Lecture 2 – Block ciphers, PRFs/PRPs, AES

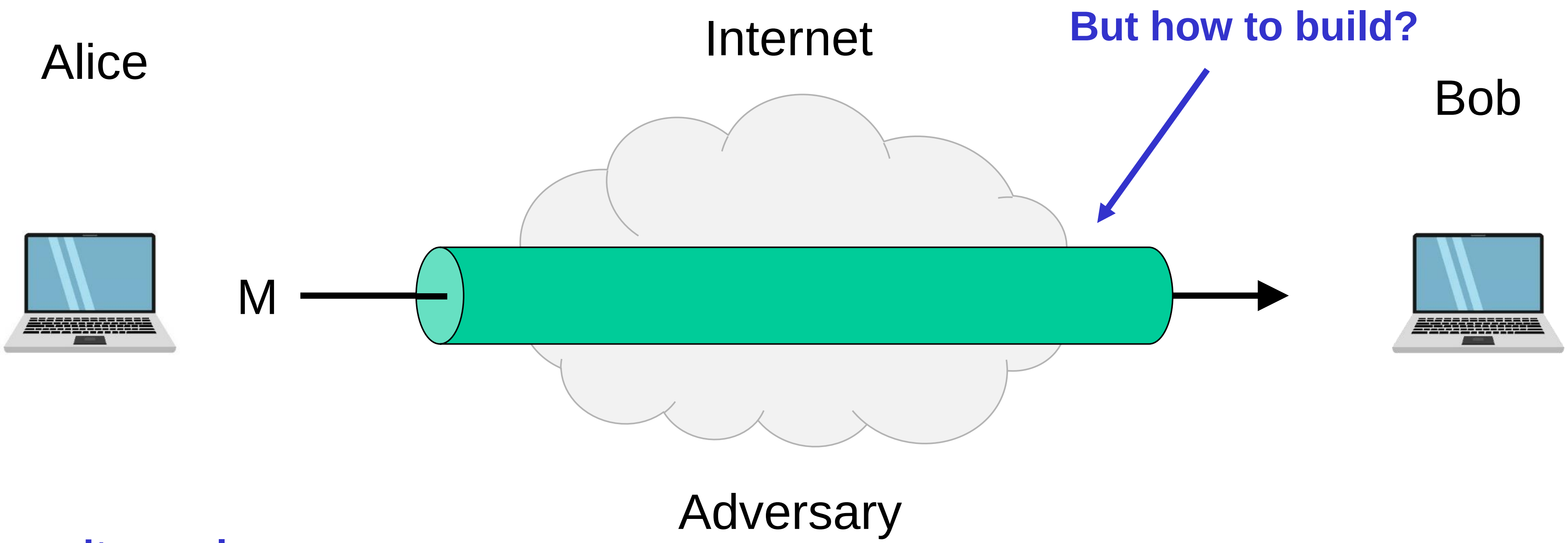
TEK4500

31.08.2022

Håkon Jacobsen

hakon.jacobsen@its.uio.no

Ideal solution: secure channels



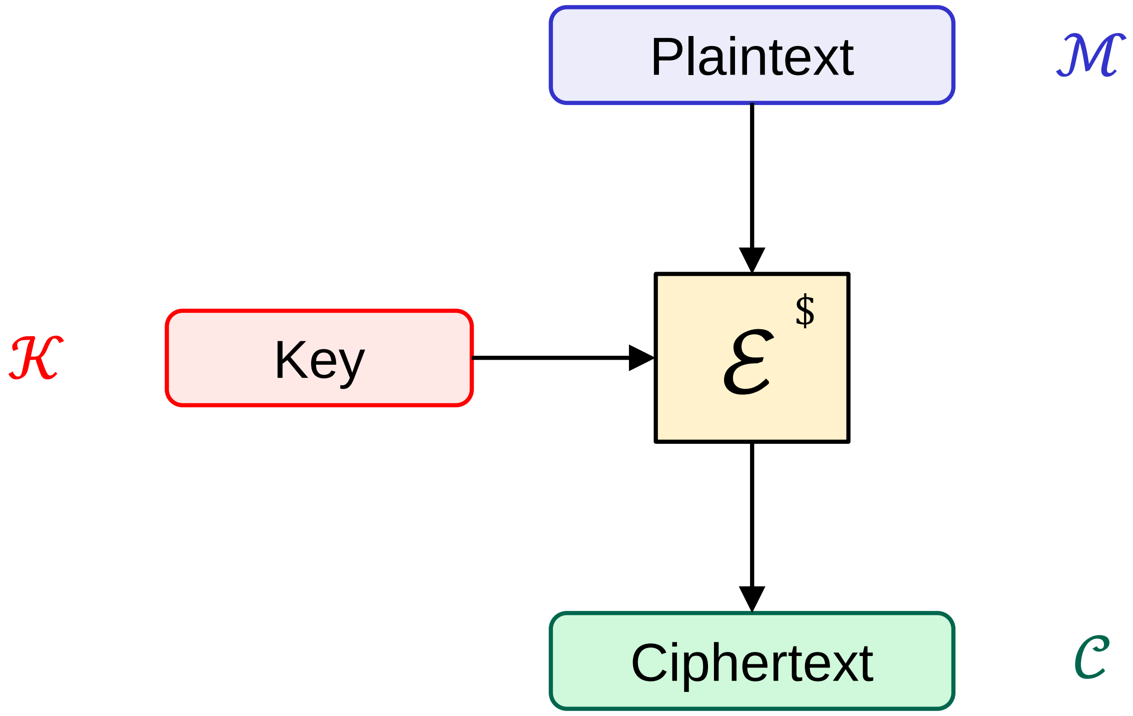
Security goals:

- **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 ✓

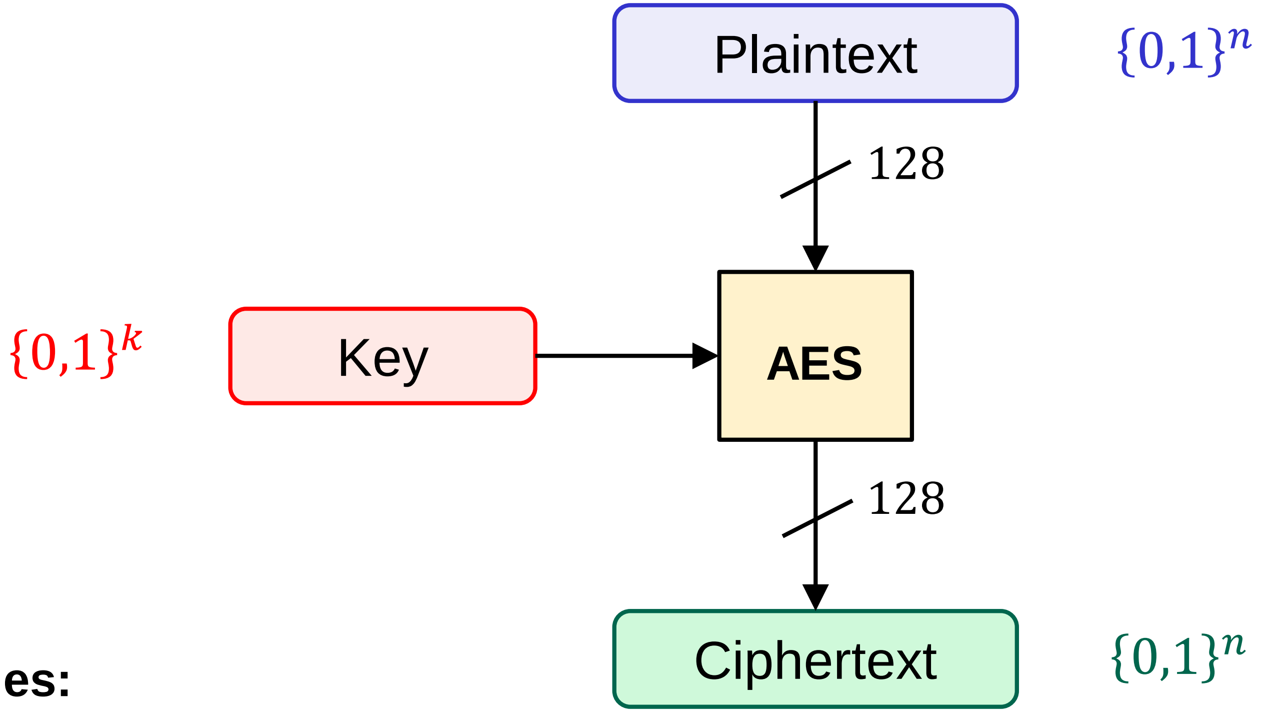
Basic goals of cryptography

	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

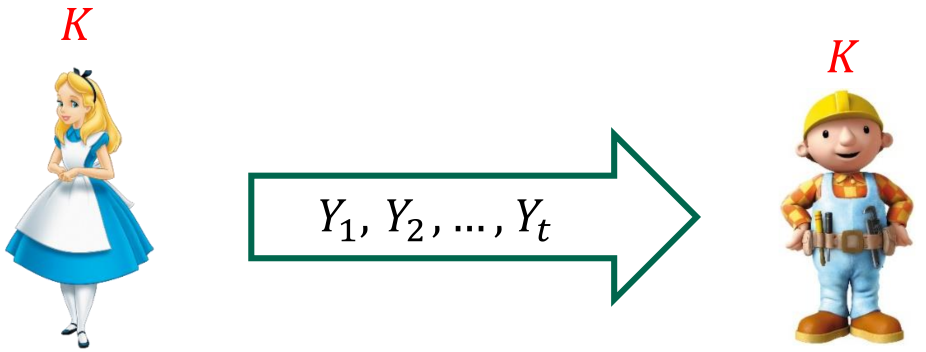


Examples:

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

Block cipher applications (1)

- Encryption of messages of 128 bits (block length)



