Authenticated Encryption

Authenticated Encryption

Active attacks on CPA-secure encryption

Recap: the story so far

Confidentiality: semantic security against a CPA attack

- Encryption secure against **eavesdropping only**

Integrity:

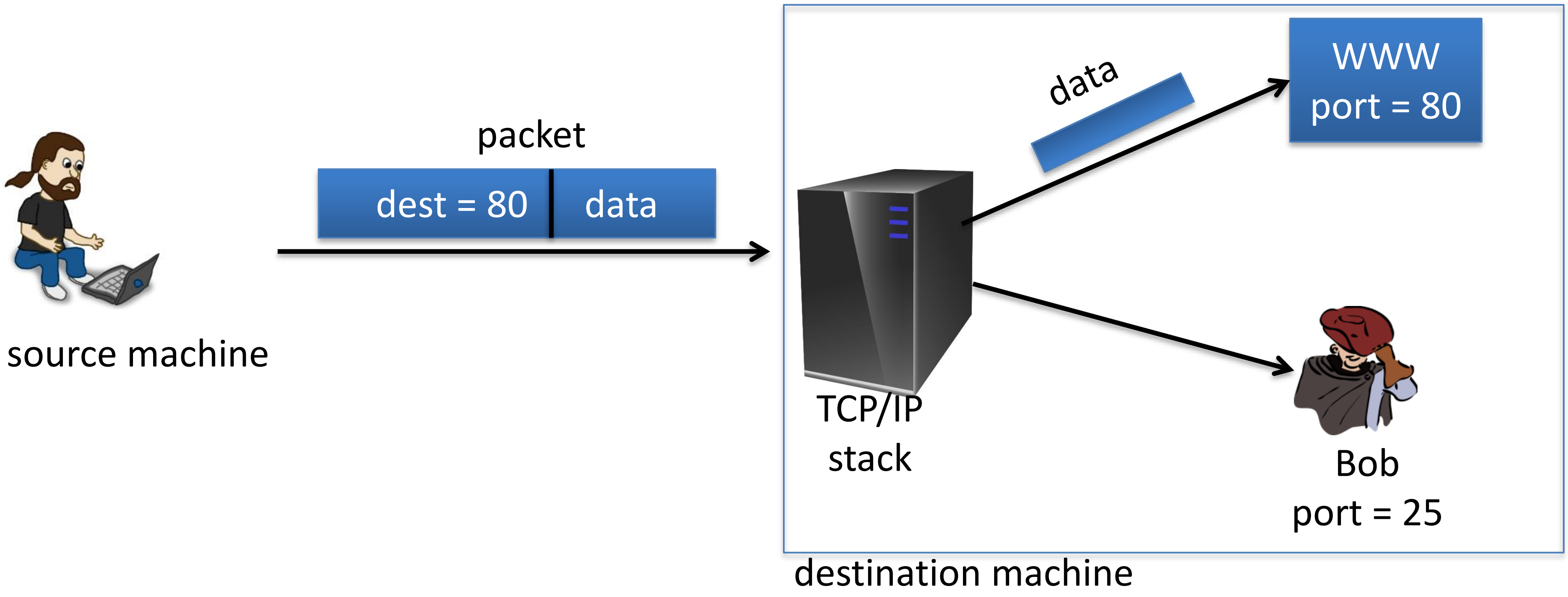
- Existential unforgeability under a chosen message attack
- CBC-MAC, HMAC, PMAC, CW-MAC

This module: encryption secure against **tampering** (active adversary)

