CIS 5560

Cryptography
Lecture 11

Course website:
pratyushmishra.com/classes/cis-5560-s24/

Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **Final Exam May 10, 2024, 9-11AM, DRLB A2**
- **Homework:**
  - **Fine to collaborate,** but **write up your own solutions**

Recap of last lecture

Formal Definition: Collision-Resistant Hash Functions

A compressing **family of functions** $\mathscr{H} = \{h : \{0,1\}^m \to \{0,1\}^n\}$ (where $m > n$) for which it is computationally hard to find collisions.

Def: $\mathscr{H}$ is collision-resistant if for every PPT algorithm A, there is a negligible function μ s.t.

$$\Pr_{h \leftarrow \mathscr{H}}\left[A\left(1^n, h\right) = \left(x, y\right) : x \neq y,\ h(x) = h\left(y\right)\right] = \mu(n)$$

Generic attack on C.R. functions

Let H: M $\longrightarrow \{0,1\}^n$ be a hash function ($|M| >> 2^n$)

Generic alg. to find a collision **in time $O(2^{n/2})$** hashes

Algorithm:

1. Choose **$2^{n/2}$** random messages in M: $m_1, \ldots, m_{2^{n/2}}$ (distinct w.h.p)
2. For i = 1, …, $2^{n/2}$ compute $t_i = H(m_i)$ $\in\{0,1\}^n$
3. Look for a collision ($t_i = t_j$). If not found, got back to step 1.

How well will this work?

The birthday paradox

Let $r_1, \ldots, r_n \in \{1, \ldots, B\}$ be IID integers.

Thm: When $n \approx \sqrt{B}$ then $\Pr[r_i = r_j \mid \exists i \neq j] \geq \frac{1}{2}$

Proof: (for uniform indep. $r_1, \ldots, r_n$)

$$\Pr[\exists i \neq j : r_i = r_j] = 1 - \Pr[\forall i \neq j : r_i \neq r_j] = 1 - \left(\frac{B-1}{B}\right)\left(\frac{B-2}{B}\right)\cdots\left(\frac{B-n+1}{B}\right) =$$

$$= 1 - \prod_{i=1}^{n-1}\left(1 - \frac{i}{B}\right) \geq 1 - \prod_{i=1}^{n-1} e^{-i/B} = 1 - e^{-\frac{1}{B}\sum_{i=1}^{n-1} i} \geq 1 - e^{-n^2/2B}$$

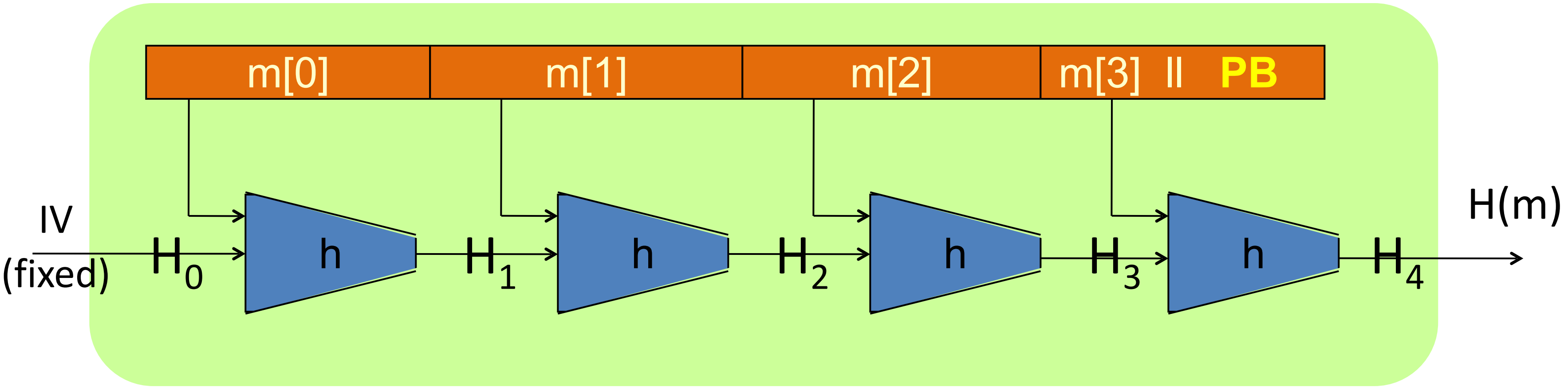
$1 - x \leq e^{-x}$

$$\geq 1 - e^{-0.72} = 0.53 > \frac{1}{2}$$

$\frac{n^2}{2B} = 0.72$

■

Merkle-Dåmgard



Given **$h: T \times X \longrightarrow T$** (compression function)

