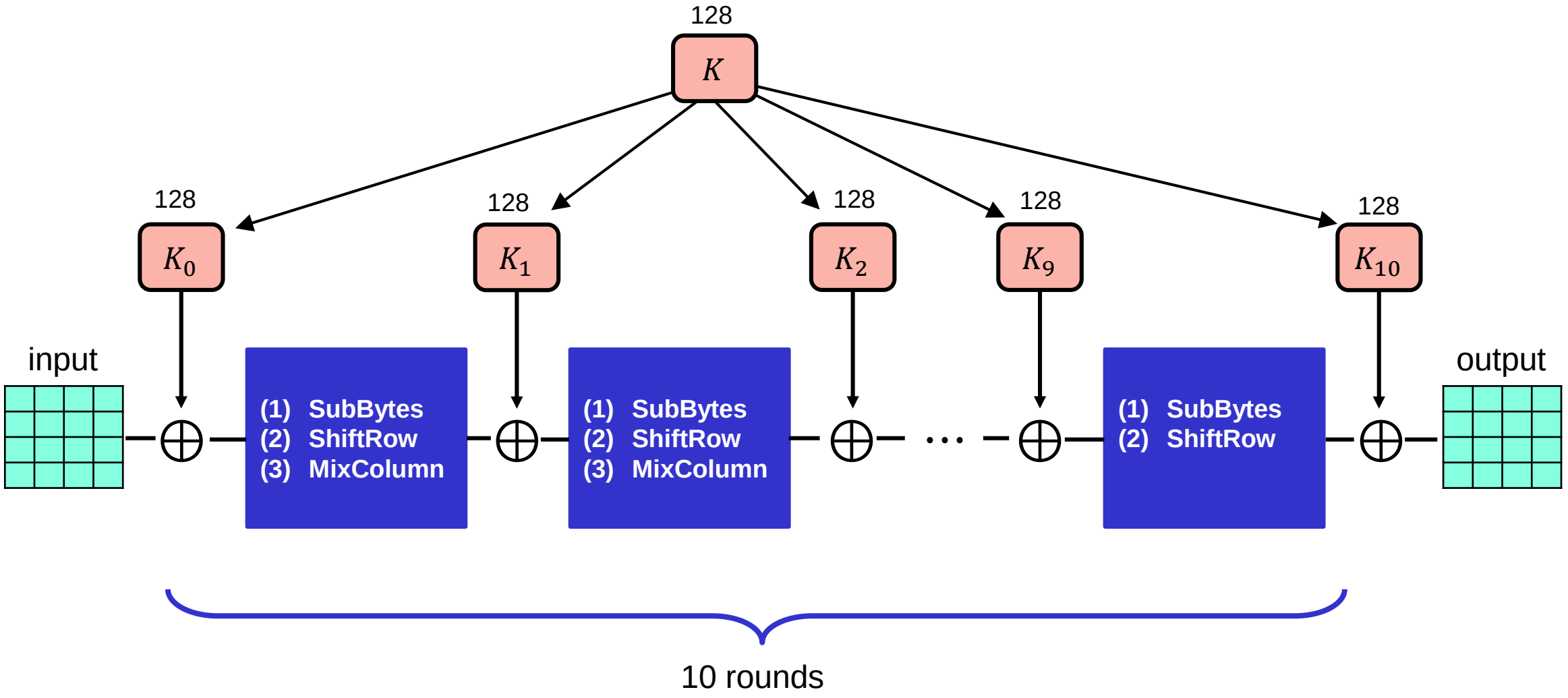
$$M\boldsymbol{a_1} = \boldsymbol{b_1}$$

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_{0,1} \\ a_{1,1} \\ a_{2,1} \\ a_{3,1} \end{pmatrix} = \begin{pmatrix} b_{0,1} \\ b_{1,1} \\ b_{2,1} \\ b_{3,1} \end{pmatrix}$$

$$1 \cdot a_{0,1} + 2 \cdot a_{1,1} + 3 \cdot a_{2,1} + 1 \cdot a_{3,1} = b_{1,1}$$

$$a_{0,1} \oplus 2 * a_{1,1} \oplus 3 * a_{2,1} \oplus a_{3,1} = b_{1,1}$$

AES-128


128
K
128
K_0
128
K_1
128
K_2
128
K_9
128
K_{10}
input
output
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
10 rounds