- Ensuring both confidentiality and integrity

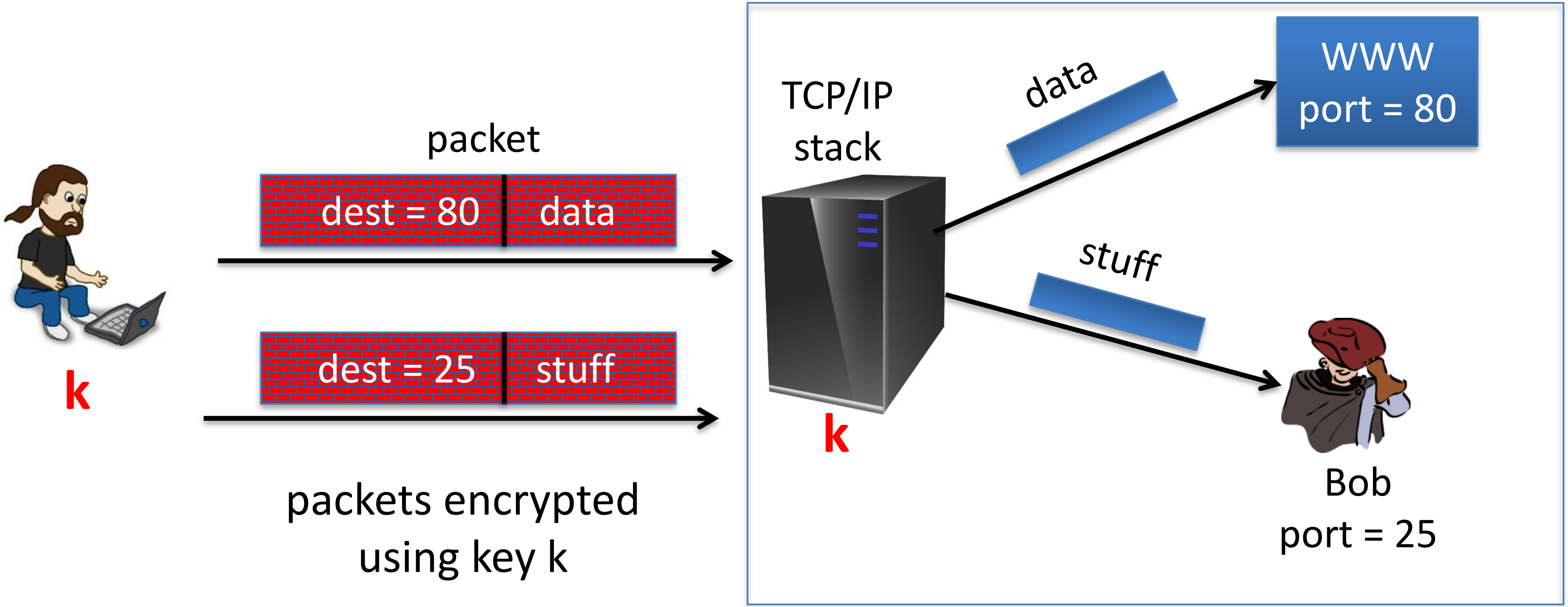
Sample tampering attacks

TCP/IP: (highly abstracted)



Sample tampering attacks

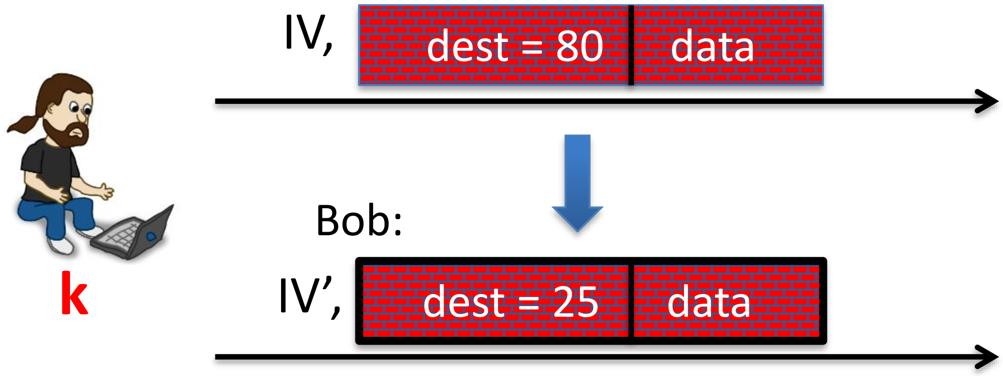
IPsec: (highly abstracted)