we obtain **$H: X^{\leq L} \longrightarrow T$** . H_i - chaining variables

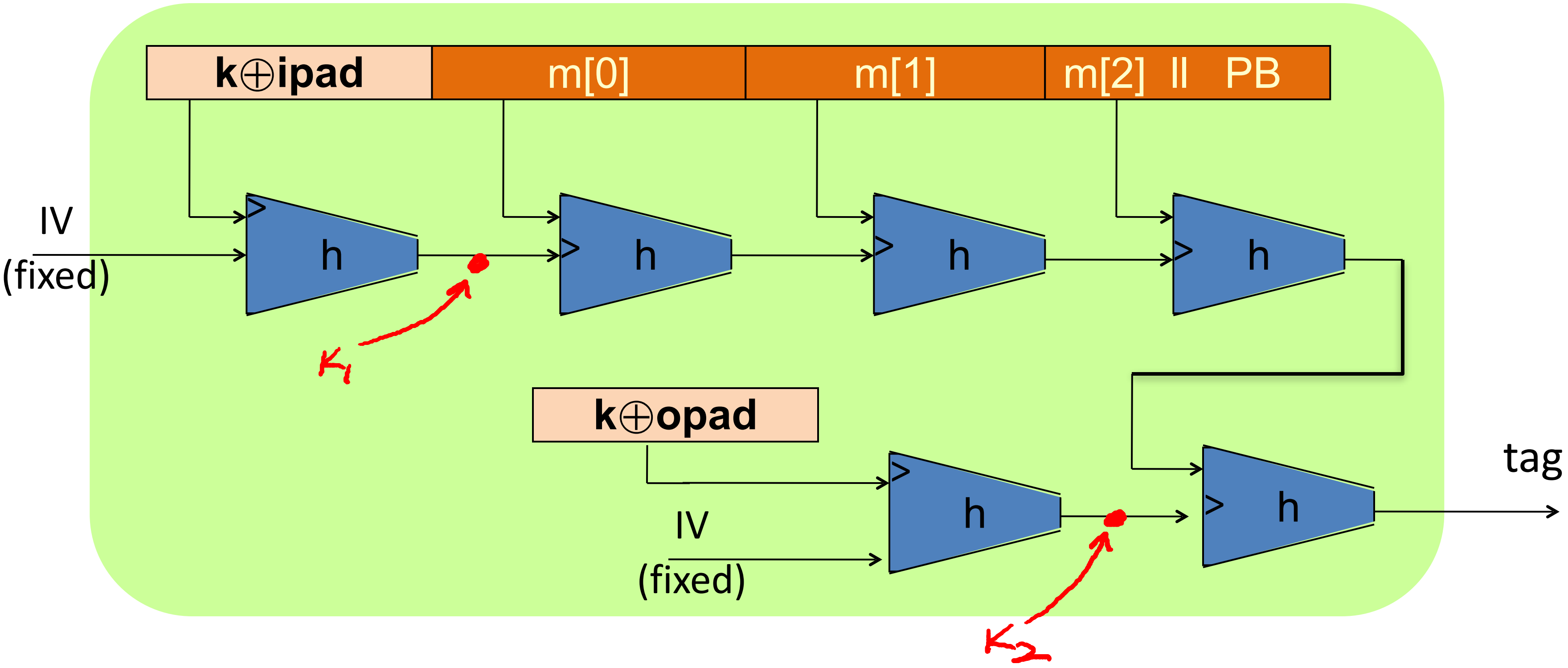
PB: padding block

1000…0 ll msg len

64 bits

If no space for PB add another block

HMAC



Today

- Encryption schemes with confidentiality *and* integrity
- Authenticated Encryption
  - IND-CPA + Ciphertext integrity
  - IND-CCA
  -

Story so far

Confidentiality: semantic security against a CPA attack

- Encryption secure against **eavesdropping only**

Integrity:

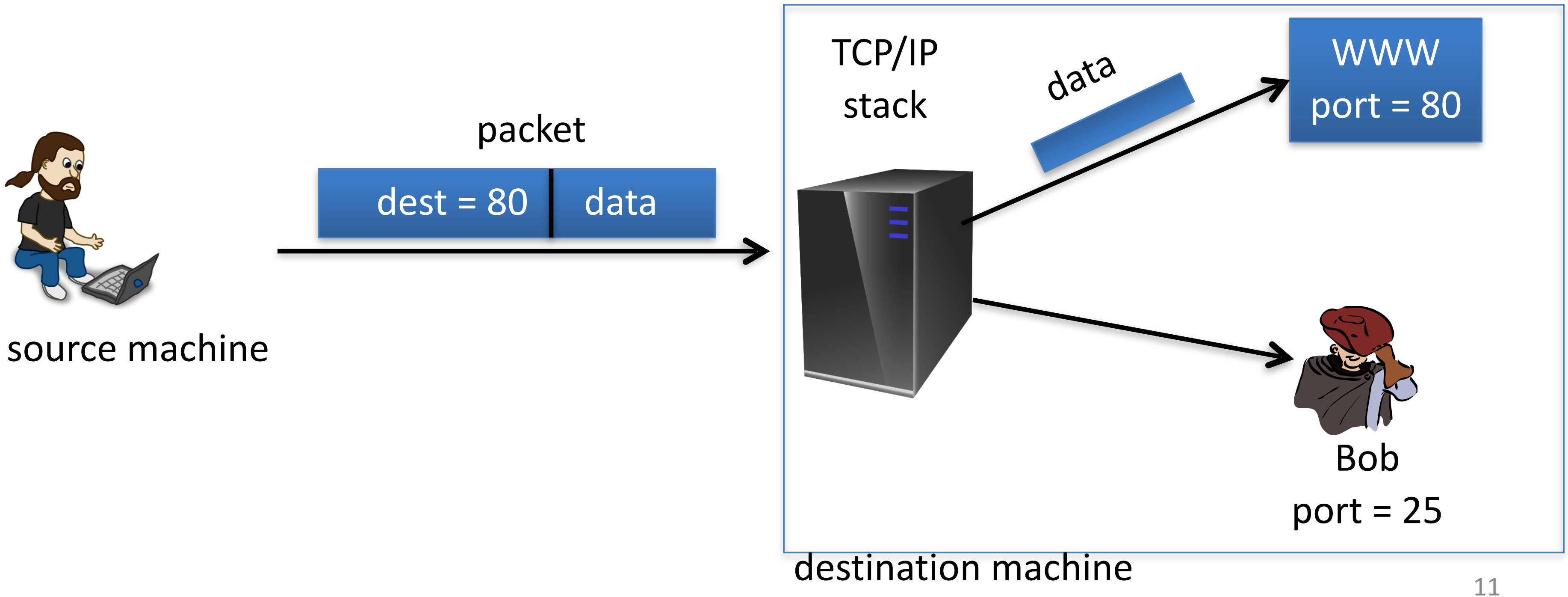
- Existential unforgeability under a chosen message attack
- CBC-MAC, HMAC, PMAC, CW-MAC

