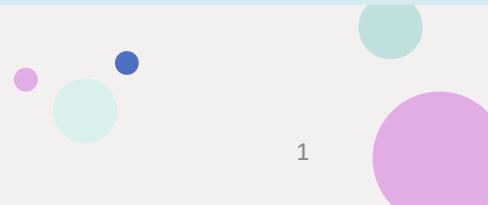




Lecture 9

CCA Security, MACs

CS409/6011: Introduction to Cryptography
Sruthi Sekar



Recap

- **CPA Security:**
 1. Encryption must be randomized in general if you want to reuse keys for multiple encryptions .
 2. CPA Security for Multiple encryptions comes for free.
 3. A CPA Secure scheme from PRFs.
- **Modes of Operation:** Practical ways of extending block ciphers for multiple encryptions.
 1. **ECB, CBC, OFB, CTR** modes (when implemented “correctly”)
 2. Some practical issues in using the above modes.

Need to go beyond eavesdroppers!

Today's Class



CCA Secure Encryption

Definition

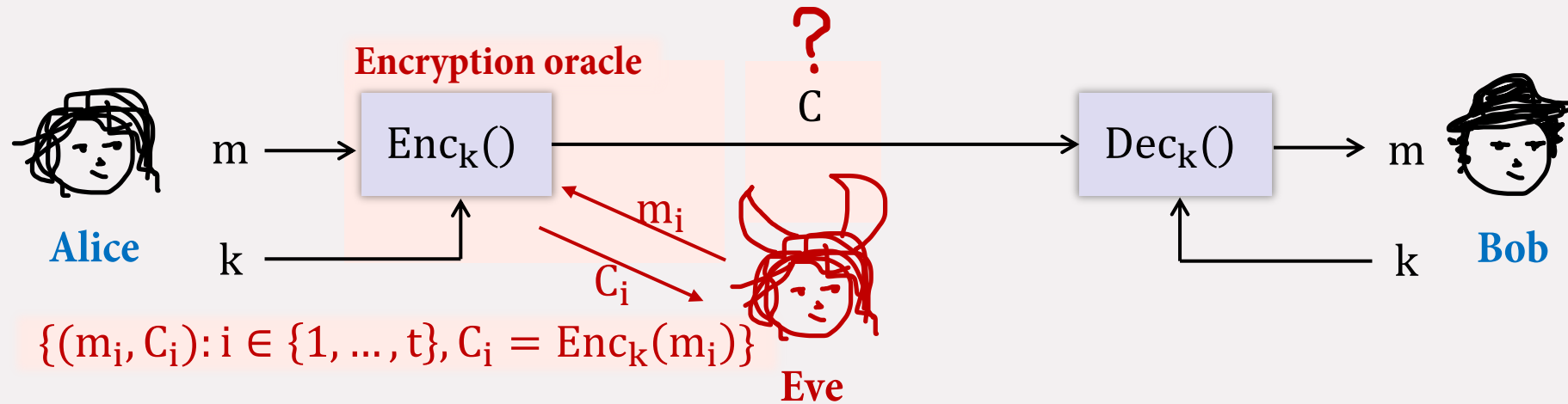
A CCA secure scheme



Message Authentication Codes

RECALL

Chosen Plaintext Attack (CPA)



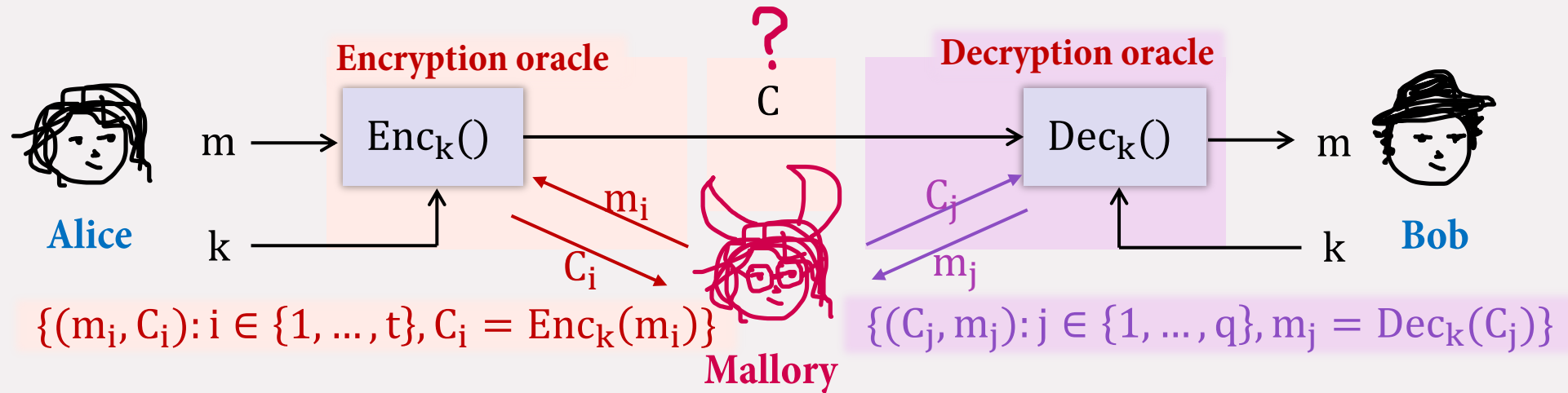
Chosen Plaintext Attack (CPA):

Eavesdropper with encryption oracle

PPT

- Eve influences Alice to encrypt messages of her choice (on the same key).
- Eve's goal is then to gain more information about message under a new ciphertext!

Chosen Ciphertext Attack (CCA)



