Lecture 2 – Block ciphers, PRFs, PRPs, AES

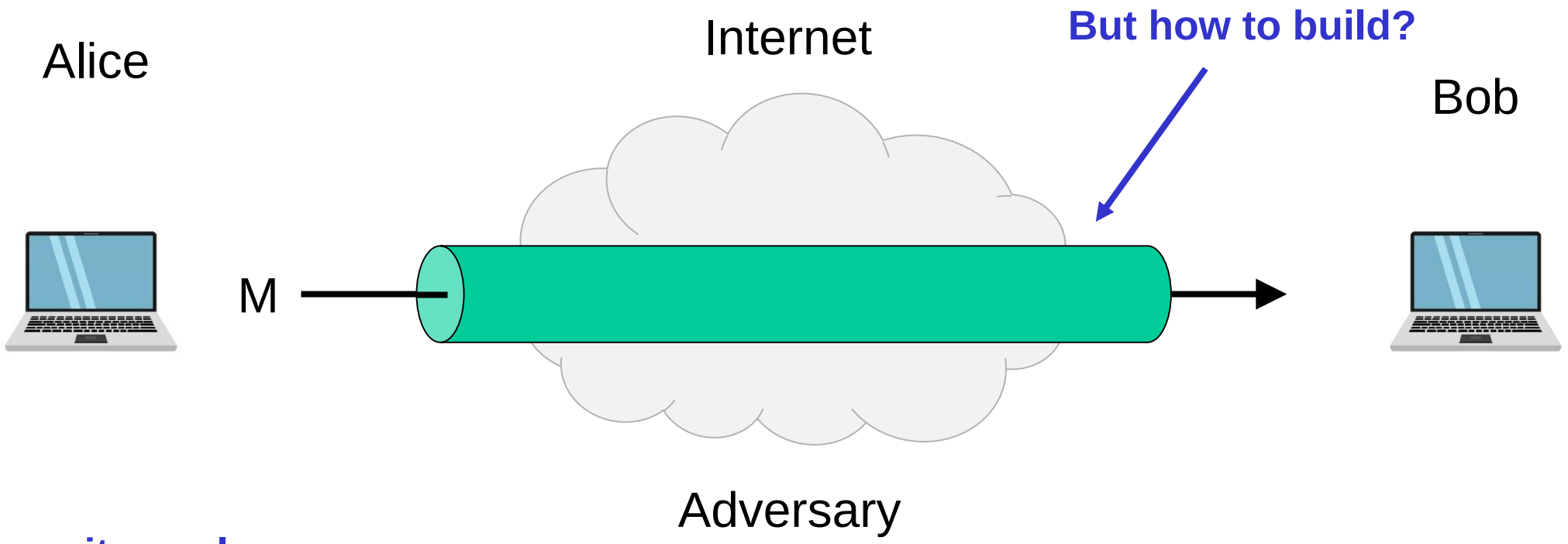
TEK4500

26.08.2026

Håkon Jacobsen

hakon.jacobsen@its.uio.no

Ideal solution: secure channels



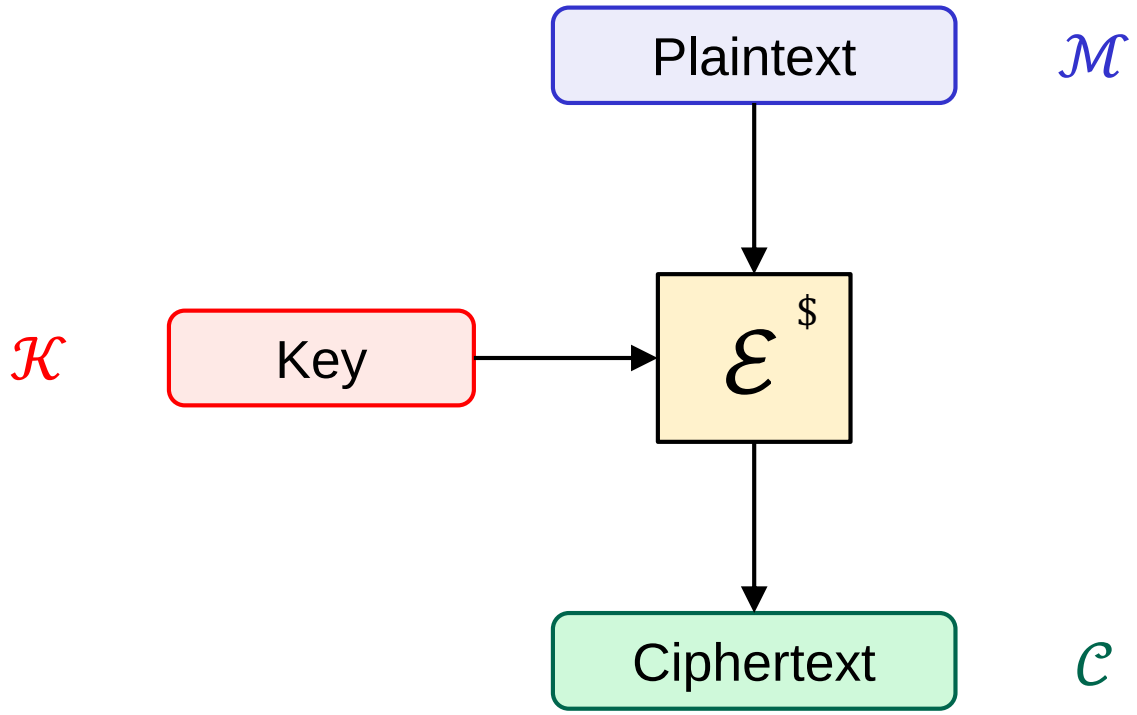
Security goals:

- **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 ✓

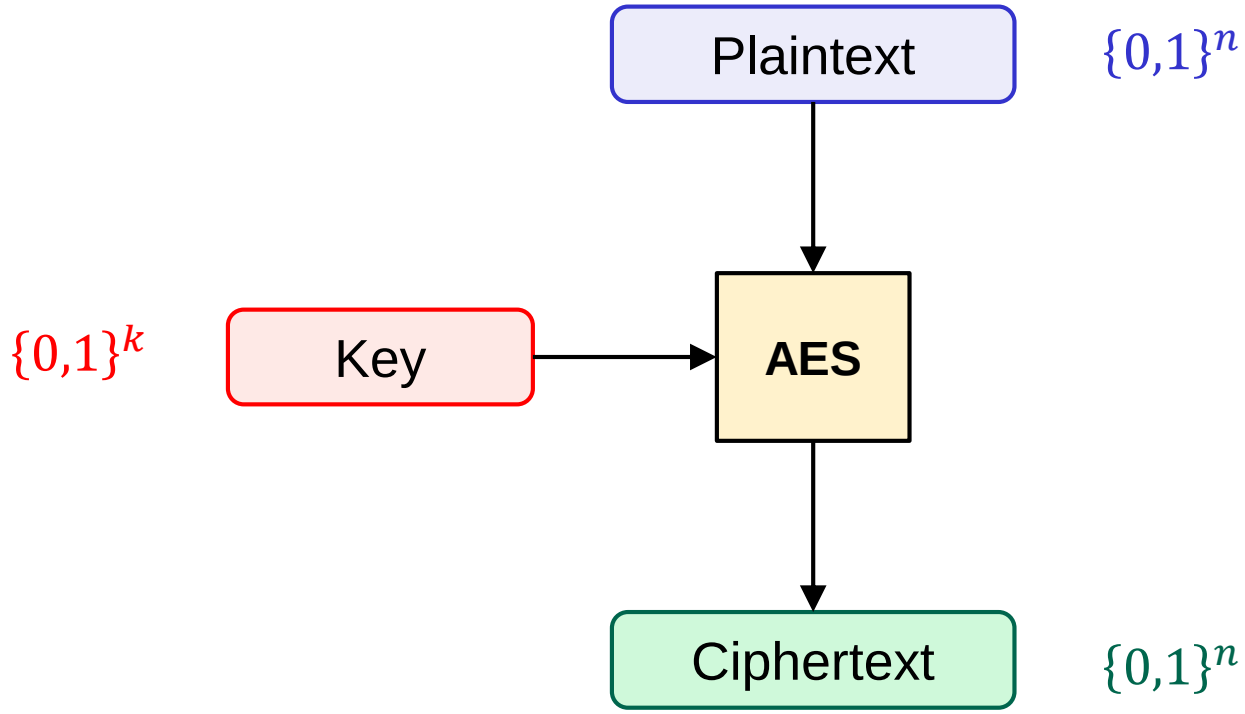
Basic goals of cryptography

	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

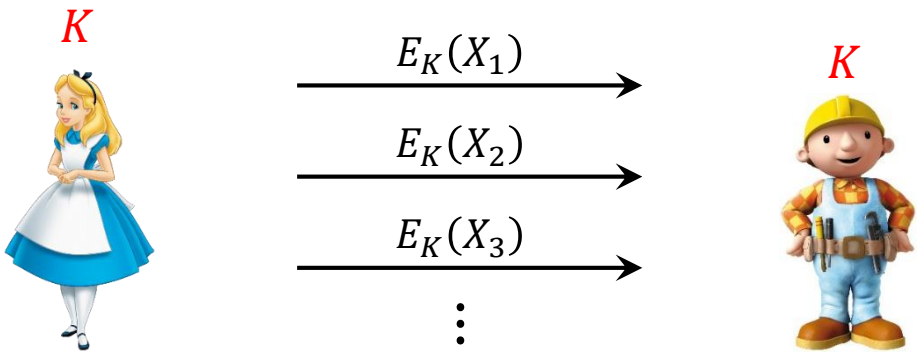


Block ciphers



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

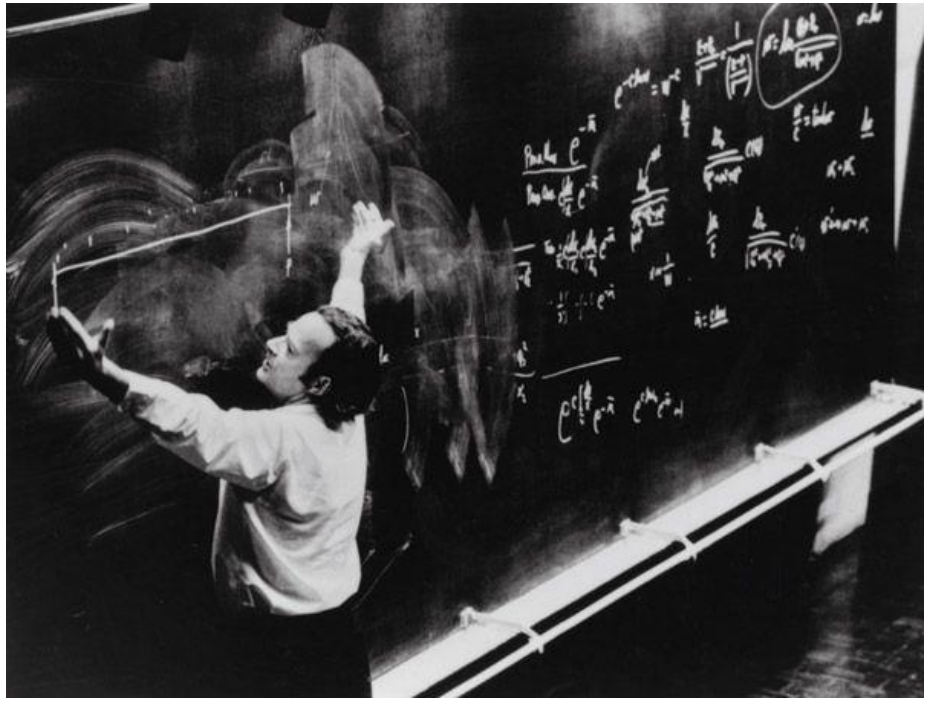
Block cipher applications



- Encryption of messages of exactly 128 bits (block length)

- 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)

- Most useful viewpoint: block ciphers are **not** encryption schemes!
  - A block cipher is a **primitive** used to construct other things
  - Goal of primitive: produce random-looking output



Pseduorandom functions (PRFs)

Pseudorandom functions (PRFs) – syntax

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

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

- k – key length
- in – input length
- out – output length

- Useful viewpoint: a PRF is a *family* of functions:
  - For each choice of $K \in \{0,1\}^k$ we get a "new" function $F_K : \{0,1\}^{in} \to \{0,1\}^{out}$ defined by $F_K(\cdot) \stackrel{\text{def}}{=} F(K, \cdot)$

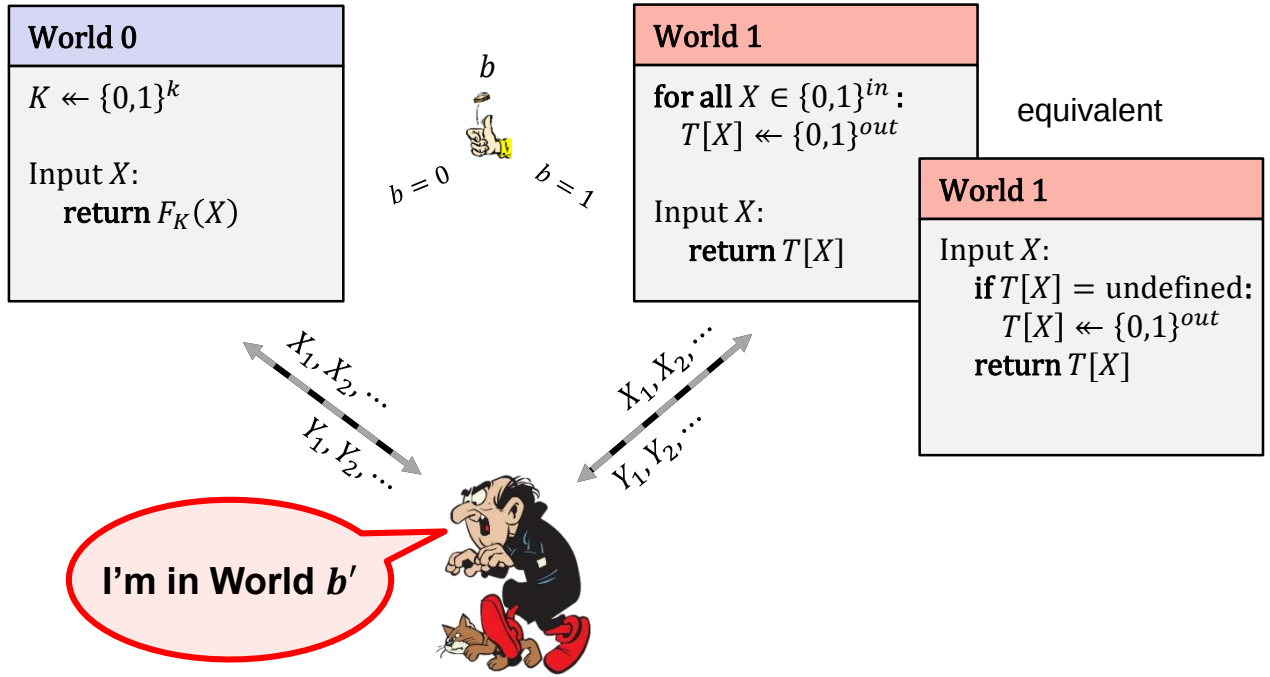
PRF – security

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

1. $b \leftarrow \{0,1\}$
2. $K \leftarrow \{0,1\}^k$
3. $T \leftarrow$ empty dictionary
4. **for all** $X \in \{0,1\}^{in}$
5. $\quad T[X] \leftarrow \{0,1\}^{out}$
6.
7. $b' \leftarrow A^{\mathcal{O}(\cdot)}$
8. **return** $b' \stackrel{?}{=} b$

$\mathcal{O}(X)$:

1. **if** $b = 0$:
2. $\quad$ **return** $F_K(X)$
3. **else:**
4. $\quad$ **return** $T[X]$



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

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

PRF – security

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

1. $b \leftarrow \{0,1\}$
2. $K \leftarrow \{0,1\}^k$
3. $T \leftarrow$ empty dictionary
4. **for all** $X \in \{0,1\}^{in}$
5. $\quad T[X] \leftarrow \{0,1\}^{out}$
6.
7. $b' \leftarrow A^{\mathcal{O}(\cdot)}$
8. **retu**