This module: encryption secure against **tampering**

- Ensuring both confidentiality and integrity

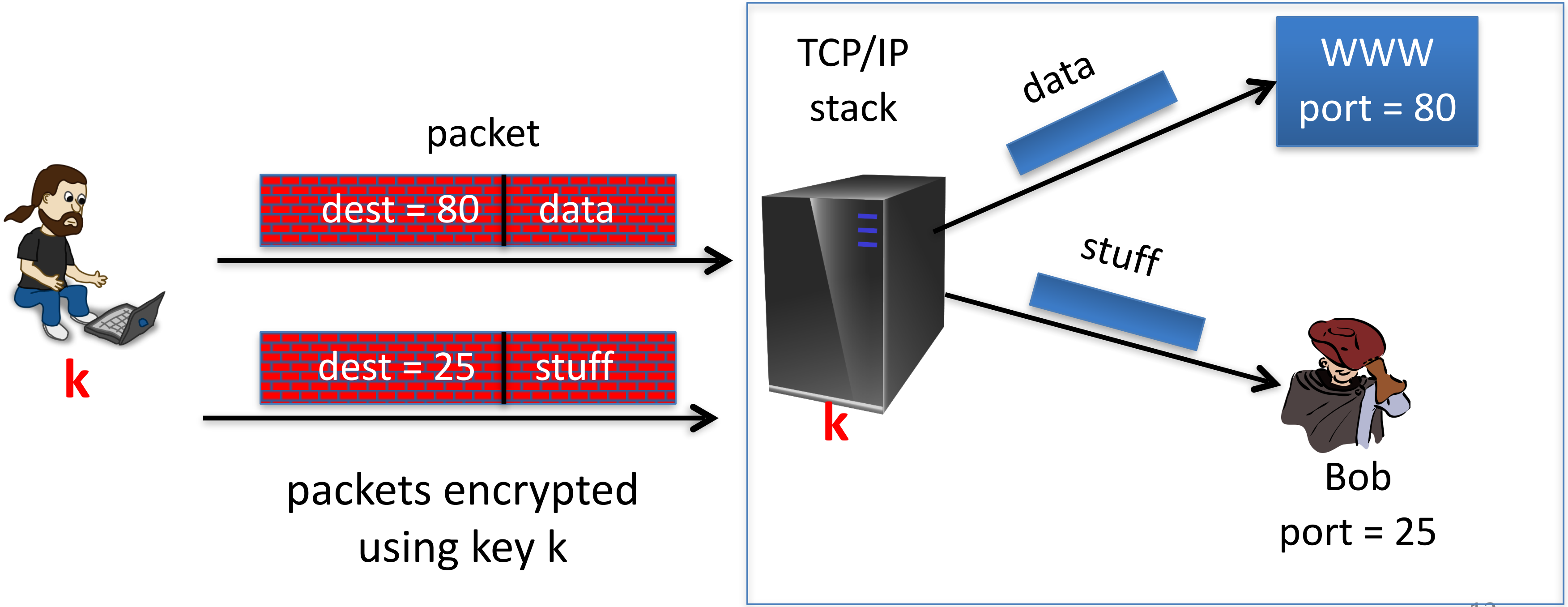
Sample tampering attacks

TCP/IP: (highly abstracted)