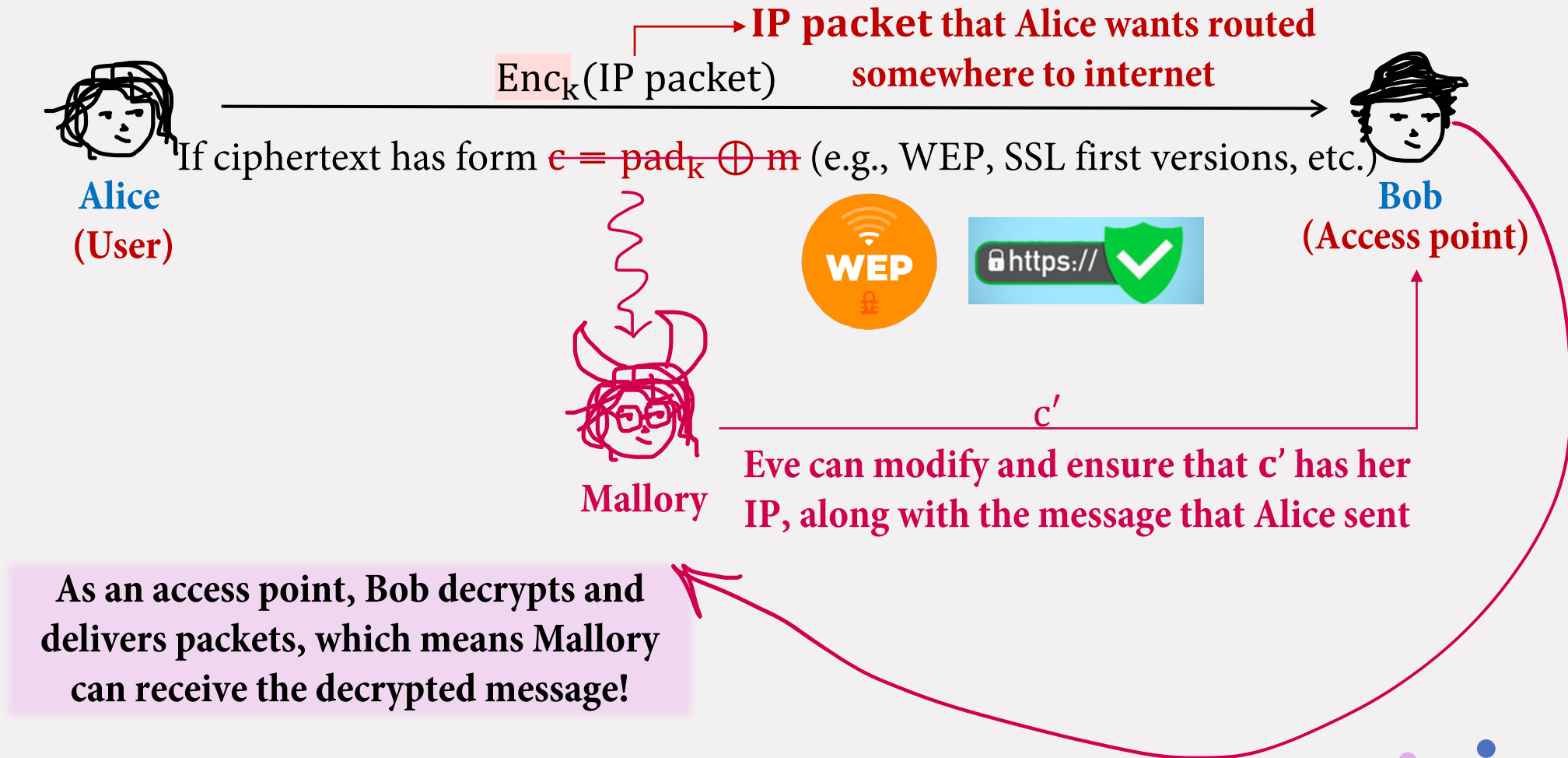
Chosen Ciphertext Attack (CCA):

~~Eavesdropper with encryption oracle~~
PPT

Active Mallory
(gets Enc & Dec oracle)

- **Mallory** influences Alice to encrypt messages of her choice (on the same key).
- **Mallory**'s goal is then to gain more information about message under a new ciphertext!
- **Mallory influences Bob to decrypt messages of her choice (on the same key).**

Decryption Oracle can be Catastrophic!



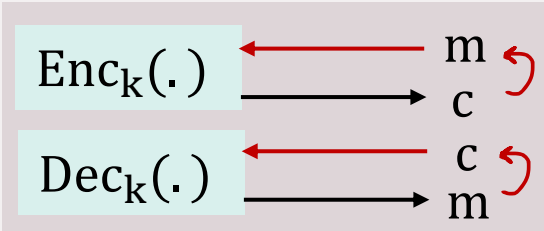
Security against CCA



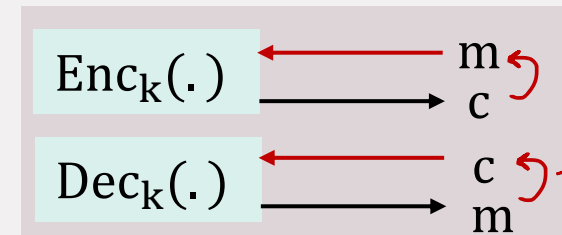
$k \leftarrow \text{Gen}(1^n)$

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

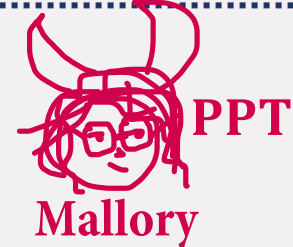
$\Pi = (\text{Gen}, \text{Enc}, \text{Dec}), \mathcal{M}$



$m_0 \neq m_1 \in \mathcal{M}$
 $|m_0| = |m_1|$
 $c_b \leftarrow \text{Enc}_k(m_b)$



$b' \in \{0,1\}$



Can include
queried messages!

Cannot
Query c_b

Eve wins if output is 1
 $\text{PrivK}_{\text{Mallory}, \Pi}^{\text{cca}}$ outputs 1, if $b' = b$
 and 0, o.w.

Definition 1

Π is **CCA secure** if, for all **PPT** adversary **Mallory**, there exists a negligible function **negl** such that $\forall n$

$$\Pr[\text{PrivK}_{\text{Mallory}, \Pi}^{\text{cca}}(n) = 1] \leq 1/2 + \text{negl}(n)$$

All CPA schemes you saw are broken!

$\text{Gen}(1^n)$: Pick $k \in_R \{0,1\}^n$ and output it.

