
Lecture 2: Symmetric Key Cryptography

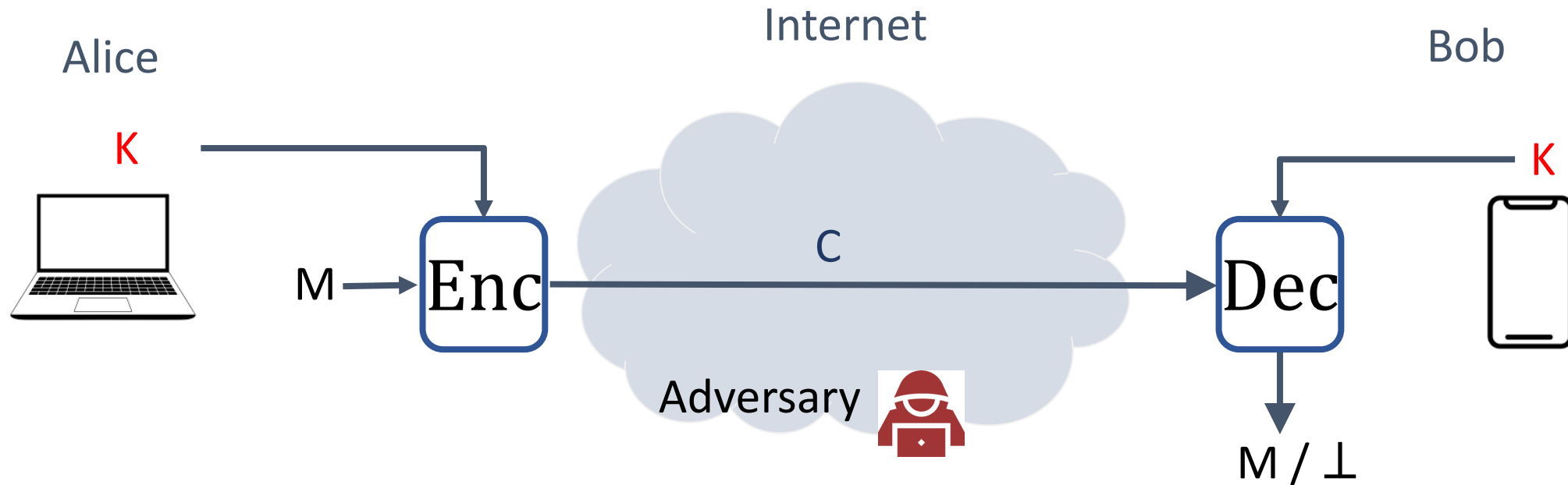
-COMP 6712 Advanced Security and Privacy

Haiyang Xue

haiyang.xue@polyu.edu.hk

2024/1/22

Symmetric-key cryptography



Enc : encryption algorithm (public)

K : shared key between Alice and Bob

Dec : decryption algorithm (public)

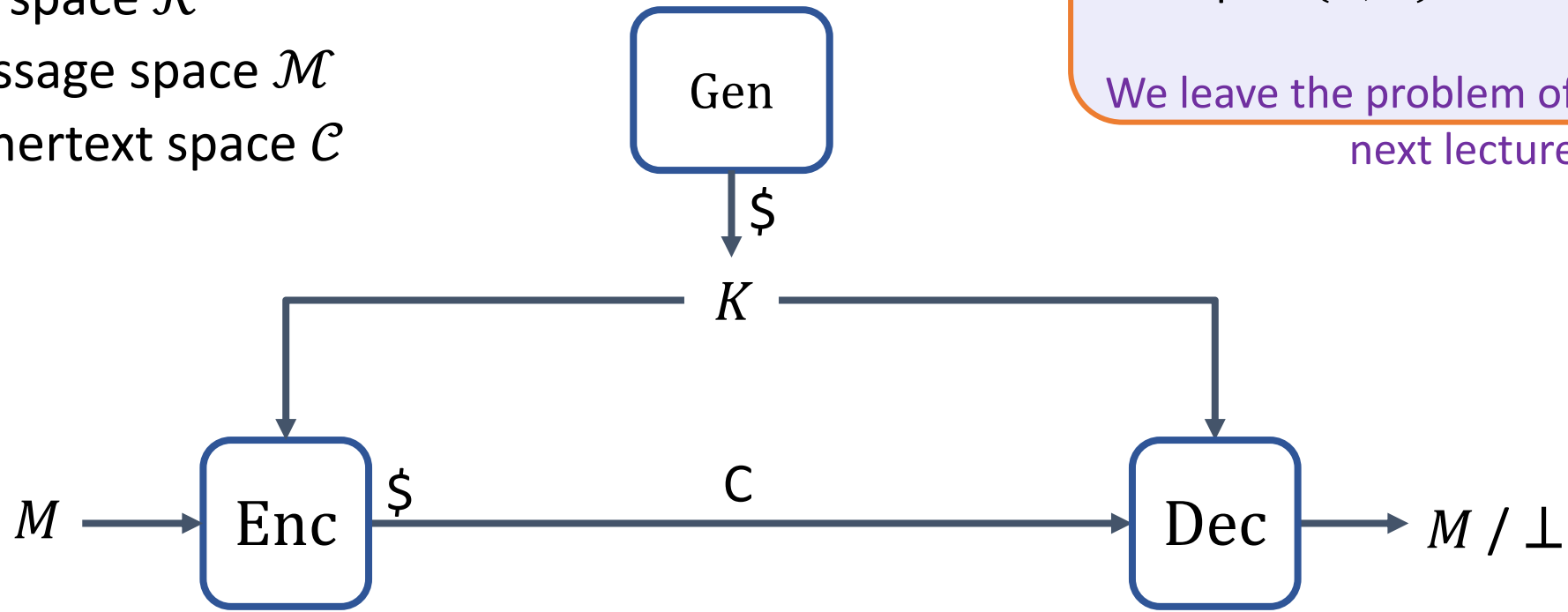
Outline of this lecture

- Syntax and security of symmetric-key cryptography
- Perfect security and one-time pad
- Stream cipher, block cipher and MAC
- Hash function
- Constructions

Syntax of symmetric encryption scheme

- A **symmetric encryption** $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ consists of three **public** algorithms:
- with
 - Key space \mathcal{K}
 - Message space \mathcal{M}
 - Ciphertext space \mathcal{C}

Key Generation: on input security parameter and randomness, Outputs (K, K) as the secret keys
We leave the problem of sending K to next lecture

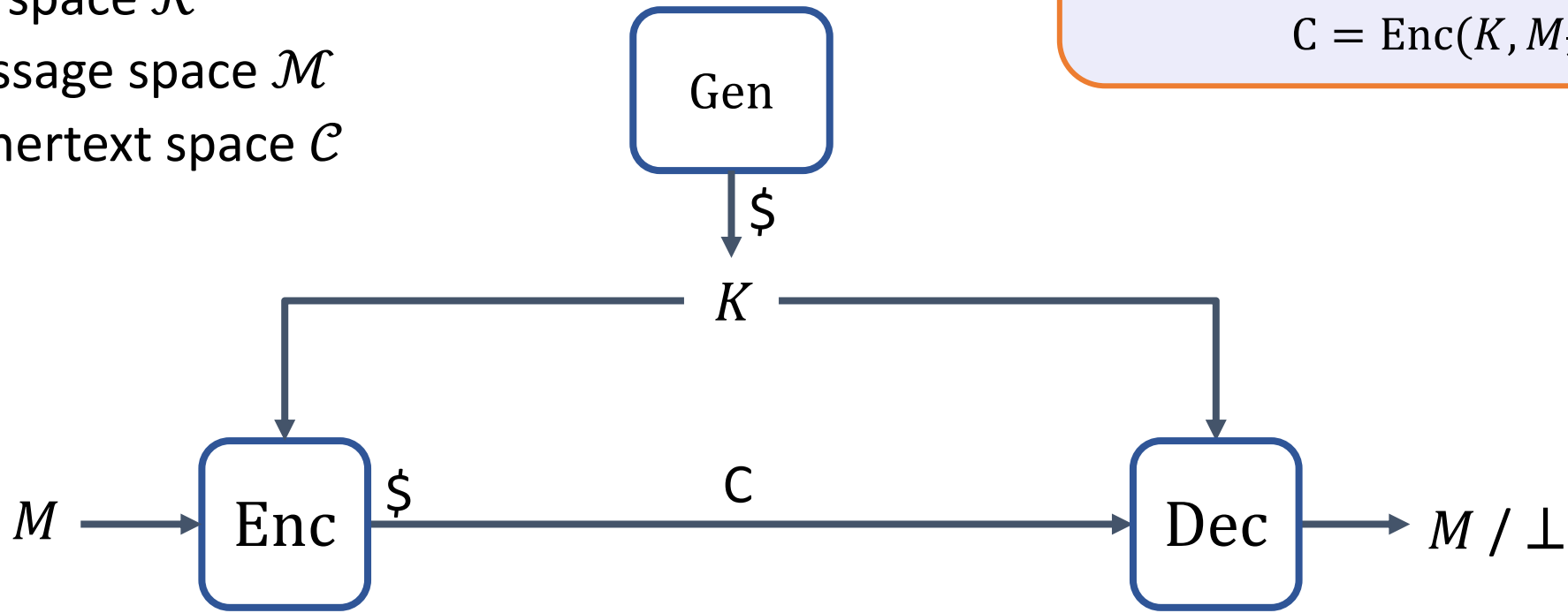


Syntax of symmetric encryption scheme

- A **symmetric encryption** $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ consists of three **public** algorithms:
- with
 - Key space \mathcal{K}
 - Message space \mathcal{M}
 - Ciphertext space \mathcal{C}

Encryption: on input M from \mathcal{M} and K ,
(and randomness r)

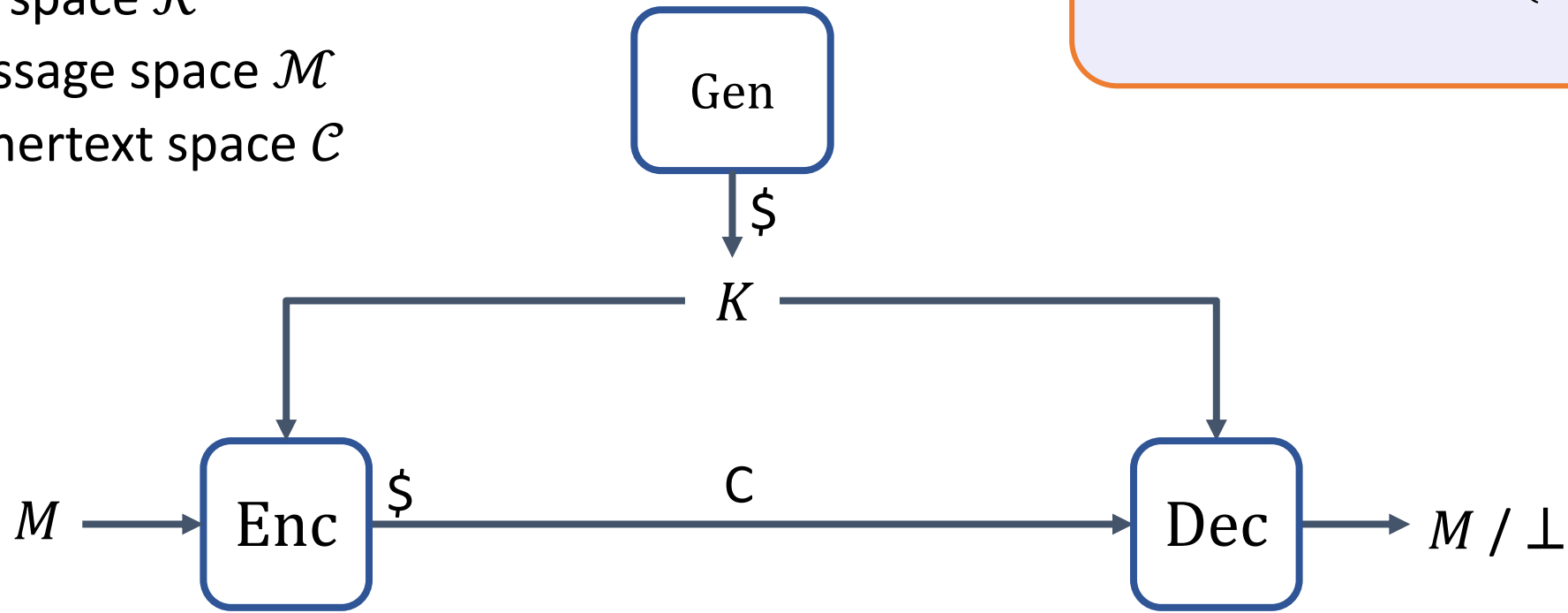
$$C = \text{Enc}(K, M, r)$$



Syntax of symmetric encryption scheme

- A **symmetric encryption** $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ consists of three **public** algorithms:
- with
 - Key space \mathcal{K}
 - Message space \mathcal{M}
 - Ciphertext space \mathcal{C}

Decryption: on input C from \mathcal{C} and K ,
 $M / \perp = \text{Dec}(K, C)$



Syntax of symmetric encryption scheme

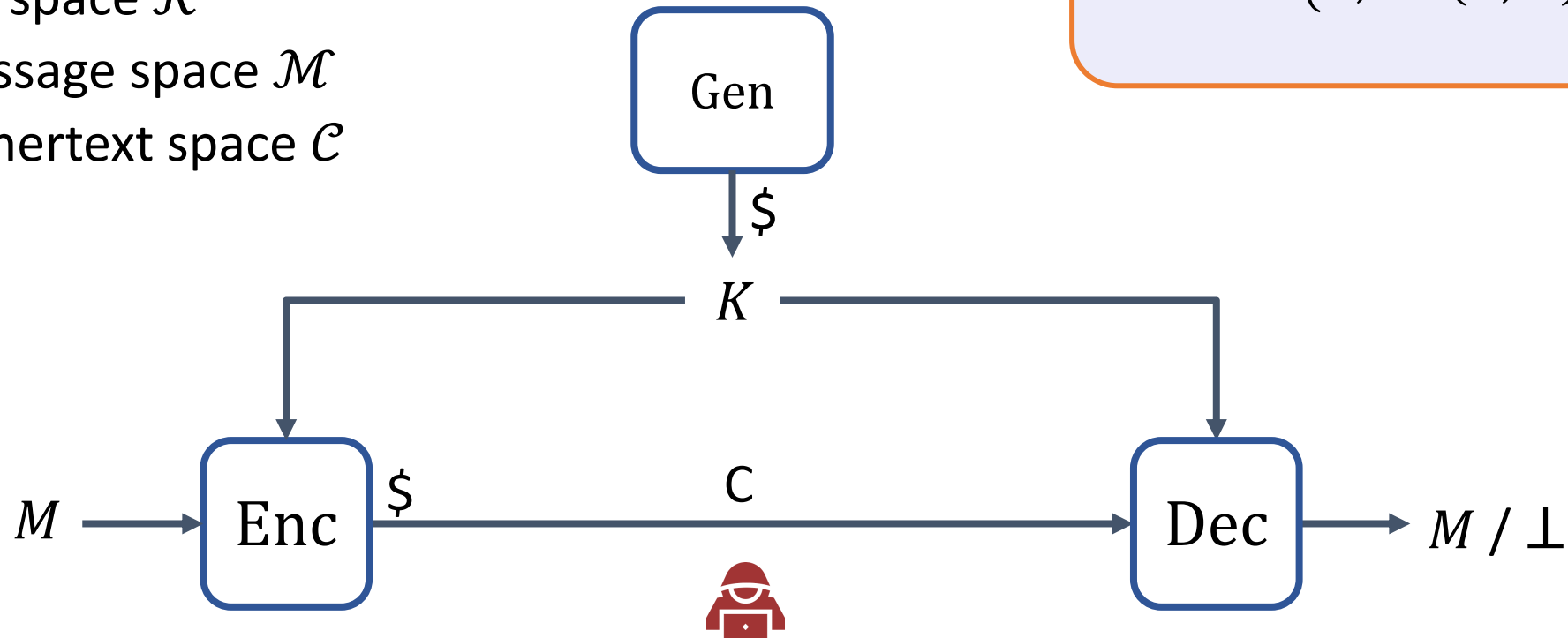
- A **symmetric encryption** $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ consists of three **public** algorithms:

- with

- Key space \mathcal{K}
- Message space \mathcal{M}
- Ciphertext space \mathcal{C}

Correctness: For all $K \leftarrow \text{Gen}$:

$$\text{Dec}(K, \text{Enc}(K, M)) = M$$



Is it possible to be secure against an adversary with unbounded computational power???