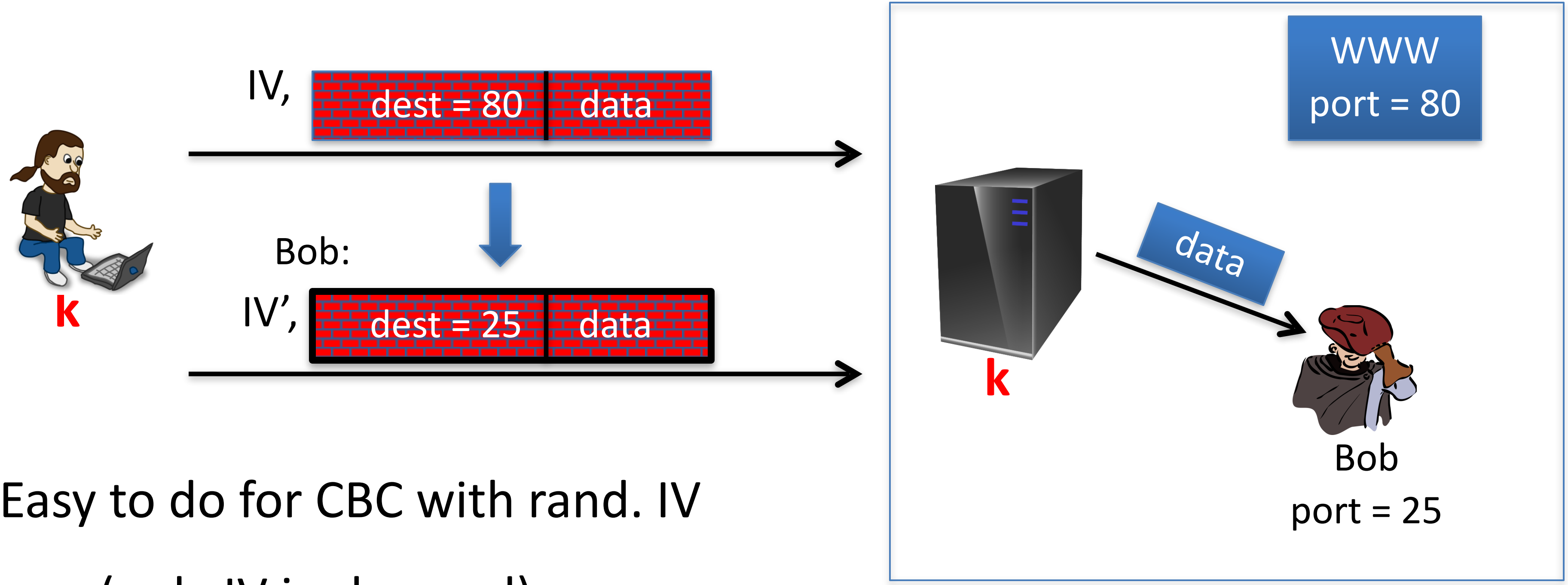
Sample tampering attacks

IPsec: (highly abstracted)



Reading someone else's data

Note: attacker obtains decryption of any ciphertext beginning with "dest=25"



Easy to do for CBC with rand. IV

(only IV is changed)

Encryption is done with CBC with a random IV.

What should IV' be?

$m[0] = D(k, c[0]) \oplus IV = \text{"dest=80..."}$

- ○ $IV' = IV \oplus (\ldots 25 \ldots)$
- ○ $IV' = IV \oplus (\ldots 80 \ldots)$
- ○ $IV' = IV \oplus (\ldots 80 \ldots) \oplus (\ldots 25 \ldots)$
- ○ It can't be done

The lesson

CPA security cannot guarantee secrecy under active attacks.

Only use one of two modes:

- If message needs integrity but no confidentiality:
  use a **MAC**

- If message needs both integrity and confidentiality:
  use **authenticated encryption** modes (this module)

Goals

An **authenticated encryption** system (Gen, Enc, Dec) is a cipher where

As usual: $\text{Enc} : \mathscr{K} \times \mathscr{M} \rightarrow \mathscr{C}$

but $\text{Dec} : \mathscr{K} \times \mathscr{C} \rightarrow \mathscr{M} \cup \{\perp\}$

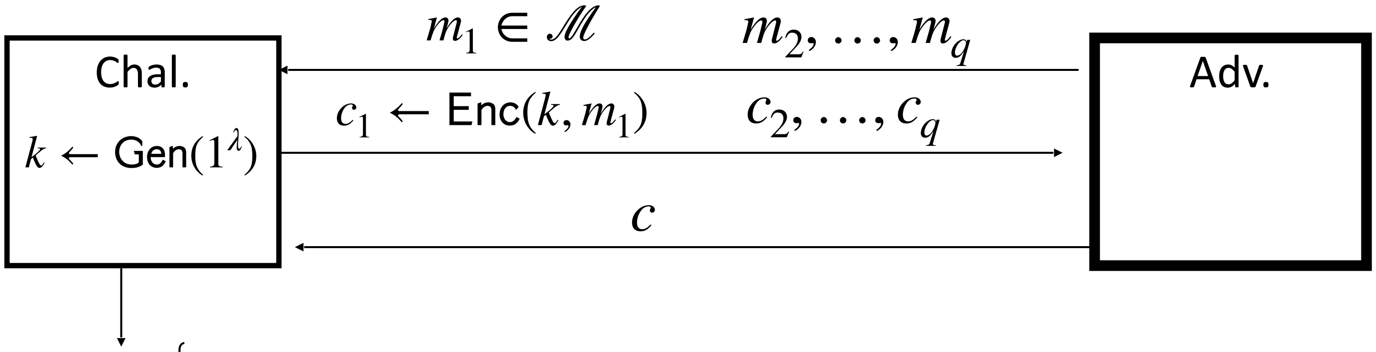
ciphertext is rejected

<u>Security</u>: the system must provide

- IND-CPA, and
- **ciphertext integrity:**
  attacker cannot create new ciphertexts that decrypt properly

Ciphertext integrity

Let (Gen, Enc, Dec) be a cipher with message space $\mathcal{M}$.



$$b \begin{cases} b = 1 & \text{if } \text{Dec}(k, c) \neq \perp \text{ and } c \notin \{c_1, \ldots, c_q\} \\ b = 0 & \text{otherwise} \end{cases}$$