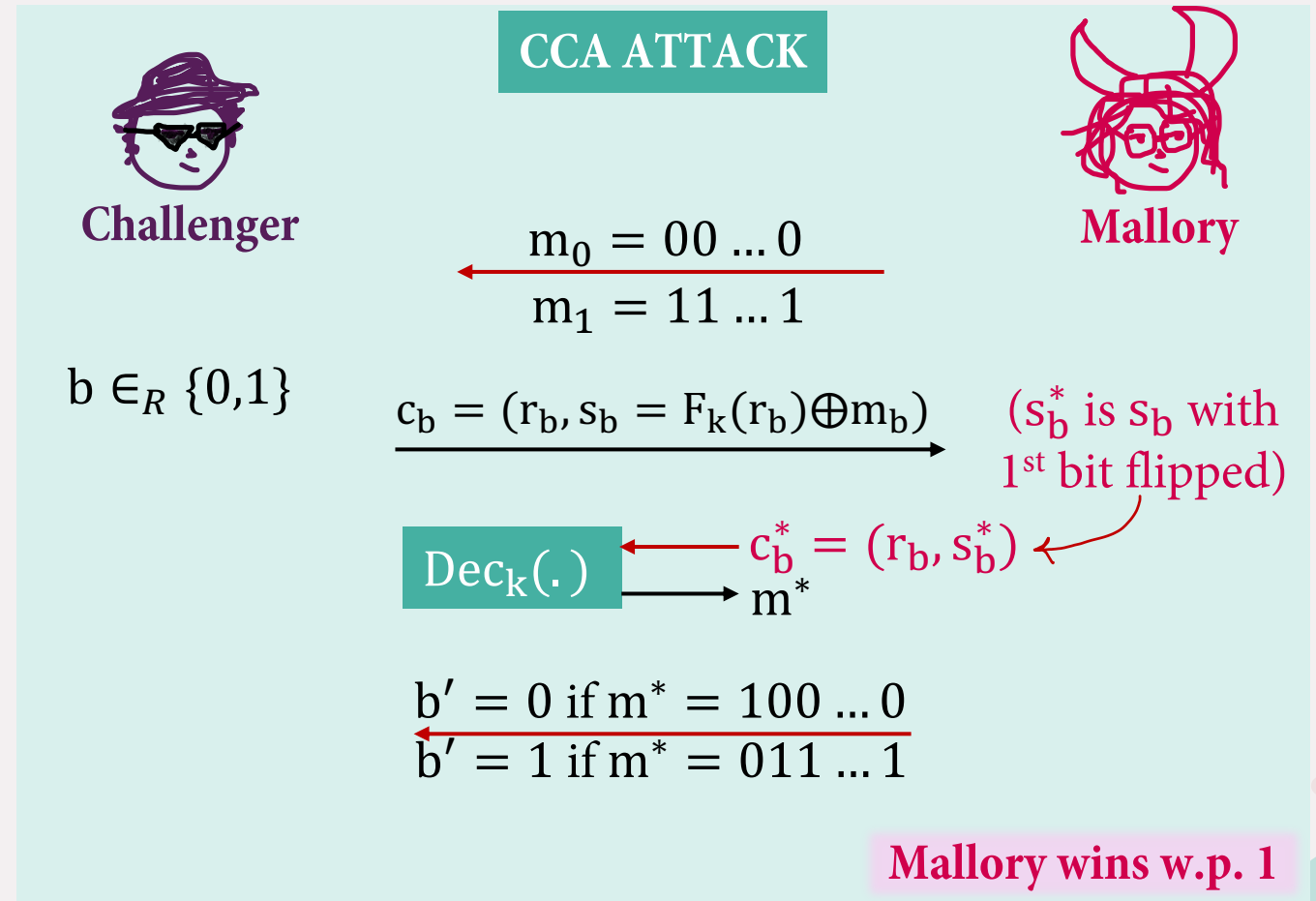
$\text{Enc}(k, m)$: Pick $r \in_R \{0,1\}^n$ and output
 $c = (r, F_k(r) \oplus m)$

$\text{Dec}(k, C)$: $C = (r, s)$, output the plaintext
 $m = F_k(r) \oplus s$

Construction 1, Lecture 8

(F is a pseudorandom function)

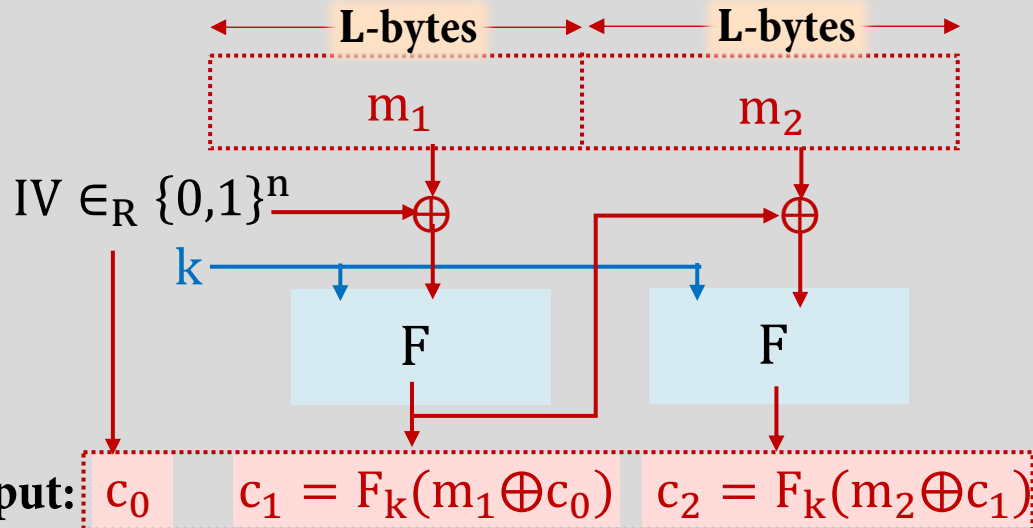
These attacks are called malleability attacks!



CBC Mode and Padding Oracle Attack

$\text{Gen}(1^n) \rightarrow k$

$\text{Enc}(k, m)$: Suppose $m = (m_1, m_2)$



$\text{Dec}(k, c = (c_0, c_1, c_2))$:

- **Output:** $m_1 = F_k^{-1}(c_1) \oplus c_0$ and $m_2 = F_k^{-1}(c_2) \oplus c_1$

m is assumed to be of length tL -bytes for some t . What if the message length $|m| \neq tL$ for some t ?

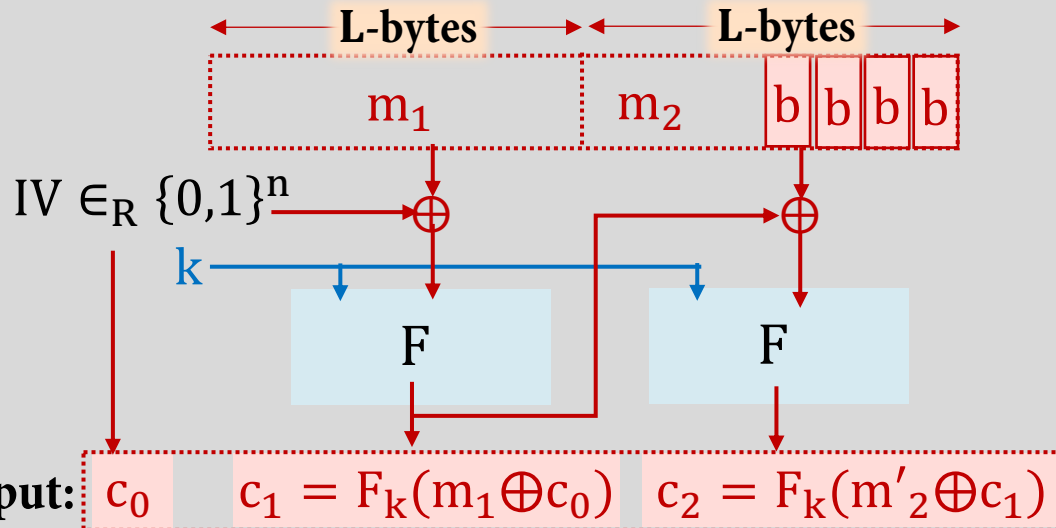
PKCS#5 padding

- Say b -bytes need to be appended to the last block of m ($1 \leq b \leq L$ is an integer).
- We append the string containing the integer b , represented in 1 byte (or 2 hexadecimal digits), repeated b times.