Perfect security and one-time pad

- If an enc is secure against **an adversary with unbounded computational power**, it satisfies **Perfect security**

Definition: $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ is said to be **perfectly secret** if for every distribution over \mathcal{M} , any $m \in \mathcal{M}$, any $c \in \mathcal{C}$

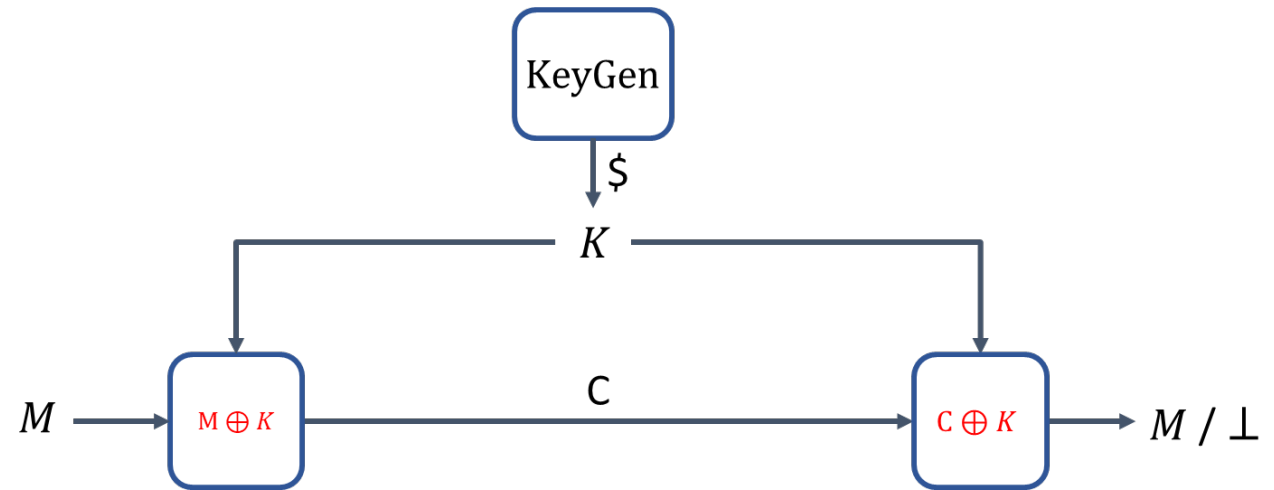
$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

with probability taken over the random choice $K \leftarrow \mathcal{K}$ and the random coins used by Enc (if any))

- The ciphertext gives nothing about the message (**even for unbounded adversary**)

Is perfect security possible? One-time Pad

- $\mathcal{K} = \{0,1\}^n$
- $\mathcal{M} = \{0,1\}^n$
- $\mathcal{C} = \{0,1\}^n$



Gen:

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

$$Enc: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$$

$$Enc(K, M) = M \oplus K$$

$$Dec: \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$$

$$Dec(K, C) = C \oplus K$$

Is perfect security possible? One-time Pad

- $\mathcal{K} = \{0,1\}^n$
- $\mathcal{M} = \{0,1\}^n$
- $\mathcal{C} = \{0,1\}^n$

Gen:

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

Enc: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

$$Enc(K, M) = M \oplus K$$

Dec: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

$$Dec(K, C) = C \oplus K$$

1110001101

$$\begin{array}{r} 0101100100 \quad M \\ \oplus 1110001101 \quad K \\ \hline = 1011101001 \quad C \end{array}$$

$$\begin{array}{r} 1011101001 \quad C \\ \oplus 1110001101 \quad K \\ \hline = 0101100100 \quad M \end{array}$$

One-time Pad

$$P(A | B) = \frac{P(A)P(B | A)}{P(B)}$$

Theorem: The One-time Pad encryption scheme has **perfect security**

- **Have to show:** $\Pr[M = m | C = c] = \Pr[M = m]$

$$\Pr[C = c | M = m] = \Pr[m \oplus K = c] = \Pr[K = m \oplus c] = \frac{1}{2^n}$$

$$\Pr[C = c] = \sum_{m \in \mathcal{M}} \Pr[C = c | M = m] \Pr[M = m] = \frac{1}{2^n} \sum_{m \in \mathcal{M}} \Pr[M = m] = \frac{1}{2^n}$$

$$\Pr[M = m | C = c] = \frac{\Pr[C = c | M = m] \Pr[M = m]}{\Pr[C = c]} = \frac{\frac{1}{2^n} \Pr[M = m]}{\frac{1}{2^n}}$$

Limitation

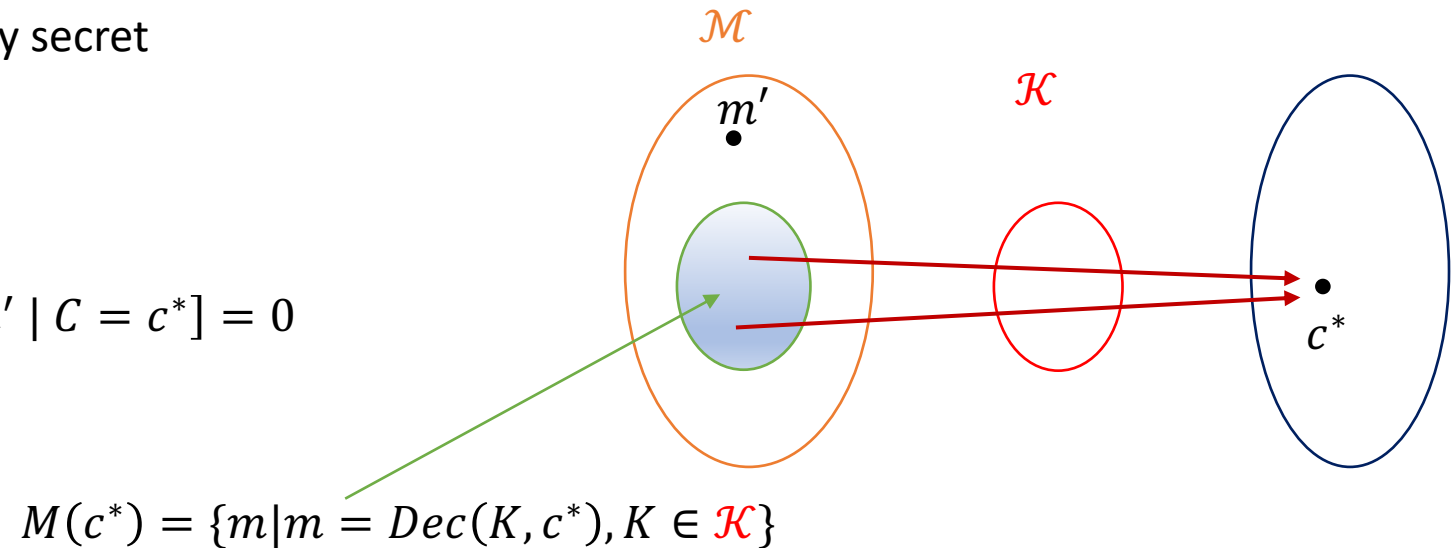
- But $|\mathcal{K} = \{0,1\}^n| = |\mathcal{M} = \{0,1\}^n|$?
- If we find a way to deliver K , why not deliver M directly?

Theorem: If Π is a perfectly secret enc with key space \mathcal{K} and message space \mathcal{M}
 $|\mathcal{K}| \geq |\mathcal{M}|$

We show: if $|\mathcal{K}| < |\mathcal{M}|$, Π can not be perfectly secret

We have $|M(c^*)| \leq |\mathcal{K}| < |\mathcal{M}|$,

$\Pr[M = m'] \neq 0$, while $\Pr[M = m' | C = c^*] = 0$



A short summary

- perfect security against the unbounded adversary
- could be achieved via the one-time pad
- Inherent limitation, key space \geq message space
- How to break the limitation?

Break the limitation

- Aim low
- ~~Unbounded adversary~~
- Guarantee against efficient adversaries that run for some feasible amount of time. (ex. probabilistic polynomial time (PPT))
- Adversaries can potentially succeed with a small probability

small probability- negligible function

Definition: A positive function f is said to be **negligible** if for **every positive** polynomial p , and sufficiently large n

$$f(n) \leq \frac{1}{p(n)}.$$

• Ex

$$2^{-n}$$

$$2^{-\sqrt{n}}$$

$$\frac{1}{n^{1000}} ??$$

Theorem: for every positive polynomial q , if f is **negligible**, so does $q(n) \cdot f(n)$.

Necessary of PPT and negligible

- probability polynomial time

- If $|\mathcal{K}| < |\mathcal{M}|$, ciphertext must leak some information to **UNBOUNDED** adversary

- Negligible success probability

- Adversary runs in constant time can win with probability $\frac{1}{|\mathcal{K}|}$

Computational security

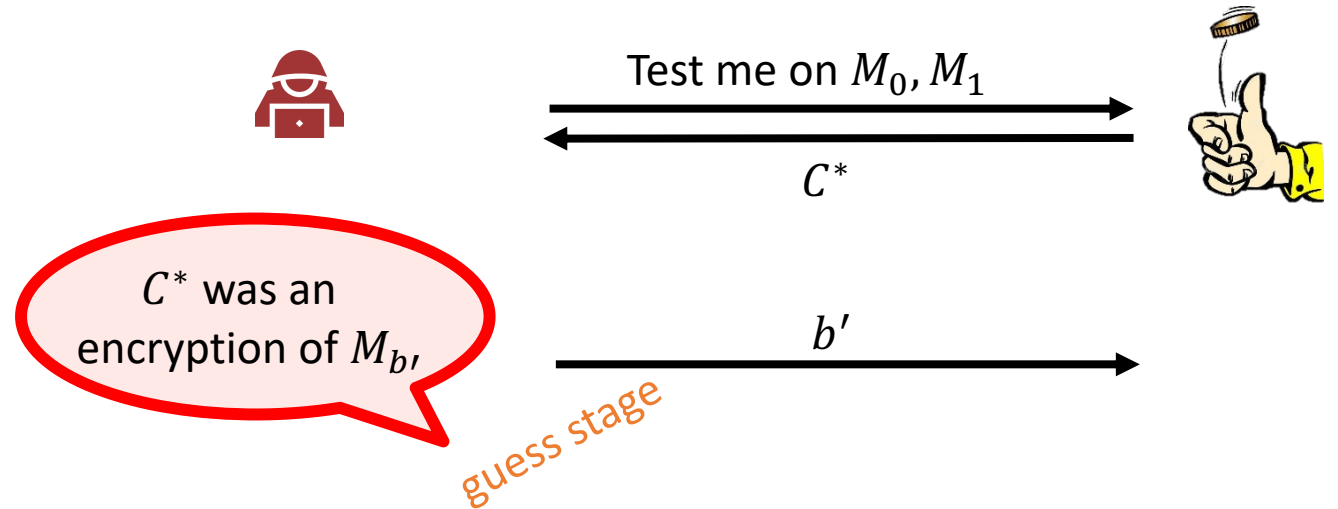
Definition: A scheme is (t, ε) -secure if any adversary running for a time **at most t** succeeds in breaking the scheme **with probability at most ε** .

Definition: A scheme Π is said to be **computationally secure** if any **PPT** adversary succeeds in **breaking** the scheme with **negligible** probability.

IND-eavesdropper

$\mathbf{Exp}_{\Pi}^{\text{ind-eav}}(A)$

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi.\text{Gen}$
3. $M_0, M_1 \leftarrow A()$ // find stage
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow \Pi.\text{Enc}(K, M_b)$
7. $b' \leftarrow A(C^*)$ // guess stage
8. return $b' \stackrel{?}{=} b$



Definition: The **IND-eav-advantage** of an adversary A is

$$\mathbf{Adv}_{\Pi}^{\text{ind-eav}}(A) = |\Pr[\mathbf{Exp}_{\Pi}^{\text{ind-eav}}(A) \Rightarrow 1] - 1/2|$$

Construction of IND-eavesdropper secure enc

- We could construct a secure enc from PRG
- PRG is generally a function to extends k random bits to $k + l$ pseudo-random bits

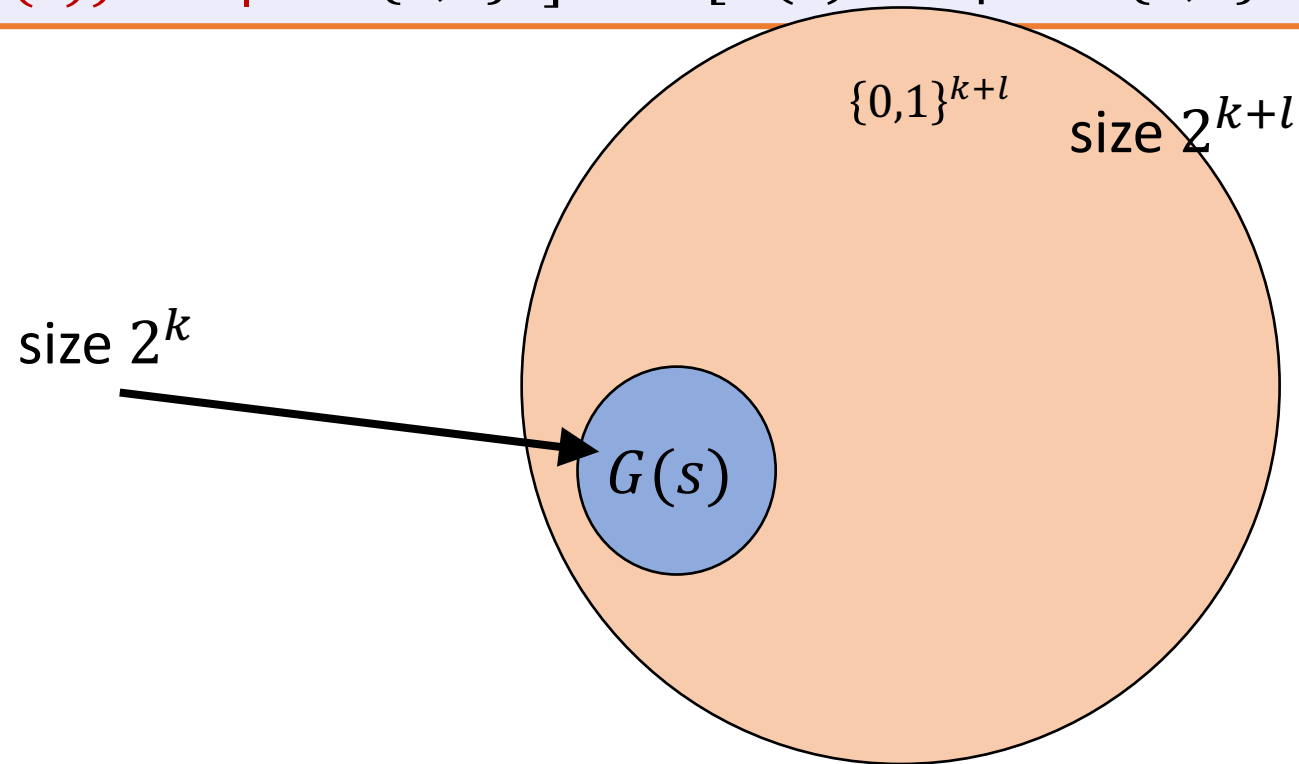
pseudo-random generator (PRG)

Definition: A pseudorandom random generator (PRG) is a function

$$G : \{0,1\}^k \rightarrow \{0,1\}^{k+l}$$

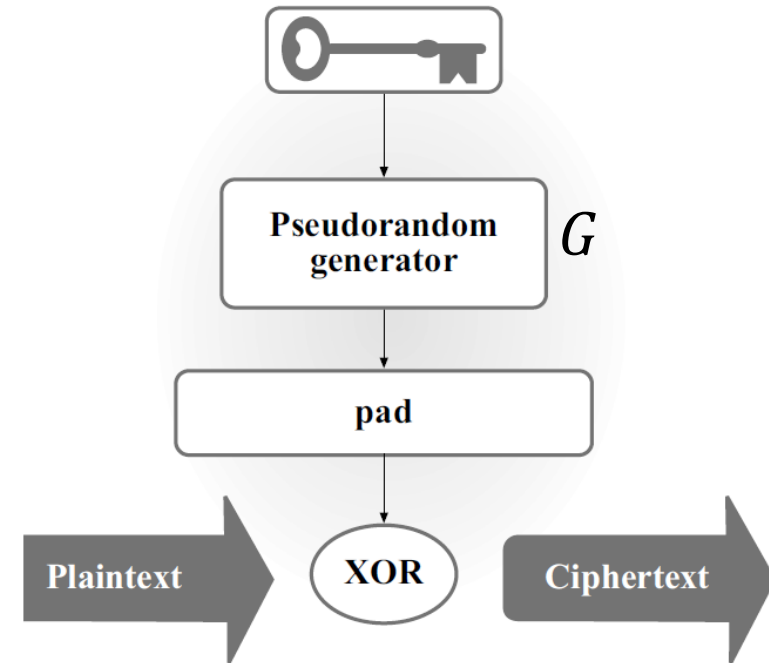
Such that

- $0 < l < \text{poly}(k)$
- For any PPT A, $\Pr[A(G(s)) = 1 | s \leftarrow \{0,1\}^k] - \Pr[A(r) = 1 | r \leftarrow \{0,1\}^{k+l}] < \text{negl}$



IND-eavesdropper Enc (with fix length) from PRG

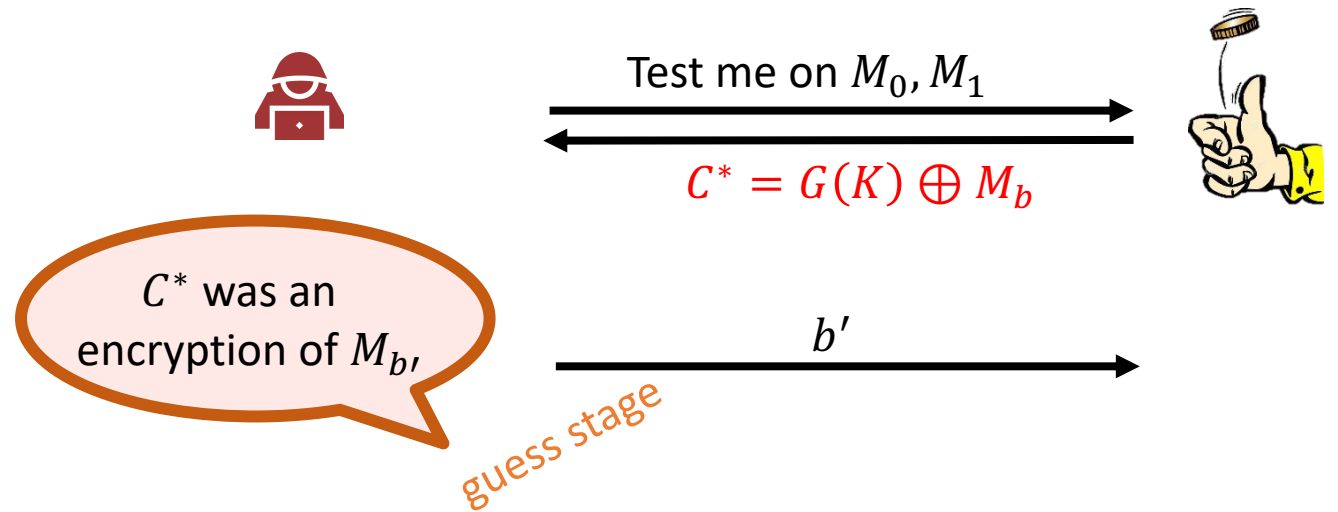
- Let $G : \{0,1\}^k \rightarrow \{0,1\}^{k+l}$ be a PRG
- Π 1. Gen: $K \leftarrow \{0, 1\}^k$
- Π 1. Enc(K, M): $C = G(K) \oplus M$
- Π 1. Dec(K, C): $M = G(K) \oplus C$