Def: (Gen, Enc, Dec) has **ciphertext integrity** if for all PPT A:

$$\text{Adv}_{\text{CI}}[A] = \Pr[b = 1] = \text{negl}(\lambda)$$

Authenticated encryption

Def: (G, E,D) provides **authenticated encryption (AE)** if it

(1) is IND-CPA secure, and

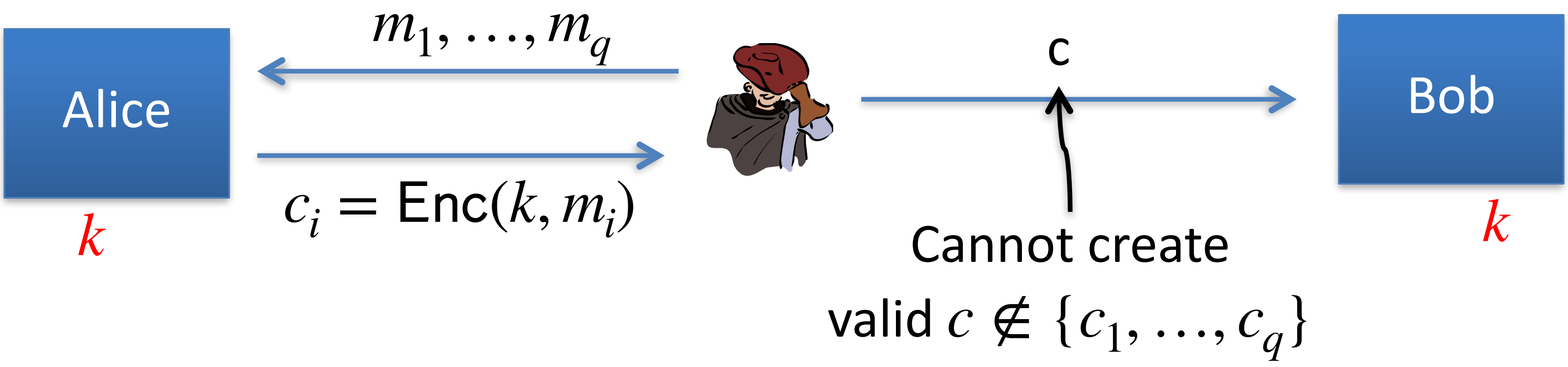
(2) has ciphertext integrity

Bad example: CBC with rand. IV does not provide AE

- $D(k,\cdot)$ never outputs $\perp$, hence adv. easily wins CI game

Implication 1: authenticity

Attacker cannot fool Bob into thinking a message was sent from Alice



$\Rightarrow$ if $\text{Dec}(k, c) \neq \perp$ Bob knows message is from someone who knows k

(but message could be a replay)

Implication 2

Authenticated encryption

↓

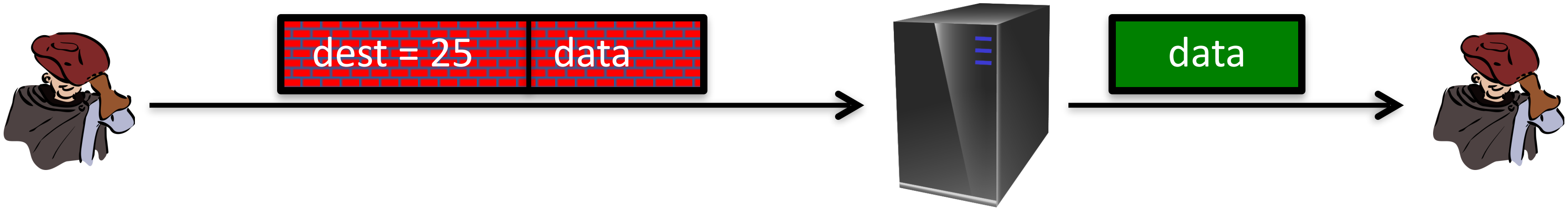
Security against **chosen ciphertext attacks**

Chosen ciphertext attacks

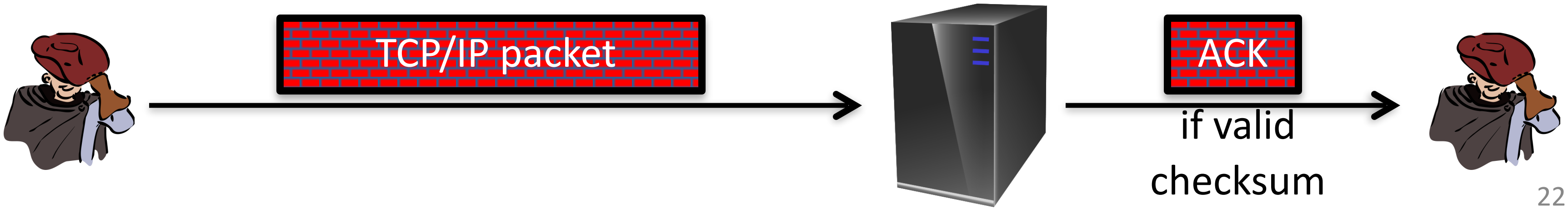
Example chosen ciphertext attacks

Adversary A has ciphertext c that it wants to decrypt

- Often, A can fool server into decrypting ***other*** ciphertexts (not c)



- Often, adversary can learn partial information about plaintext



Chosen ciphertext security

Adversary’s power: both CPA and CCA

- Can obtain the encryption of arbitrary messages of his choice
- Can decrypt any ciphertext of his choice, other than challenge

(conservative modeling of real life)

Adversary’s goal:

Learn partial information about challenge plaintext

Chosen ciphertext security: definition

Let (Gen, Enc, Dec) be a cipher with message space $\mathscr{M}$

Challenger

$k \leftarrow \mathsf{Gen}(1^\lambda)$

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

for $i \in \{1, \ldots, q\}$:

(1) **CPA query:**

$m_{i,0}, m_{i,1} \in \mathscr{M} : |m_{i,0}| = |m_{i,1}|$

$c_i \leftarrow \mathsf{Enc}(k, m_{i,b})$

(2) **CCA query:**

$c_j \in \mathscr{C} \quad : c_j \notin \{c_1, \ldots, c_i\}$

$m_j \leftarrow \mathsf{D}(k, c_j) : m_j \in \mathscr{M} \cup \{ \perp \}$

Adversary

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

Chosen ciphertext security: definition

E is CCA secure if for all "efficient" A: $\Pr[b = b'] = 1/2 + \mu(\lambda)$

Question: Is CBC with rand. IV CCA-secure?