$\mathcal{O}($

1. **i**
2. $\quad$ **retu**
3. **else:**
4. $\quad$ **return** $T[X]$



World 0

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

Input X:
$\quad$ **return** $F_K(X)$

World 1

for all $X \in \{0,1\}^{in}$ **:**
$\quad T[X] \leftarrow \{0,1\}^{out}$

Input X:
$\quad$ **return** $T[X]$

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

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

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

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

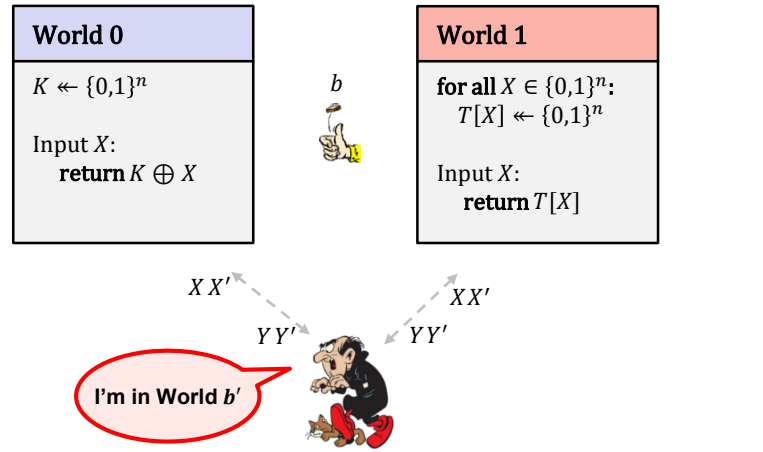
$$\mathbf{Adv}_F^{\mathrm{prf}}(A) \stackrel{\mathrm{def}}{=} 2 \cdot |\Pr[b' = b] - 1/2|$$

Example

- Let $F(K, X) = K \oplus X$ (i.e., the OTP)
- **Claim:** F is *not* a secure PRF

A

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



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

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

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

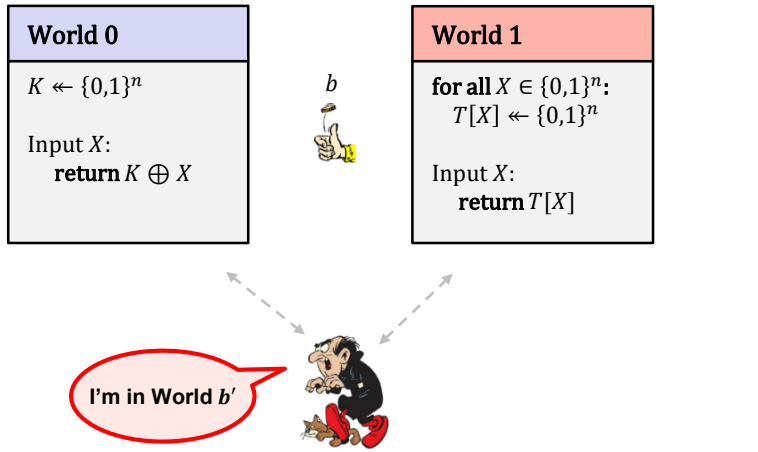
(law of total probability)

Example

- Let $F(K, X) = K \oplus X$ (i.e., the OTP)
- **Claim:** F is *not* a secure PRF

A

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



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

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

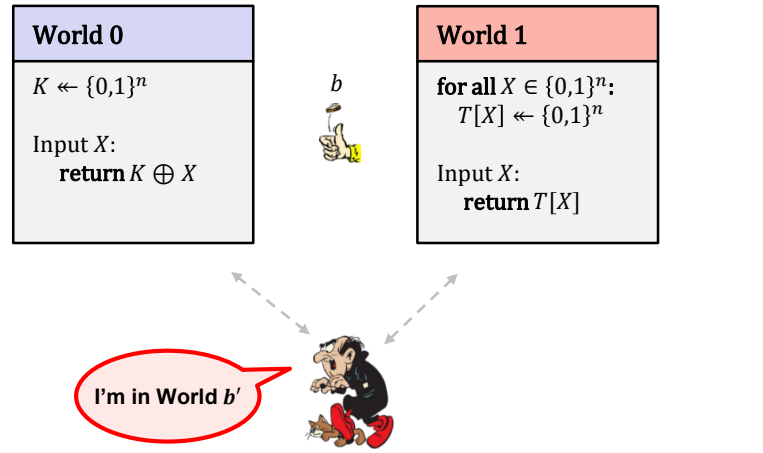
$$\begin{aligned}
\Pr[b' = b] &= \Pr[\, b' = 0 \mid b = 0\,] \cdot \Pr[\, b = 0] + \Pr[\, b' = 1 \mid b = 1\,] \cdot \Pr[\, b = 1] && \text{(law of total probability)} \\
&= \Pr[\, b' = 0 \mid b = 0\,] \cdot 1/2 + \Pr[\, b' = 1 \mid b = 1\,] \cdot 1/2 \\
&= \Pr[Y' = Y \oplus 1^n \mid b = 0] \cdot 1/2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

Example

- Let $F(K, X) = K \oplus X$ (i.e., the OTP)
- **Claim:** F is *not* a secure PRF

A

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



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

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

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

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

Understanding "advantage"

- F is a **secure PRF** if $\mathbf{Adv}_F^{\mathrm{prf}}(A)$ is "*small*" for *all* adversaries A using a "*reasonable*" amount of resources
  - Does not depend on the adversary's *strategy* (since we consider *all* possible A)
  - Only the amount of resources it uses: running time, memory usage, number of oracle calls (calls to $\mathcal{O}$), …

- What does *small* and *reasonable* mean?

  - **Example: 128-bit** security:

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) \leq \frac{2^{80}}{2^{128}} = 2^{-48} \approx 0.0000000000000036$$

    for all adversaries A that run in time $t \leq 2^{80}$

  - **Example:** a PRF is *insecure* if we can come up with *one* efficient adversary with good advantage

Huge numbers, tiny numbers

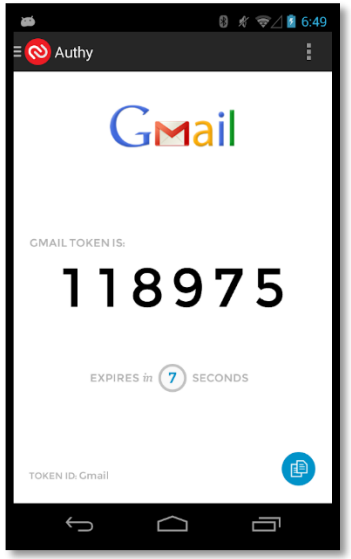
Cycles	Approx cost* (USD)	Point of reference
2^{50}	$3.50	Cup of coffee
2^{55}	$100	Good tickets to a Portland Trailblazers game
2^{65}	$130,000	Median home price in Chickasaw County, Iowa
2^{75}	$130 million	Budget of one of the Harry Potter movies
2^{85}	$140 billion	GDP of Hungary
2^{92}	$20 trillion	GDP of the United States
2^{99}	$2 quadrillion	All human economic activity since 300,000 BCE
2^{128}	a lot!	A billion human civilation's worth of effort

Prob.	Event
$2^{-28.1}$	Winning the jackpot in the US Powerball lottery
$2^{-33.4}$	Longest streak in **roulette**: red 32 times in a row. Probability: $(18/38)^{31}$
$2^{-40.6}$	Longest recorded streak in **craps**: rolled 2 dice 154 times without getting a 7. Probabilty: $(30/36)^{154}$
$2^{-56.2}$	Winning the jackpot in the US Powerball lottery …**two** weeks in a row
$2^{-84.3}$	Winning the jackpot in the US Powerball lottery …**three** weeks in a row