PROOF idea: IND-eavesdropper

$\text{Exp}_{\Pi_1}^{\text{ind-eav}}(A)$

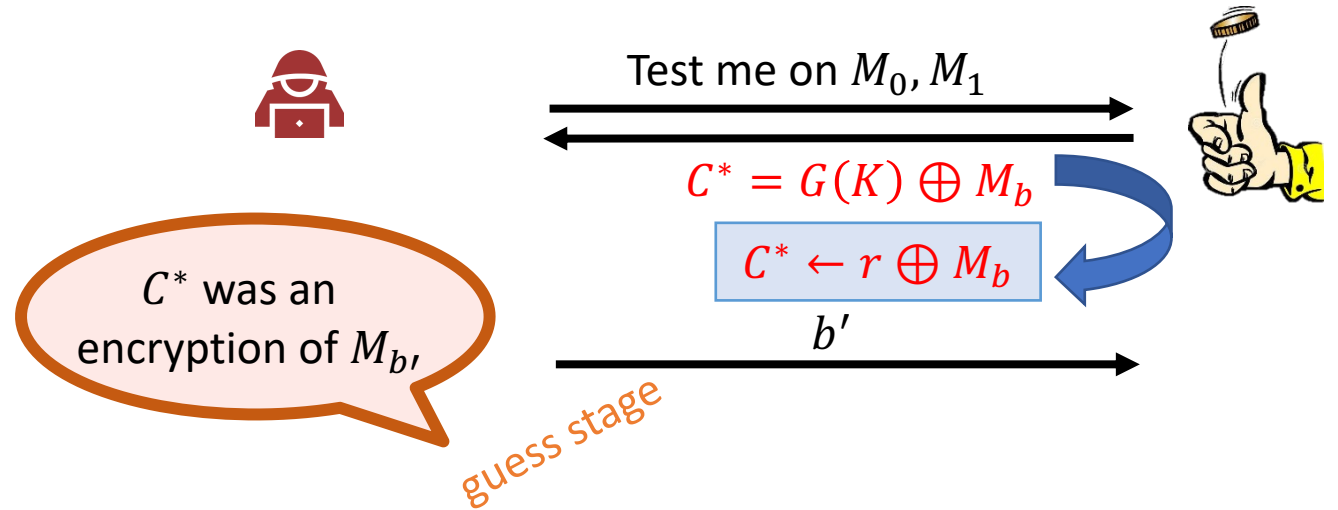
1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi_1.\text{Gen}$
3. $M_0, M_1 \leftarrow A()$ // find stage
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow G(K) \oplus M_b$
7. $b' \leftarrow A(C^*)$ // guess stage
8. return $b' \stackrel{?}{=} b$



PROOF idea: IND-eavesdropper

Exp _{Π_1} ^{ind-eav}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi_1.\text{Gen}$
3. $M_0, M_1 \leftarrow A()$ // find stage
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow G(K) \oplus M_b$ $C^* \leftarrow r \oplus M_b$
7. $b' \leftarrow A(C^*)$ // guess stage
8. return $b' \stackrel{?}{=} b$



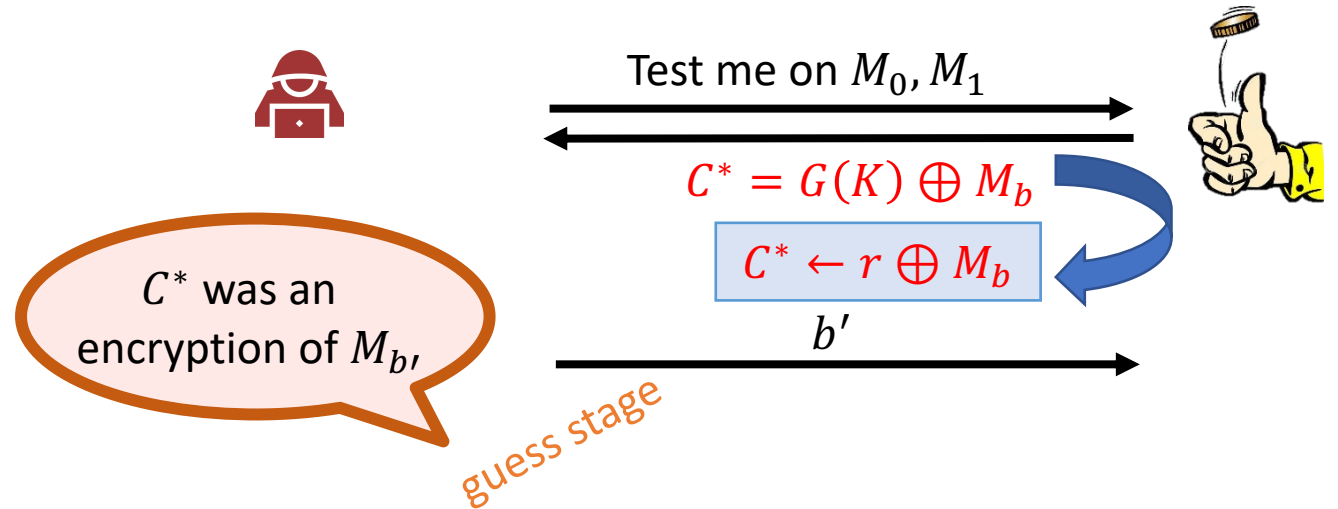
Now, this is an **one-time pad** and the **IND-eav-advantage** of an adversary A is

$$\text{Adv}_{\Pi_1}^{\text{ind-eav}}(A) = 0$$

PROOF idea: IND-eavesdropper

$\text{Exp}_{\Pi_1}^{\text{ind-eav}}(A)$

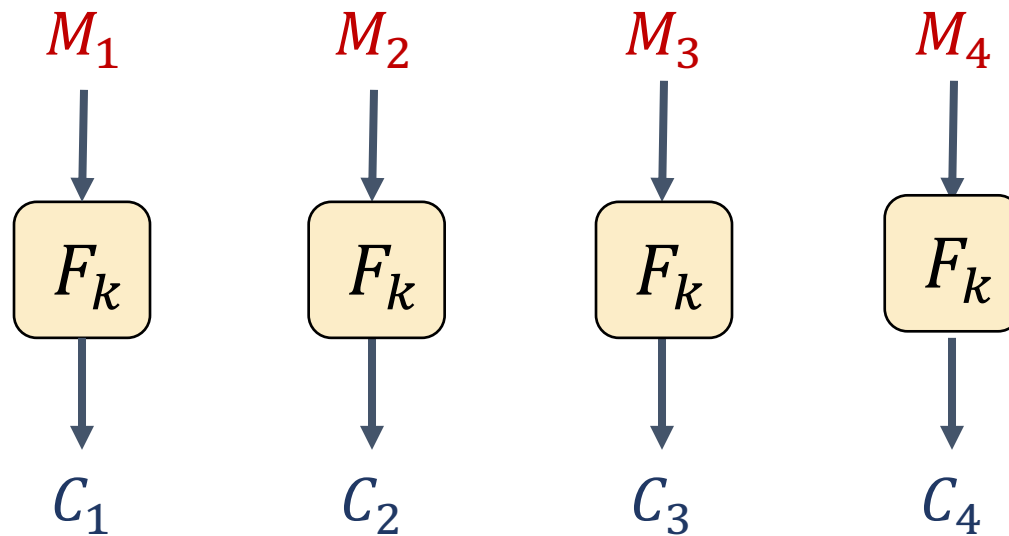
1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi_1.\text{Gen}$
3. $M_0, M_1 \leftarrow A()$ // find stage
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow G(K) \oplus M_b$ $C^* \leftarrow r \oplus M_b$
7. $b' \leftarrow A(C^*)$ // guess stage
8. return $b' \stackrel{?}{=} b$



Any PPT adversary can not find the switch, since G is a PRG

Electronic Code Book (ECB) mode (for longer message)

- Given a block cipher $F_k: \{0,1\}^n \rightarrow \{0,1\}^n$ which is the encryption of Π_1
- $\text{ECB}[F_k] = (\text{Gen}, \text{Enc}, \text{Dec})$



Weakness of ECB

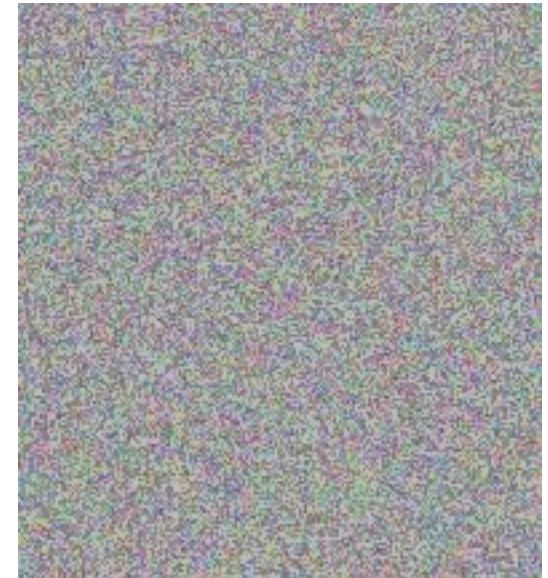
Plaintext



ECB encrypted

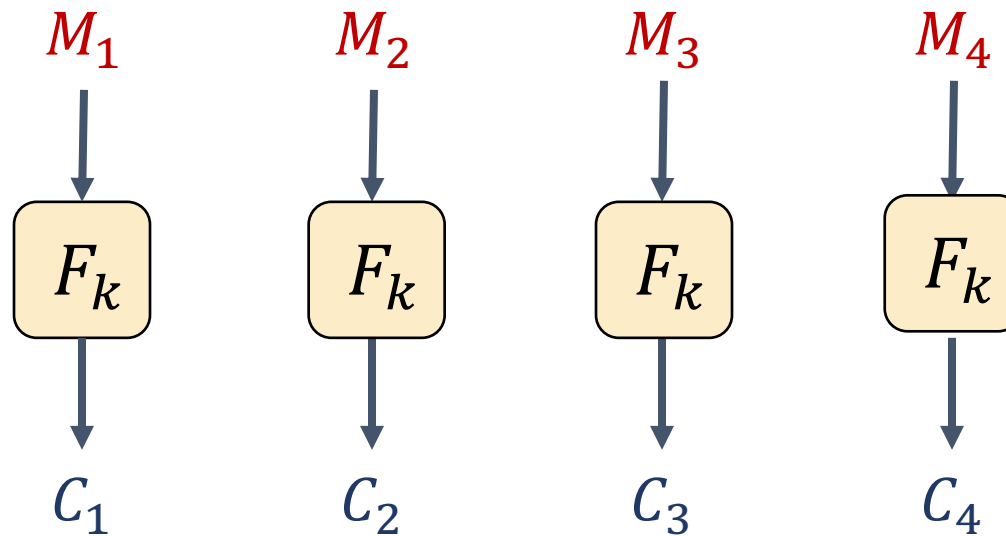


Properly encrypted

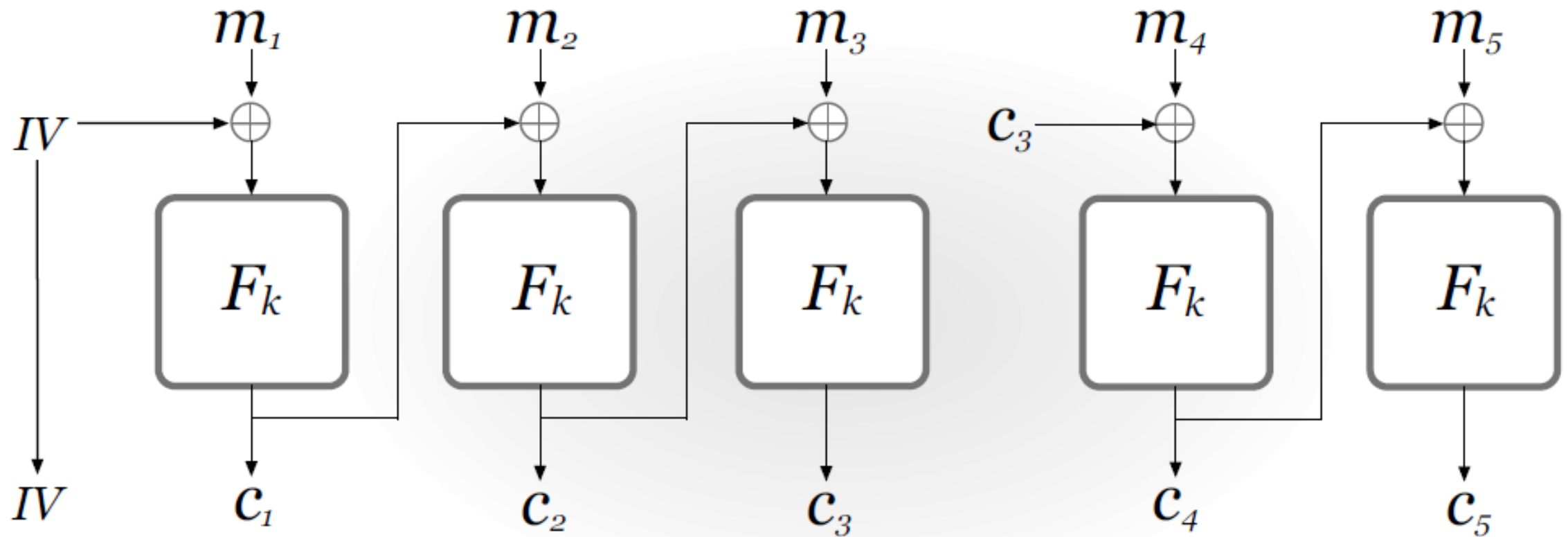


Electronic Code Book (ECB) mode (for longer message)

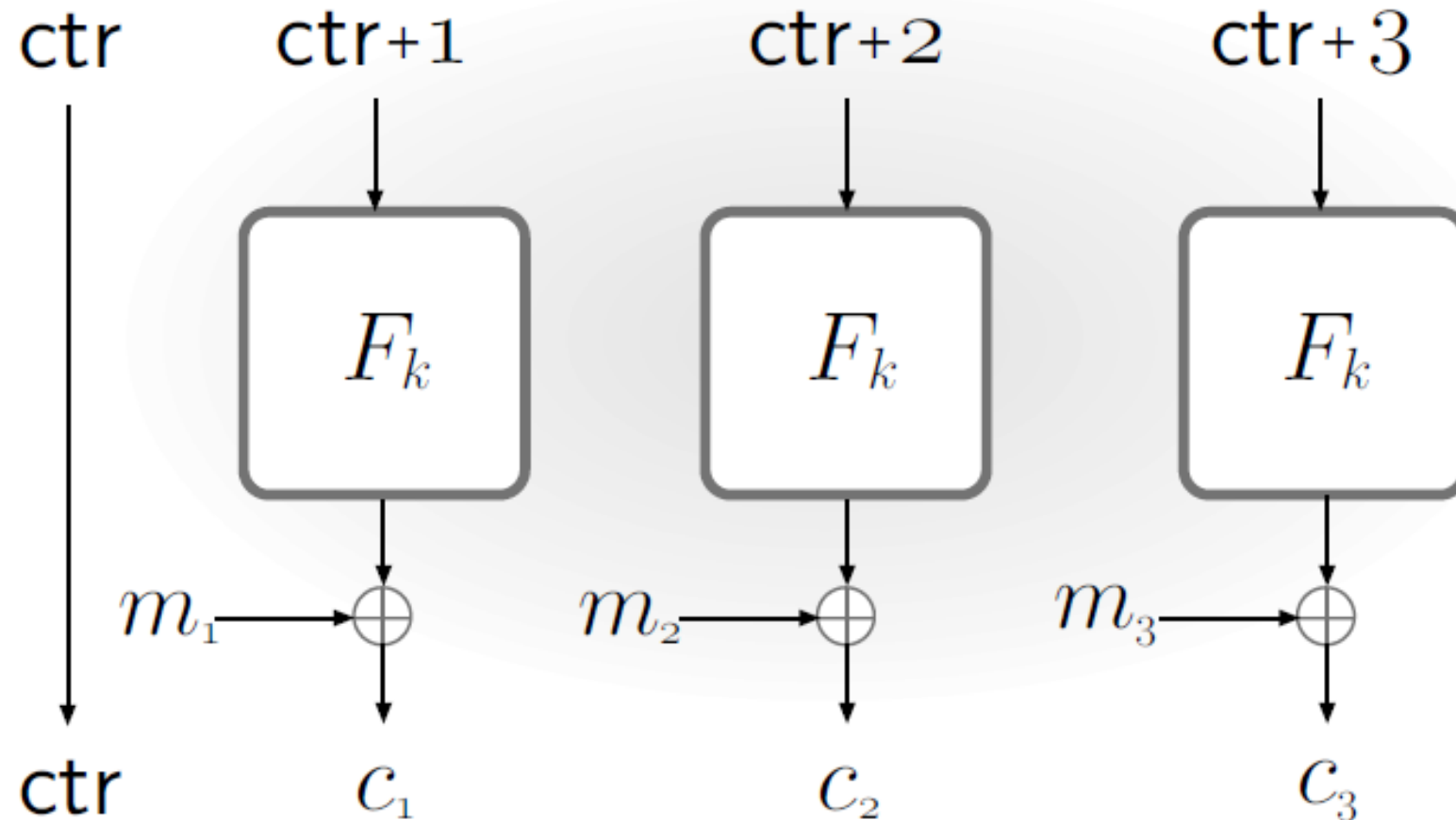
- This is because if $M_1 = M_2$, then $C_1 = C_2$