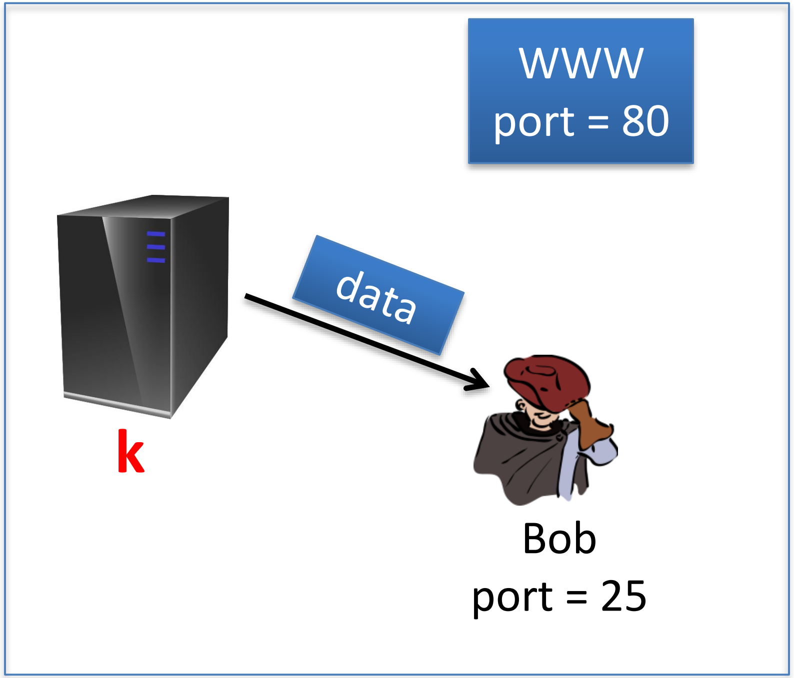


packets encrypted using key k

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?

$m[0] = D(k, c[0]) \oplus IV$ = "dest=80…"

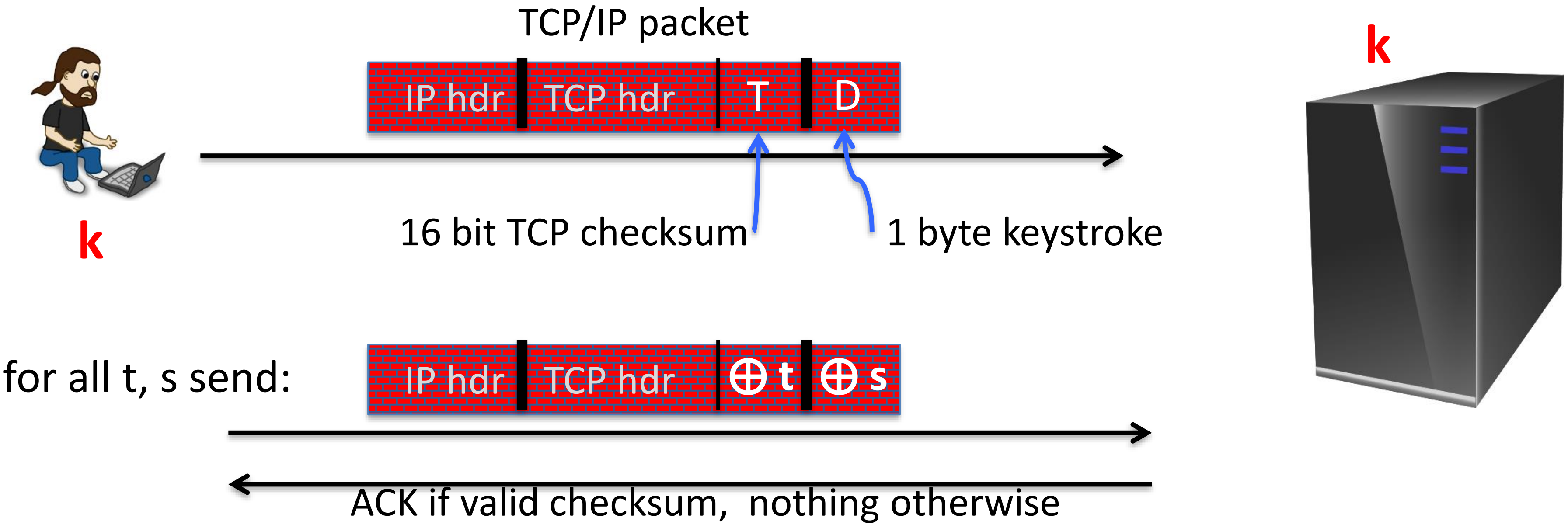
- IV' = IV ⊕ (…25…)
- IV' = IV ⊕ (…80…)
- IV' = IV ⊕ (…80…) ⊕ (…25…)
- It can't be done

$D(k, c[0]) \oplus IV' = D(k, c[0]) \oplus IV \oplus 80 \oplus 25$

$= \ldots 25 \ldots$

An attack using only network access

Remote terminal app.: each keystroke encrypted with CTR mode



$\{$ checksum(hdr, D) $= t \oplus$ checksum(hdr, $D \oplus s$) $\}$ $\Rightarrow$ can find D

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)