*running on cheapest available Amazon EC2 cloud service

Source: https://joyofcryptography.com/comp/

PRF applications: one-time tokens



$T \leftarrow$ current_time
return $F_K(T)$

PRF

$F : \{0,1\}^k \times \{0,1\}^{32} \rightarrow \{0,1,2, \ldots, 999{,}999\}$

K





$641966 \leftarrow F_K(T)$

$\text{OTT} = 641966$

$\vdots$

$\text{OTT} \stackrel{?}{=} F_K(T' - 2)$

$\text{OTT} \stackrel{?}{=} F_K(T' - 1)$

$\text{OTT} \stackrel{?}{=} F_K(T')$

$\text{OTT} \stackrel{?}{=} F_K(T' + 1)$

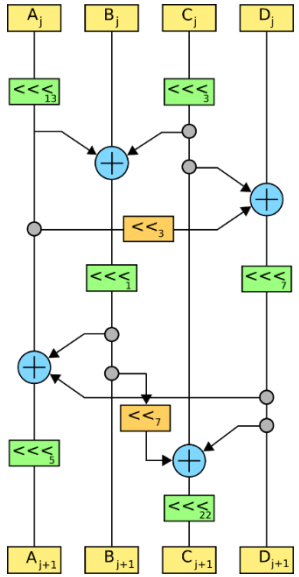
$\text{OTT} \stackrel{?}{=} F_K(T' + 2)$

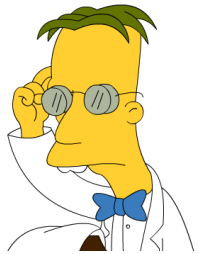
$\vdots$

PRF applications

- The “work horse” of symmetric crypto

- Can be used to build:
  - One-time tokens
  - Encryption of arbitrary length messages (including stream ciphers)
  - Message authentication codes
  - Authenticated encryption
  - Hash functions
  - Pseudorandom generators (PRNGs)
  - Key derivation functions





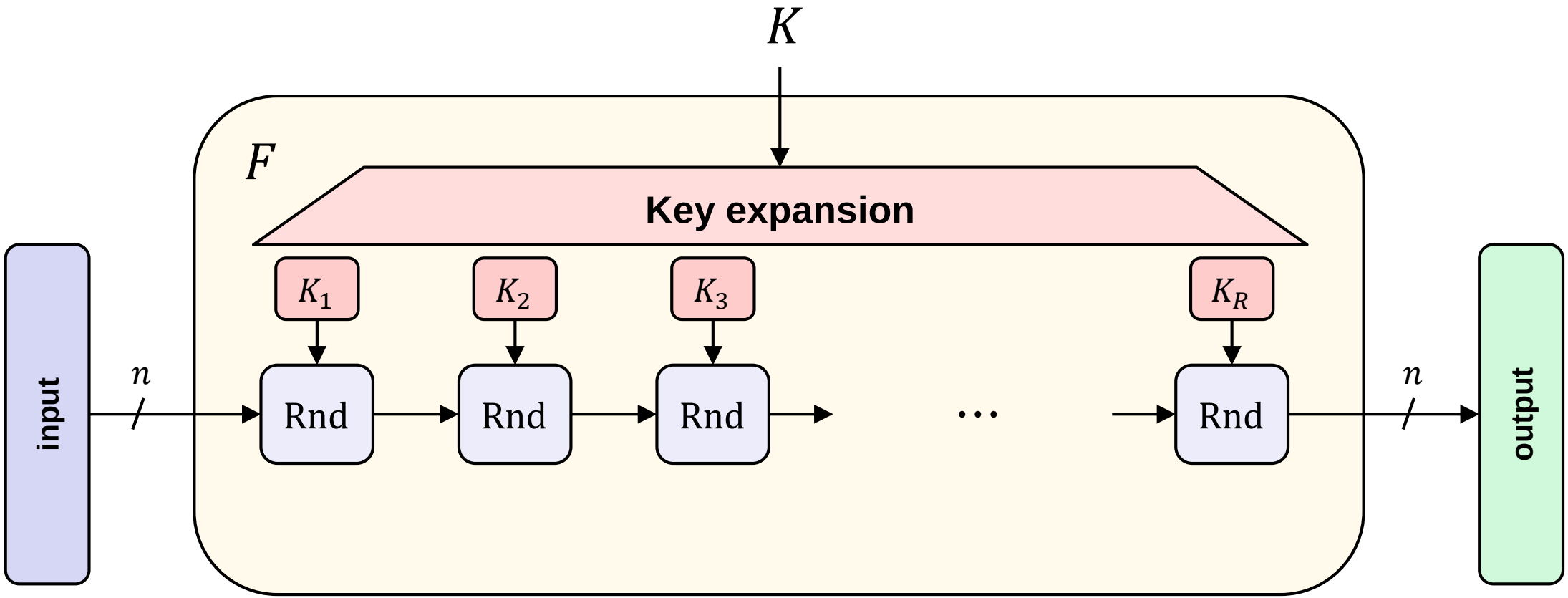
Constructing PRFs and block ciphers

Principles for designing PRFs / block ciphers

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

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

PRFs and block ciphers in practice



Rnd = **round function** (iterated R times)

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



Advanced Encryption Standard

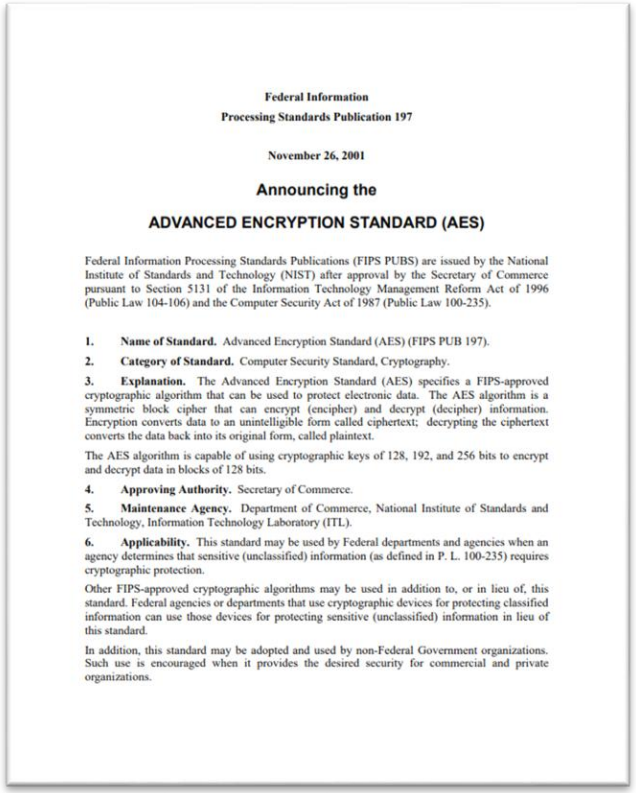
AES – history

- 1970-1990's: Data Encryption Standard (DES)
  - Key length too short (56 bits)
  - Input 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).

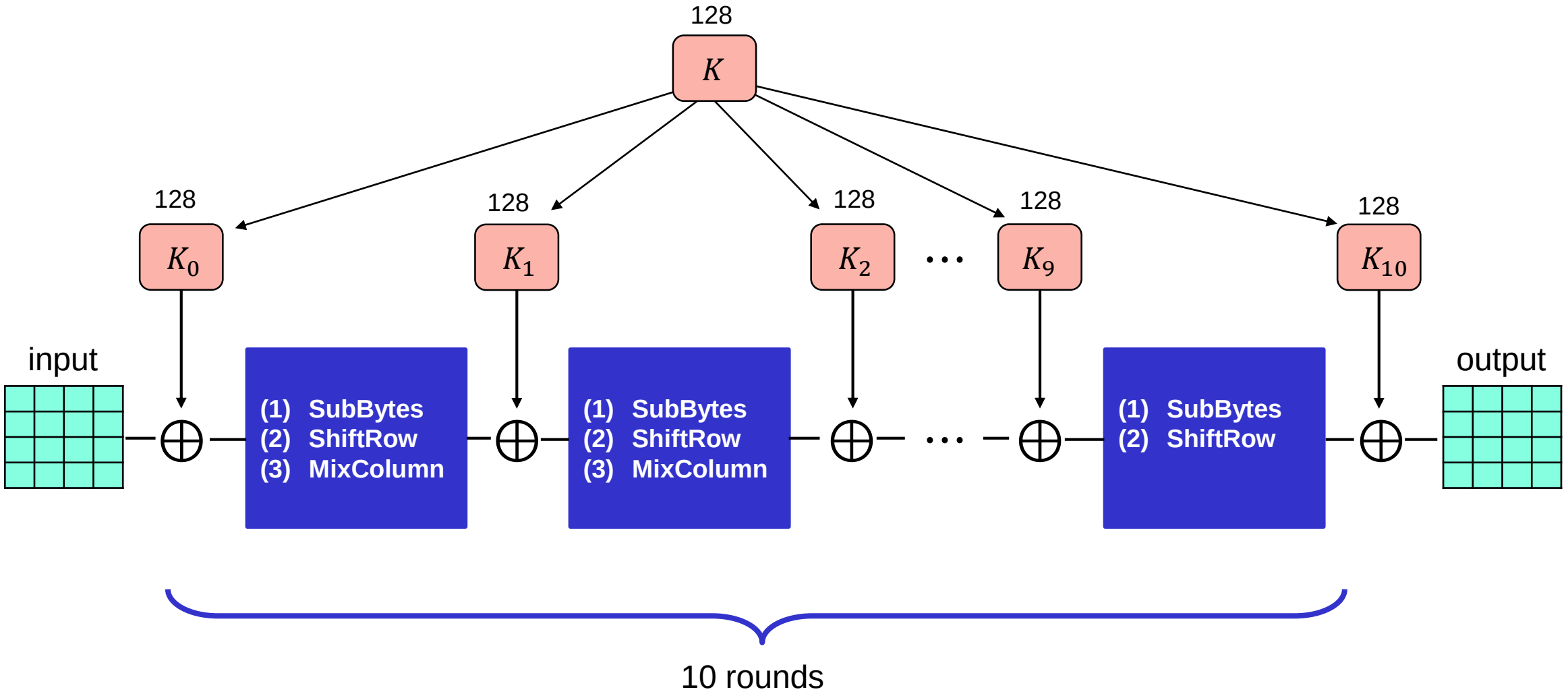
1. **Name of Standard.** Advanced Encryption Standard (AES) (FIPS PUB 197).
2. **Category of Standard.** Computer Security Standard, Cryptography.
3. **Explanation.** The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. 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.