Cipher Block Chaining (CBC) mode

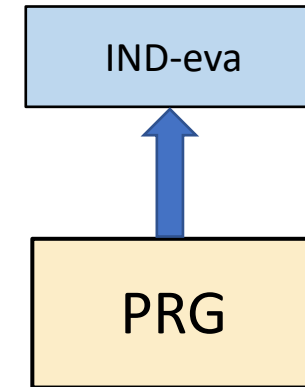


Counter (CTR) mode

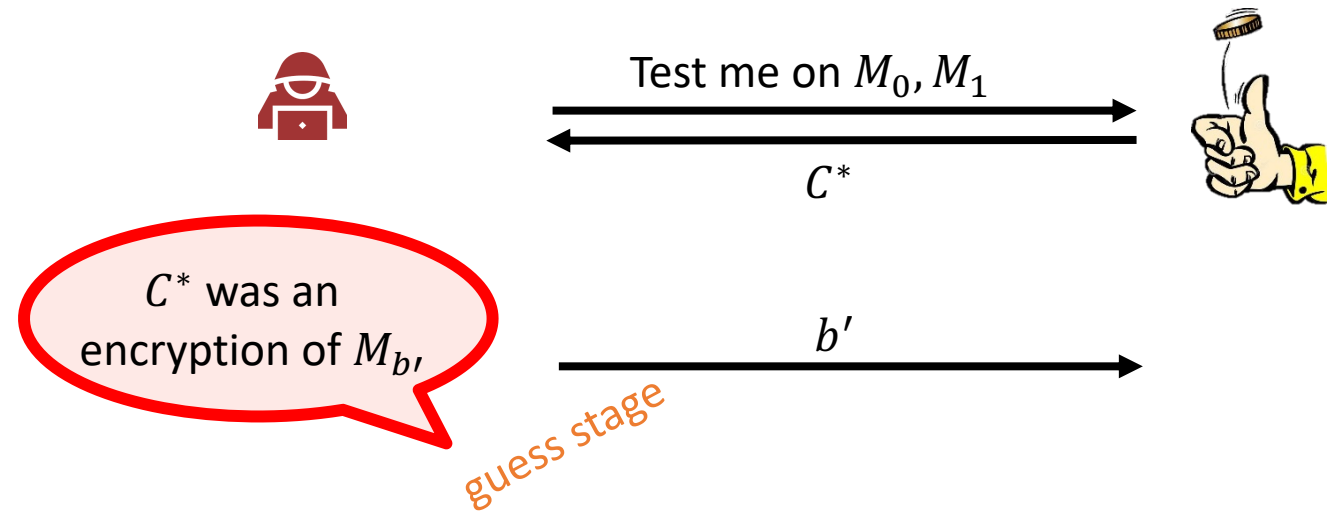


A short summary

- With aim of computational security, we can encrypt a long message with a short key
- With PRG, we could build IND-eavesdropper Enc
- We can further encrypt a longer message by splitting the message in blocks. It may operate in several models, EBC, CBC, CTR etc.
- IND-eavesdropper is a very weak security aim.



IND-eavesdropper is weak



Definition: The **IND-eav-advantage** of an adversary A is

$$\mathbf{Adv}_{\Pi}^{\text{ind-eav}}(A) = |\Pr[\mathbf{Exp}_{\Pi}^{\text{ind-eav}}(A) \Rightarrow 1] - 1/2|$$

Strong Security: IND-CPA

- In World War II
- British placed naval mines at certain locations, knowing that the Germans—when finding those mines—would encrypt the locations and send them back to Germany
- $C = \text{Enc}(\text{location of mines})$



https://en.wikipedia.org/wiki/Naval_mine

An adversary may have the capability to choose a message and get the ciphertext

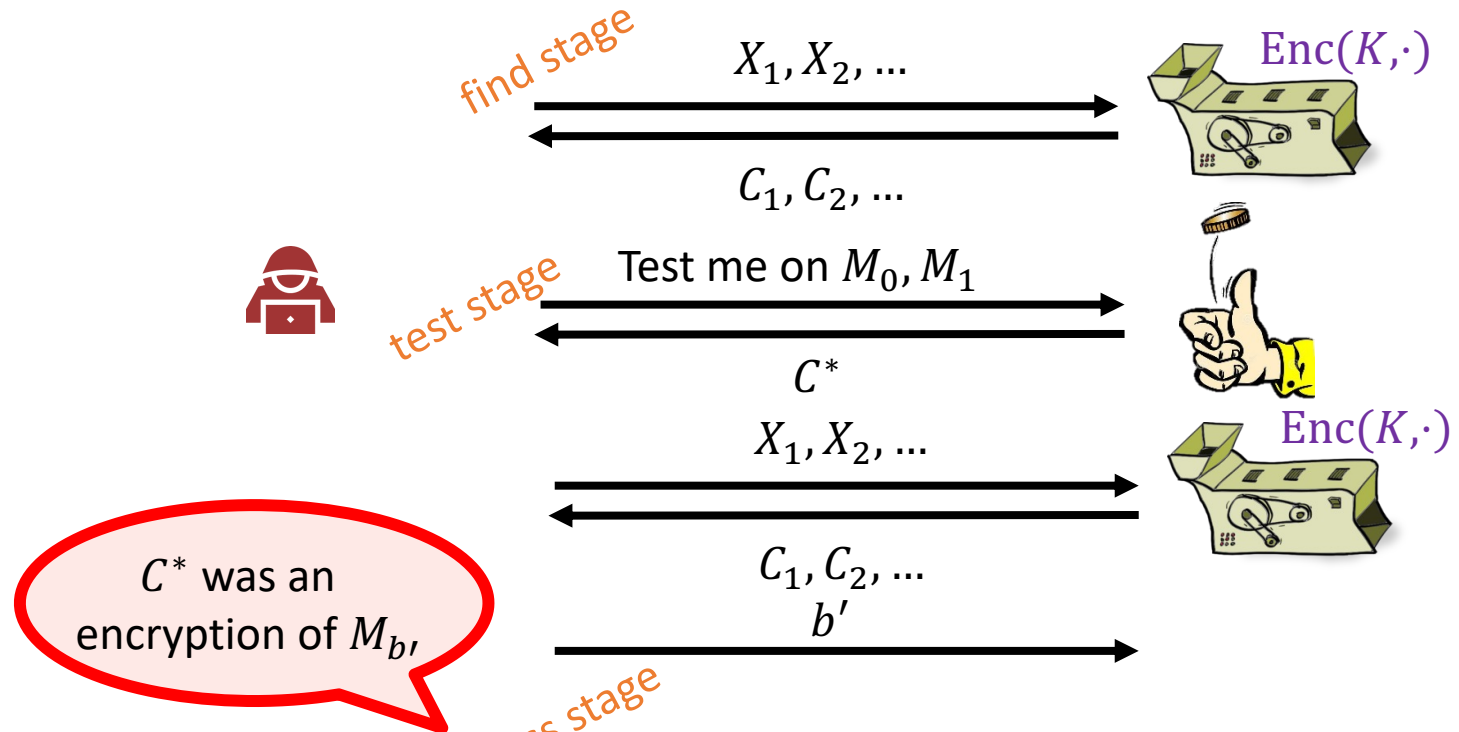
IND-CPA (choose plaintext attack)

Exp_Π^{ind-cpa}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi. \text{Gen}$
3. $M_0, M_1 \leftarrow A^{\text{Enc}(K, \cdot)}$ // find stage
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow \Pi. \text{Enc}(K, M_b)$ // test stage
7. $b' \leftarrow A^{\text{Enc}(K, \cdot)}(C^*)$ // guess stage
8. return $b' \stackrel{?}{=} b$

Enc(K, M)

-
1. return $\Pi. \text{Enc}(K, M)$



Definition: The **IND-CPA-advantage** of an adversary A is

$$\text{Adv}_{\Pi}^{\text{ind-cpa}}(A) = \left| \Pr \left[\text{Exp}_{\Pi}^{\text{ind-cpa}}(A) \Rightarrow 1 \right] - 1/2 \right|$$

IND-CPA Insecurity of Π_1

Adversary A

1. Query $C \leftarrow \Pi_1.\text{Enc}(K, 0^{128})$ in the find stage
2. Submit $M_0 = 0^{128}$ and $M_1 = 1^{128}$
3. Receive challenge C^*
4. if $C^* = C$ output 0
5. else, output 1

Actually, this attack works for any DETERMINISTIC Enc

Construction of IND-CPA secure enc

- We could construct an IND-CPA secure enc from PRF
- PRF generalizes the notion of PRG
- instead of considering “random-looking” strings we consider “random-looking” functions

pseudorandom function (PRF)

Definition: A pseudorandom function (PRF) is a function

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

satisfying security in next page

- k, in, out are called **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 $F_K : \{0,1\}^{in} \rightarrow \{0,1\}^{out}$ defined by $F_K(X) = F(K, X)$

Secure PRFs

- Let $F : \{0,1\}^k \times \{0,1\}^{in} \rightarrow \{0,1\}^{out}$
- $S_F = \{ F_K \mid K \in \{0,1\}^k \} \subseteq \text{Func}[in, out]$
- $\text{Func}[in, out]$: the set of *all* functions from $\{0,1\}^{in}$ to $\{0,1\}^{out}$

- F_K is **secure** if

$$\Pr[A^{F_K(\cdot)}(\cdot) = 1 \mid F_K \leftarrow S_F] - \Pr[A^{\tilde{F}(\cdot)}(\cdot) = 1 \mid \tilde{F} \leftarrow \text{Func}[in, out]] < \text{negl}$$

- Size of $\text{Func}[in, out]$

size $2^{out} \cdot 2^{in}$

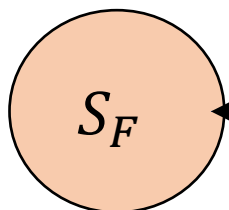


2^{in}

If $out = in = 128$, size is $(2^{128}) \cdot 2^{128}$

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

out



size 2^k

AES-128: 2^{128}

Concrete PRF

- AES-128/256/512
- $S_F = 2^{128}, 2^{256}, 2^{512}$

IND-CPA secure Π_2

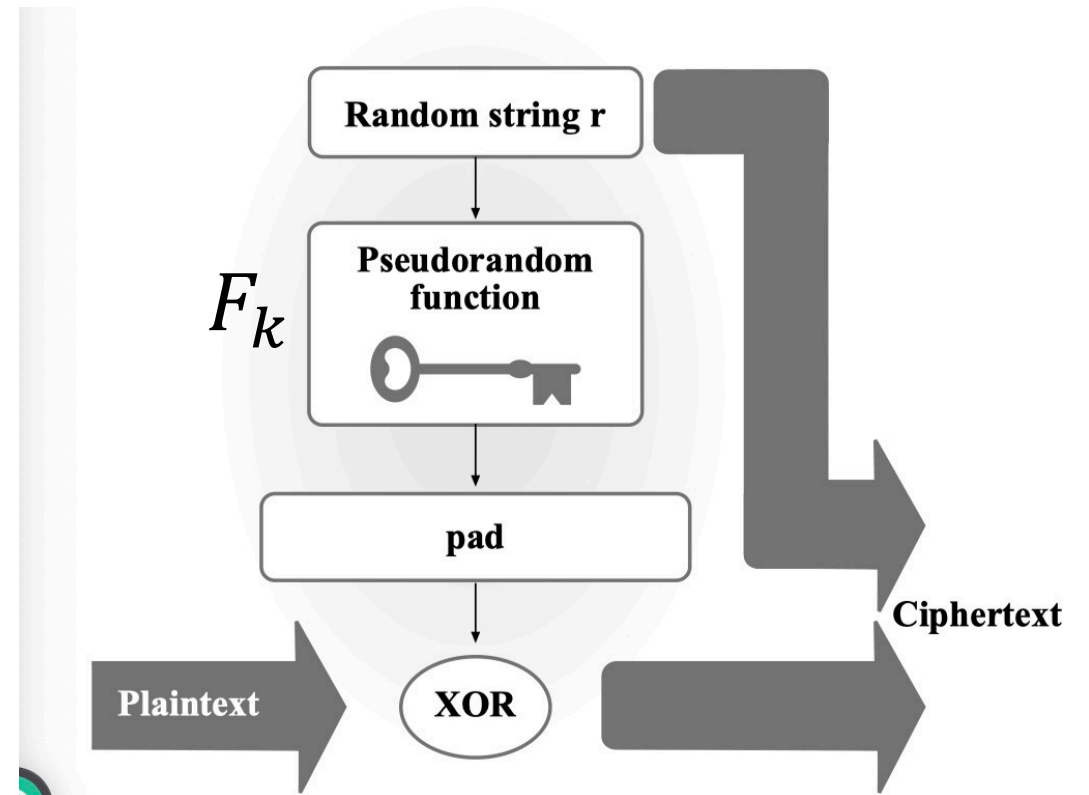
Let F_k be a PRF

Alg Π_2 . Enc(K, M)

1. $r \leftarrow \{0, 1\}^n$
2. $c_2 = F_k(r) \oplus M$
3. **return** $\langle r, c_2 \rangle$

Alg Π_2 . Dec(K, C)

1. **return** $c_2 \oplus F_k(r)$



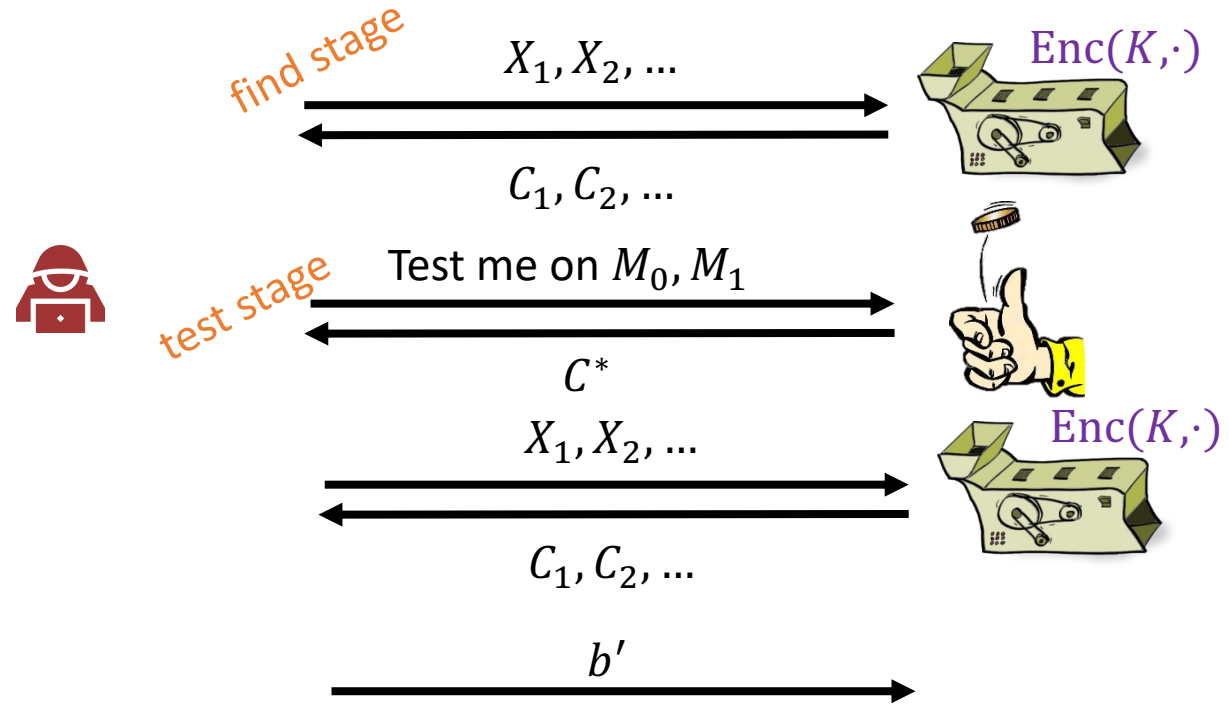
Proof idea: IND-CPA (choose plaintext attack)

Exp _{Π_2} ^{ind-cpa}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi_2.\text{Gen}$
3. $M_0, M_1 \leftarrow A^{\text{Enc}(K, \cdot)}$ // find stage
- 4.
- 5.
6. $C^* \leftarrow \langle r^*, F_K(r^*) \oplus M_b \rangle$ // test stage
7. $b' \leftarrow A^{\text{Enc}(K, \cdot)}(C^*)$ // guess stage
8. **return** $b' \stackrel{?}{=} b$