$Y_i \leftarrow E_K(X_i)$ $\qquad$ $X_i \leftarrow E_K^{-1}(Y_i)$

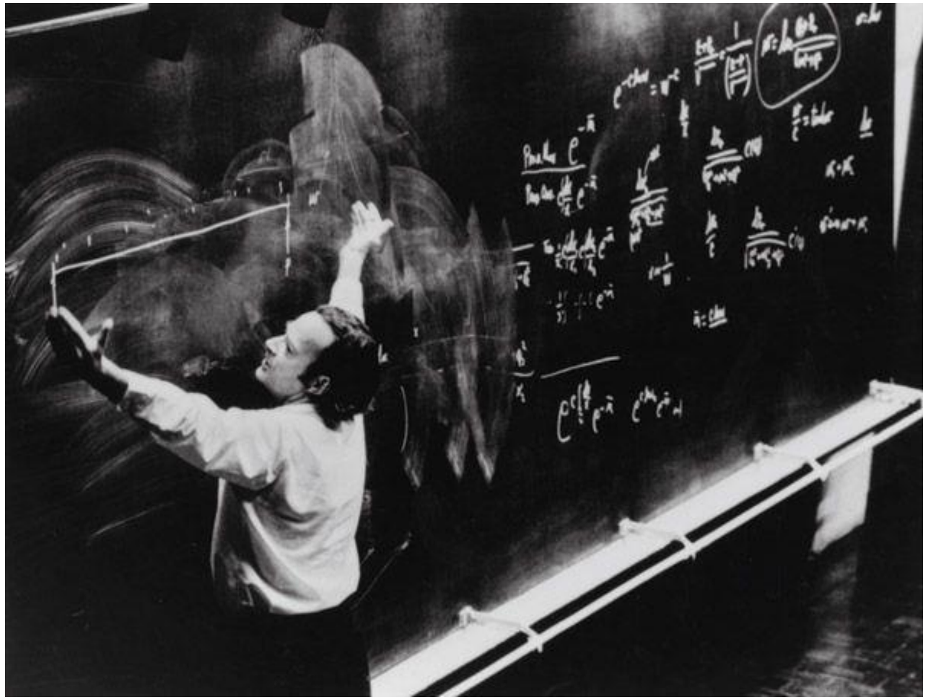
- **However:** actually want to encrypt messages of *arbitrary* length!
  - Splitting the message into multiple 128 bit blocks (like above) is **not secure!**
  - A **mode-of-operation** is needed (covered later in the course)

- 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

Pseudorandom functions (PRFs) and permutations (PRP)

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

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

- 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 $F_K : \{0,1\}^{in} \to \{0,1\}^{out}$ de

PRP = block cipher

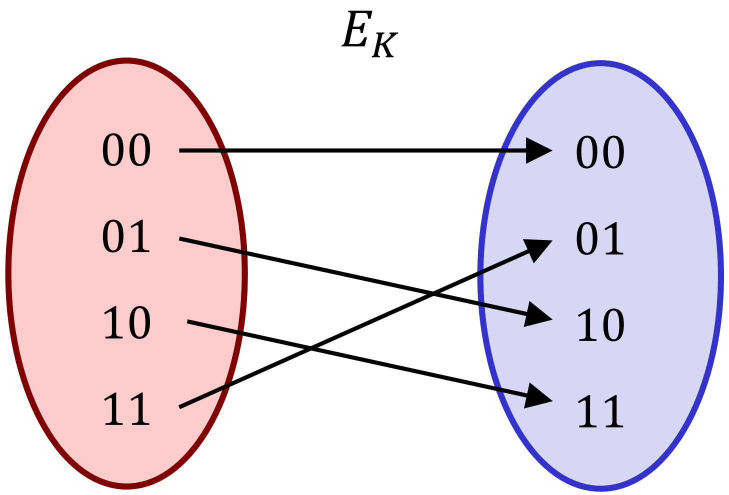
note: all PRPs are PRFs
(but not all PRFs are PRPs!)

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

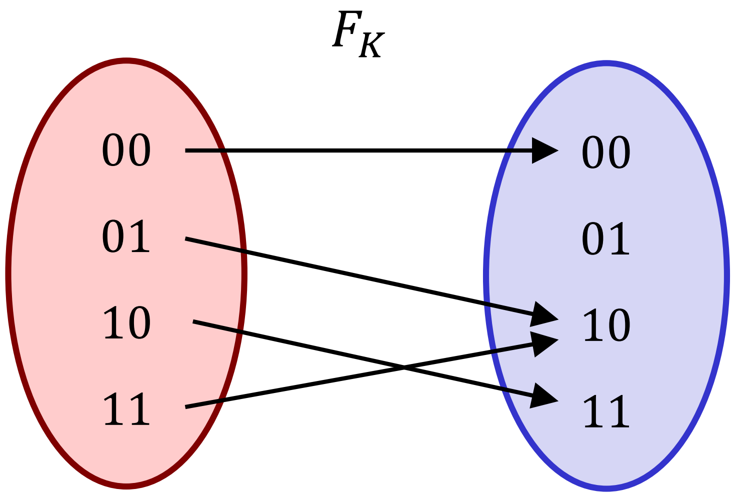
$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

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

Permutations vs. functions

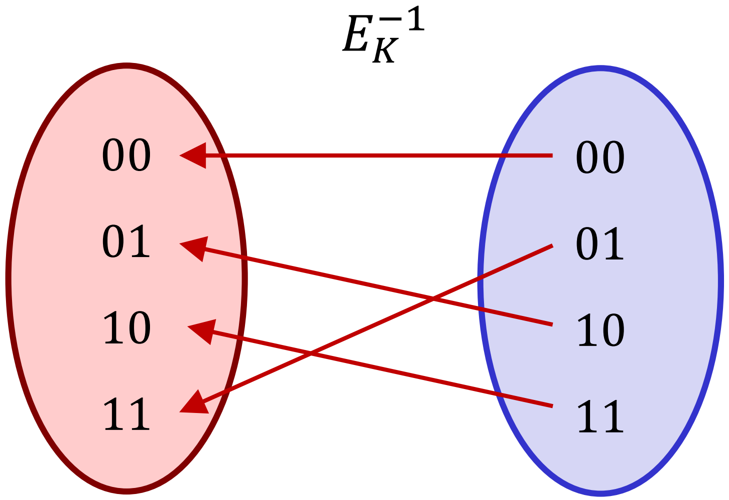


Permutation

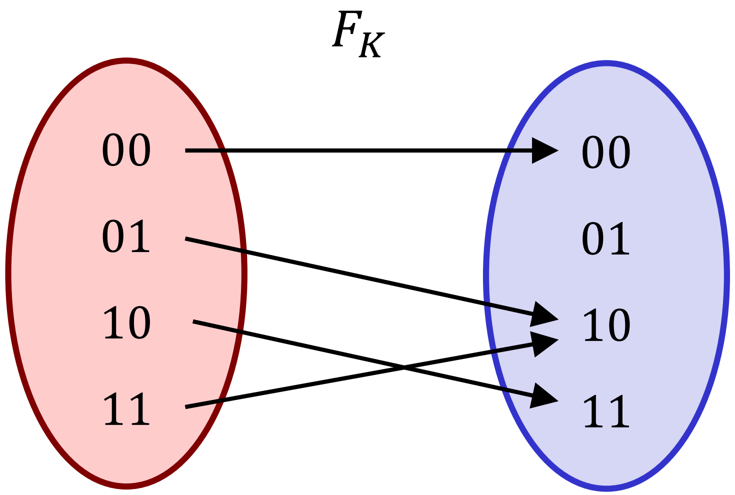


Not a permutation

Permutations vs. functions



Permutation



Not a permutation

Important:

$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is **not** a permutation

$E_K : \{0,1\}^n \to \{0,1\}^n$ **is** a permutation

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 $E_K(X)$
  - ~~**P6:** Should be hard to detect *repetitions* among $X_1, X_2, \ldots$ from $E_K(X_1), E_K(X_2), \ldots$~~
  - **P7:** …

Not good enough!

Impossible!