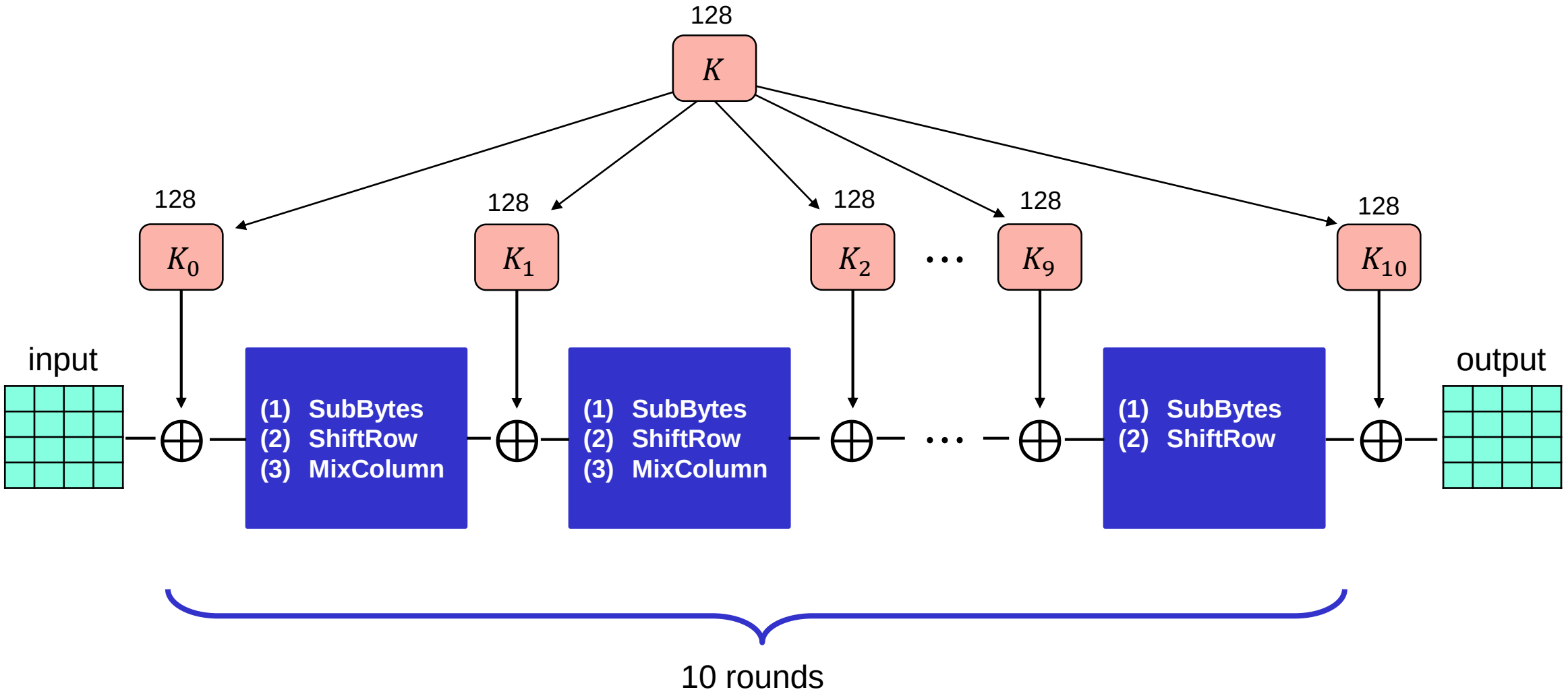
   Other FIPS-approved cryptographic algorithms may be used in addition to, or in lieu of, this standard. Federal agencies or departments that use cryptographic devices for protecting classified information can use those devices for protecting sensitive (unclassified) information in lieu of this standard.

   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.

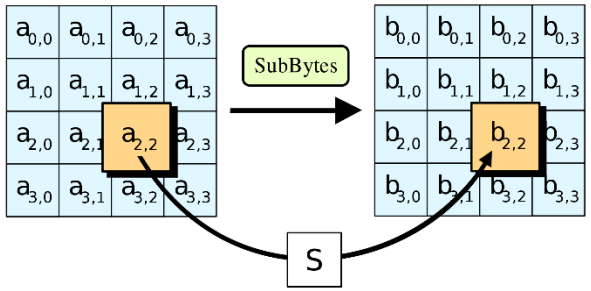
AES-128

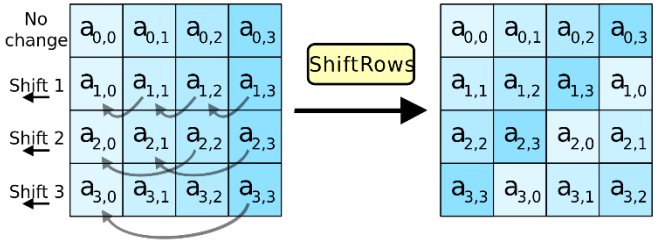

128
K
128
K0
128
K1
128
K2
K9
128
K10
input
output
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
(3) MixColumn
(1) SubBytes
(2) ShiftRow
10 rounds