AES round

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

SubBytes	$b_{i,j} = S[a_{i,j}]$
ShiftRows	$\begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix} = \begin{bmatrix} b_{0,j} \\ b_{1,j-1} \\ b_{2,j-2} \\ b_{3,j-3} \end{bmatrix}$
MixColumns	$\begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix}$
AddRoundKey	$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S[a_{0,j}] \\ S[a_{1,j-1}] \\ S[a_{2,j-2}] \\ S[a_{3,j-3}] \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$$

$$= \left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \cdot S[a_{0,j}] \right) \oplus \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \cdot S[a_{1,j-1}] \right)$$

$$\oplus \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \cdot S[a_{2,j-2}] \right) \oplus \left(\begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \cdot S[a_{3,j-3}] \right) \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$$

$T_0[x] = \left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \cdot S[x] \right)$	$T_1[x] = \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_2[x] = \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_3[x] = \left(\begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \cdot S[x] \right)$

AES round

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

SubBytes	$b_{i,j} = S[a_{i,j}]$
ShiftRows	$\begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix} = \begin{bmatrix} b_{0,j} \\ b_{1,j-1} \\ b_{2,j-2} \\ b_{3,j-3} \end{bmatrix}$
MixColumns	$\begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix}$
AddRoundKey	$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = T_0[a_{0,j}] \oplus T_1[a_{1,j-1}] \oplus T_2[a_{2,j-2}] \oplus T_3[a_{3,j-3}] \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$$

$T_0[x] = \left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \cdot S[x] \right)$	$T_1[x] = \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_2[x] = \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_3[x] = \left(\begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} \cdot S[x] \right)$

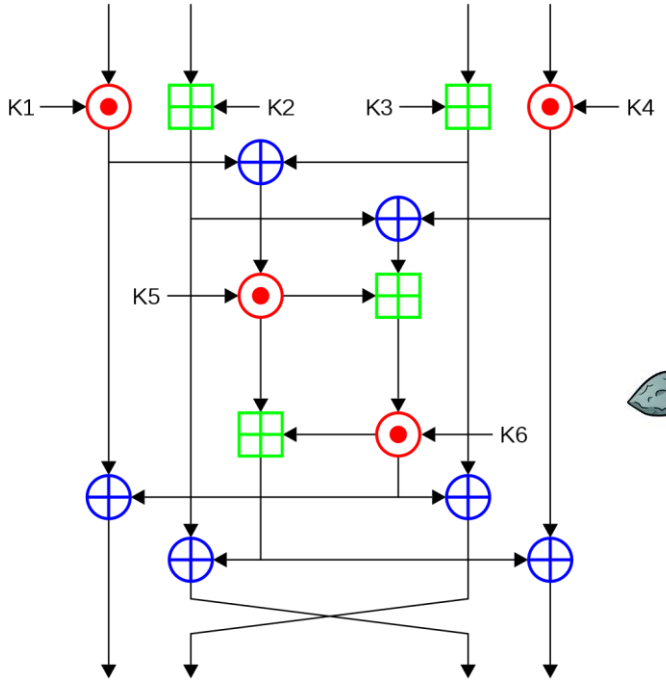
AES performance

- AES is reasonably efficient in software
  - T-table implementation very fast (but not secure!)
  - Hard to implement fast and constant-time

	Throughput (my laptop)
AES-128 (in software)	0.27 GB/s
AES-128 (w/AES-NI)	3.45 GB/s

- Intel introduced dedicated AES instructions into their CPUs (AES-NI):
  - **aesenc**, **aesenclast**: do one round of AES in one cycle
  - **aeskeygenassist:** do AES key expansion
  - **aesdec, aesdeclast:** do one round of AES decryption in one cycle
  - **aesimc:** do AES inverser MixColumns

  - Now standard in all modern CPUs



Attacking block ciphers

Attacks on block ciphers

- Brute force attacks: search through every possible key in key space
  - Generic: works for all block ciphers
  - Not practical for large key spaces

- Advanced attacks: try to exploit the concrete details of the block cipher
  - Differential cryptanalysis ('90, but known by the designers of DES + NSA since mid '70)
  - Linear cryptanalysis ('92)
  - AES designed to resist both

- Implementation attacks: vulnerabilities due to implementation characteristics
  - Power draw
  - Timing
  - Cache misses

Summary

- Block ciphers are very important **primitives** (building blocks) – but they are not encryption schemes!

- Correct abstraction: block ciphers = PRPs

- Right security notion for PRFs/PRPs:

  indistinguishability from random function/permutation

- Concrete block cipher designs: DES and AES