$\text{Enc}(K, M)$

-
1. **return** $\langle r, F_K(r) \oplus M \rangle$



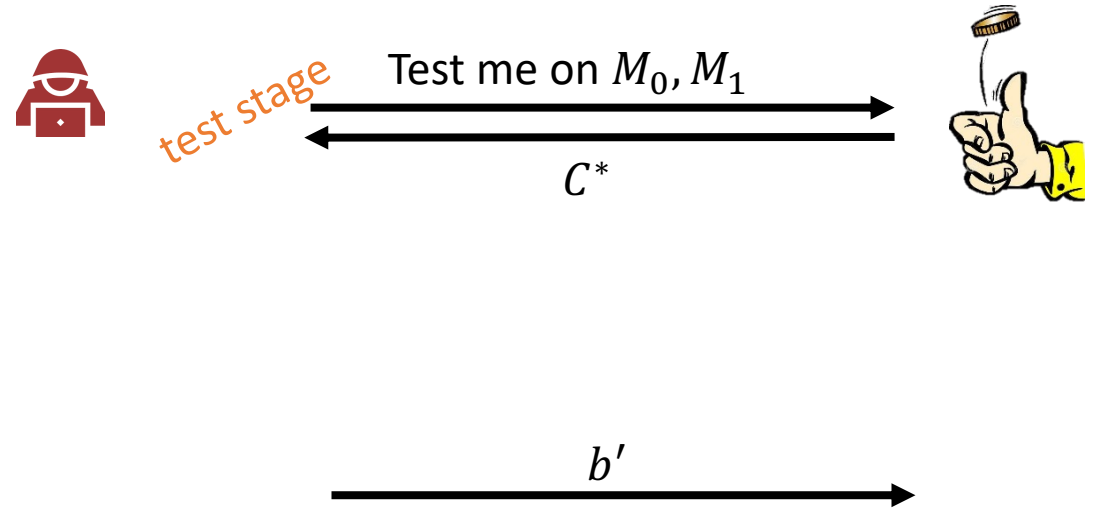
Proof idea: IND-CPA (choose plaintext attack)

Exp_{Π2}^{ind-cpa}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi2. \text{Gen}$
3. $M_0, M_1 \leftarrow A^{\text{Enc}(K, \cdot)}$ // find stage
- 4.
- 5.
6. $C^* \leftarrow \langle r^*, F_K(r^*) \oplus M_b \rangle$ // test stage
7. $b' \leftarrow A^{\text{Enc}(K, \cdot)}(C^*)$ // guess stage
8. **return** $b' \stackrel{?}{=} b$

Enc(K, M)

-
1. **return** $\langle r, F_K(r) \oplus M \rangle$ $\langle r, \tilde{F}(r) \oplus M \rangle$



Step 1: Due to PRF

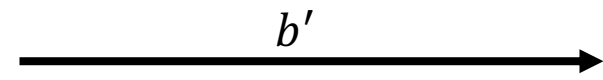
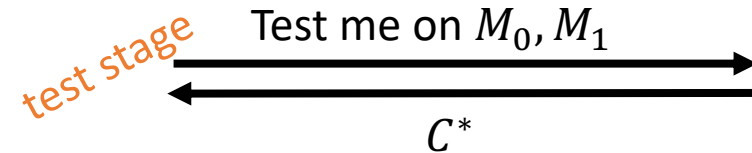
Proof idea: IND-CPA (choose plaintext attack)

Exp_{Π2}^{ind-cpa}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi2.\text{Gen}$
3. $M_0, M_1 \leftarrow A^{\text{Enc}(K, \cdot)}$ // find stage
- 4.
- 5.
6. $C^* \leftarrow \langle r^*, F_K(r^*) \oplus M_b \rangle$ / $\langle r^*, \tilde{F}(r^*) \oplus M_b \rangle$
7. $b' \leftarrow A^{\text{Enc}(K, \cdot)}(C^*)$ // guess stage
8. **return** $b' \stackrel{?}{=} b$

Enc(K, M)

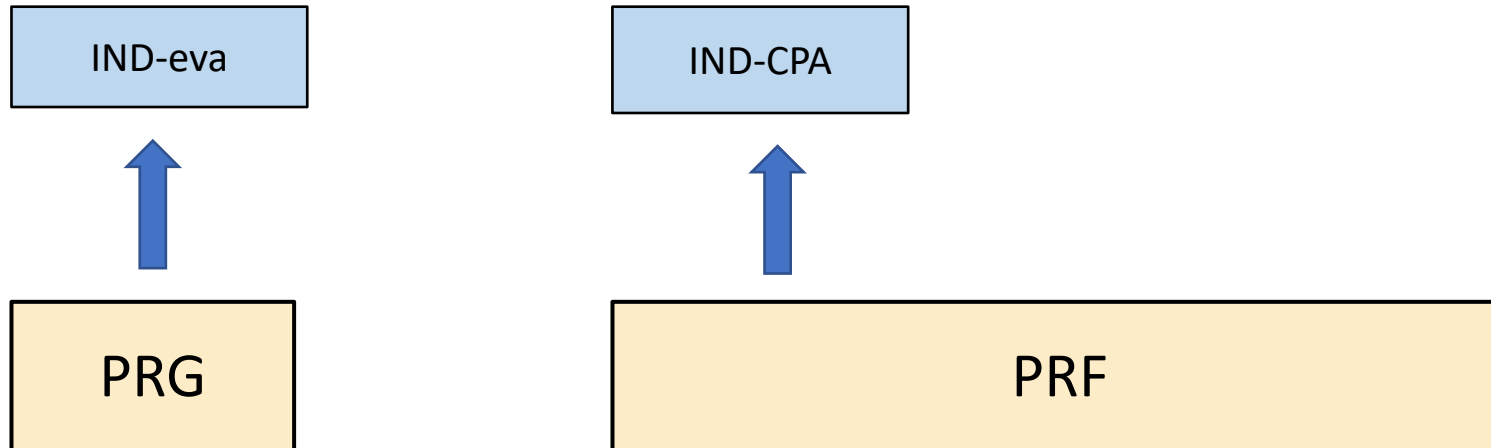
-
1. **return** $\langle r, F_K(r) \oplus M \rangle$ / $\langle r, \tilde{F}(r) \oplus M \rangle$



Step 1: Due to PRF

Step 2: Due to PRF

A short summary

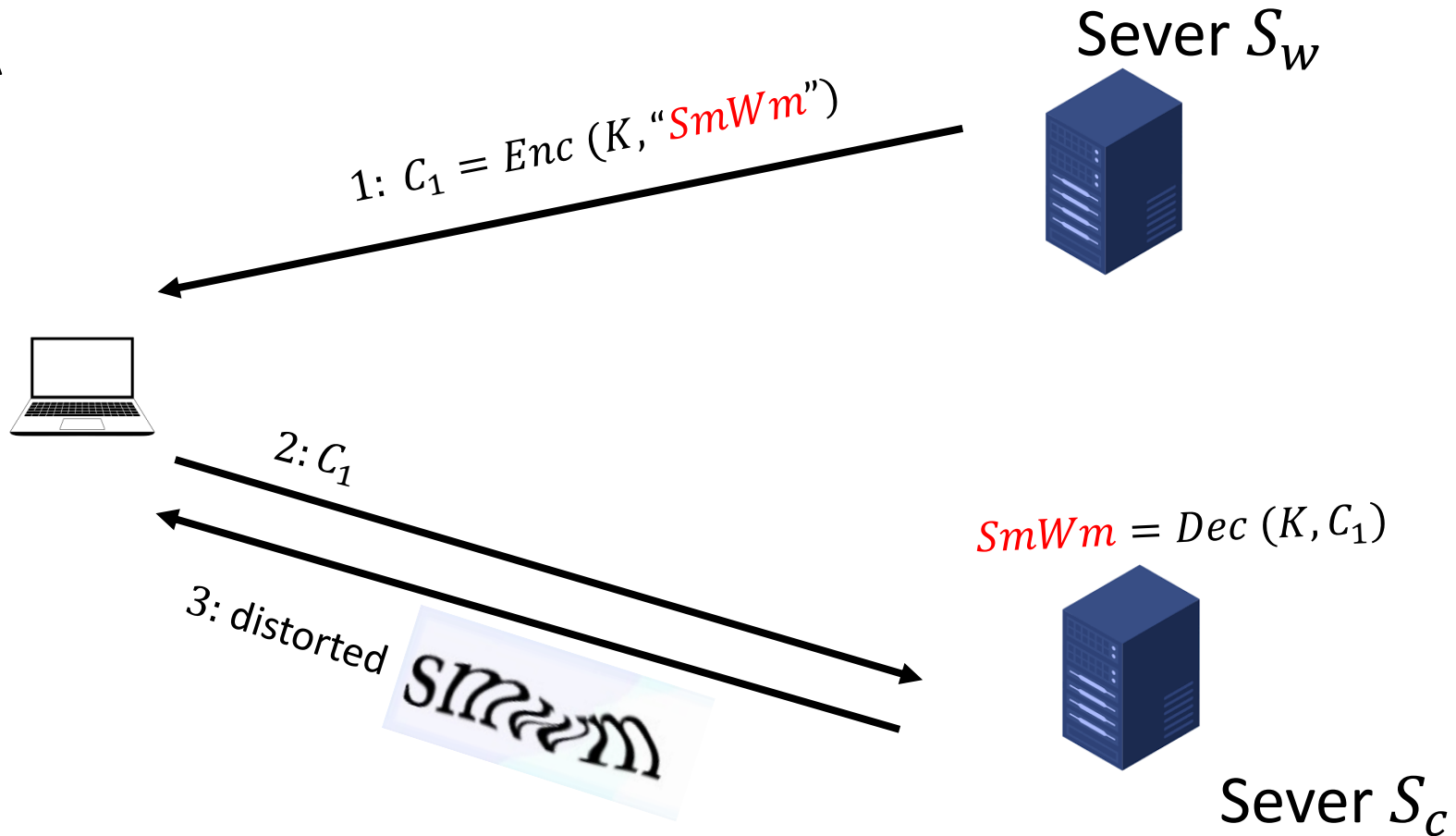
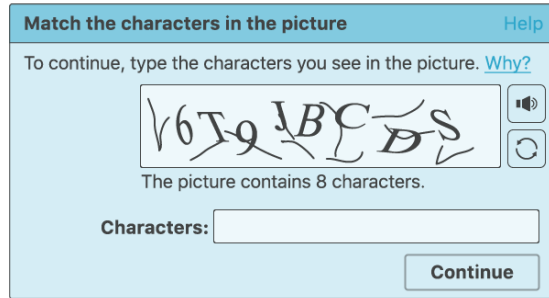


A short summary

- Define IND-CPA is necessary
- Π_1 is not IND-CPA secure
- With PRF in hand, we can construct generic IND-CPA secure Enc
- Stronger security????

Stronger Security: IND-CCA

- Example CAPTCHA



An adversary may have the capability to choose a ciphertext and get the message

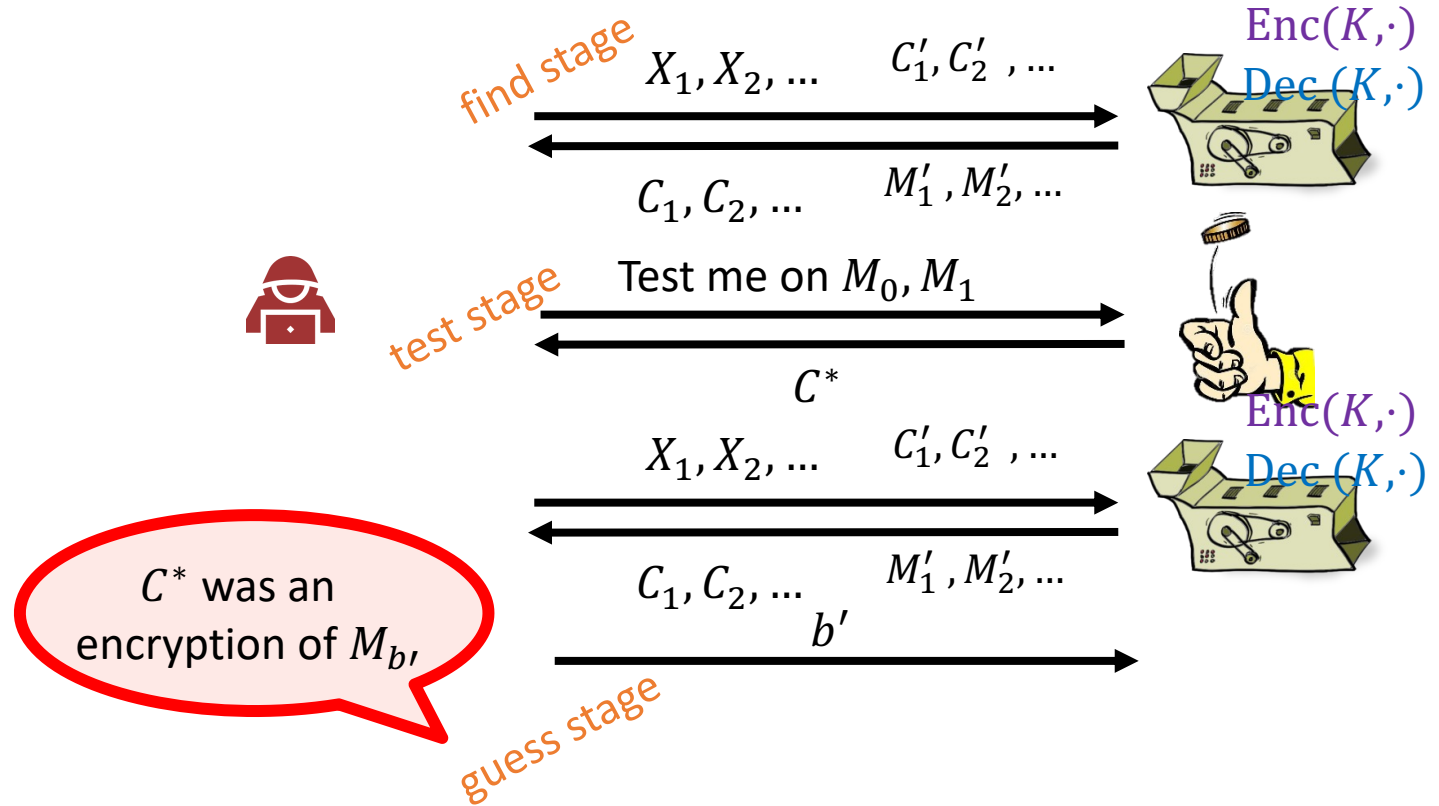
IND-CCA (choose ciphertext attack)

Exp_Π^{ind-cpa}(A)

1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi. \text{Gen}$
3. $M_0, M_1 \leftarrow A^{Enc(K, \cdot)}$ // find
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow \Pi. \text{Enc}(K, M_b)$ // test
7. $b' \leftarrow A^{Enc(K, \cdot)}(C^*)$ // guess
8. return $b' \stackrel{?}{=} b$

$Enc(K, M)$

-
1. return $\Pi. \text{Enc}(K, M)$



IND-CCA (choose ciphertext attack)

$\mathbf{Exp}_{\Pi}^{\text{ind-cca}}(A)$

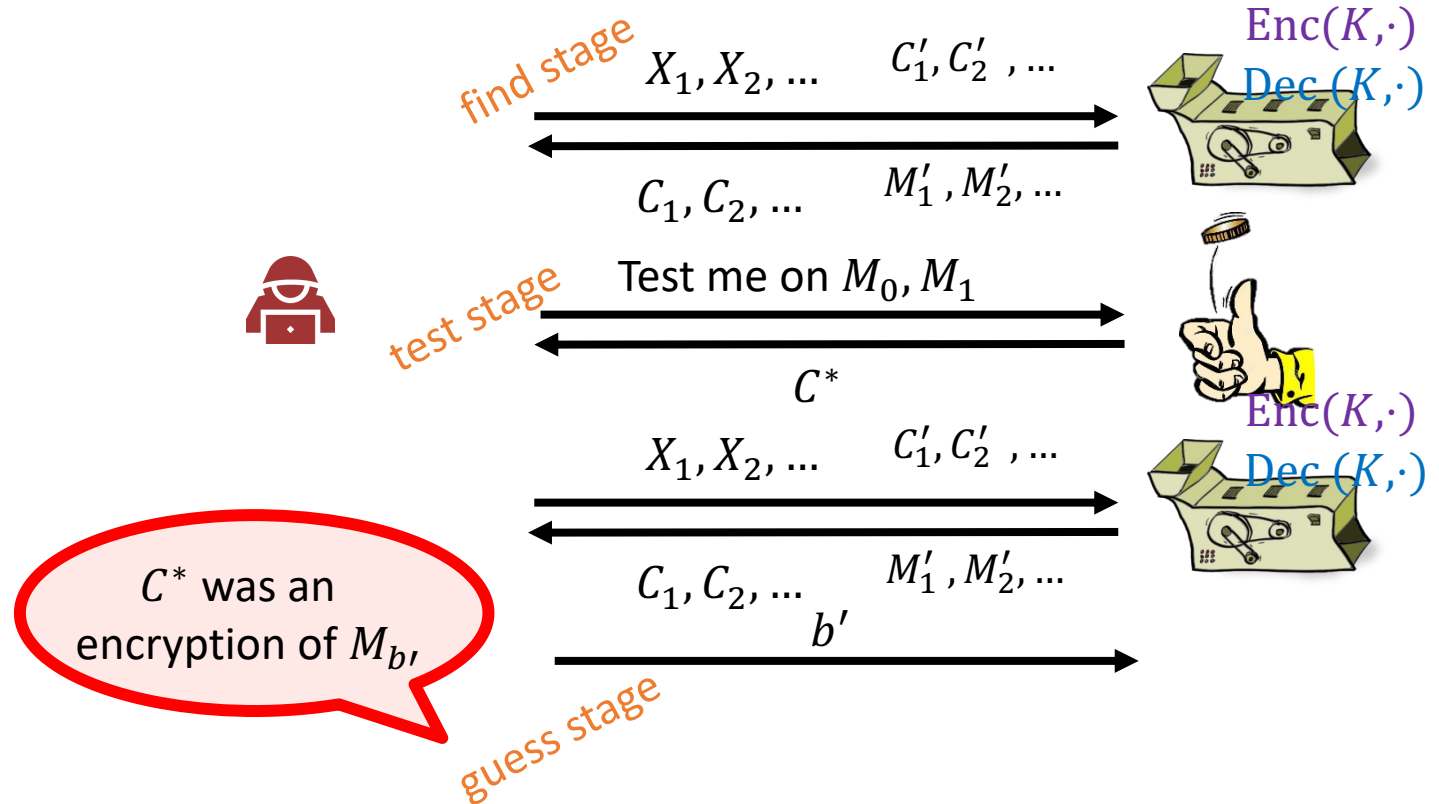
1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi.\text{Gen}$
3. $M_0, M_1 \leftarrow A^{\text{Enc}(K,\cdot)\text{Dec}(K,\cdot)}$ // find
4. if $|M_0| \neq |M_1|$ then
5. return \perp
6. $C^* \leftarrow \Pi.\text{Enc}(K, M_b)$ // test
7. $b' \leftarrow A^{\text{Enc}(K,\cdot)\text{Dec}(K,\cdot)}(C^*)$ // guess
8. return $b' \stackrel{?}{=} b$