End of Segment

Authenticated Encryption

Definitions

Goals

An **authenticated encryption** system (E,D) is a cipher where

As usual: $E: K \times M \times N \longrightarrow C$

but $D: K \times C \times N \longrightarrow M \cup \{\bot\}$

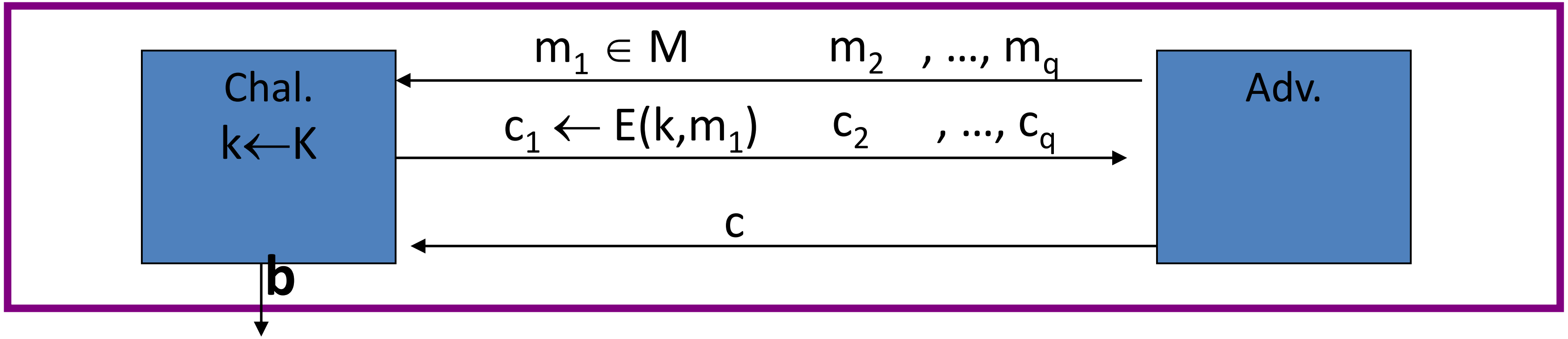
ciphertext is rejected

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

- sem. security under a CPA attack, and
- **ciphertext integrity:**
  attacker cannot create new ciphertexts that decrypt properly

Ciphertext integrity

Let (E,D) be a cipher with message space M.



$$\mathbf{b}=\begin{cases} 1 & \text{if } D(k,c) \neq \bot \text{ and } c \notin \{c_1, \dots, c_q\} \\ 0 & \text{otherwise} \end{cases}$$

Def: (E,D) has **<u>ciphertext integrity</u>** if for all "efficient" A:

$Adv_{CI}[A,E]$ = Pr[Chal. outputs 1] is "negligible."

Authenticated encryption

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

(1) semantically secure under CPA, and

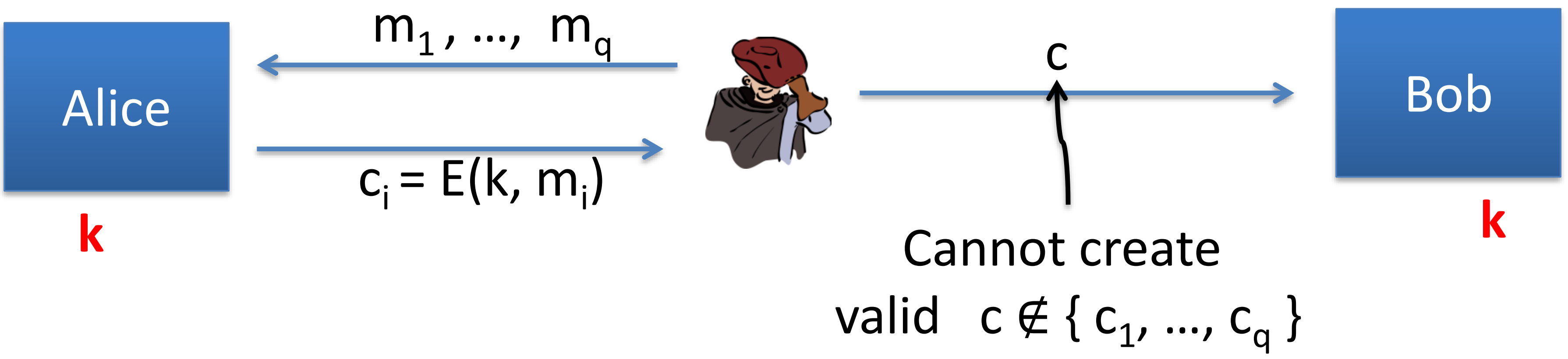
(2) has ciphertext integrity

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 $D(k,c) \neq \perp$ Bob knows message is from someone who knows k

(but message could be a replay)

Implication 2

Authenticated encryption $\Rightarrow$

Security against **chosen ciphertext attacks**
(next segment)