CBC Mode and Padding Oracle Attack

$\text{Gen}(1^n) \rightarrow k$

$\text{Enc}(k, m)$: Suppose $m = (m_1, m_2)$



$\text{Dec}(k, c = (c_0, c_1, c_2))$:

- Get $m_1 = F_k^{-1}(c_1) \oplus c_0$ and $m'_2 = F_k^{-1}(c_2) \oplus c_1$
- Read the final byte value b .
- If the last b bytes of m'_2 all have b then strip off padding and **output: m_1, m_2**
- Else, **output: “bad padding”**

m is assumed to be of length tL -bytes for some t . What if the message length $|m| \neq tL$ for some t ?

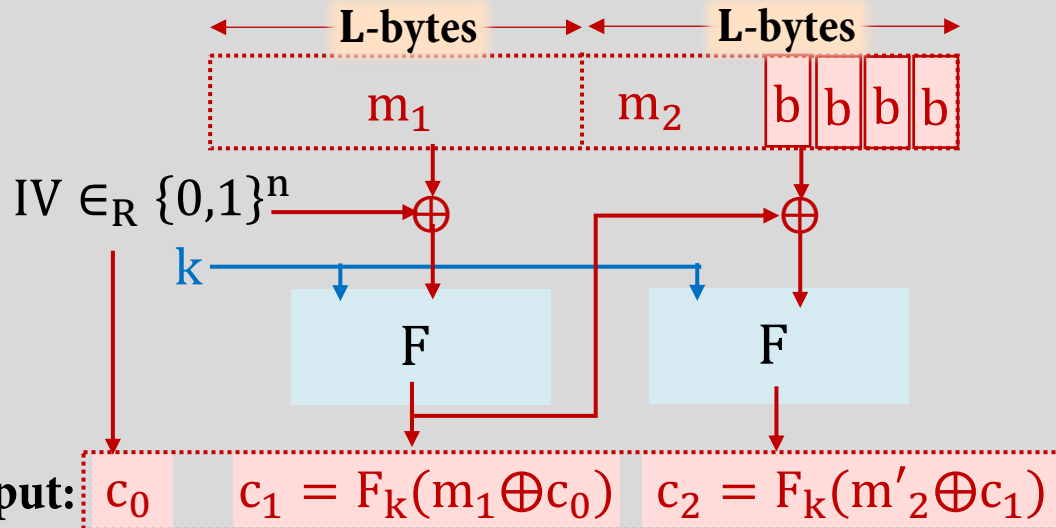
PKCS#5 padding

- Say b -bytes need to be appended to the last block of m ($1 \leq b \leq L$ is an integer).
- We append the string containing the integer b , represented in 1 byte (or 2 hexadecimal digits), repeated b times.

CBC Mode and Padding Oracle Attack

$\text{Gen}(1^n) \rightarrow k$

$\text{Enc}(k, m)$: Suppose $m = (m_1, m_2)$



$\text{Dec}(k, c = (c_0, c_1, c_2))$:

- Get $m_1 = F_k^{-1}(c_1) \oplus c_0$ and $m'_2 = F_k^{-1}(c_2) \oplus c_1$
- Read the final byte value b .
- If the last b bytes of m'_2 all have b then strip off padding and **output: m_1, m_2**
- Else, **output: “bad padding”**

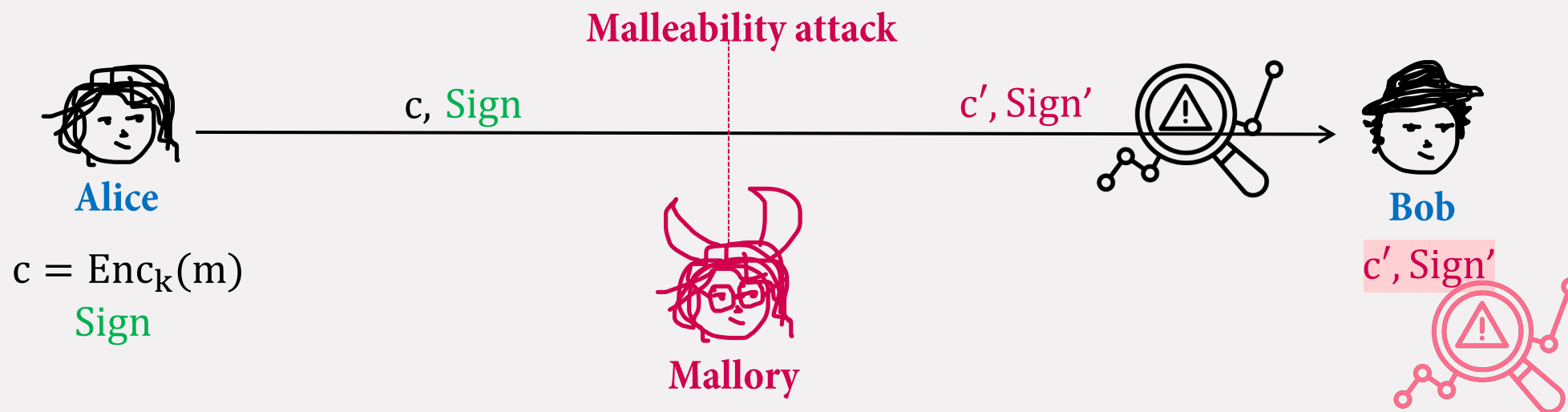
CCA Attack: Padding Oracle Attack [Vaudenay, 2002]

What to query the decryption oracle to help you recover the message underlying any $c = (IV, c_1, c_2)$?

- If $c'_1 = c_1 \oplus \delta$ for any string δ , then $\text{Dec}(k, (IV, c'_1, c_2))$ computes $m'_2 = m_2 \oplus \delta$.
- Use the “**bad padding**” output to your advantage now!