Random functions

$\tilde{F} : \{0,1\}^{in} \to \{0,1\}^{out}$

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

2^{in}

out

Random functions

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

X	$\widetilde{F}(X)$
$\vdots$	$\vdots$

$T \leftarrow [\,]$

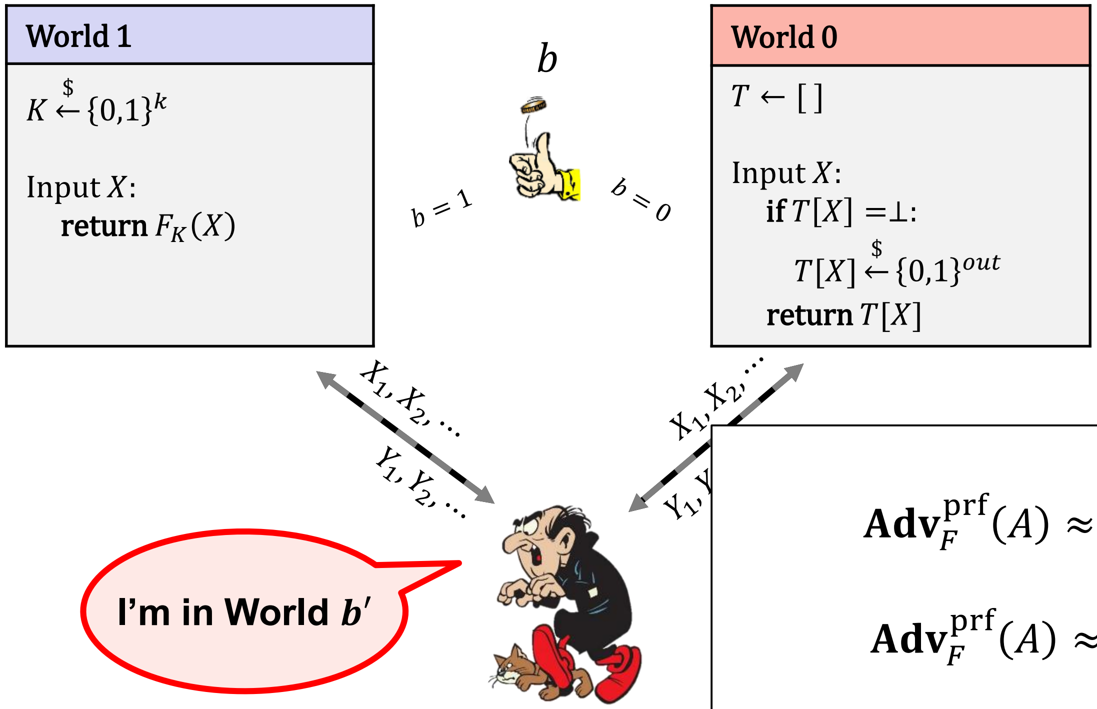
<u>$\widetilde{F}(X)$:</u>

if $T[X] = \perp$**:**

$\quad T[X] \xleftarrow{\$} \{0,1\}^{out}$

return $T[X]$

PRF – security; formal definition



$\mathbf{Adv}_F^{\mathrm{prf}}(A) \approx 1/2$ = adversary is doing well

$\mathbf{Adv}_F^{\mathrm{prf}}(A) \approx 0$ = adversary is doing poorly

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

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

PRF – security; formal definition

World 1

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

Input X:
    return $F_K(X)$

b

World 0

$T \leftarrow [\,]$

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

Intuitive idea: F is a **secure PRF** if $\mathbf{Adv}_F^{\mathrm{prf}}(A)$ is "*small*" for all "*reasonable*" A

I'm in World b'

$\mathbf{Adv}_F^{\mathrm{prf}}(A) \approx 1$ = adversary is doing well

$\mathbf{Adv}_F^{\mathrm{prf}}(A) \approx 0$ = adversary is doing poorly

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:

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

for all adversaries A that run in time $\leq t$

  - **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. Query $X = 0^n$

World 1

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

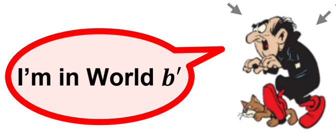
Input X:
  return $K \oplus X$

b

World 0

$T \leftarrow [\,]$

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



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

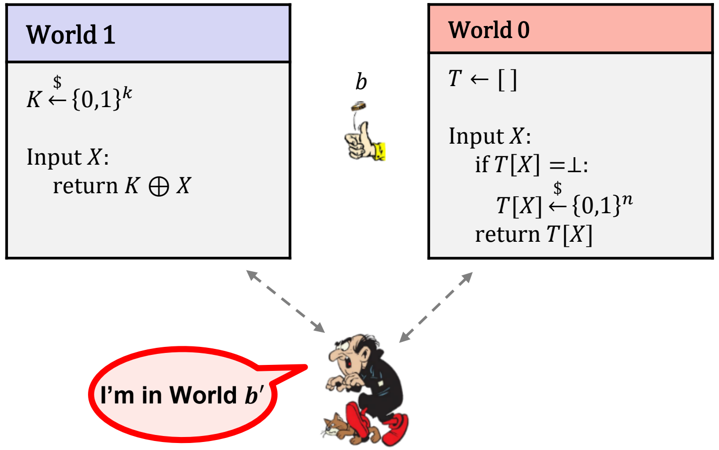
$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = |2 \cdot \Pr[b' = b] - 1|$$

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

Example

- Define $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ by $F(K, X) = K \oplus X$

- **Claim:** F is not a secure PRF



A

1. Query $X = 0^n$ // receive back $Y = K \oplus 0^n = K$ or $Y \xleftarrow{\$} \{0,1\}^n$
2. Query $X' = 1^n$ // receive back $Y' = K \oplus 1^n$ or $Y' \xleftarrow{\$} \{0,1\}^n$
3. Output $b' = \begin{cases} 1, & \text{if } Y \oplus X' = Y' \\ 0, & \text{otherwise} \end{cases}$

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

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

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

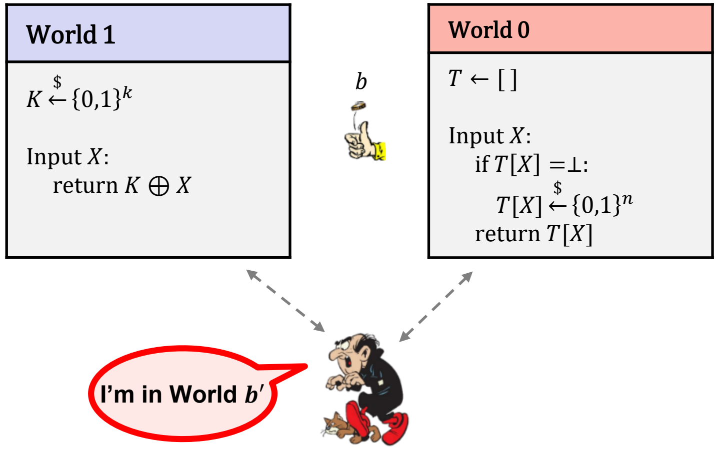
Example

- 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. Query $X = 0^n$ // receive back $Y = K \oplus 0^n = K$ or $Y \xleftarrow{\$} \{0,1\}^n$
2. Query $X' = 1^n$ // receive back $Y' = K \oplus 1^n$ or $Y' \xleftarrow{\$} \{0,1\}^n$
3. Output $b' = \begin{cases} 1, & \text{if } Y \oplus X' = Y' \\ 0, & \text{otherwise} \end{cases}$



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

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

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

Example

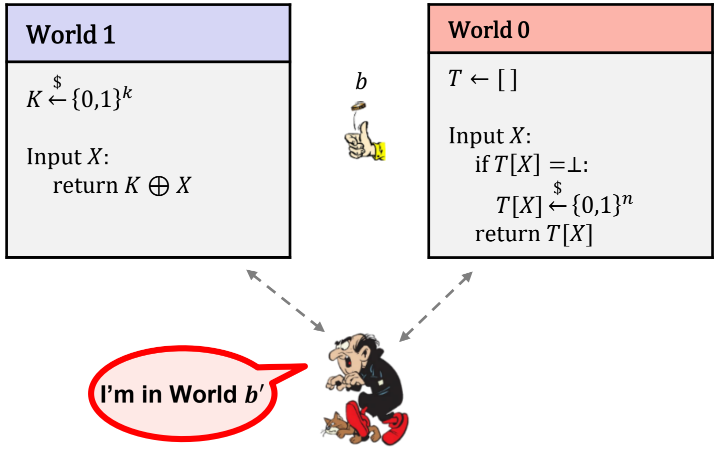
- Define $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $F(K, X) = K \oplus X$

- **Claim:** F is not a secure PRF

A		
1.	Query $X = 0^n$	// receive back $Y = K \oplus 0^n = K$ or $Y \xleftarrow{\$} \{0,1\}^n$
2.	Query $X' = 1^n$	// receive back $Y' = K \oplus 1^n$ or $Y' \xleftarrow{\$} \{0,1\}^n$
3.	Output $b' = \begin{cases} 1, & \text{if } Y \oplus X' = Y' \\ 0, & \text{otherwise} \end{cases}$	

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

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



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

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

Why is the PRF definition good?

- **P1:** Should be hard to obtain K from $F_K(X)$ for secret K

A

1. Query X

(X_3) …

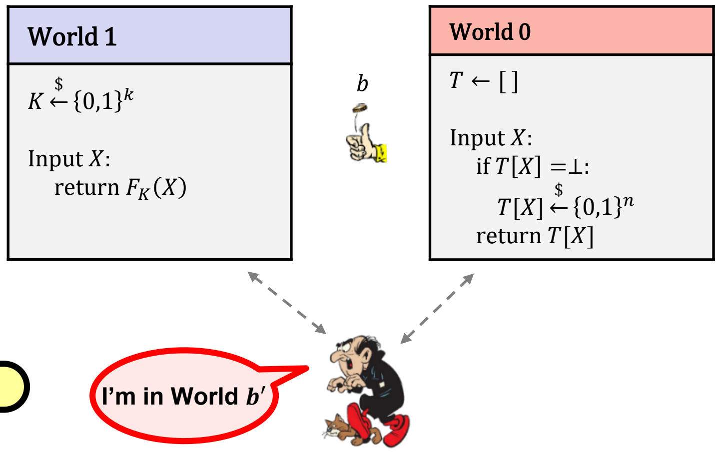
$), F_K(X_3)$ …

Logic 101

$A \Rightarrow B$

is *equivalent to:*

not $A \Leftarrow$ 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 $\Longrightarrow F$ has properties P1 – P5, P7, …

F is ***not*** PRF secure $\Longleftarrow F$ 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~~ PRP – security

World 1

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

Input X:
    return $F_K(X)$

b

World 0

$T \leftarrow [\,]$

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

I'm in World b'

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

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

PRP security $\Longrightarrow$ PRF security

Theorem: PRP security $\Longrightarrow$ PRF security. For all A making at most q oracle queries:

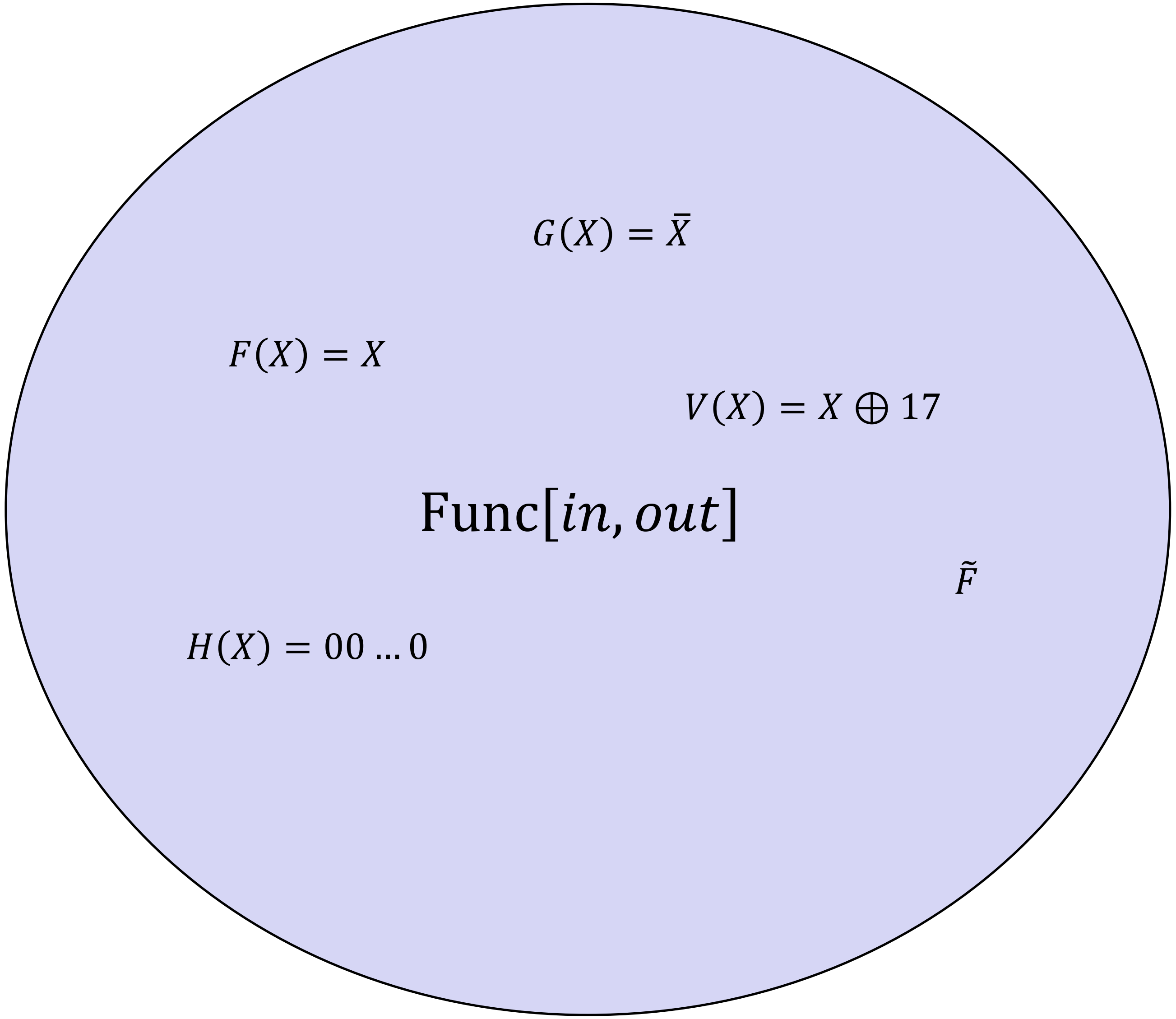
$$\mathbf{Adv}_E^{\mathrm{prf}}(A) \leq \mathbf{Adv}_E^{\mathrm{prp}}(A) + \frac{2q^2}{2^n}$$

Proof sketch:

$$\begin{aligned}
\mathbf{Adv}_E^{\mathrm{prf}}(A) = 2 \cdot \Pr[b' = b] - 1 &= 2 \cdot \left(\Pr\left[b' = b \wedge \overline{\mathrm{coll}}\right] + \Pr[b' = b \wedge \mathrm{coll}] \right) - 1 \\
&\leq 2 \cdot \left(\Pr\left[b' = b \wedge \overline{\mathrm{coll}}\right] + \Pr[\mathrm{coll}] \right) - 1 \\
&\leq 2 \cdot \Pr\left[b' = b \wedge \overline{\mathrm{coll}}\right] + 2 \cdot \frac{q^2}{2^n} - 1 \\
&= \mathbf{Adv}_E^{\mathrm{prp}}(A) + \frac{2q^2}{2^n}
\end{aligned}$$

PRF – security; equivalent view

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



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

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

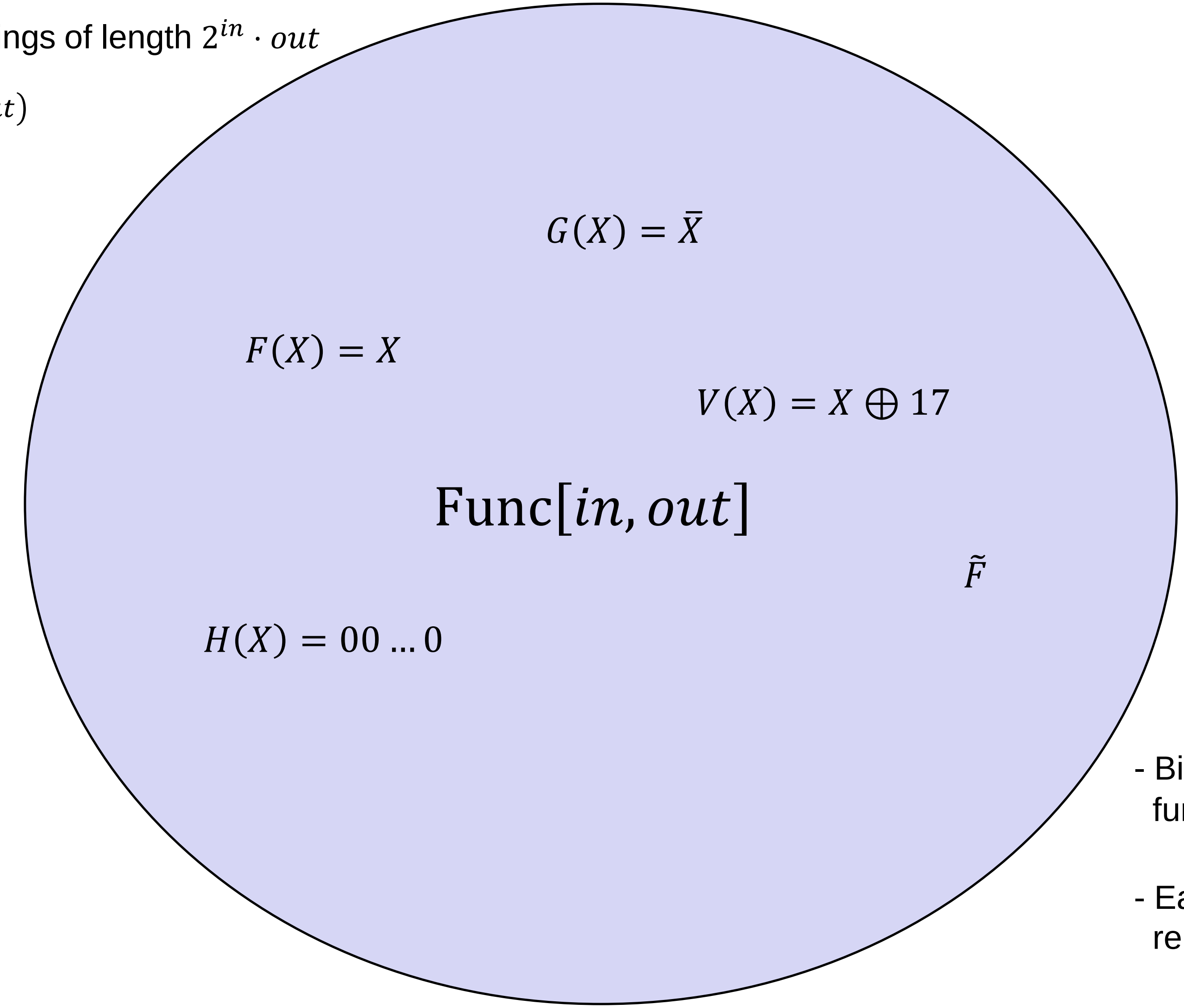
$= 2^{(2^{in} \cdot out)}$

Example:

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

$= 2^{16}$

$= 65536$



$\tilde{F}(X)$
$101 \dots 111$
$001 \dots 001$
$111 \dots 100$
$101 \dots 000$
$\vdots$
$001 \dots 001$

2^{in} rows; width out

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

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

PRF – security; equivalent view

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

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

Example:

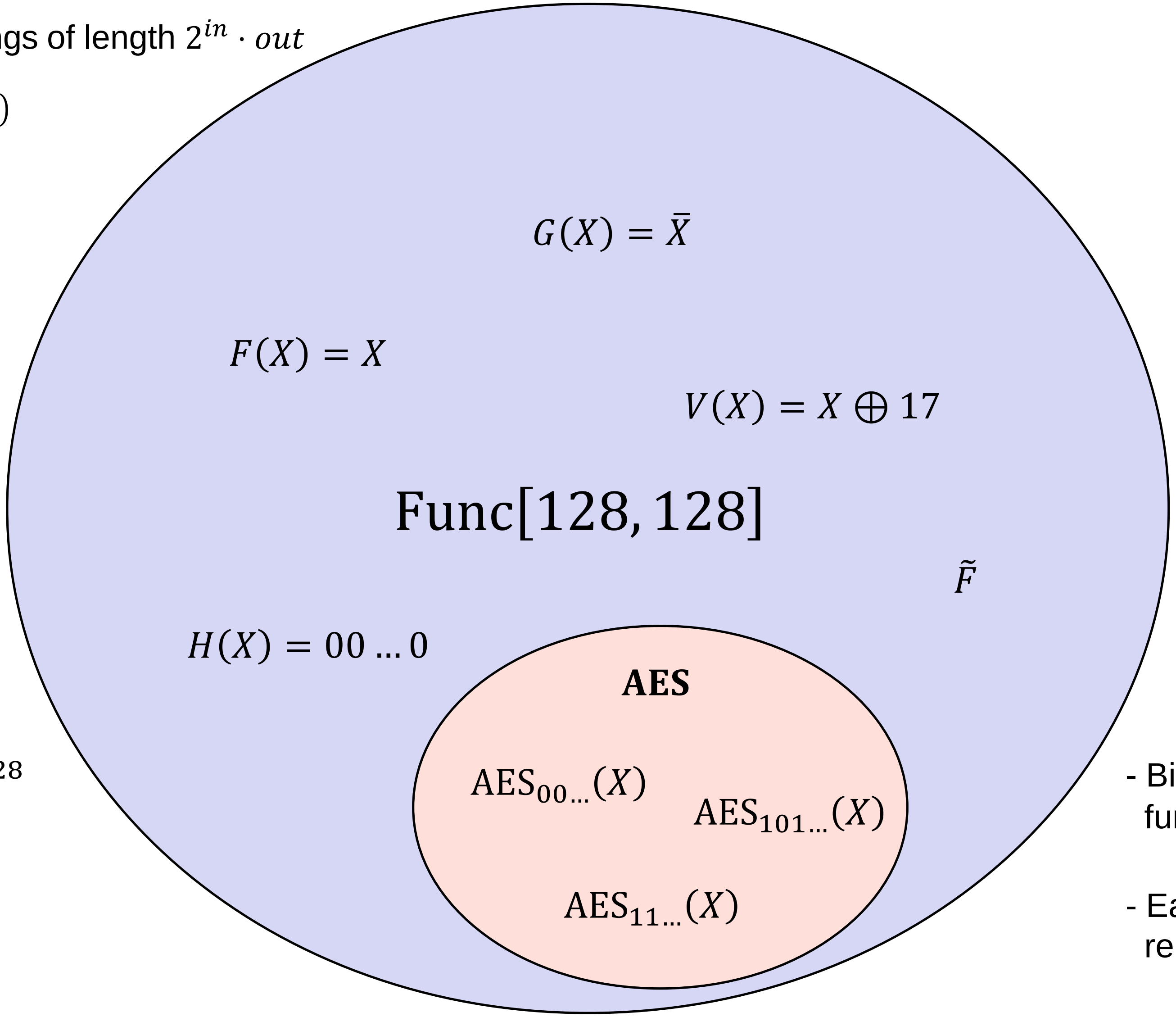
$|\mathrm{Func}[3,2]| = 2^{2^3 \cdot 2}$

$= 2^{16}$

$= 65536$

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

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



$\tilde{F}(X)$
$101 \ldots 111$
$001 \ldots 001$
$111 \ldots 100$
$101 \ldots 000$
$\vdots$
$001 \ldots 001$

2^{in}

out

- Bits needed to specify *one* function: $2^{in} \cdot out$
- Each *unique* bitstring of length $2^{in} \cdot out$ represents a *unique* function

PRF – security; equivalent view

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

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

Example:

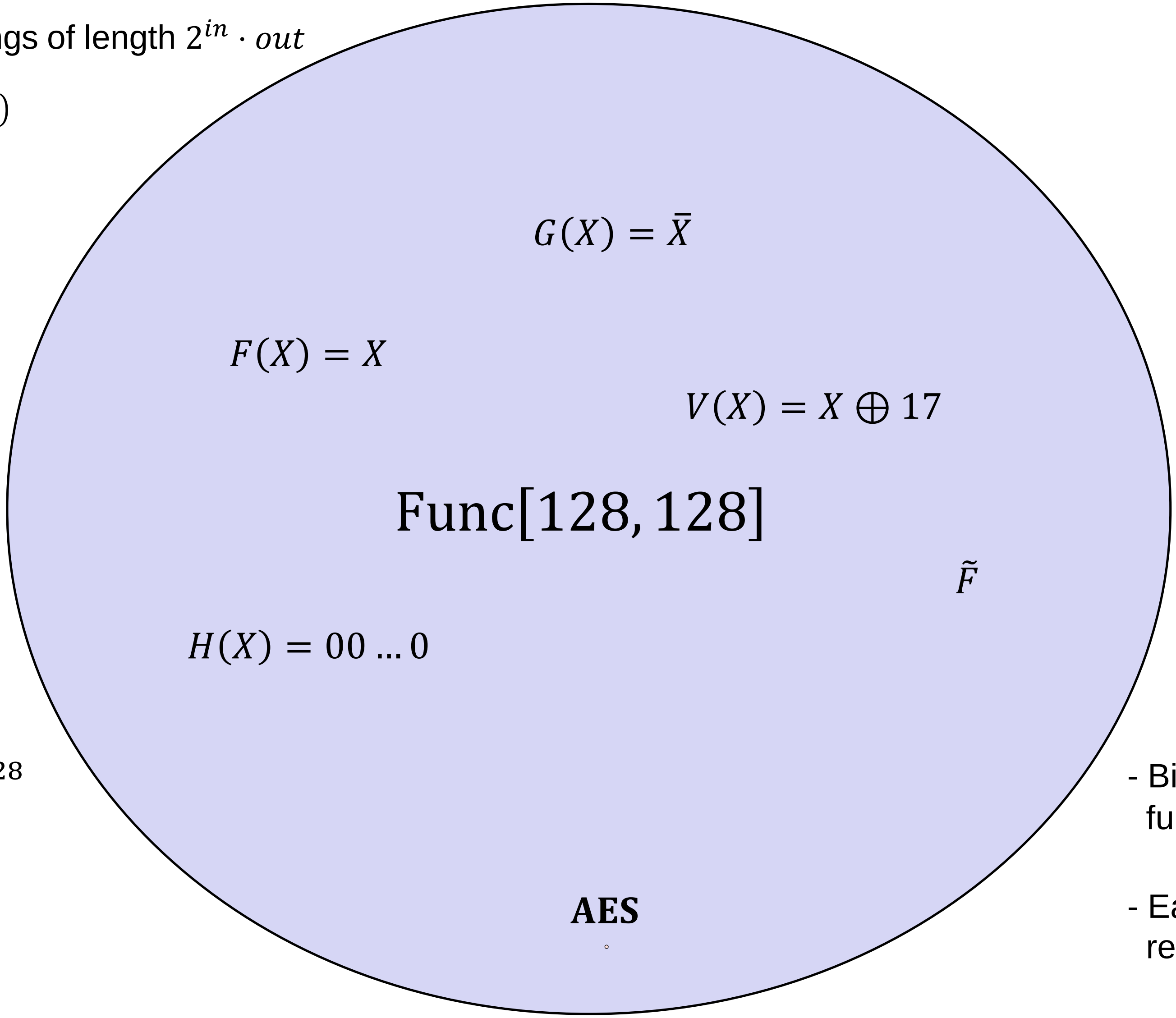
$|\text{Func}[3,2]| = 2^{2^3 \cdot 2}$

$= 2^{16}$

$= 65536$

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

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



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

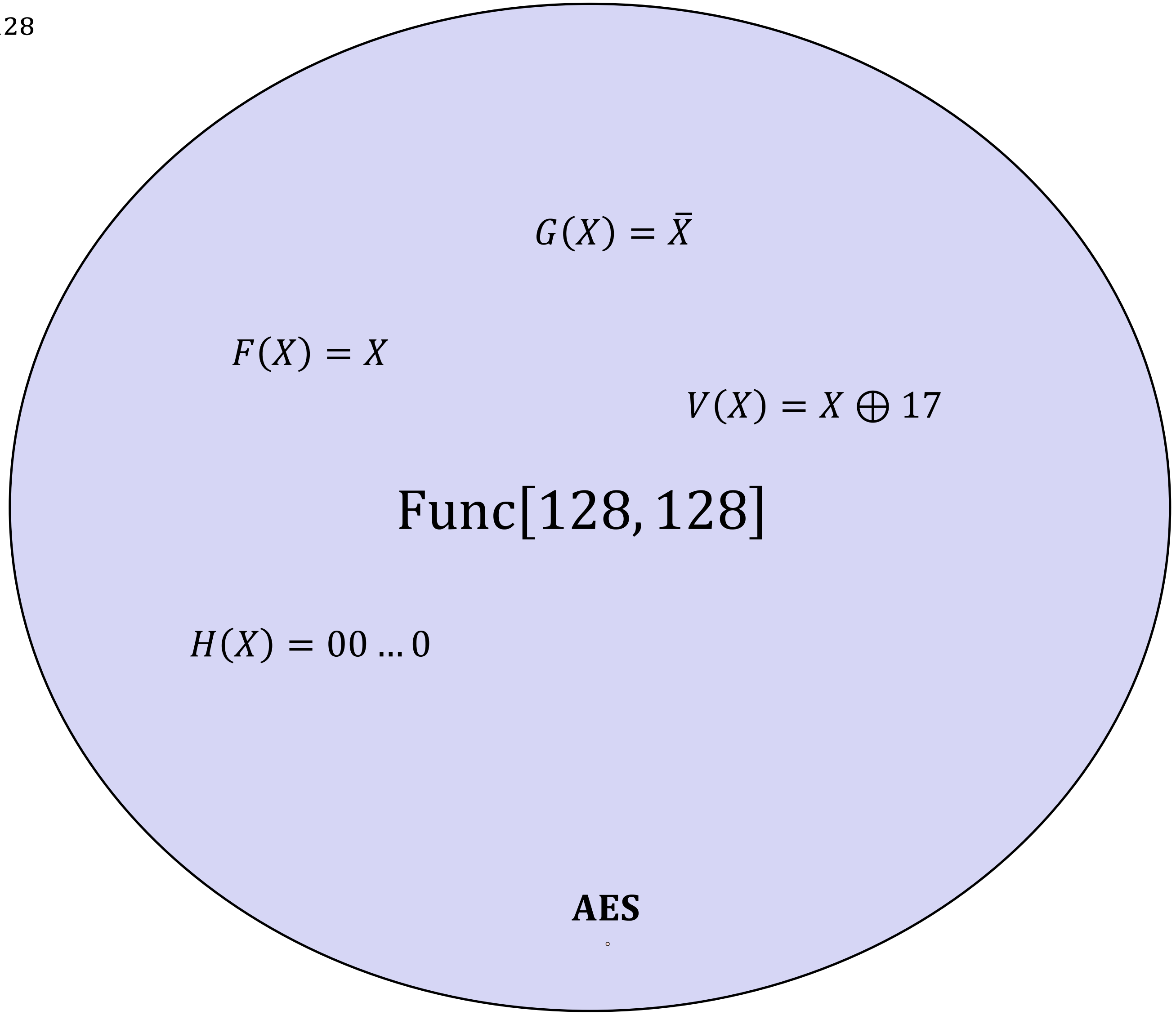
2^{in}

out

- Bits needed to specify *one* function: $2^{in} \cdot out$
- Each *unique* bitstring of length $2^{in} \cdot out$ represents a *unique* function