Authenticated enc. ⇒ CCA security

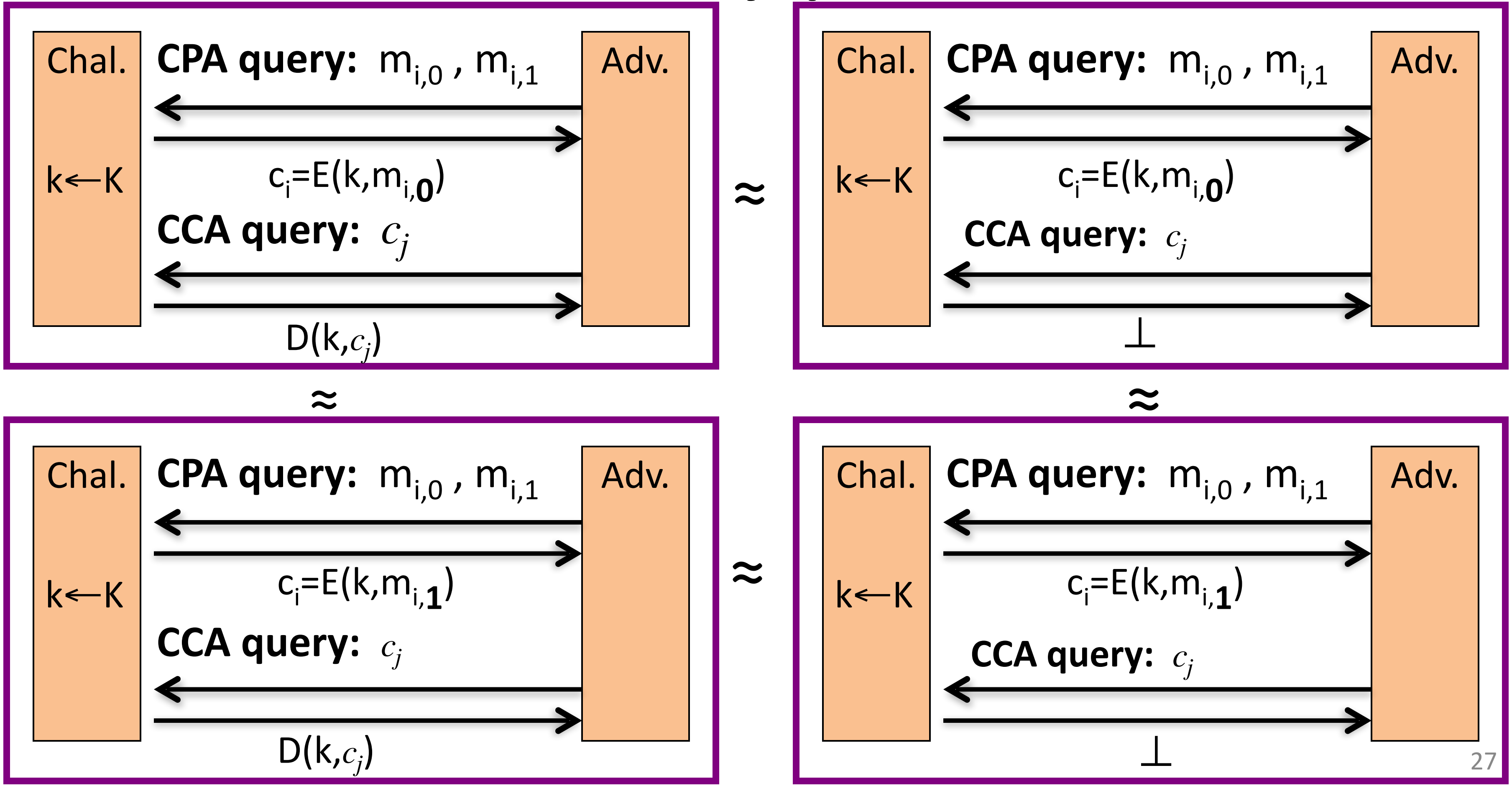
Thm: Let (E,D) be a cipher that provides AE.

Then (E,D) is CCA secure !

In particular, for any q-query eff. A there exist eff. B_1, B_2 s.t.

$$Adv_{CCA}[A,E] \leq 2q \cdot Adv_{CI}[B_1,E] + Adv_{CPA}[B_2,E]$$

Proof by pictures



So what?

Authenticated encryption:

- ensures confidentiality against an active adversary that can decrypt some ciphertexts

Limitations:

- does not prevent replay attacks
- does not account for side channels (timing)

Constructions of AE

… but first, some history

Authenticated Encryption (AE): introduced in 2000 [KY'00, BN'00]

Crypto APIs before then:

- Provide API for CPA-secure encryption (e.g. CBC with rand. IV)
- Provide API for MAC (e.g. HMAC)

Every project had to combine the two itself without a well defined goal

- Not all combinations provide AE …

Combining MAC and ENC (CCA)

Encryption key k_E. MAC key = k_M

Option 1: (SSL)

msg m ⇨ msg m | tag t ⇨ [ciphertext]

MAC(k_M, m) — Enc$(k_E, m || t)$

always correct

Option 2: (IPsec)

msg m ⇨ [ciphertext] ⇨ [ciphertext] | tag t

Enc(k_E, m) — MAC(k_M, c)

Option 3: (SSH)

msg m ⇨ [ciphertext] ⇨ [ciphertext] | tag t

Enc(k_E, m) — MAC(k_M, m)

A.E. Theorems

Let (E,D) be CPA secure cipher and (S,V) secure MAC. Then:

1. **Encrypt-then-MAC**: always provides A.E.