HW 1: Complete the CCA attack on CBC (and think about OFC and CTR)

Towards achieving CCA Security



- Mallory can change the ciphertext; we cannot prevent that!
- Can Bob at least detect that Mallory has mauled the ciphertext?
- Append some additional “Sign” that helps Bob detect?

MESSAGE AUTHENTICATION CODES

Today's Class

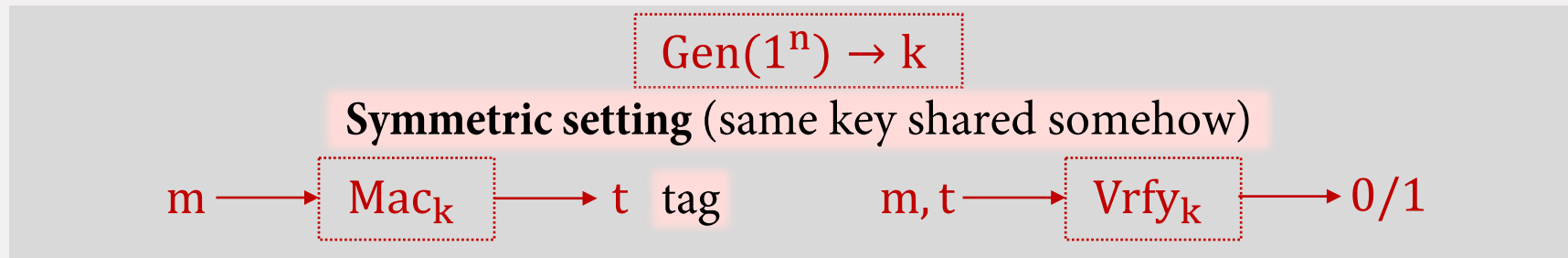


CCA Secure Encryption
Definition
A CCA secure scheme



Message Authentication Codes

Message Authentication Codes (MAC)



$$\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy}), \mathcal{M} = \{0,1\}^*$$

Correctness

For every $n \in \mathbb{N}$, every $k \leftarrow \text{Gen}(1^n)$ and every $m \in \mathcal{M}$ it holds that

$$\Pr[\text{Vrfy}_k(m, \text{Mac}_k(m)) = 1] = 1$$

Security

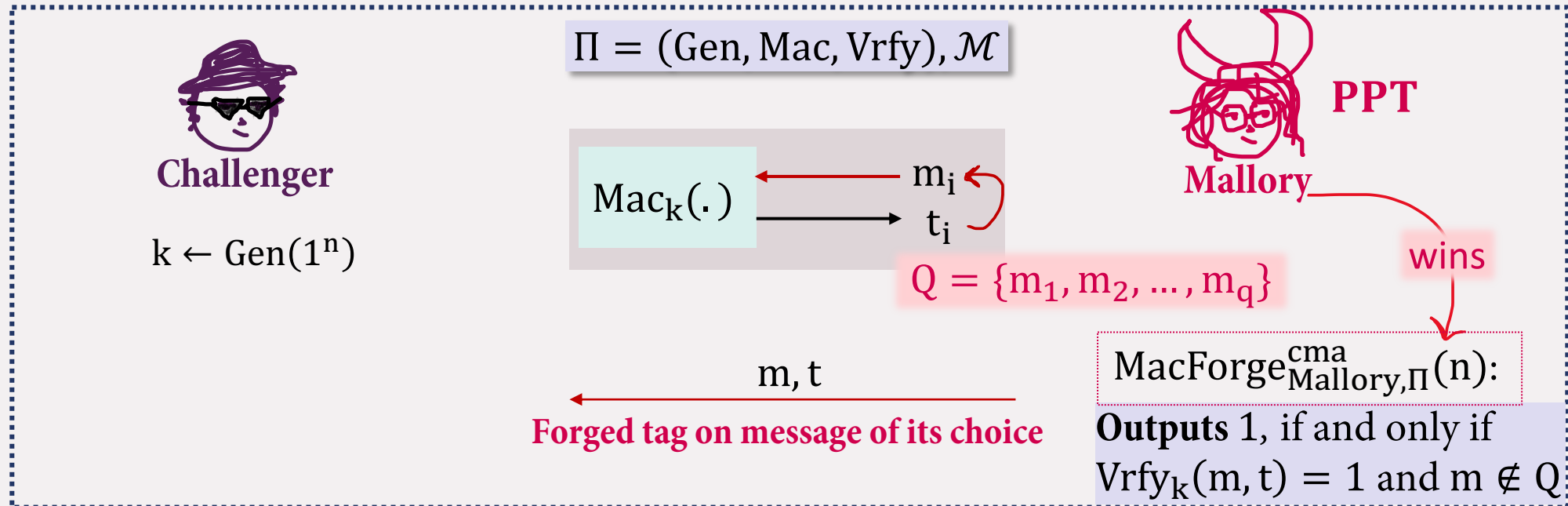


Mallory must be unable to forge a valid new tag from previously seen tags “on messages of its choice”. The forged new tag can be on any message of its choice.

Chosen-Message Attack (CMA)

Existential Unforgeability (EU)