$\text{Enc}(K, M)$

1. return $\Pi.\text{Enc}(K, M)$

$\text{Dec}(K, C), C \neq C^*$

1. return $\Pi.\text{Dec}(K, C)$



Definition: The IND-CCA-advantage of an adversary A is

$$\mathbf{Adv}_{\Pi}^{\text{ind-cca}}(A) = |\Pr[\mathbf{Exp}_{\Pi}^{\text{ind-cca}}(A) \Rightarrow 1] - 1/2|$$

IND-CCA Insecurity of Π_2

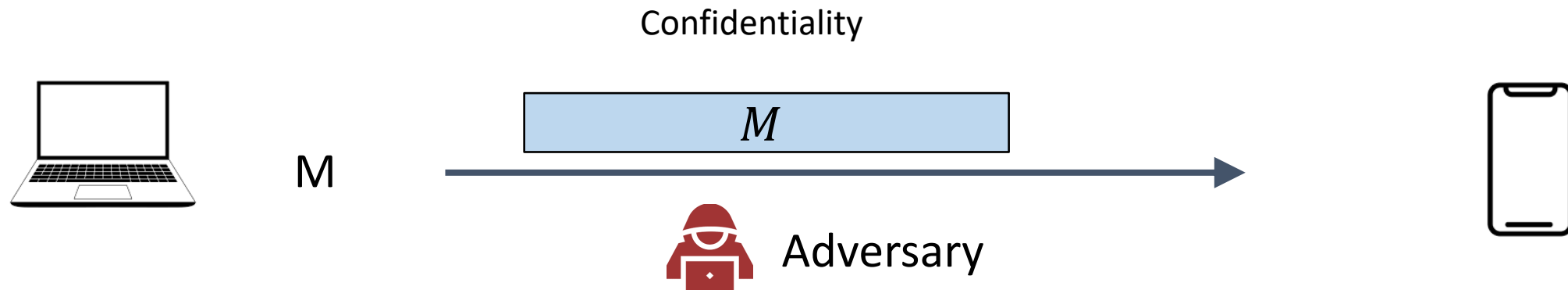
Adversary A

1. On receiving $C^* = \langle r^*, F_K(r^*) \oplus M_b \rangle$
2. Query $C = \langle r^*, F_K(r^*) \oplus M_b \oplus M_0 \rangle$ to Dec
3. On receiving $M_0 \oplus M_0$, set $b=0$
4. otherwise, $b=1$

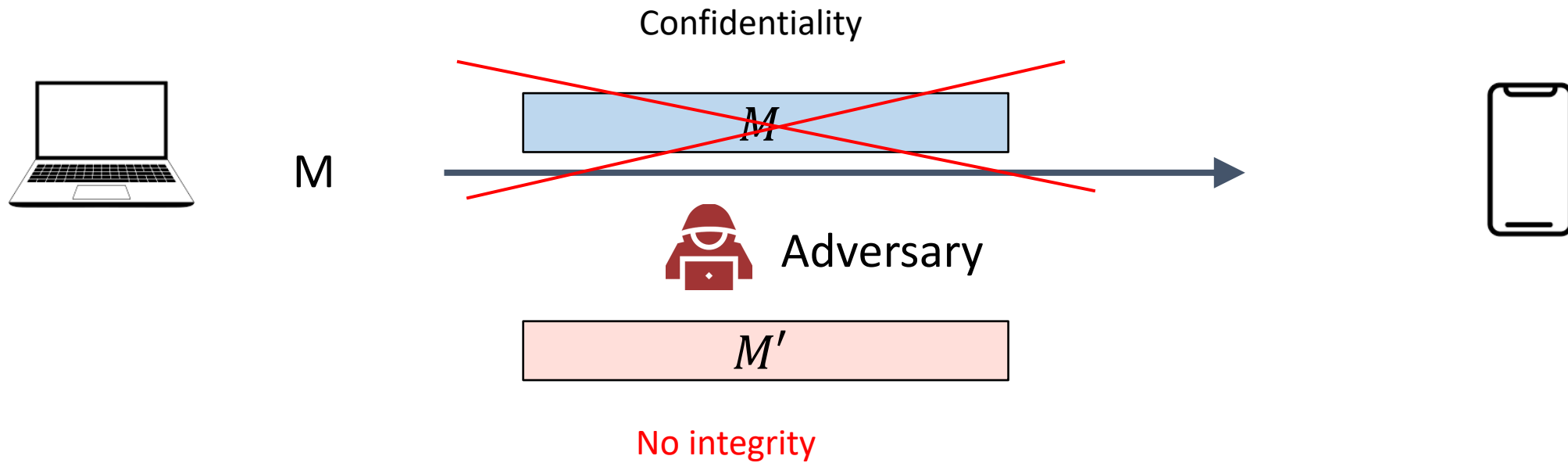
Constructions

- We leave the construction of CCA secure Enc in the following part
- after introducing MAC

Message Authenticated Code

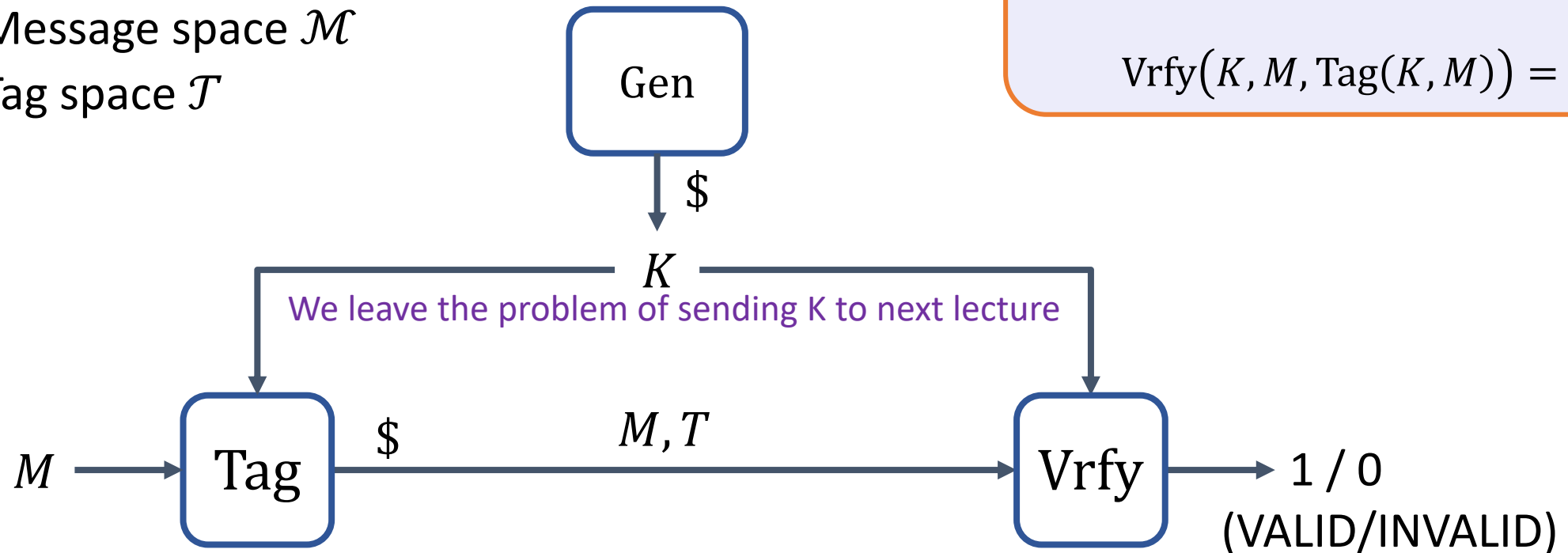


Message Authenticated Code



Message authentication code (MAC)– syntax

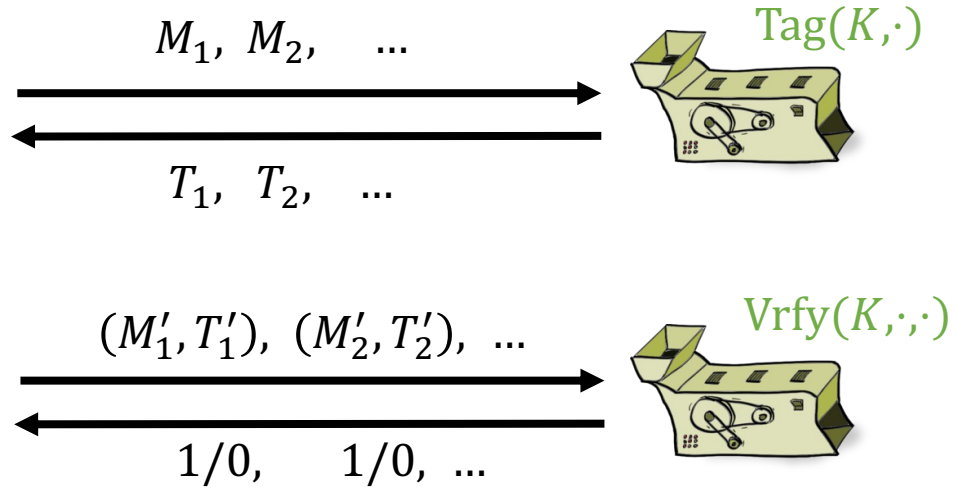
- A **message authentication scheme** $\Pi = (\text{Gen}, \text{Tag}, \text{Vrfy})$ consists of three public algorithms:
- Associated to Π :
 - Key space \mathcal{K}
 - Message space \mathcal{M}
 - Tag space \mathcal{T}



UF-CMA secure MAC



Challenger

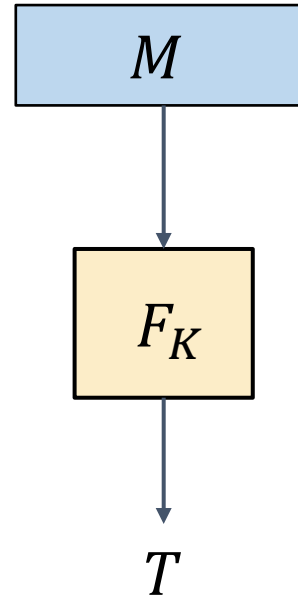


Adversary *wins* if a pair (M'_i, T'_i) is valid,
and was not among the pairs $(M_1, T_1), (M_2, T_2), \dots$

PRFs are good MACs

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

PRF



Alg Σ_{PRF} . Tag(K, M)

1. if $M \notin \{0,1\}^{in}$ then
2. return \perp
3. return $F_K(M)$

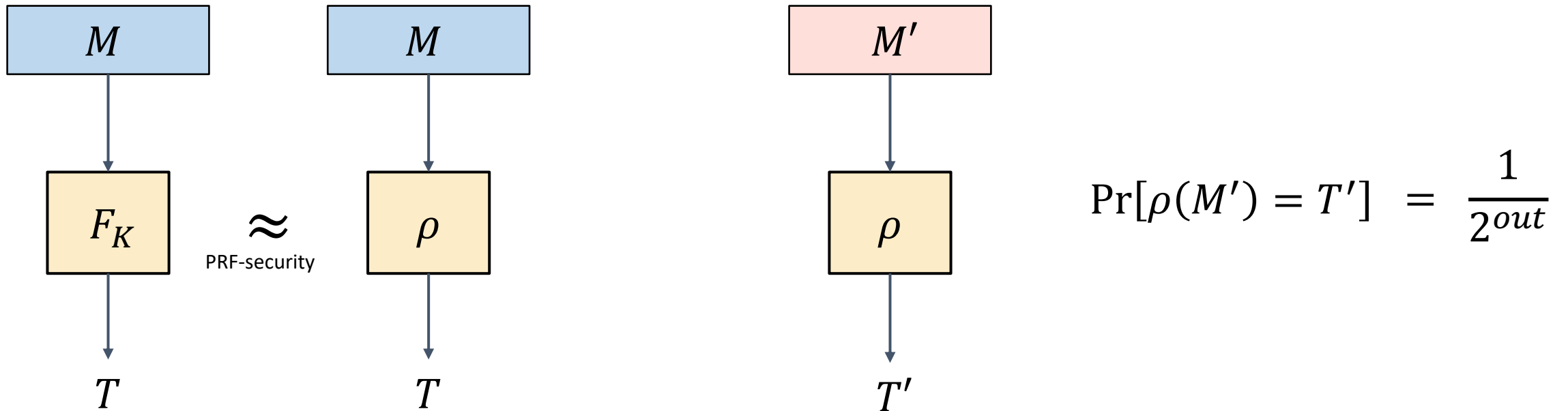
Alg Σ_{PRF} . Vrfy(K, M, T)

1. $T' \leftarrow F_K(M)$
2. return $T' \stackrel{?}{=} T$

Theorem: If F is a secure PRF then Σ_{PRF} is UF-CMA secure for *fixed-length* messages $M \in \{0,1\}^{in}$

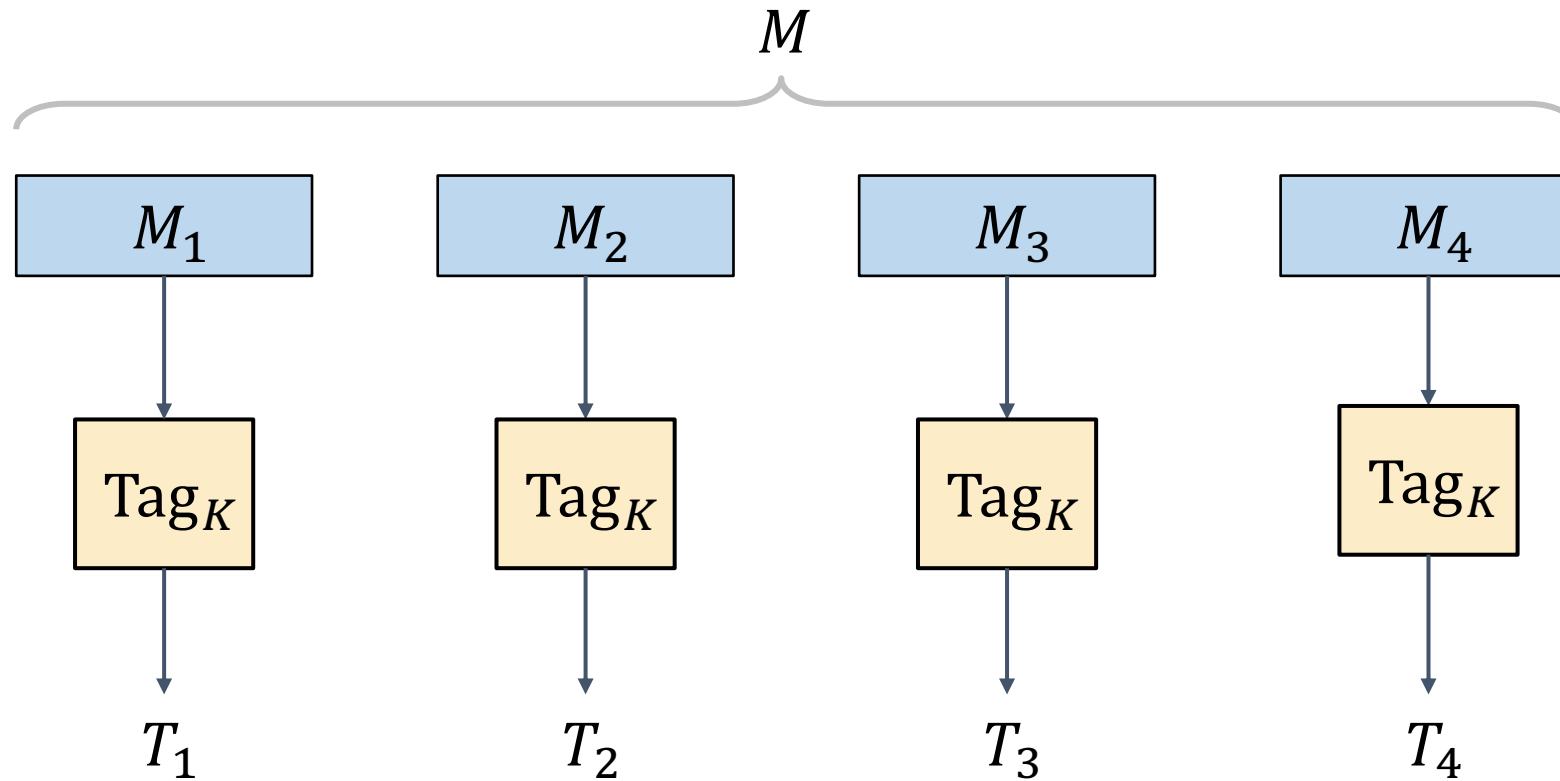
PRFs are good MACs – proof sketch

Theorem: If F is a secure PRF then Σ_{PRF} is UF-CMA secure for *fixed-length* messages $M \in \{0,1\}^{in}$