End of Segment

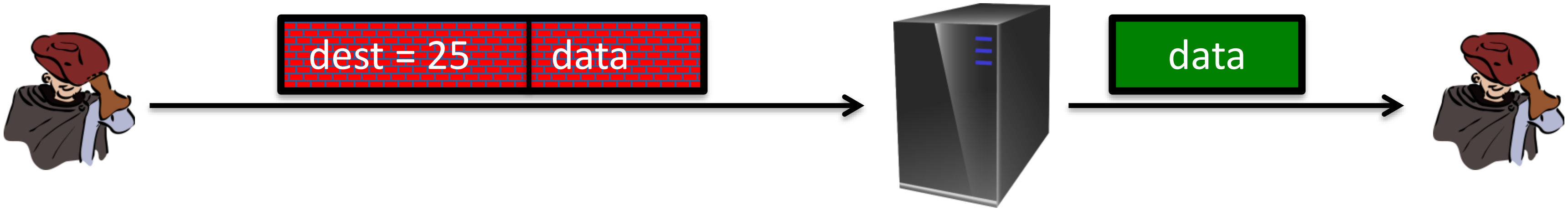
Authenticated Encryption

Chosen ciphertext attacks

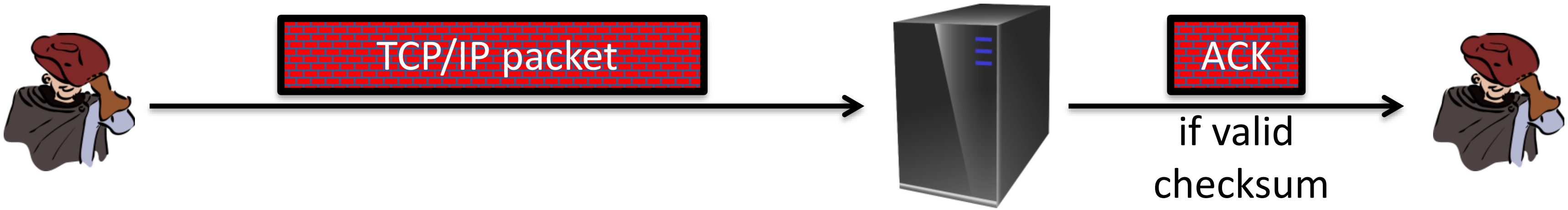
Example chosen ciphertext attacks

Adversary has ciphertext c that it wants to decrypt

- Often, adv. can fool server into decrypting **certain** ciphertexts (not c)



- Often, adversary can learn partial information about plaintext



Chosen ciphertext security

Adversary's power: both CPA and CCA

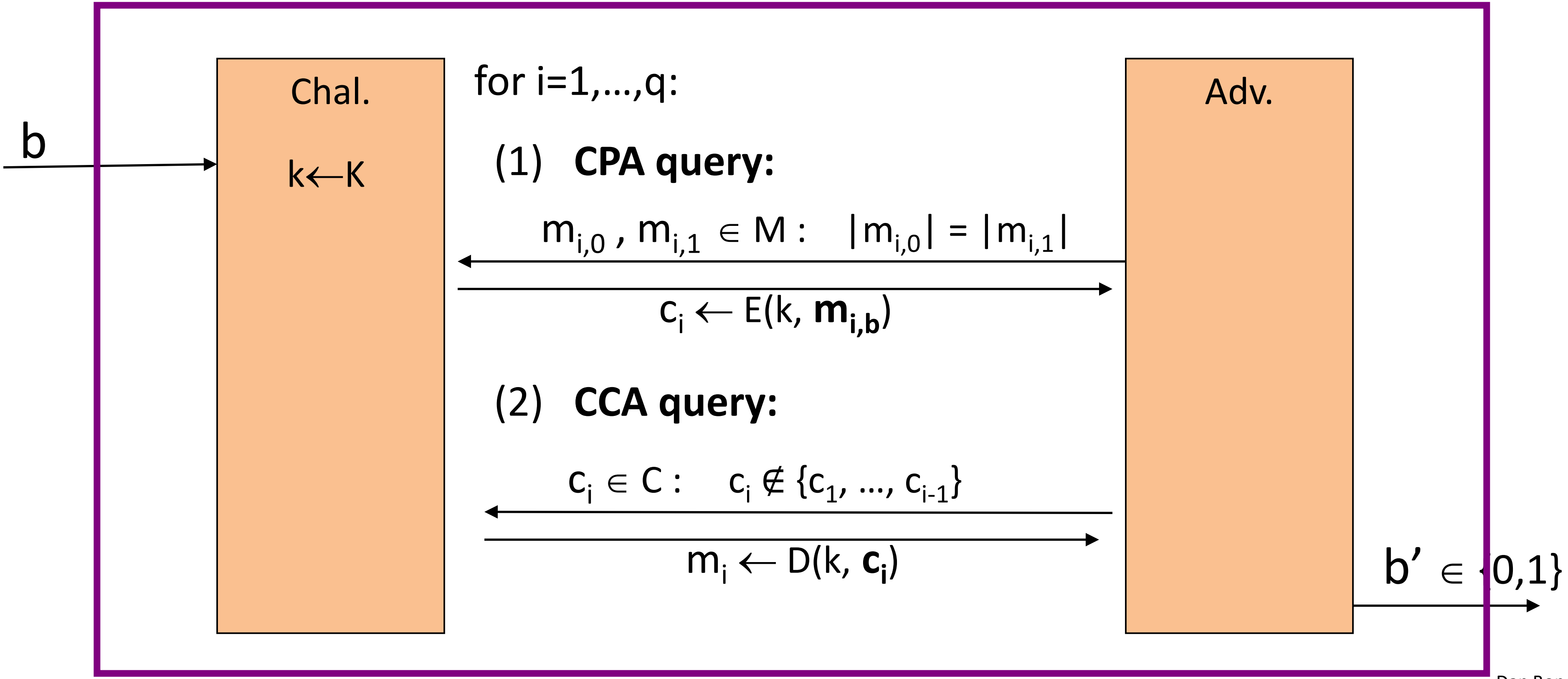
- Can obtain the encryption of arbitrary messages of his choice
- Can decrypt any ciphertext of his choice, other than challenge