Defining EU-CMA Security for MAC

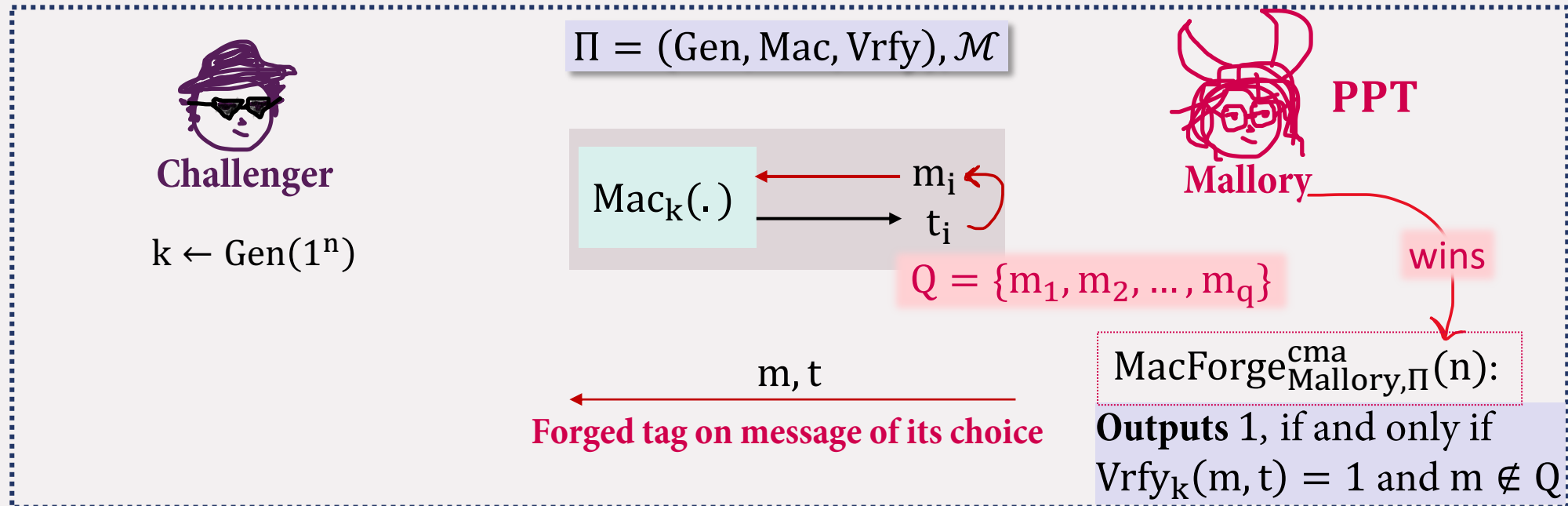


Definition 2

Π is **existential unforgeable under an adaptive chosen-message attack (EU-CMA)**, if, for all **PPT** adversary **Mallory**, there exists a negligible function **negl** such that $\forall n$

$$\Pr[\text{MacForge}_{\text{Mallory}, \Pi}^{\text{cma}}(n) = 1] \leq \text{negl}(n)$$

Quiz time



MAC or not? $((\text{Gen}, \text{Enc}, \text{Dec})$ is a CPA SKE and $(\text{Gen}, \text{Mac}, \text{Vrfy})$ is an EU-CMA secure MAC)

Encrypt to MAC: $\text{Mac}'_k(m) = \text{Enc}_k(m)$; $\text{Vrfy}'_k(m, t) := \text{output } 1, \text{ if } \text{Dec}_k(t) = m \text{ and } 0, \text{ o. w.}$ ✗

Append-0 MAC: $\text{Mac}'_k(m) = \text{Mac}_k(m) || 0$; $\text{Vrfy}'_k(m, tb) := \text{output } \text{Vrfy}_k(m, t)$ ✓

Constructing an EU-CMA MAC

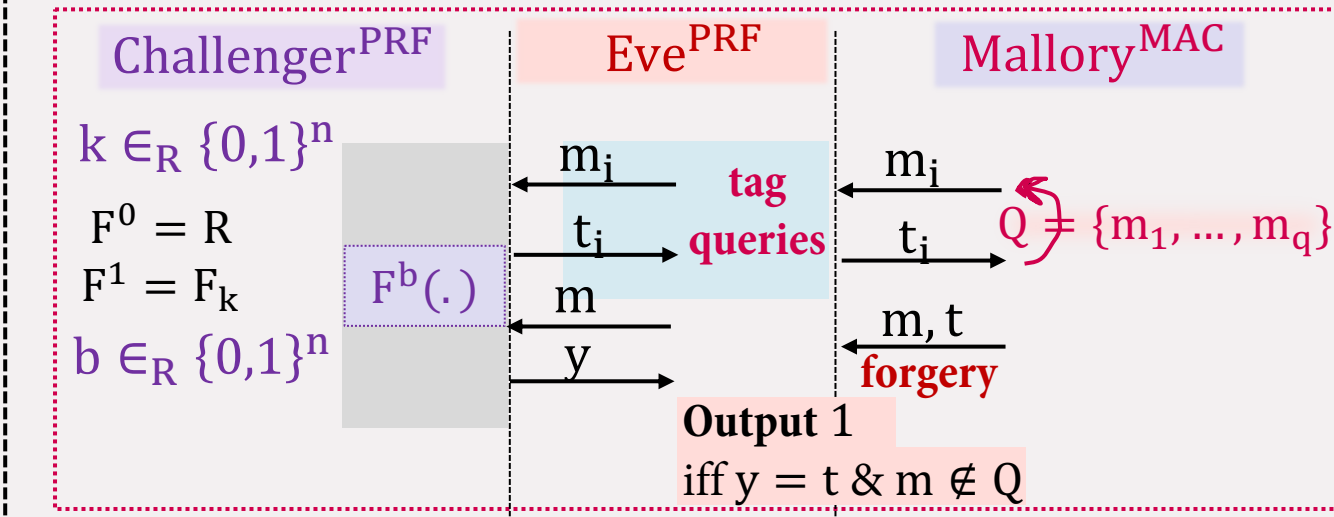
Gen(1^n): $k \in_R \{0,1\}^n$ Mac(k, m): output $t := F_k(m)$ Vrfy(k, m, t): output 1, if $t := F_k(m)$, and 0, o. w.

Construction 2 ($\mathcal{M} = \{0,1\}^n$, F is a PRF)

Theorem 1

If F is a pseudorandom function, then Construction 2 is a EU-CMA secure MAC for n -bit messages.

Proof Sketch: $\exists$ Mallory^{MAC} against Construction 2 **To prove:** $\exists$ Eve^{PRF} that breaks PRF security



If $b = 1$ (PRF World)

Mallory^{MAC} gets correct tags
 $\therefore$ her winning $\Rightarrow$ Eve^{PRF} outputs 1

If $b = 0$ (Random World)

Mallory^{MAC} gets random tags
 $\Rightarrow$ her winning prob $\leq 1/2^n$

Today's Class



CCA Secure Encryption
Definition

A CCA secure scheme



Message Authentication Codes

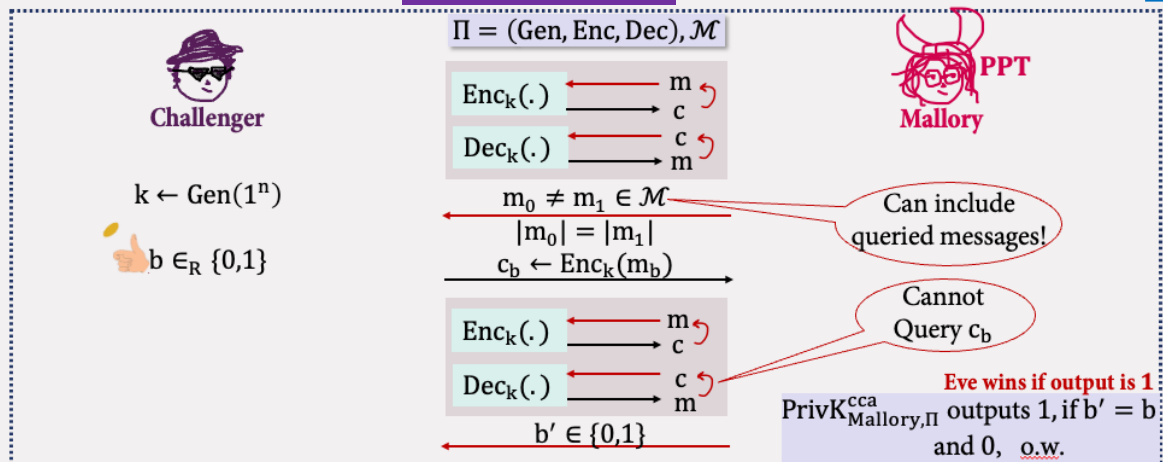
Authenticated Encryption (AE)