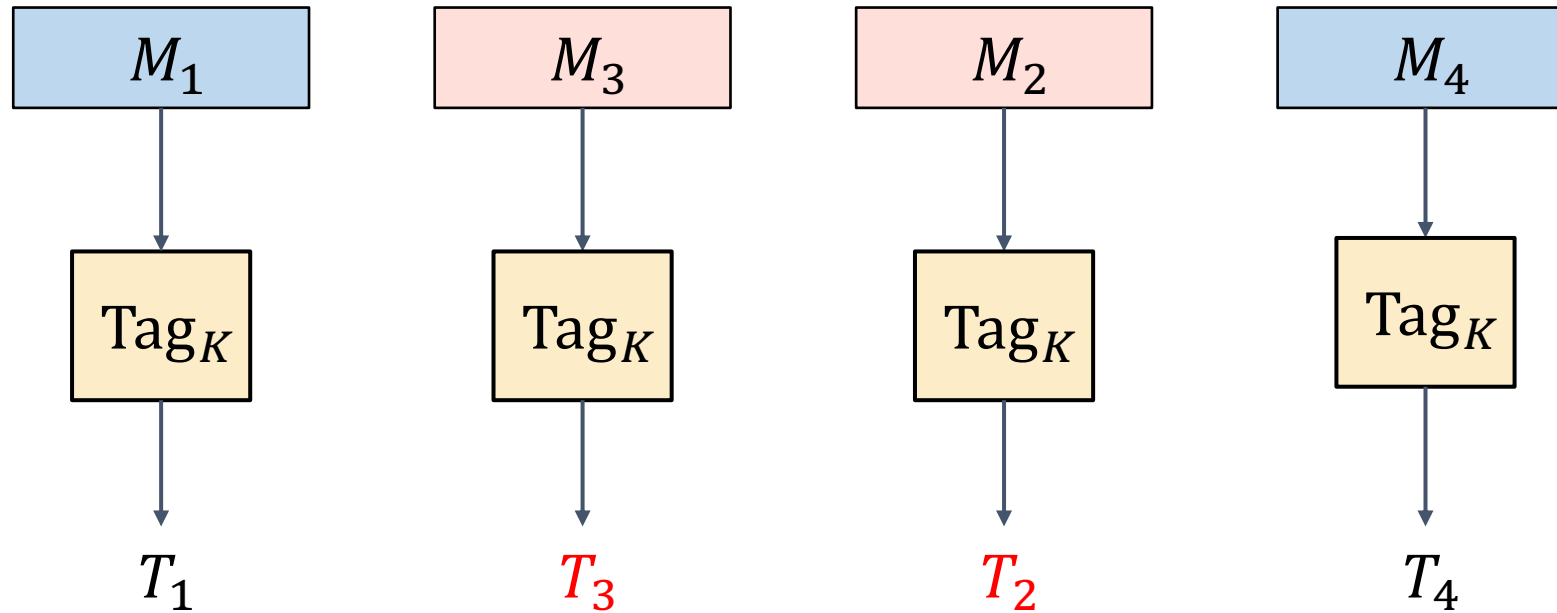
$$\rho \stackrel{\$}{\leftarrow} \text{Func}[in, out]$$

MAC for longer message Attempt 1:EBC



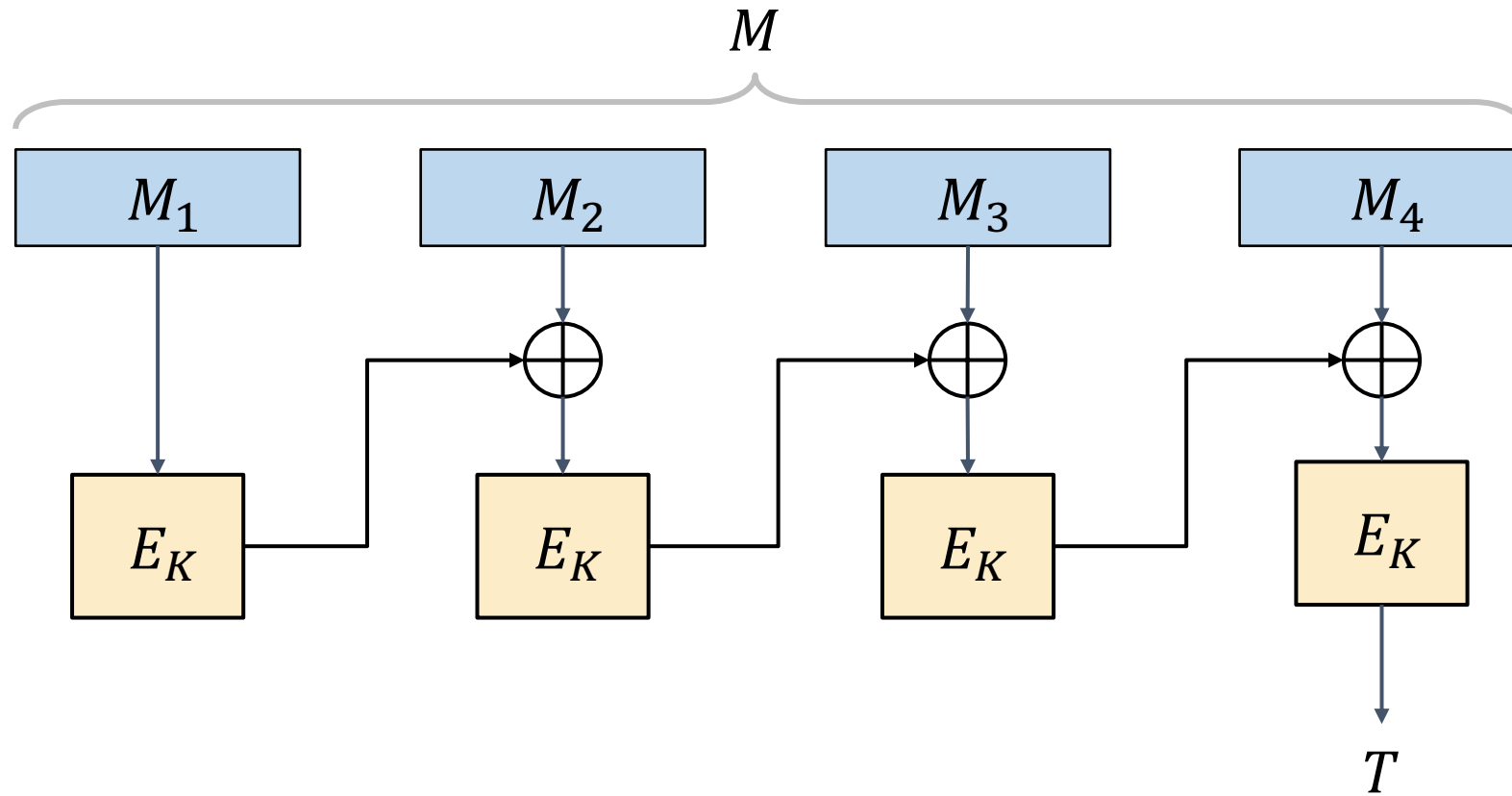
$$T = T_1 || T_2 || T_3 || T_4$$

Attempt 1 – an attack



$$T = T_1 || T_3 || T_2 || T_4$$

CBC-MAC

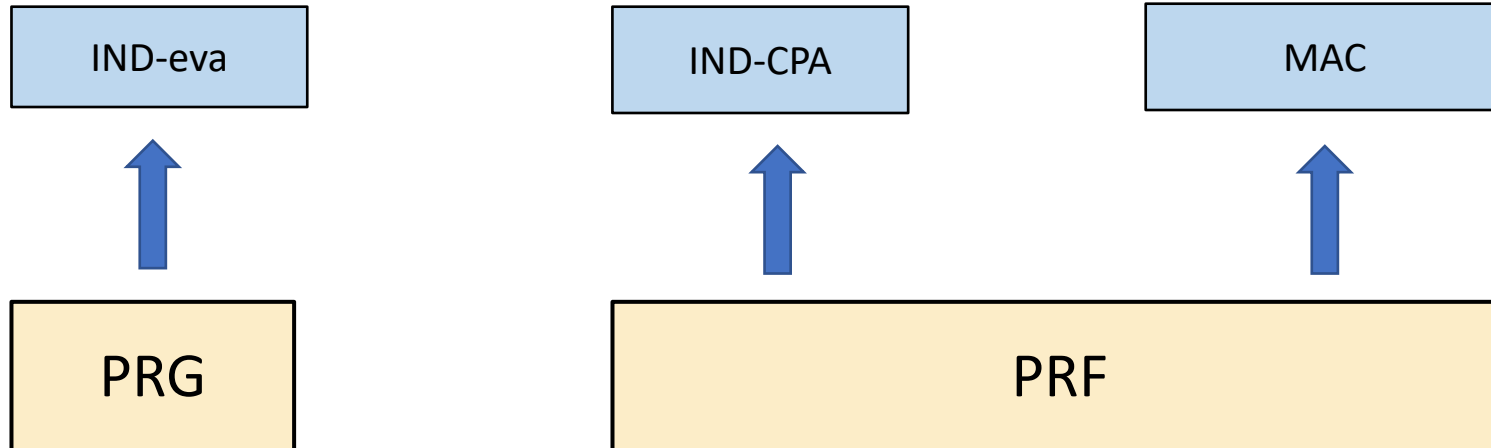


✓ Secure

A short summary

- IND-CCA security is necessary
- Existing studied schemes are not IND-CCA secure
- MAC could be used to provide integrity.
- With IND-CPA enc and MAC, we are ready to construct IND-CCA

A short summary



Recall IND-CCA

Exp_Π^{ind-cpa}(A)

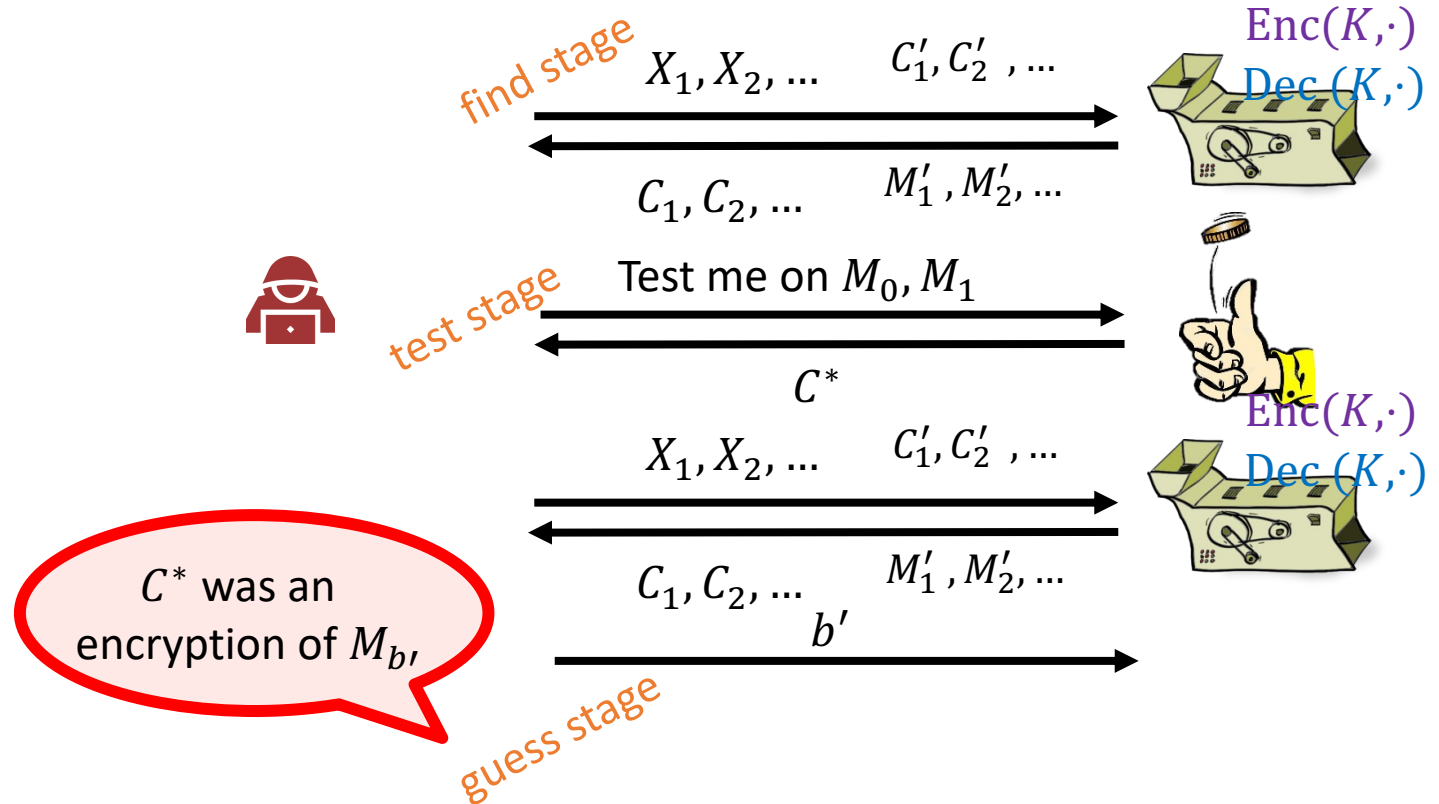
1. $b \xleftarrow{\$} \{0,1\}$
2. $K \xleftarrow{\$} \Pi. \text{Gen}$
3. $M_0, M_1 \leftarrow A^{Enc(K,\cdot)Dec(K,\cdot)}$ // find
- 4.
- 5.
6. $C^* \leftarrow \Pi. \text{Enc}(K, M_b)$ // test
7. $b' \leftarrow A^{Enc(K,\cdot)Dec(K,\cdot)}(C^*)$ // guess
8. **return** $b' \stackrel{?}{=} b$

Enc(K, M)

-
1. **return** $\Pi. \text{Enc}(K, M)$

*Dec(K, C), C ≠ C**

-
1. **return** $\Pi. \text{Dec}(K, C)$

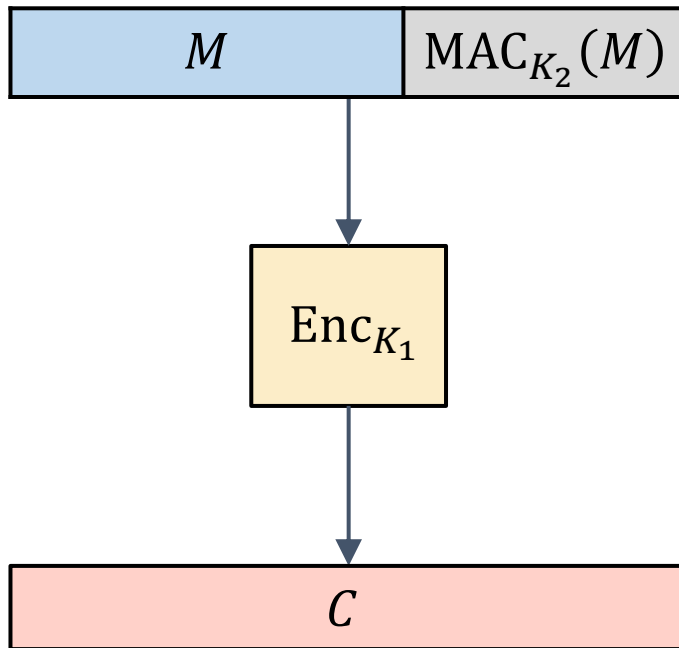


Definition: The IND-CCA-advantage of an adversary A is

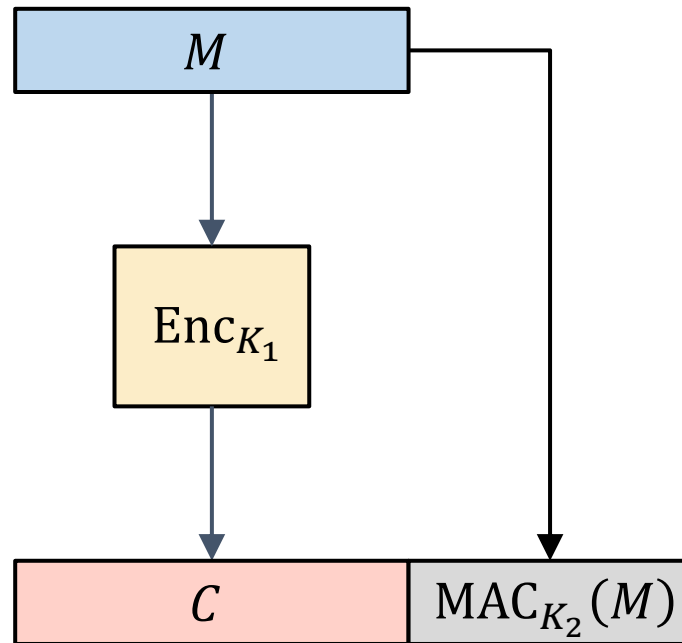
$$\text{Adv}_{\Pi}^{\text{ind-cca}}(A) = |\Pr[\mathbf{Exp}_{\Pi}^{\text{ind-cca}}(A) \Rightarrow 1] - 1/2|$$

Generic composition: IND-CPA + MAC? \rightarrow IND-CCA

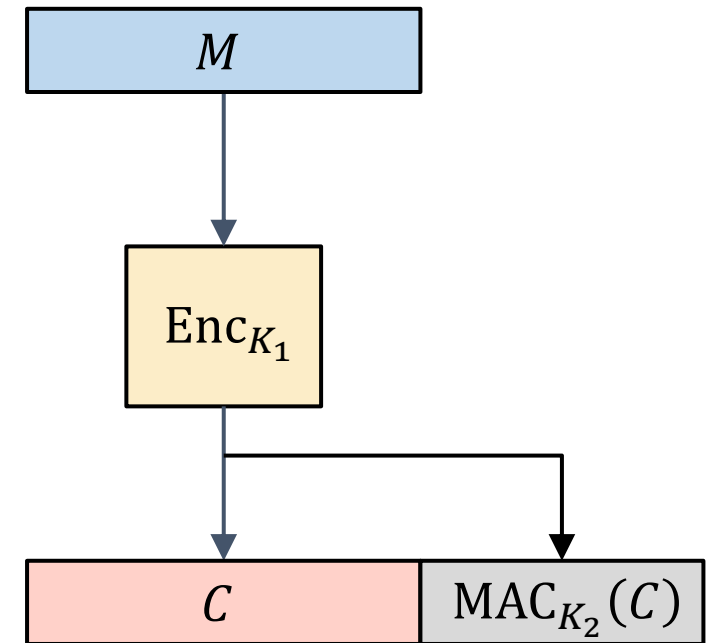
MAC-then-Encrypt (MtE)