Block ciphers – security

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



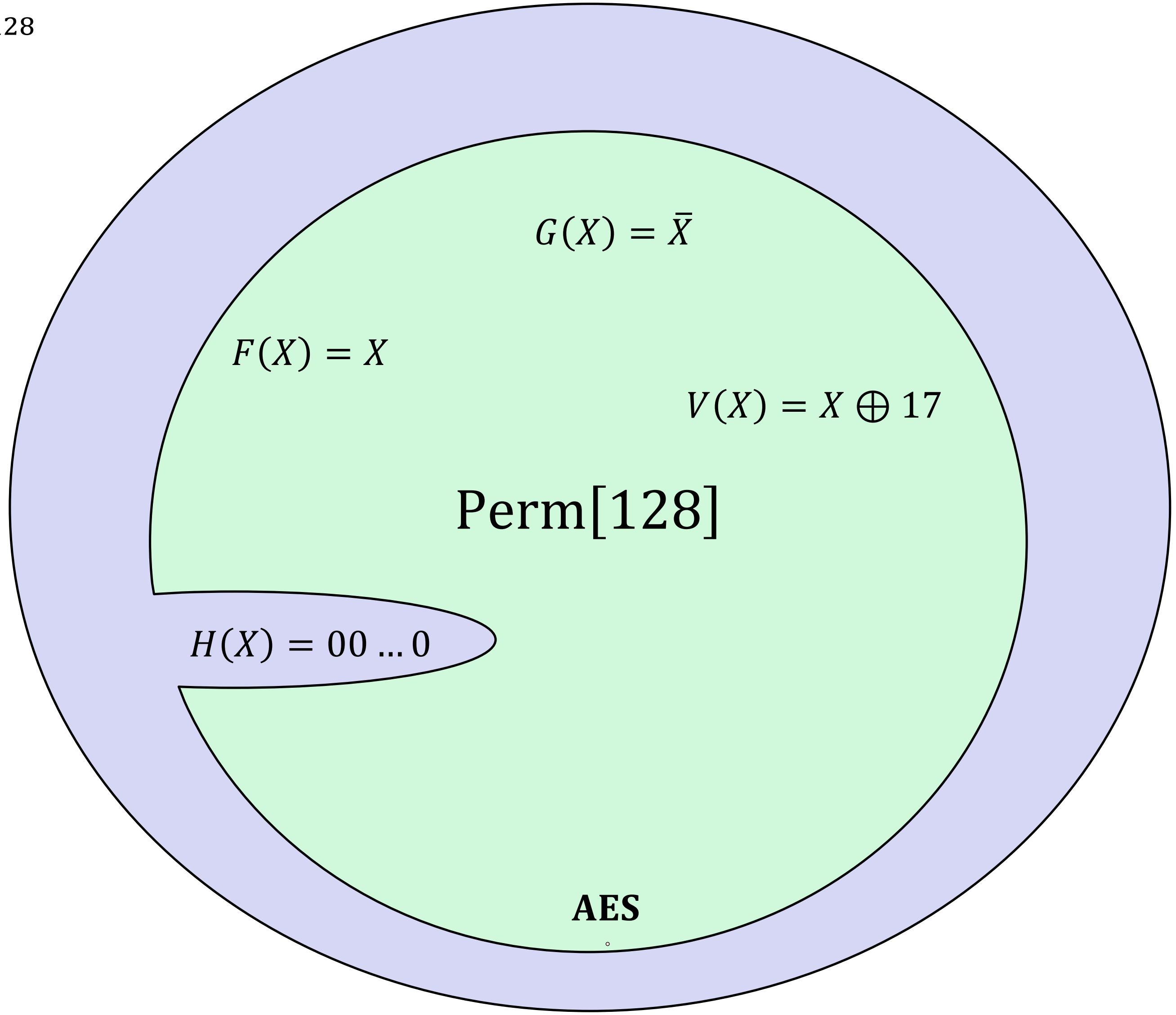
$|\mathrm{AES}| = 2^{128}$

Block ciphers – security

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

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

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



PRF – security

World 1

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

Input X:
    return $F_K(X)$

b

World 0

$T \leftarrow [\,]$

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

I'm in World b'

PRF – security; equivalent view

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

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

World 1

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

Input X:
  return $F_K(X)$

b

World 0

$\tilde{F} \xleftarrow{\$} \mathrm{Func}[in, out]$

Input X:
  return $\tilde{F}(X)$

I'm in World 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; equivalent view

$\mathbf{Exp}_F^{\cancel{\mathrm{prp}}\,\mathrm{prp}}(A)$

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

$F_0 \xleftarrow{\$} \cancel{\mathrm{Func}[in, out]}$ $\mathrm{Perm}[n]$

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

$F_1 \leftarrow F_K$

$b' \leftarrow A^{F_b(\cdot)}$

return $b' \stackrel{?}{=} b$

World 1

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

Input X:
  return $F_K(X)$

b

World 0

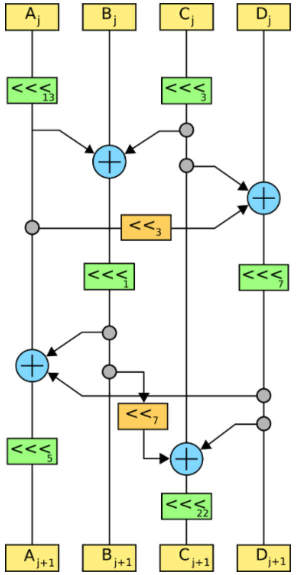
$\tilde{F} \xleftarrow{\$} \cancel{\mathrm{Func}[in, out]}$ $\mathrm{Perm}[n]$

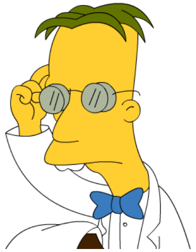
Input X:
  return $\tilde{F}(X)$

I'm in World b'