$$\Pi = (\text{Gen}, \text{Enc}, \text{Dec}), \mathcal{M}$$

SECRECY



CIPHERTEXT INTEGRITY



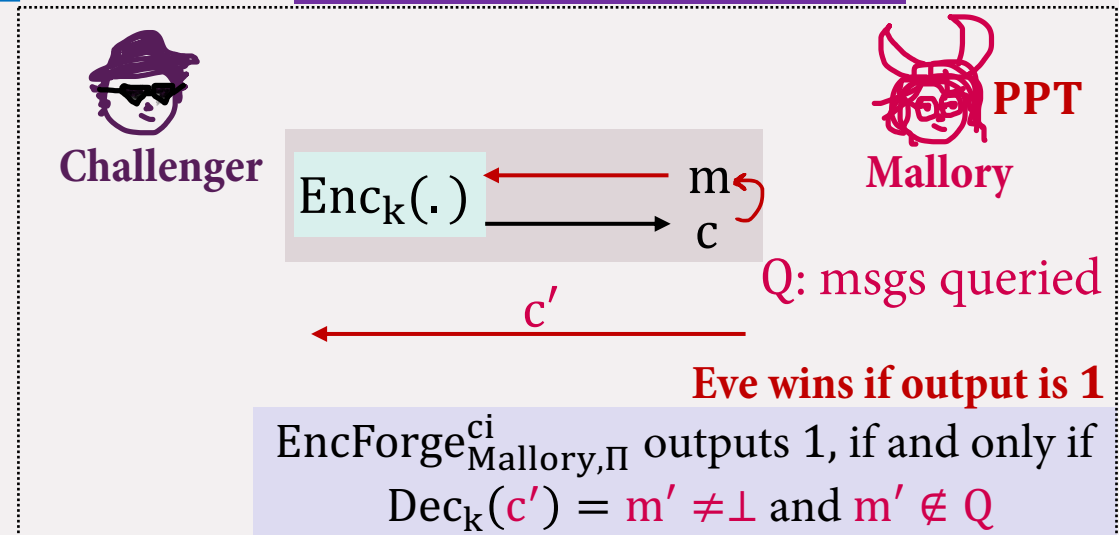
Definition 1

Π is **CCA secure** if, for all PPT adversary **Mallory**, there exists a negligible function **negl** such that $\forall n$

$$\Pr[\text{PrivK}_{\text{Mallory}, \Pi}^{\text{cca}}(n) = 1] \leq 1/2 + \text{negl}(n)$$

CCA Security

Mallory can come up with a valid ciphertext of some message on her own (given all the encryption queries she wants)



Definition 3

Π is **unforgeable** if, for all PPT adversary **Mallory**, $\exists$ negligible function **negl** such that $\forall n$

$$\Pr[\text{EncForge}_{\text{Mallory}, \Pi}^{\text{ci}}(n) = 1] \leq \text{negl}(n)$$

How do we build an AE?

TOOLS WE HAVE

How do we combine them to get an AE?



$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$
CPA Secure Encryption

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$
EU-CMA secure MAC



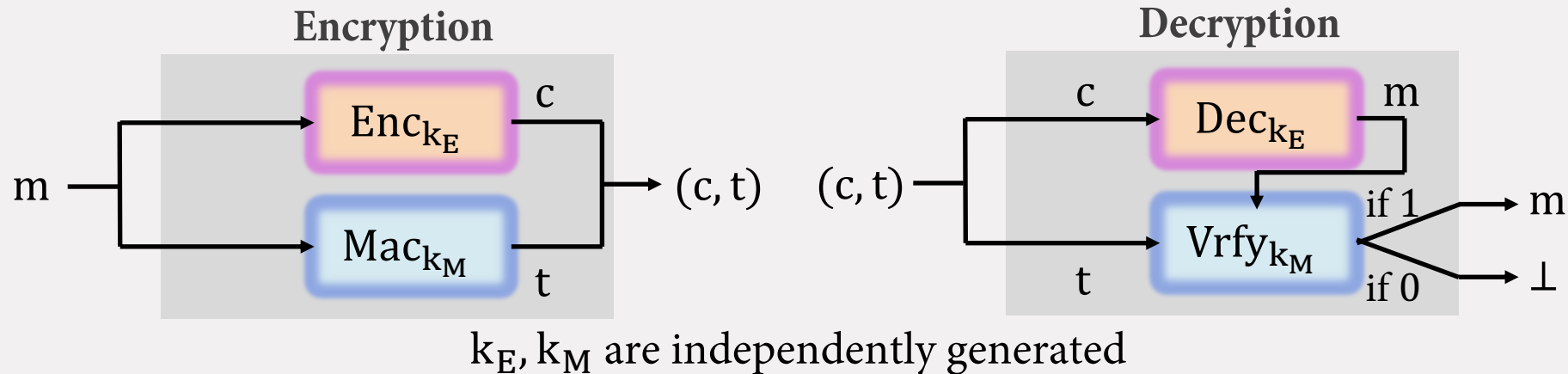
PRF



Attempt I: Encrypt and Authenticate

$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$
CPA Secure Encryption

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$
EU-CMA secure MAC



Used in SSH! Is this secure?