Encrypt-and-MAC (E&M)



Encrypt-then-MAC (EtM)



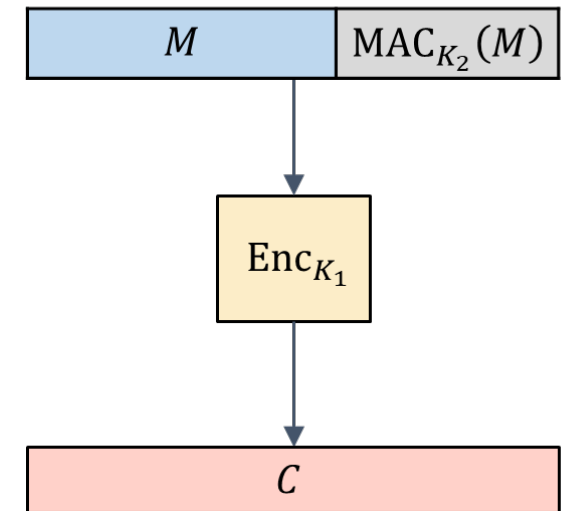
First Attempt: MAC-then-Encrypt (MtE)

- If $Enc(K, M)$ is IND-CPA secure,
- $r || Enc(K, M)$ is also IND-CPA secure, where r is a random bit
- If $Enc_K(\cdot) = r || Enc(K, \cdot)$

CCA Adversary A

1. Query $\bar{r} || Enc(K, M, MAC_{K_2}(M))$ to Dec

MAC-then-Encrypt (MtE)

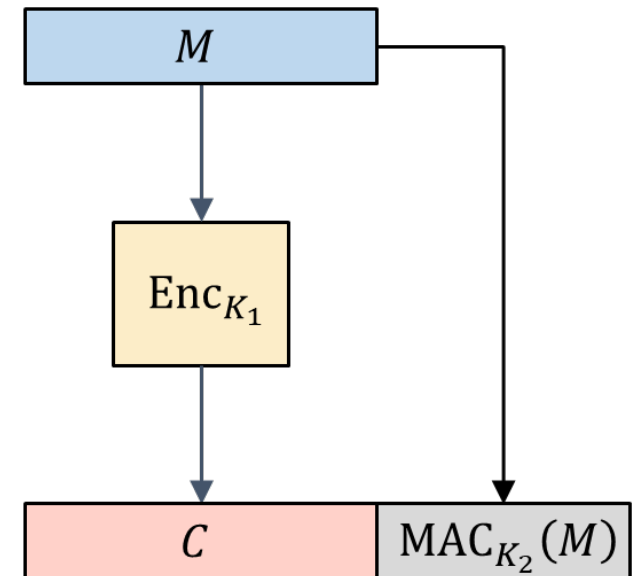


Second Attempt: Encrypt-and-MAC (E&M)

- If $MAC_k(M)$ is a UF secure MAC,
- $M || MAC_k(M)$ is also a UF secure MAC

MAC does not provide confidentiality to the input

Encrypt-and-MAC (E&M)



Encrypt-then-MAC (EtM)

Let $\Pi_2 = (\text{Enc}, \text{Dec})$ be an IND-CPA enc

Let $\Pi_m = (\text{Tag}, \text{Vrfy})$ be a secure MAC

Alg $\Pi_3.$ Gen

1. **return** random $K = (K_1, K_2)$

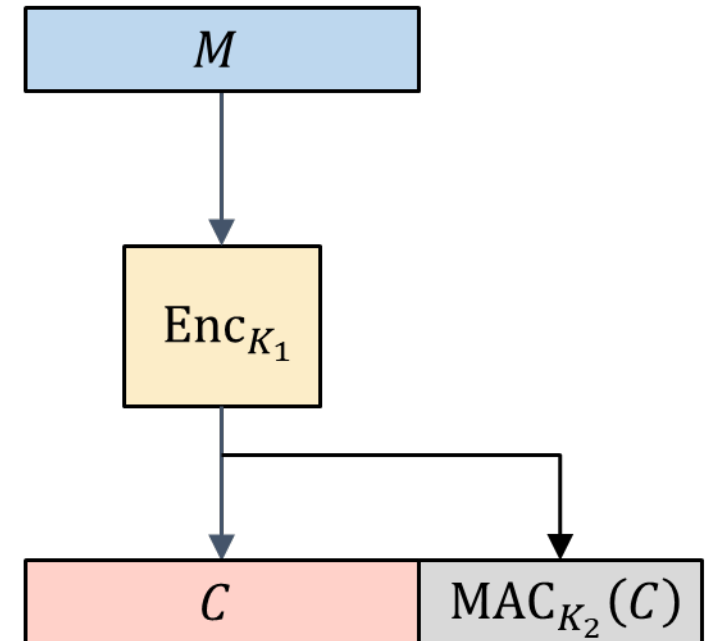
Alg $\Pi_3.$ Enc(K, M)

1. $C = \Pi_2.$ Enc(K_1, M)
2. **return** $\langle C, \text{Tag}(K_2, C) \rangle$

Alg $\Pi_3.$ Dec($K, c_1 || c_2$)

1. **return** $\Pi_2.$ Dec(K_2, c_1) if $\text{Vrfy}(K_2, c_1, c_2) = 1$

Encrypt-then-MAC (EtM)



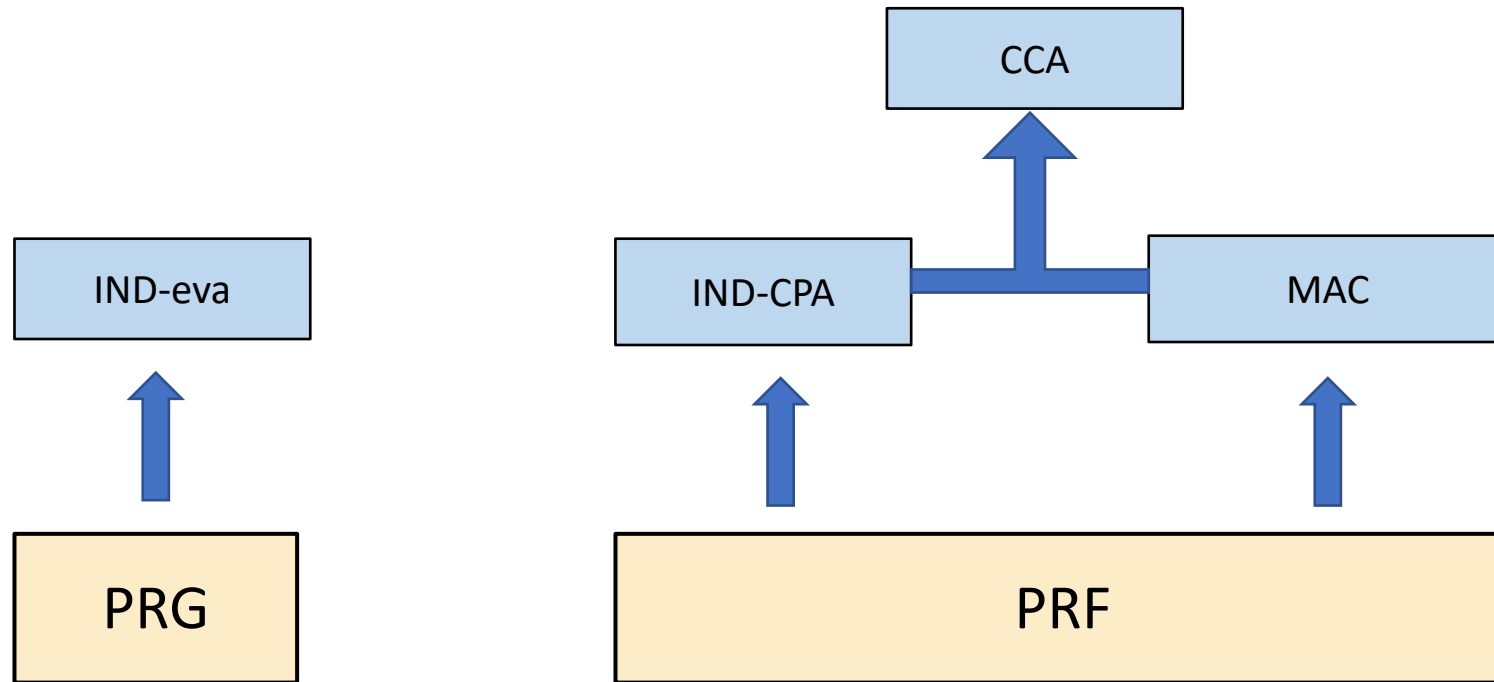
Proof idea: IND-CCA

Please refer to [KL20, Theorem 4.19] for the proof

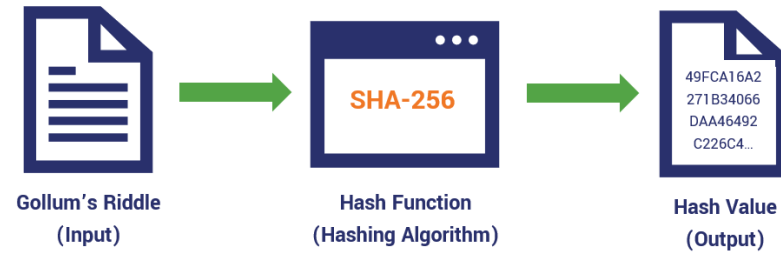
A short summary

- IND-CCA security is necessary
- We could construct an IND-CCA secure scheme from IND-CAP + MAC using Encrypt-then-MAC (EtM)

A short summary



Hash function



<https://www.thesslstore.com/blog/what-is-a-hash-function-in-cryptography-a-beginners-guide/>

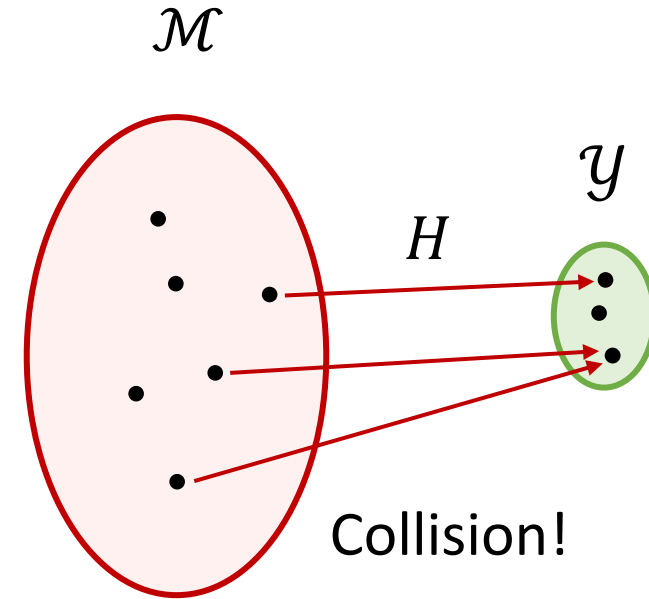
Hash functions

$$H : \mathcal{M} \rightarrow \mathcal{Y}$$

Keyless function

$$|\mathcal{M}| \gg |\mathcal{Y}|$$

Compressing



- SHA1 *: $\{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{160}$
- SHA2-256 : $\{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{256}$
- SHA3-512 : $\{0,1\}^{<2^{128}} \rightarrow \{0,1\}^{512}$

Collision Resistant

One way

Collision resistance

Exp_H^{cr}(A)

1. $(X_1, X_2) \leftarrow A_H$
2. **if** $X_1 \neq X_2$ **and** $H(X_1) = H(X_2)$ **then**
3. **return** 1
4. **else**
5. **return** 0

A

1. Output (X_1, X_2) where X_1, X_2 is a collision for H

X_1, X_2 must *exist* since $|\mathcal{M}| \gg |y|$

hence $\mathbf{Adv}_H^{\text{cr}}(A) = 1$ for unbounded A

...but how do we actually find X_1, X_2 ?!

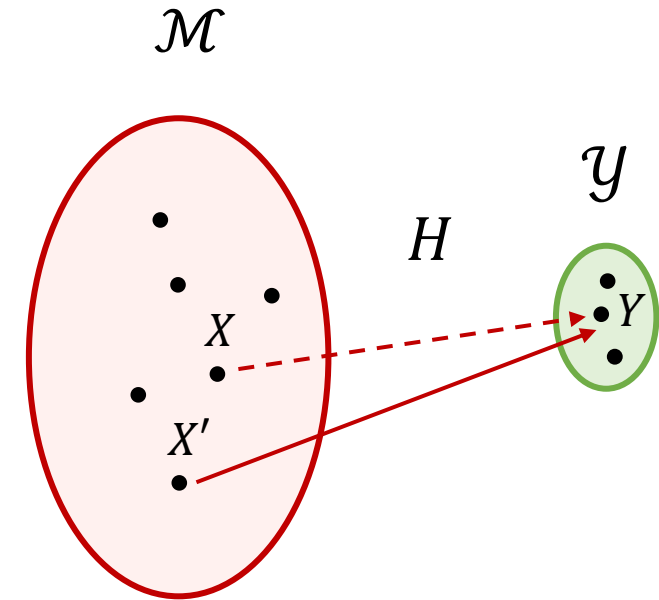
Definition: The **CR-advantage** of an adversary A against H is

$$\mathbf{Adv}_H^{\text{cr}}(A) = \Pr[\mathbf{Exp}_H^{\text{cr}}(A) \Rightarrow 1]$$

One-way security

$\mathbf{Exp}_H^{\text{ow}}(A)$

1. $X \stackrel{\$}{\leftarrow} \mathcal{M}$
2. $Y \leftarrow H(X)$
3. $X' \leftarrow A_H(Y)$
4. **return** $H(X') \stackrel{?}{=} Y$



Definition: The **OW-advantage** of an adversary A against H is

$$\mathbf{Adv}_H^{\text{ow}}(A) = \Pr[\mathbf{Exp}_H^{\text{cr}}(A) \Rightarrow 1]$$

Relation between notions

Exp_H^{cr}(A)

1. $(X_1, X_2) \leftarrow A_H$
2. **if** $X_1 \neq X_2$ **and** $H(X_1) = H(X_2)$ **then**
3. **return** 1
4. **else**
5. **return** 0

Exp_H^{ow}(A)

1. $X \xleftarrow{\$} \mathcal{M}$
2. $Y \leftarrow H(X)$
3. $X' \leftarrow A_H(Y)$
4. **return** $H(X') \stackrel{?}{=} Y$

Collision-resistance \implies One-wayness

Proof idea: suppose A_{ow} is an algorithm that breaks one-wayness

1. Pick $X \xleftarrow{\$} \mathcal{M}$ and give $Y \leftarrow H(X)$ to A_{ow}
2. A_{ow} outputs X'
3. output (X, X') as a collision ($H(X') = Y = H(X)$)

Problem: what if $X' = X$? Very unlikely assuming $|\mathcal{M}| \gg |\mathcal{Y}|$

Relation between notions

Exp_H^{cr}(A)

1. $(X_1, X_2) \leftarrow A_H$
2. **if** $X_1 \neq X_2$ **and** $H(X_1) = H(X_2)$ **then**
3. **return** 1
4. **else**
5. **return** 0

Exp_H^{ow}(A)

1. $X \xleftarrow{\$} \mathcal{M}$
2. $Y \leftarrow H(X)$
3. $X' \leftarrow A_H(Y)$
4. **return** $H(X') \stackrel{?}{=} Y$

Collision-resistance \implies One-wayness

Collision-resistance $\not\Leftarrow$ One-wayness

Suppose $H : \mathcal{M} \rightarrow \{0,1\}^{256}$ is one-way. Define

$$H'(X) = \begin{cases} 0^{256} & \text{if } X = 0 \text{ or } X = 1 \\ H(X) & \text{otherwise} \end{cases}$$

H' is one-way

H' is **not** collision-resistant

Application– MAC domain extension (HMAC)

$$\text{MAC} : \mathcal{K} \times \{0,1\}^n \rightarrow \mathcal{T} \quad H : \{0,1\}^* \rightarrow \{0,1\}^n$$

$$\text{MAC}' : \mathcal{K} \times \{0,1\}^* \rightarrow \mathcal{T}$$

$$\text{MAC}'(K, M) = \text{MAC}(K, H(M)) \quad \leftarrow \text{Hash-then-MAC paradigm}$$

Theorem: If H is collision-resistant and MAC is UF-CMA secure, then MAC' is UF-CMA secure

A short summary

- Hash functions are compressing functions
- Collision resistance and one-wayness are two properties of hash function
- Hash could be used to build HMAC

Summary

- Syntax and security of symmetric-key cryptography
- Perfect security and one-time pad
- Stream cipher, block cipher and MAC
- Hash function
- Constructions

Recap

Primitives	Security	Examples
Pseudorandom function (PRF)	Indistinguishability from random function	AES-128/256/512 HMAC
Encryption	IND-eva IND-CPA IND-CCA	PRG \$+PRF Enc-t-Mac
MAC	Integrity	PRF CBC-MAC HMAC
Authenticated Encryption	IND-CCA (+ unforgeable encryption)	IND-CPA+MAC AES-256-GCM
Hash function	Collision-resistance + one-wayness	SHA2-256 SHA2-512 SHA3

-
- Symmetric key encryption assumes two parties have a shared key K

We will talk in the next lecture **the problem of sending K**

Thank you

Questions