(conservative modeling of real life)

Adversary's goal: Break semantic security

Chosen ciphertext security: definition

$\mathbb{E} = (E,D)$ cipher defined over (K,M,C). For $b=0,1$ define EXP(b):

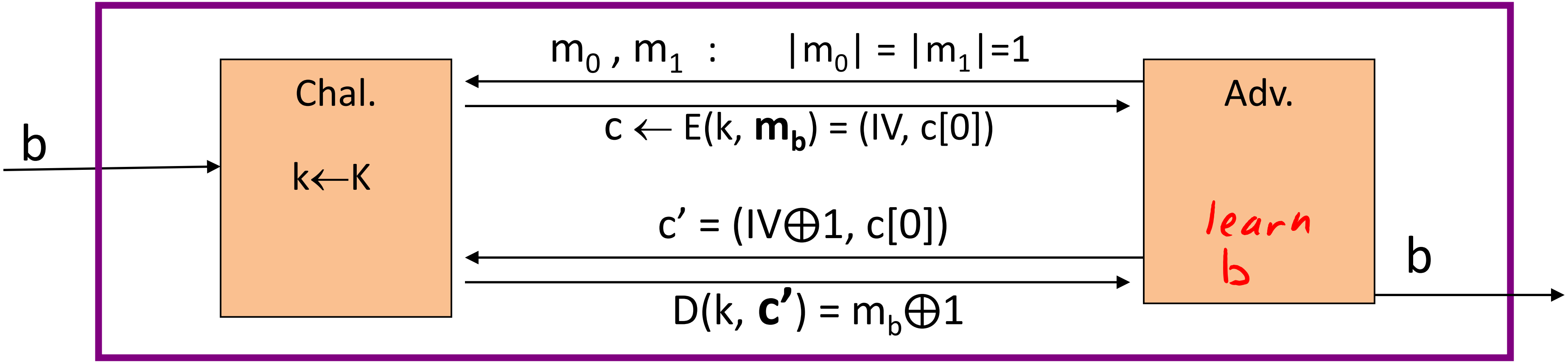


Chosen ciphertext security: definition

$\mathbb{E}$ is CCA secure if for all "efficient" A:

$$\text{Adv}_{\text{CCA}}[A,\mathbb{E}] = \left| \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] \right|$$ is "negligible."