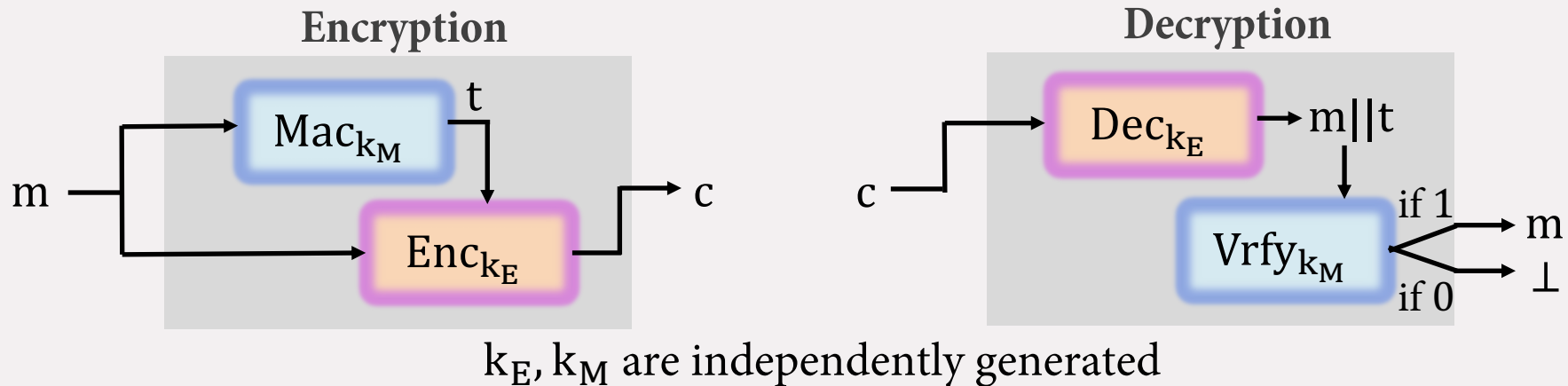
Π_M does not guarantee any privacy of m !
(It can output, say, first bit of m)^{*}

^{*}For any secure MAC, MAC' defined by $\text{Mac}'_k(m) := m_1 || \text{Mac}_k(m)$ (m_1 is first bit of m) is also secure!

Attempt II: Authenticate then Encrypt

$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$
CPA Secure Encryption

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$
EU-CMA secure MAC



Used in **SSL** Is this secure?

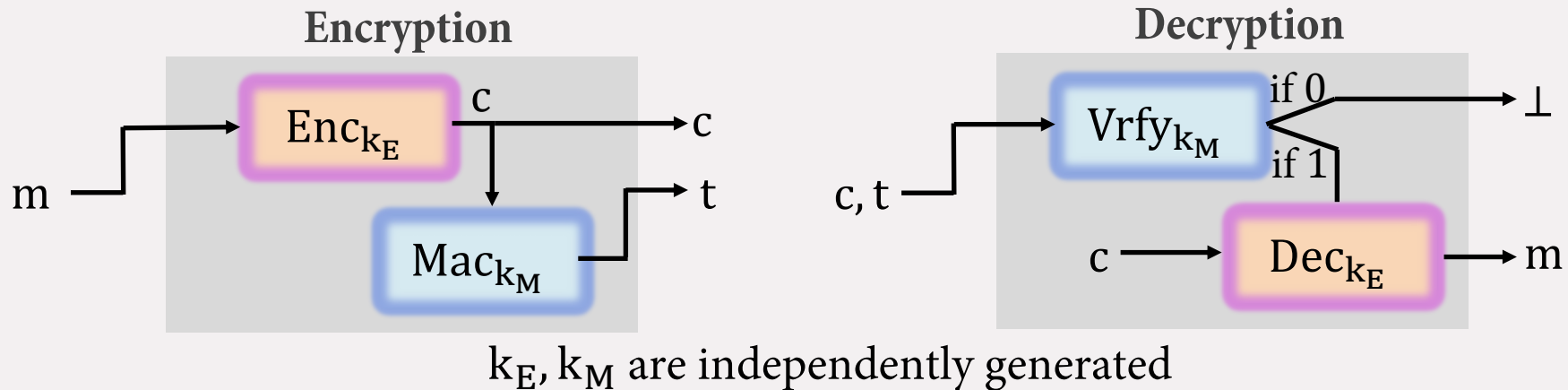
There exists a CPA secure scheme such that
for any secure MAC this scheme is not an AE!
(POODLE Attack on: CBC mode + MAC)

Simpler counter example: for secure Π_E take $\text{Enc}'_k(m) := 0 || \text{Enc}_k(m)$; Dec' ignores first bit.

Attempt III: Encrypt then Authenticate

$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$
CPA Secure Encryption

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$
EU-CMA secure MAC



Is this secure?



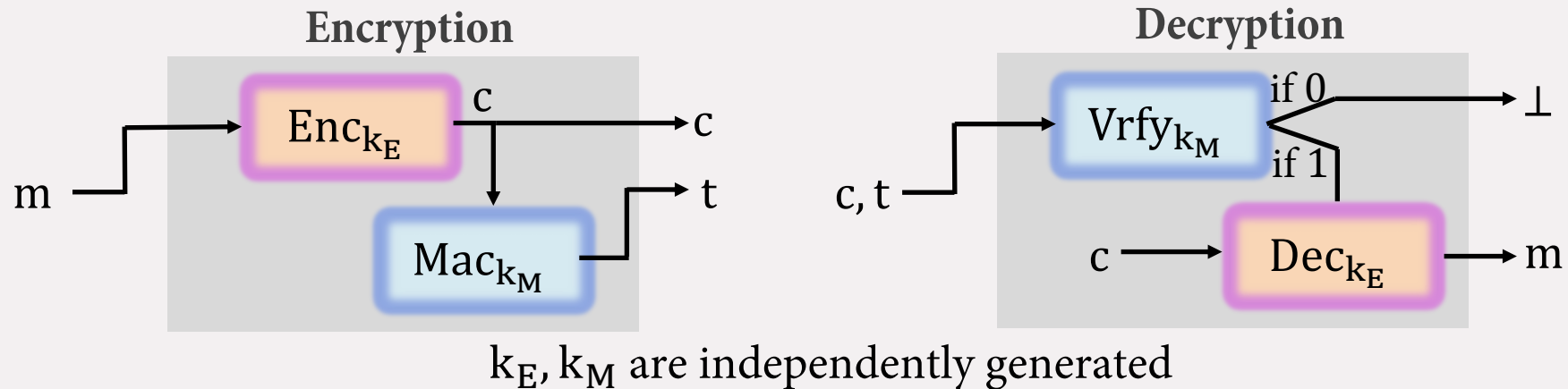
Not with the CMA security of MAC!
(Mallory can generate a new valid tag on an already queried c)

we need a stronger MAC security to prevent this!

Attempt III: Encrypt then Authenticate

$\Pi_E = (\text{Gen}, \text{Enc}, \text{Dec})$
CPA Secure Encryption

$\Pi_M = (\text{Gen}, \text{Mac}, \text{Vrfy})$
EU-SCMA secure MAC



Used in IPSec! Is this secure? ✓

Secure if you use CPA with
strong-CMA secure MAC

Next Lecture

- Strong CMA security for MAC
- Encrypt then Authenticate is a secure Authenticated Encryption (**and hence CCA secure**)
- How to authenticate messages of arbitrary length (**domain extension of MAC**)
--motivates a new important primitive, Hash functions!



THANK YOU