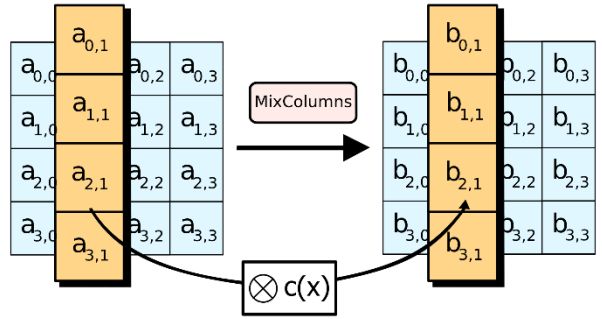
AES-128


128
K
128
K_0
128
K_1
128
K_2
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

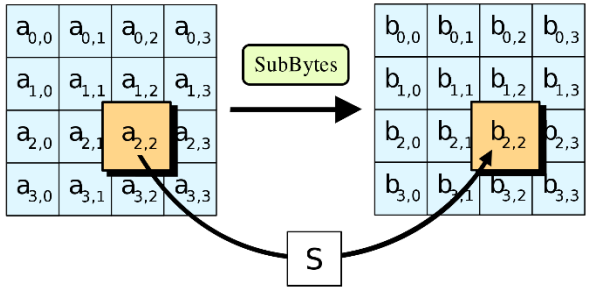


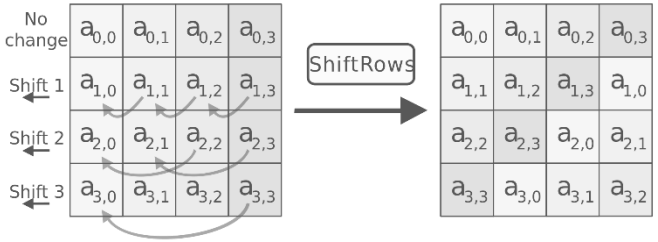


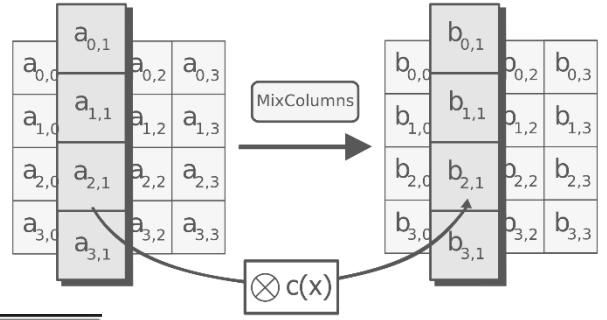


(1) SubBytes
(2) ShiftRow
(3) MixColumn

AES round function - SubBytes





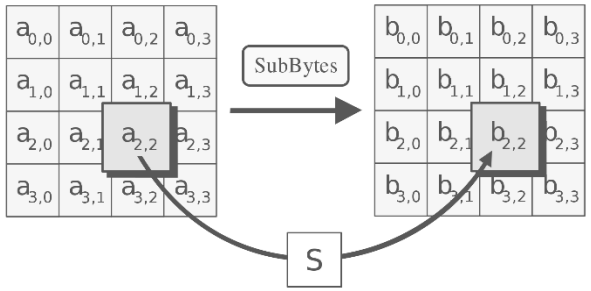


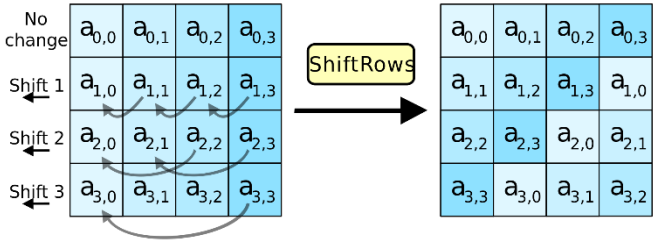
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC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

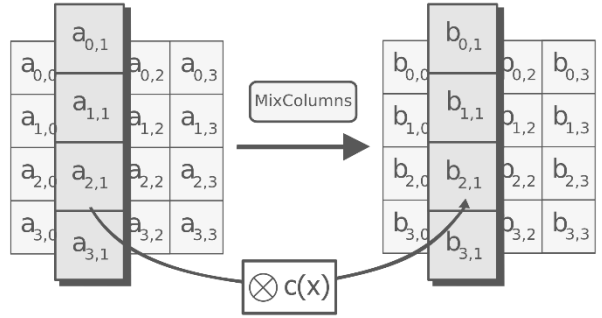
$a_{2,2} = 8A$

$b_{2,2} = 7E$

AES round function - ShiftRows

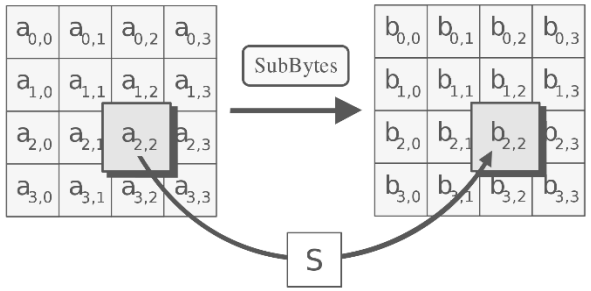


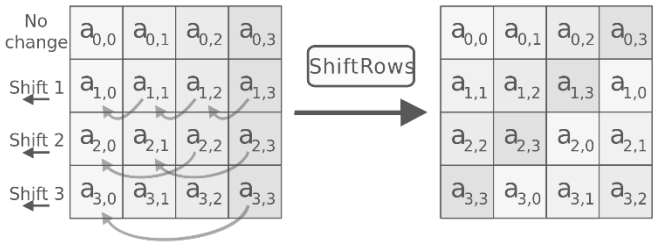


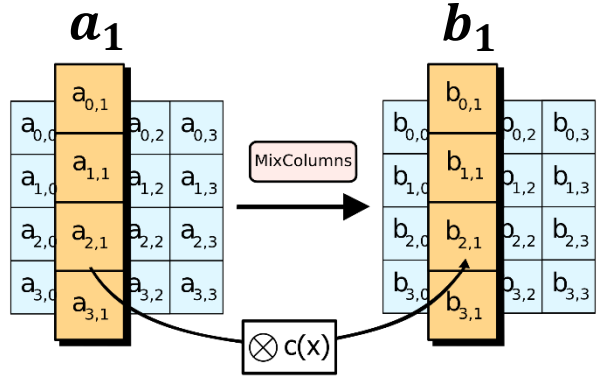


(1) SubBytes
(2) ShiftRow
(3) MixColumn

AES round function - MixColumns







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

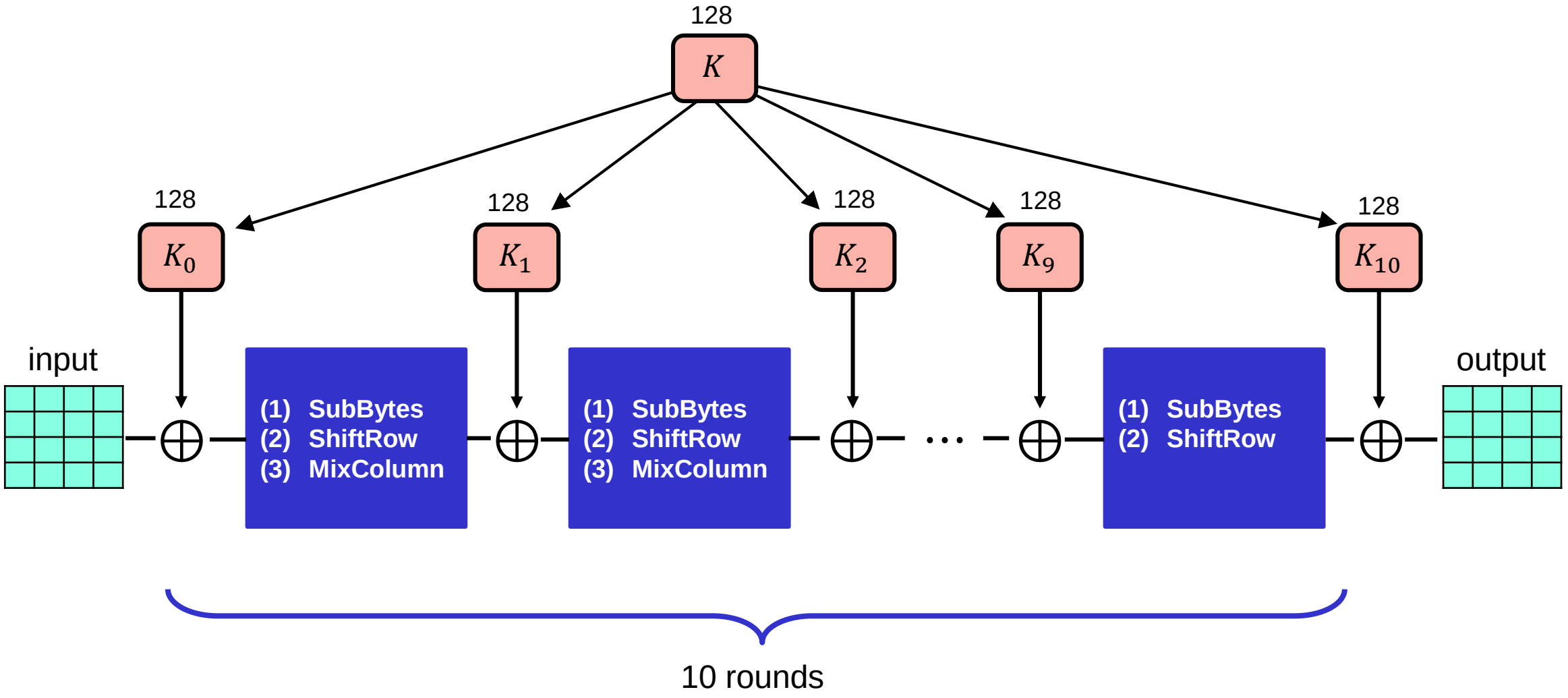
$$M\boldsymbol{a_1} = \boldsymbol{b_1}$$