Example: CBC with rand. IV is not CCA-secure



Authenticated enc. ⇒ CCA security

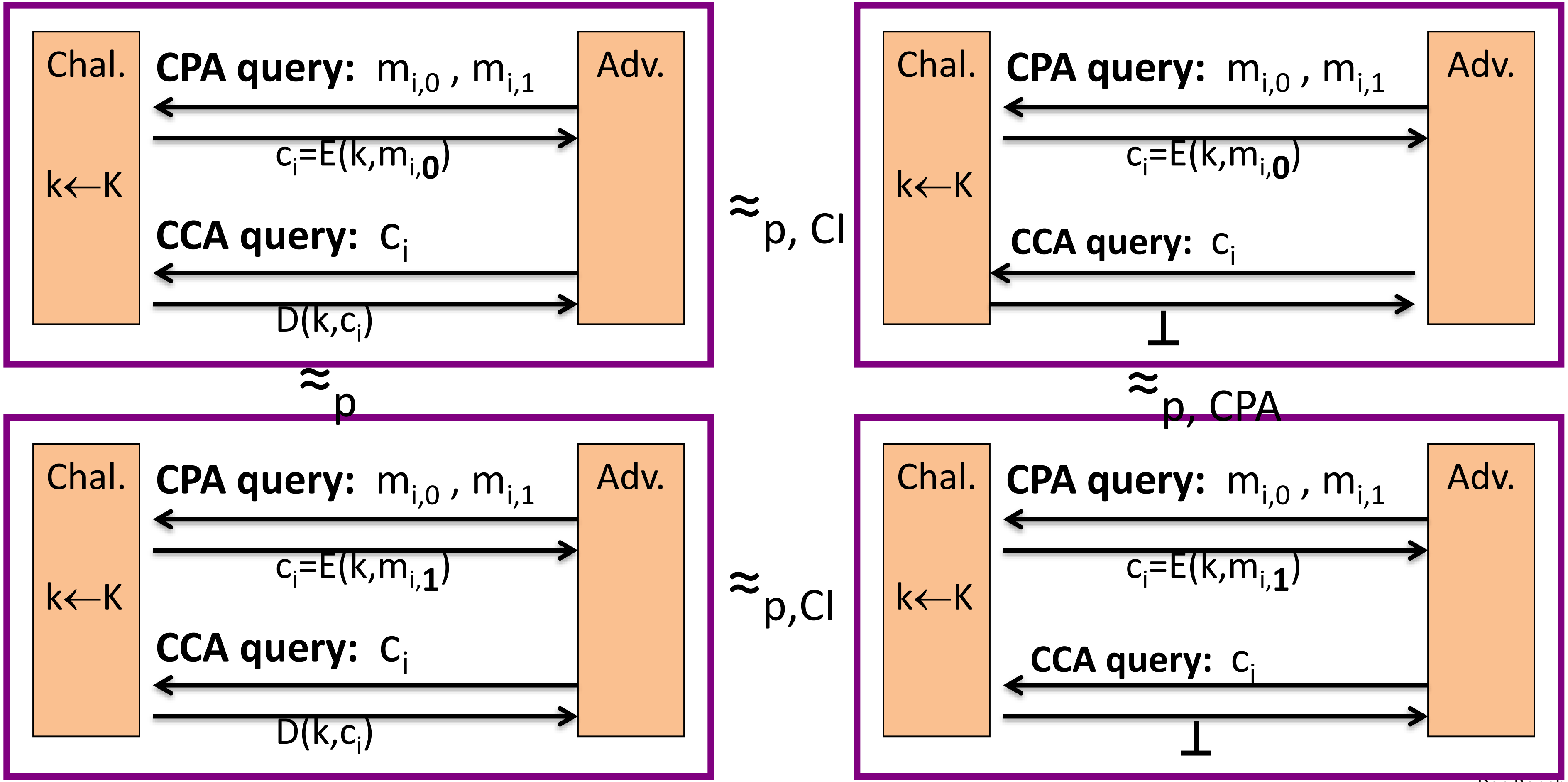
Thm: Let (E,D) be a cipher that provides AE.

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)

End of Segment

Authenticated Encryption

Constructions from ciphers and MACs

… but first, some history

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

Crypto APIs before then: (e.g. MS-CAPI) crypto API

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

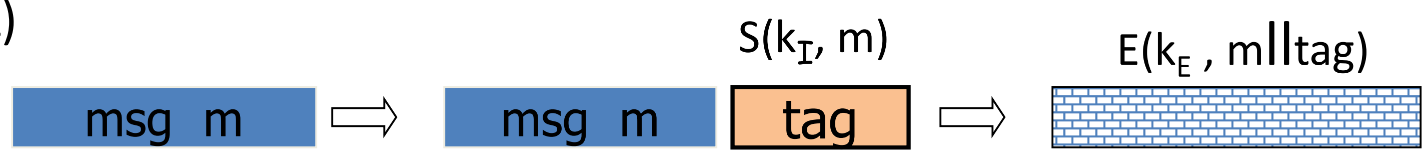
Every project had to combine the two itself without a well defined goal

- Not all combinations provide AE …

Combining MAC and ENC (CCA)