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

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





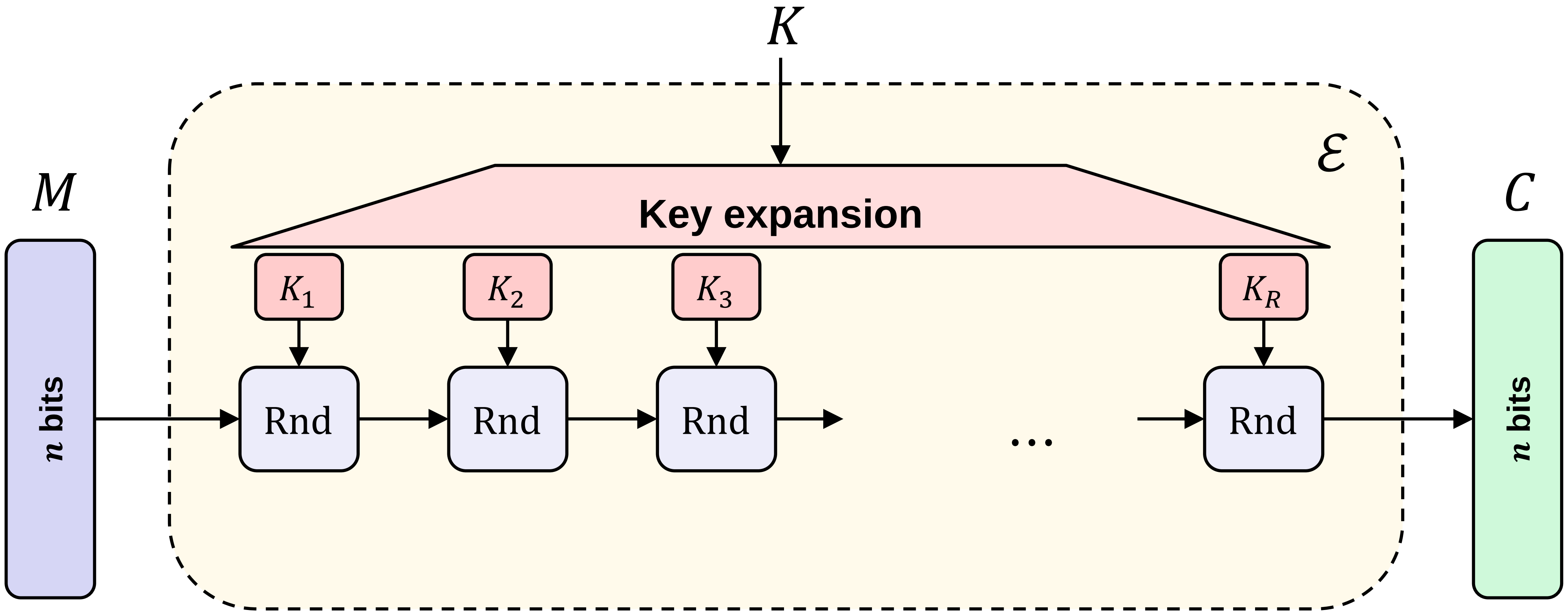
Constructing block ciphers

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**

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

Advanced Encryption Standard

AES

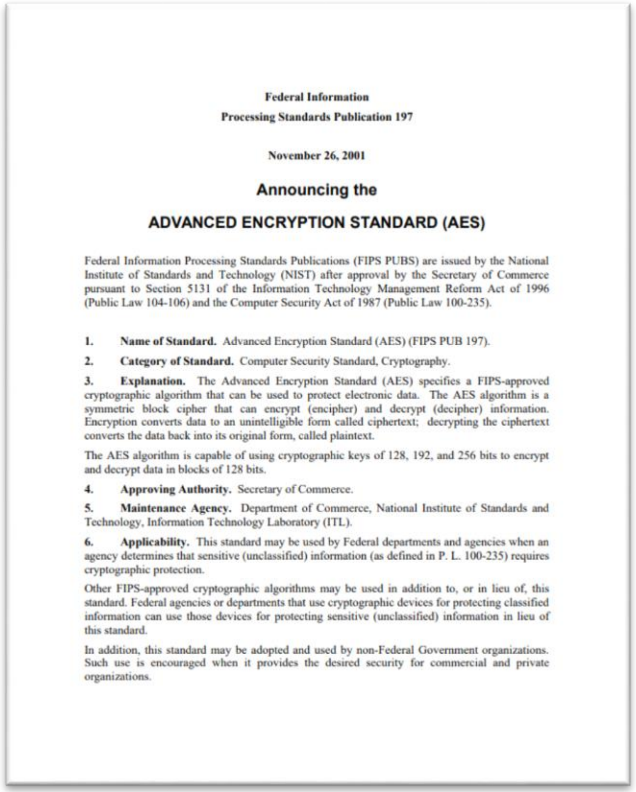
- History
  - Date Encryption Standard (DES) used since the 70's
  - Key length too short (56 bits)
  - Block length too short (64 bits)
  - Too slow in software
  - 1997: NIST hosts a competition to find a new block cipher
  - 2001: Rijndael is announced the winner → renamed AES

- If you remember nothing else:

$$\text{AES-128} : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$$

$$\text{AES-192} : \{0,1\}^{192} \times \{0,1\}^{128} \to \{0,1\}^{128}$$

$$\text{AES-256} : \{0,1\}^{256} \times \{0,1\}^{128} \to \{0,1\}^{128}$$



Federal Information
Processing Standards Publication 197

November 26, 2001

Announcing the

ADVANCED ENCRYPTION STANDARD (AES)

Federal Information Processing Standards Publications (FIPS PUBS) are issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce pursuant to Section 5131 of the Information Technology Management Reform Act of 1996 (Public Law 104-106) and the Computer Security Act of 1987 (Public Law 100-235).

1. **Name of Standard.** Advanced Encryption Standard (AES) (FIPS PUB 197).

2. **Category of Standard.** Computer Security Standard, Cryptography.

3. **Explanation.** The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext.

The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits.

4. **Approving Authority.** Secretary of Commerce.

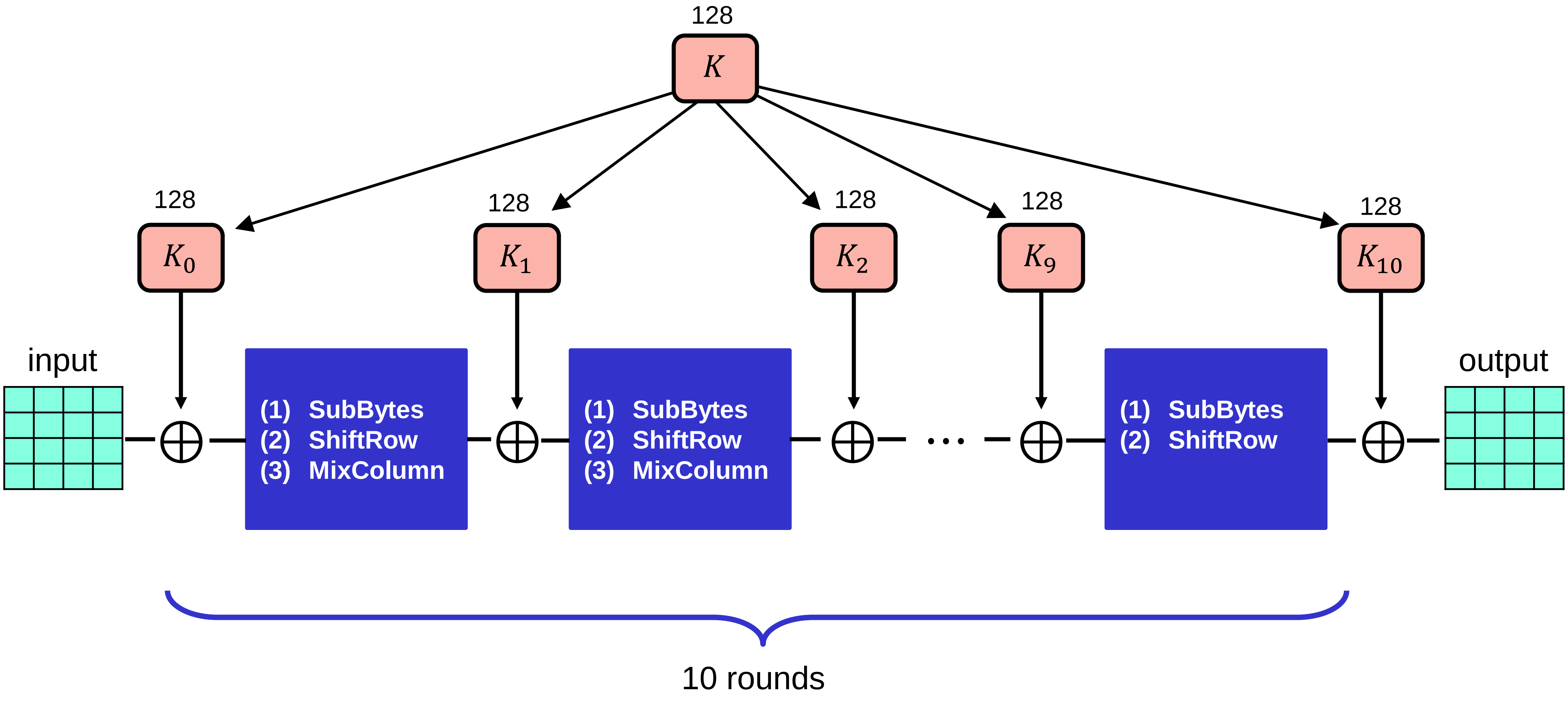
5. **Maintenance Agency.** Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL).

6. **Applicability.** This standard may be used by Federal departments and agencies when an agency determines that sensitive (unclassified) information (as defined in P. L. 100-235) requires cryptographic protection.

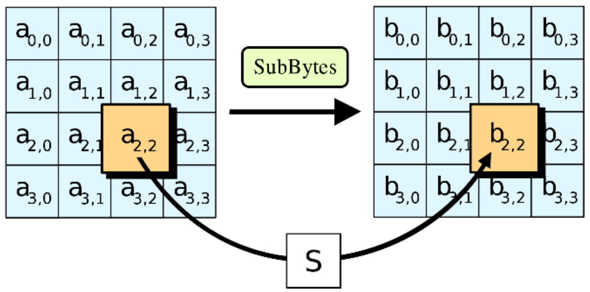
Other FIPS-approved cryptographic algorithms may be used in addition to, or in lieu of, this standard. Federal agencies or departments that use cryptographic devices for protecting classified information can use those devices for protecting sensitive (unclassified) information in lieu of this standard.

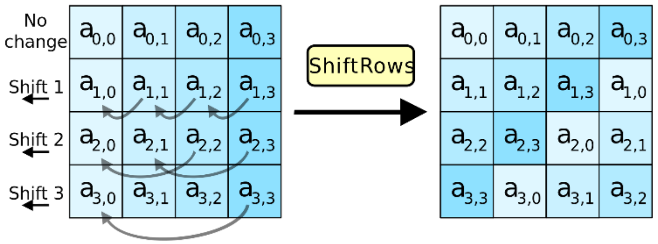
In addition, this standard may be adopted and used by non-Federal Government organizations. Such use is encouraged when it provides the desired security for commercial and private organizations.