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

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

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

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 summary

- AES is believed to be a secure PRF
  - 128/192/256-bit security for AES-128/192/256
  - Based on the best currently known attacks
  - Not formally proven

- Reasonably efficient in software
  - Table-based implementation is very fast (but not secure!)
  - Difficult to implement AES both fast *and* securely in SW

- Intel introduced dedicated AES instructions into their CPUs (AES-NI):
  - **aesenc:** do one round of AES
  - **aesenclast:** do final round of AES
  - **aeskeygenassist:** do AES key expansion
  - **aesdec:** do one round of AES decryption
  - **aesdeclast:** do final round of AES decryption
  - **aesimc:** do inverse MixColumns
  - Now standard in all modern CPUs

	Throughput (my laptop)
AES-128 (no AES-NI)	0.27 GB/s
AES-128 (with AES-NI)	3.45 GB/s

Block ciphers, a.k.a. pseudorandom *permutations* (PRP)

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

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

Special case:

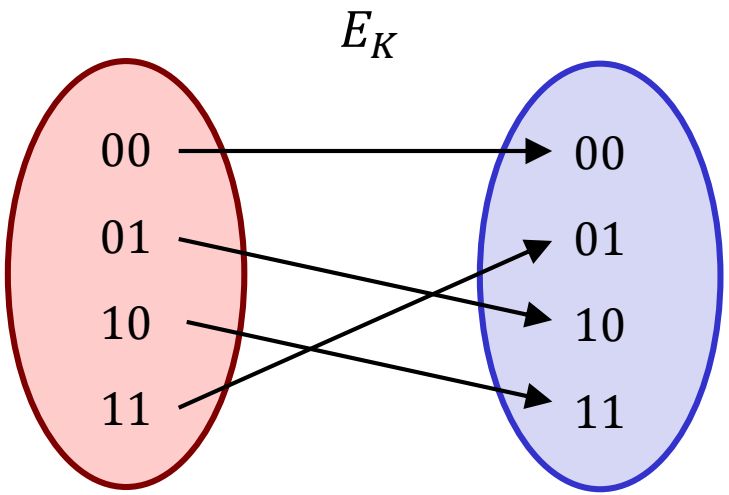
- $in = out = n$
- F_K is invertible

Definition: A **block cipher**, or **pseudorandom *permutation* (PRP)**, is function:

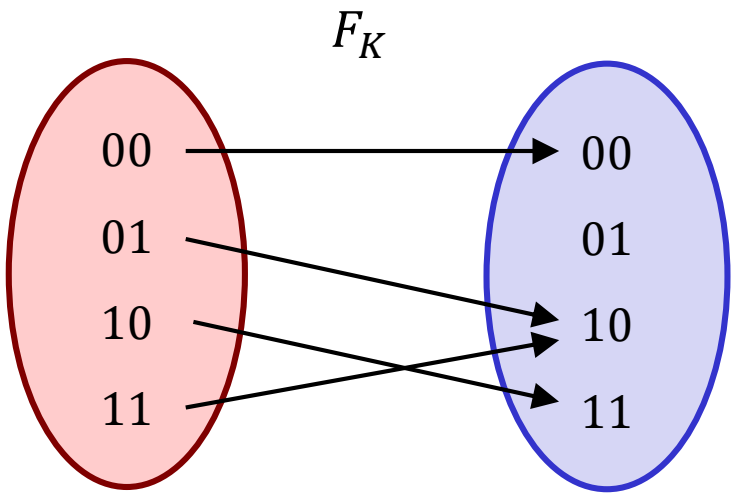
$$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

where $E_K : \{0,1\}^n \rightarrow \{0,1\}^n$ is *invertible* for every $K \in \{0,1\}^k$

Permutations vs. functions

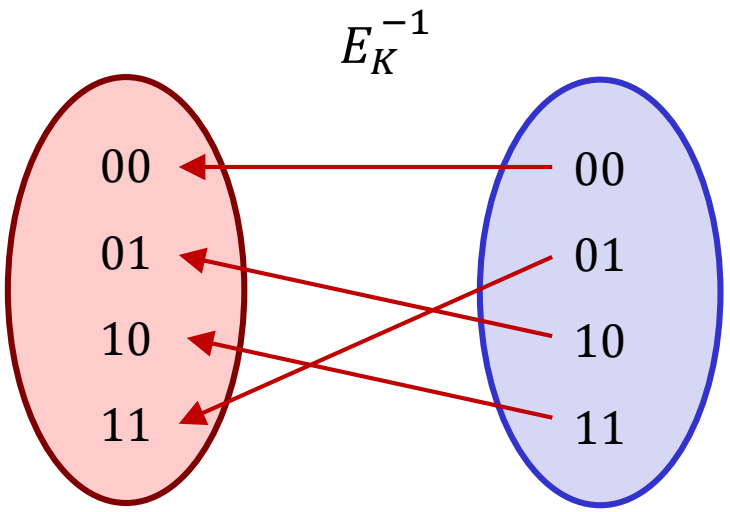


Permutation

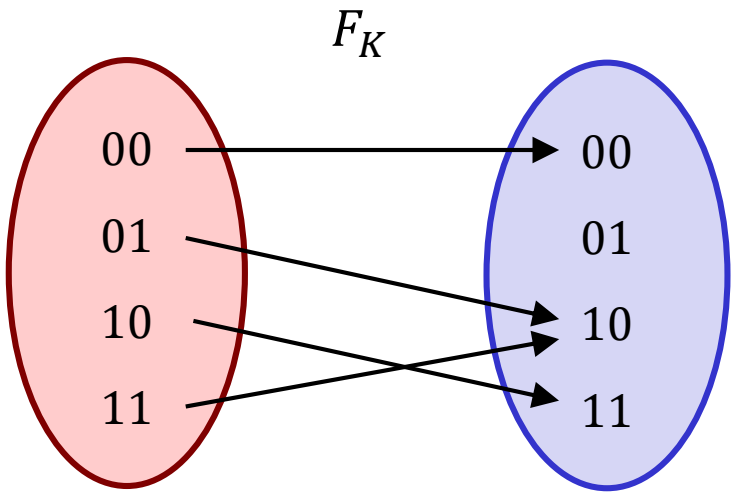


Not a permutation

Permutations vs. functions



Permutation



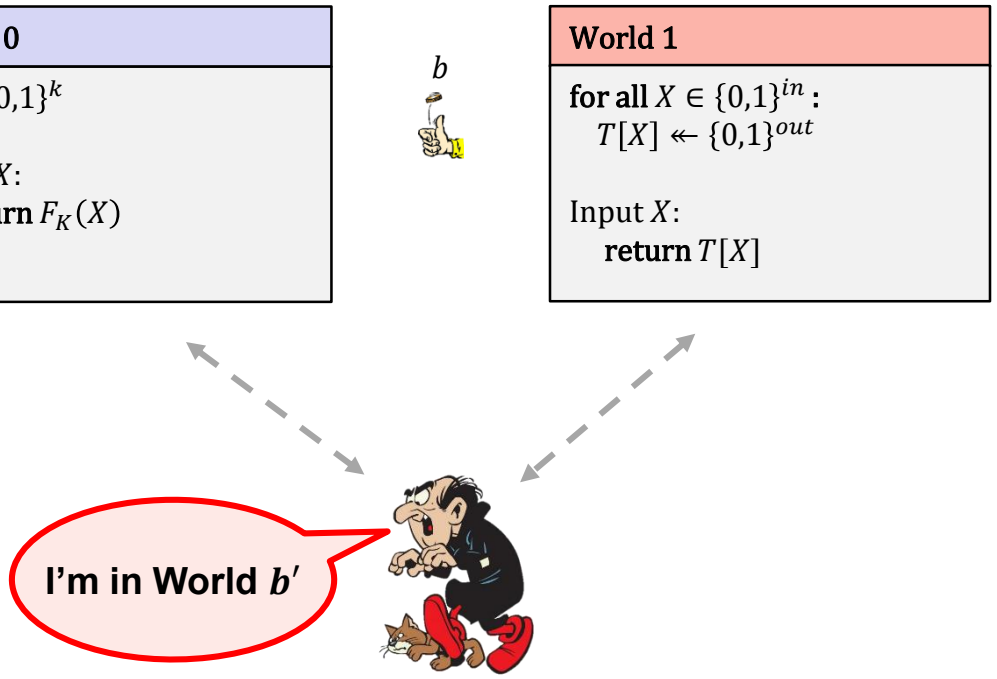
Not a permutation

Pedantic note:

$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is **not** a permutation (but a block cipher/PRP)

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

PRF – security



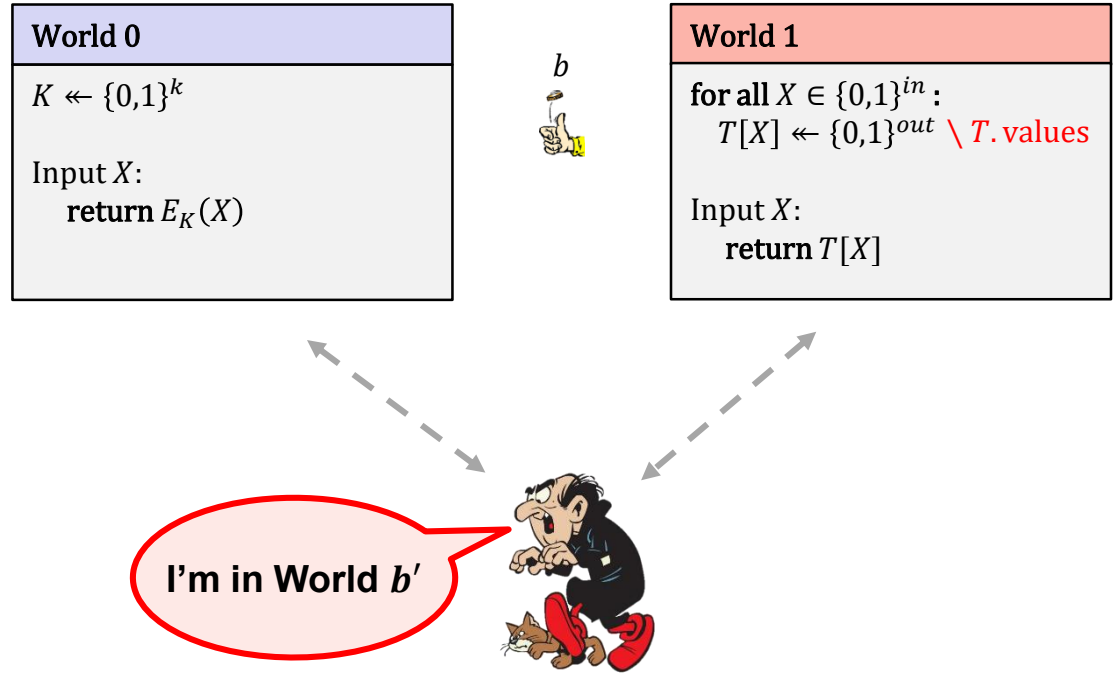
~~PRF~~ PRP – security

$\mathbf{Exp}_E^{\text{prp}}(A)$

1. $b \twoheadleftarrow \{0,1\}$
2. $K \twoheadleftarrow \{0,1\}^k$
3. $T \leftarrow$ empty dictionary
4. **for all** $X \in \{0,1\}^{in}$
5. $\quad T[X] \twoheadleftarrow \{0,1\}^{out} \setminus T.\text{values}$
6.
7. $b' \leftarrow A^{\mathcal{O}(\cdot)}$
8. **return** $b' \stackrel{?}{=} b$

$\mathcal{O}(X)$:

1. **if** $b = 0$:
2. $\quad$ **return** $E_K(X)$
3. **else:**
4. $\quad$ **return** $T[X]$



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

$$\mathbf{Adv}_E^{\text{prp}}(A) \stackrel{\text{def}}{=} 2 \cdot |\Pr[b' = b] - 1/2|$$

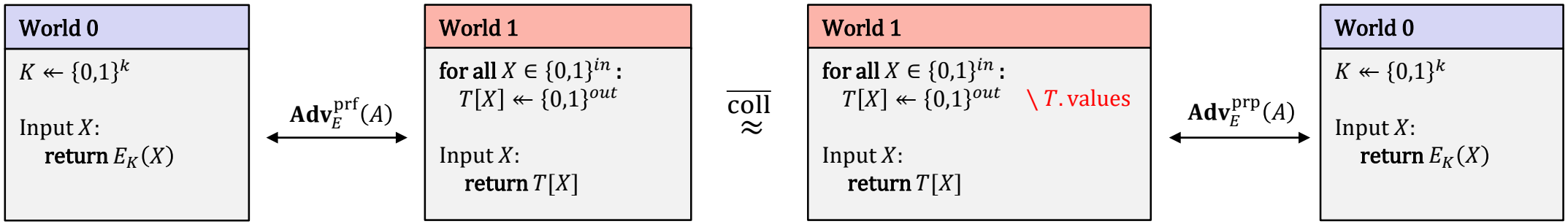
PRF security ⟺ PRP security

PRF/PRP Switching Lemma: Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a PRP. For all A making exactly q oracle queries:

$$\left|\mathbf{Adv}_E^{\mathrm{prf}}(A) - \mathbf{Adv}_E^{\mathrm{prp}}(A)\right| \leq \frac{q^2}{2^{n+1}}$$

Proof sketch:

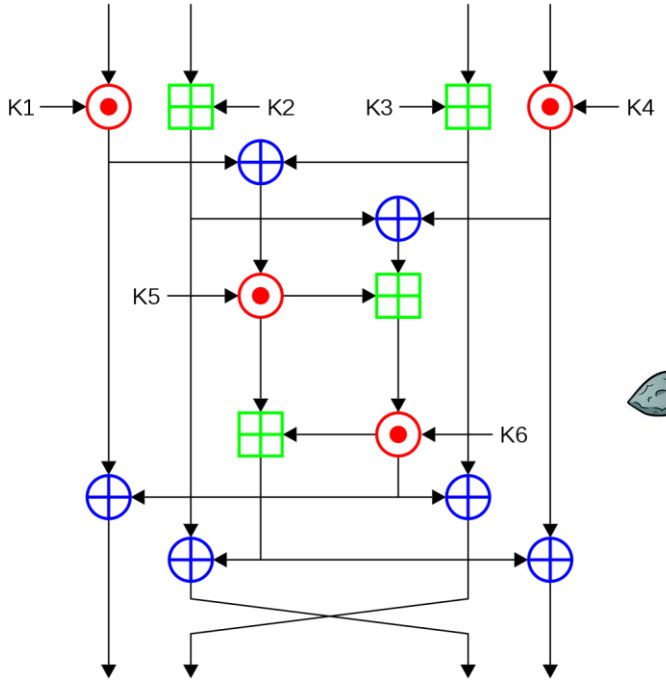
Event **coll:** two (or more) outputs from PRF oracle $\mathcal{O}(\cdot)$ *collide* (are equal)



$$\Pr[\mathrm{coll}] = \Pr[\mathrm{coll}_1 \cup \mathrm{coll}_2 \cup \cdots \cup \mathrm{coll}_q] \leq \Pr[\mathrm{coll}_1] + \Pr[\mathrm{coll}_2] + \Pr[\mathrm{coll}_3] + \cdots + \Pr[\mathrm{coll}_q]$$

$$\leq \frac{0}{2^n} + \frac{1}{2^n} + \frac{2}{2^n} + \cdots + \frac{q-1}{2^n} = \frac{q(q-1)}{2^n \cdot 2} < \frac{q^2}{2^{n+1}}$$

output from call i collides with any of the previous outputs





Attacking PRF/PRPs

Attacks on PRF/PRPs

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

- **Specialized attacks:** try to exploit the concrete details of the block cipher
  - Differential cryptanalysis ('90, but known by the designers of DES and 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
- More general (and useful) notion: PRFs
- Security notion for PRFs: output indistinguishable from random
- Concrete PRF/block cipher designs: DES, AES, …
- Next week: how to create encryption schemes from PRFs