2. **MAC-then-encrypt**: may be insecure against CCA attacks

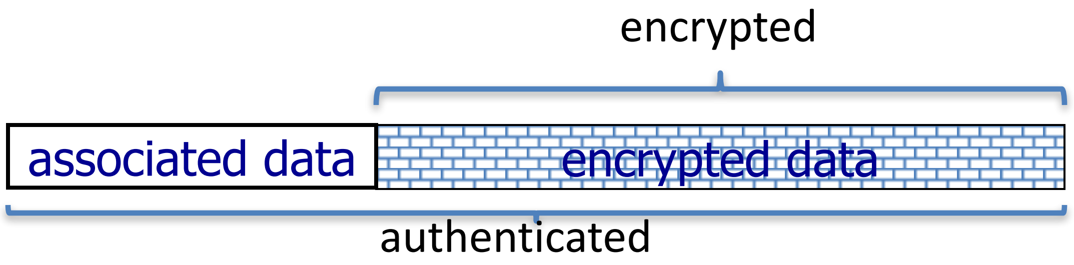
   however: when (E,D) is rand-CTR mode or rand-CBC
   M-then-E provides A.E.

Security of Encrypt-then-MAC

Standards (at a high level)

- **GCM**: CTR mode encryption then CW-MAC (accelerated via Intel's PCLMULQDQ instruction)
- **CCM**: CBC-MAC then CTR mode encryption (802.11i)
- **EAX**: CTR mode encryption then CMAC

All support AEAD: (auth. enc. with associated data). All are nonce-based.



CBC paddings attacks

Recap

Authenticated encryption: CPA security + ciphertext integrity

- Confidentiality in presence of **active** adversary
- Prevents chosen-ciphertext attacks

Limitation: cannot help bad implementations … (this segment)

Authenticated encryption modes:

- Standards: GCM, CCM, EAX
- General construction: encrypt-then-MAC

The TLS record protocol (CBC encryption)

Decryption: **dec($k_{b\to s}$, record, $ctr_{b\to s}$)** :

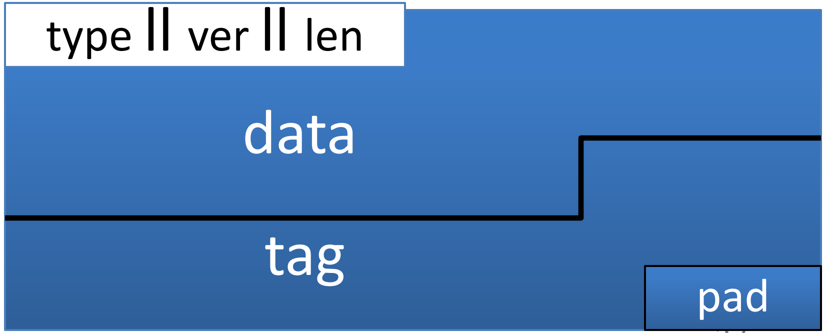
step 1: CBC decrypt record using k_{enc}

step 2: check pad format: abort if invalid

step 3: check tag on [$++ctr_{b\to s}$ ll header ll data]
abort if invalid

Two types of error:

- **padding error**
- **MAC error**



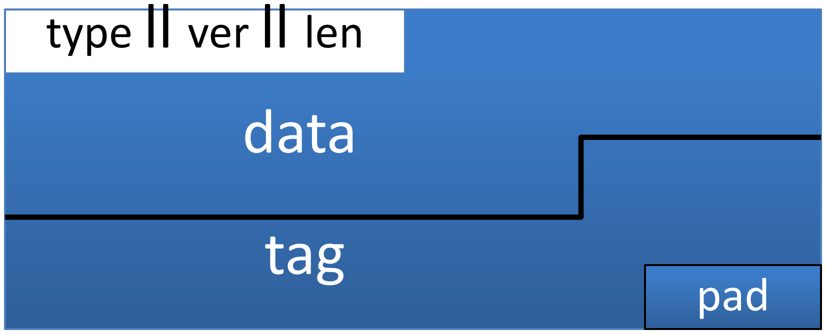
Padding oracle

Suppose attacker can differentiate the two errors
(pad error, MAC error):

⇒ **Padding oracle**:

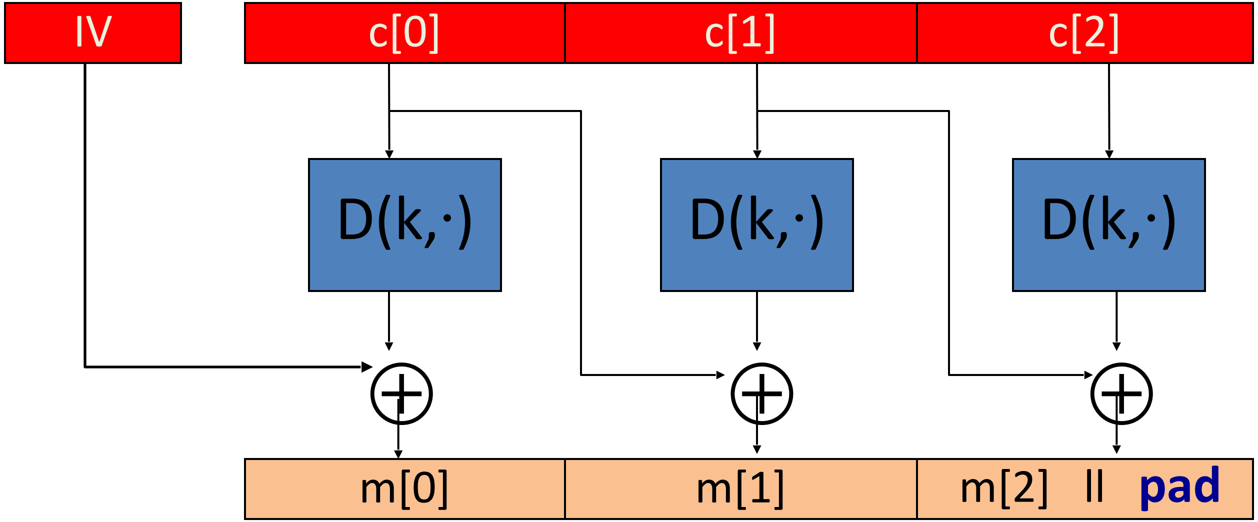
attacker submits ciphertext and learns if last bytes of plaintext are a valid pad

Nice example of a **chosen ciphertext attack**



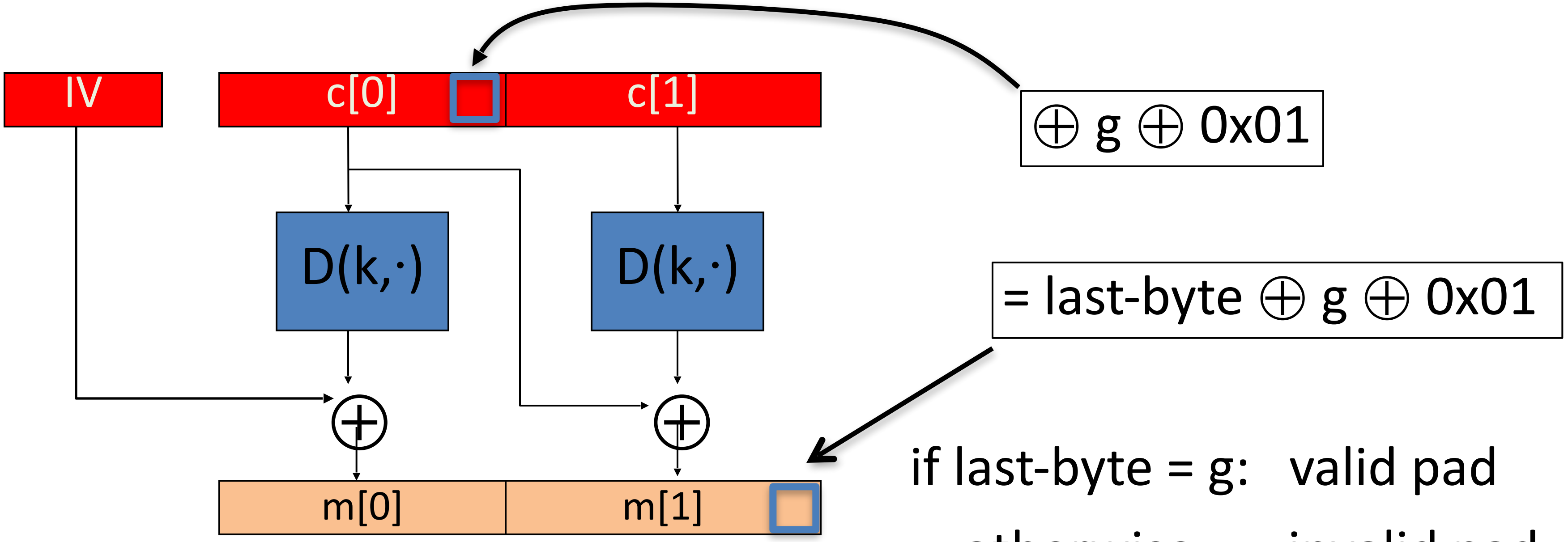
Using a padding oracle (CBC encryption)

Attacker has ciphertext **c = (c[0], c[1], c[2])** and it wants **m[1]**



Using a padding oracle (CBC encryption)

step 1: let **g** be a guess for the last byte of m[1]



if last-byte = g: valid pad

otherwise: invalid pad

Using a padding oracle (CBC encryption)

Attack: submit **(IV, c'[0], c[1])** to padding oracle

⇒ attacker learns if last-byte = g

Repeat with g = 0,1, …, 255 to learn last byte of m[1]

Then use a (02, 02) pad to learn the next byte and so on …

Lesson

1. Encrypt-then-MAC would completely avoid this problem:

   MAC is checked first and ciphertext discarded if invalid

2. MAC-then-CBC provides A.E., but padding oracle destroys it

Will this attack work if TLS used counter mode instead of CBC?

(i.e. use MAC-then-CTR)

- ○ Yes, padding oracles affect all encryption schemes
- ○ It depends on what block cipher is used
- ○ No, counter mode need not use padding ⇐
- ○