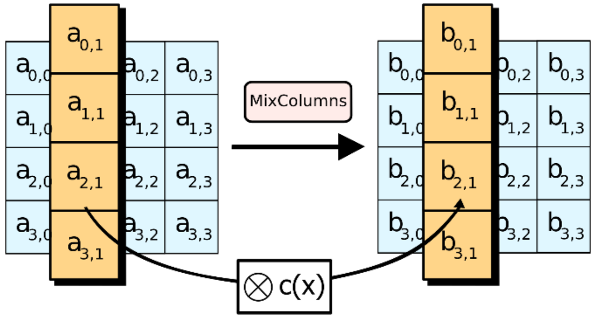
AES-128


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

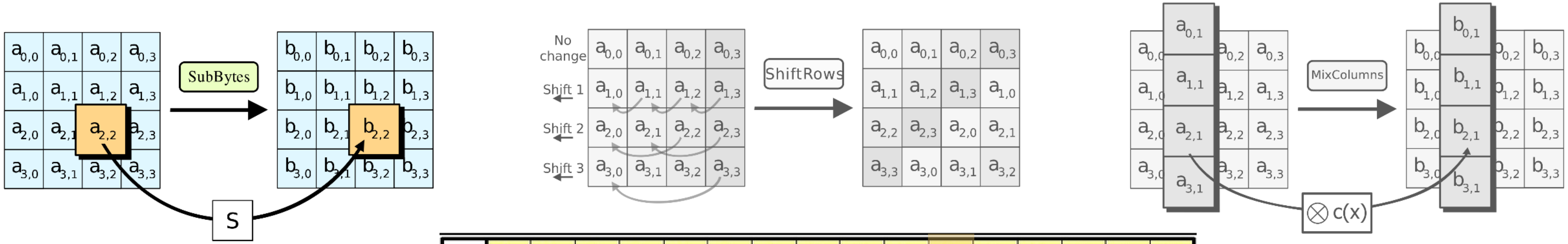






(1) SubBytes
(2) ShiftRow
(3) MixColumn

AES round function - SubBytes

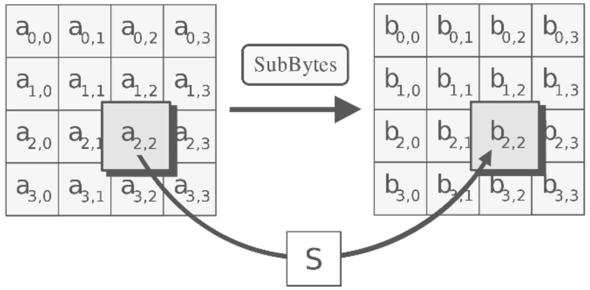


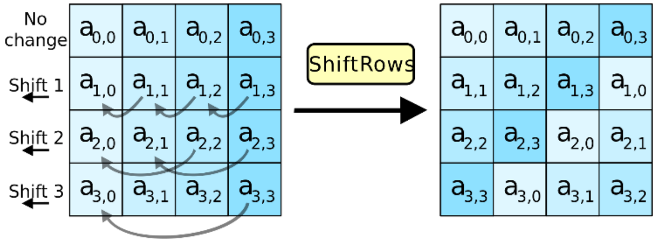
$a_{2,2} = \text{8A}$

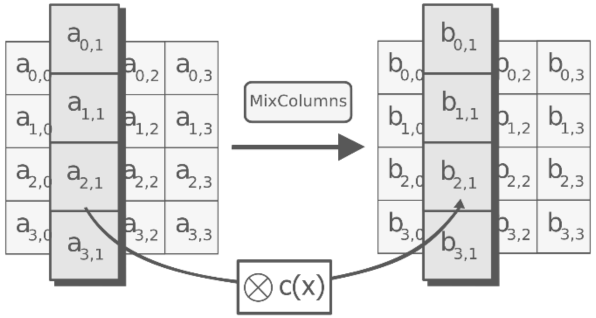
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	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

$b_{2,2} = \text{7E}$

AES round function - ShiftRows

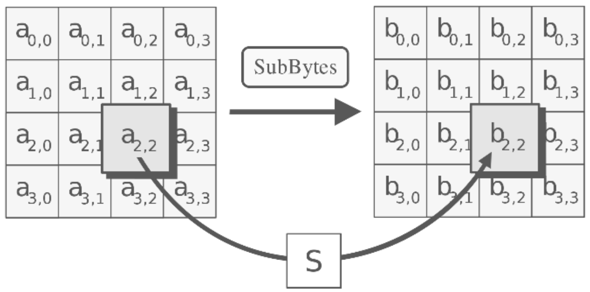


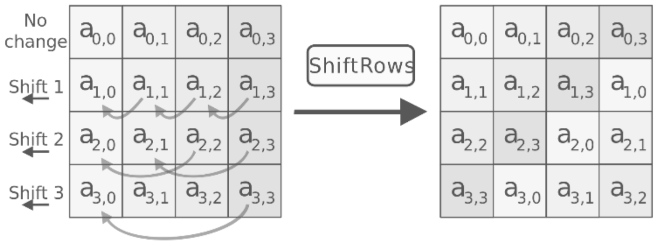


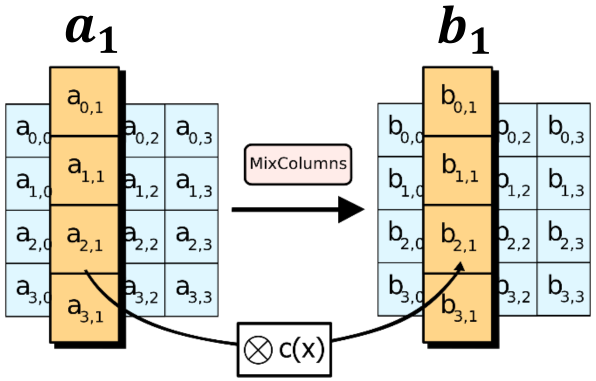


(1) **SubBytes**
(2) **ShiftRow**
(3) **MixColumn**

AES round function - MixColumns







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

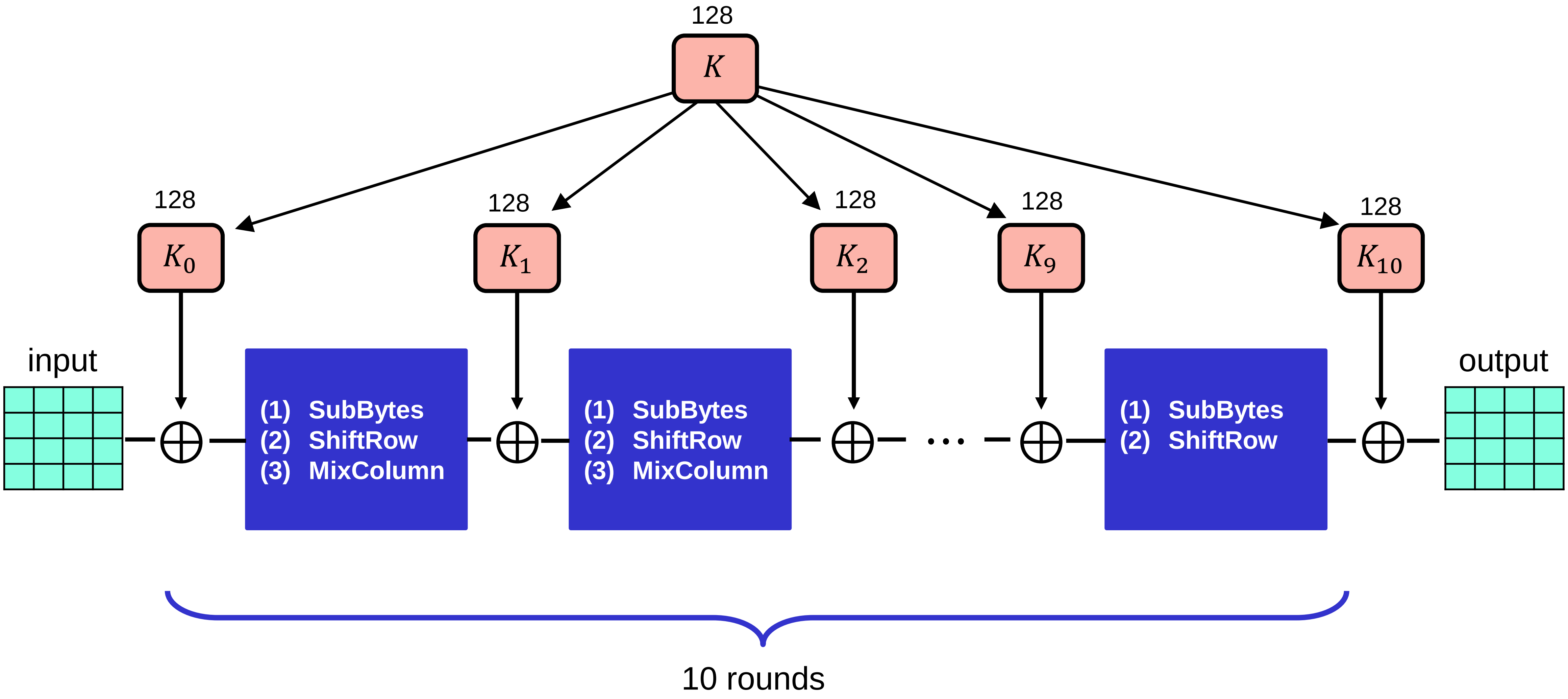
$$M\boldsymbol{a_1} = \boldsymbol{b_1}$$

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

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

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

AES-128



128
K
128
K_0
128
K_1
128
K_2
128
K_9
128
K_{10}
input
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
(3) MixColumn
···
(1) SubBytes
(2) ShiftRow
output
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	$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}$

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

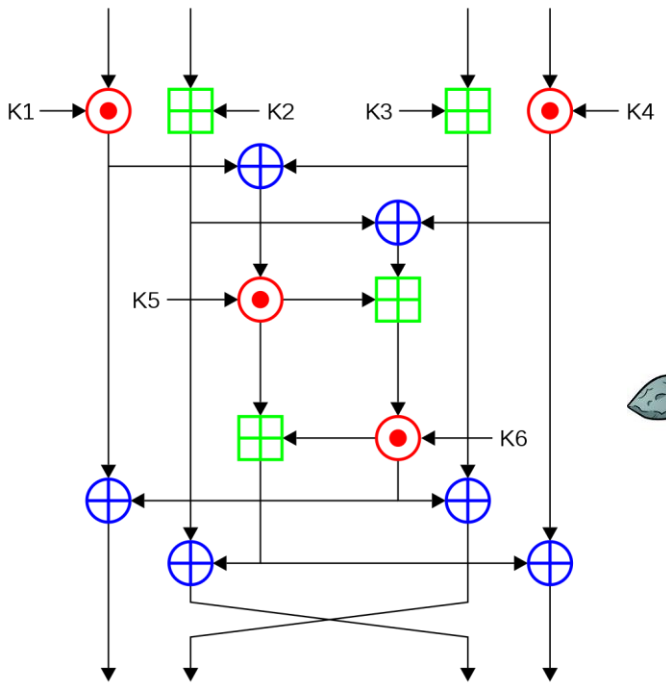
$T_0[x] = \left(\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} \cdot S[x] \right)$	$T_1[x] = \left(\begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_2[x] = \left(\begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} \cdot S[x] \right)$	$T_3[x] = \left(\begin{bmatrix} 01 \\ 01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

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) – not encryption schemes!
- Correct abstraction: block ciphers = PRPs
- Right security notion for PRFs/PRPs:

  indistinguishability from random function/permutation
- Concrete block cipher design: DES, AES, …