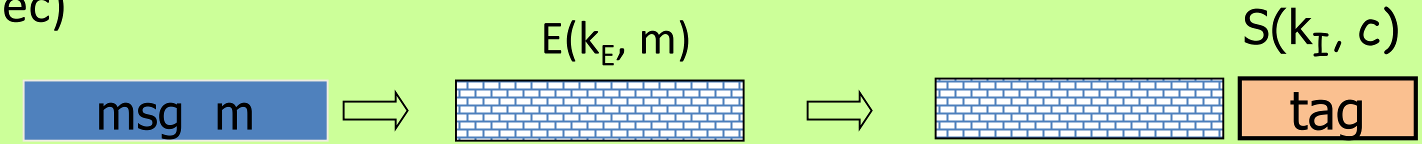
Encryption key k_E.   MAC key = k_I

Option 1: (SSL)

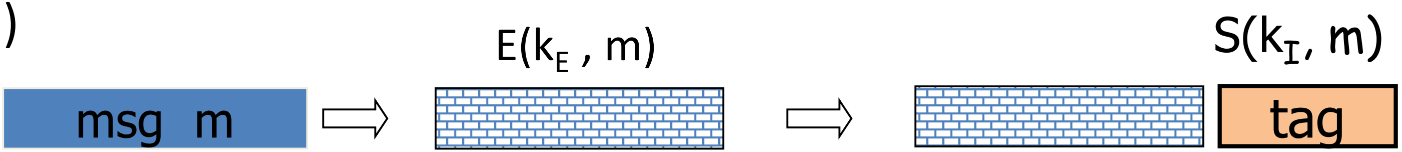


Option 2: (IPsec)

always correct



Option 3: (SSH)



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

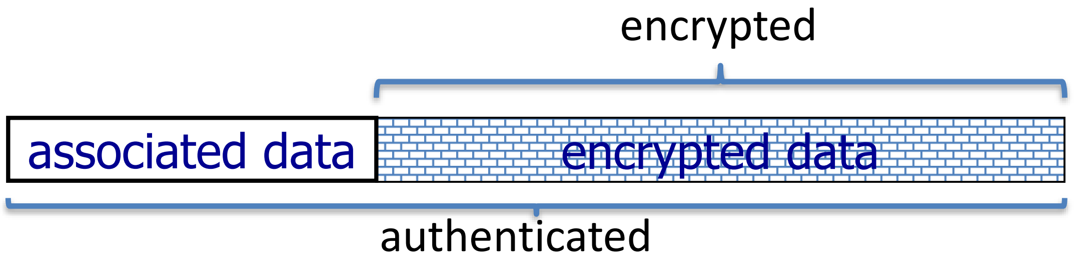
   however: when (E,D) is rand-CTR mode or rand-CBC
   M-then-E provides A.E.

   for rand-CTR mode, one-time MAC is sufficient

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.

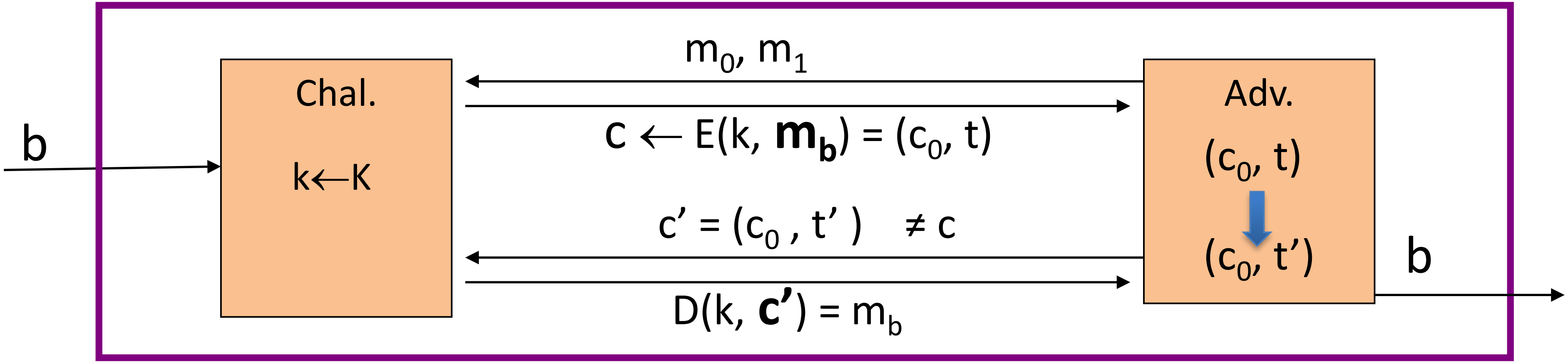


MAC Security -- an explanation

Recall: MAC security implies $(m, t) \not\Rightarrow (m, t')$

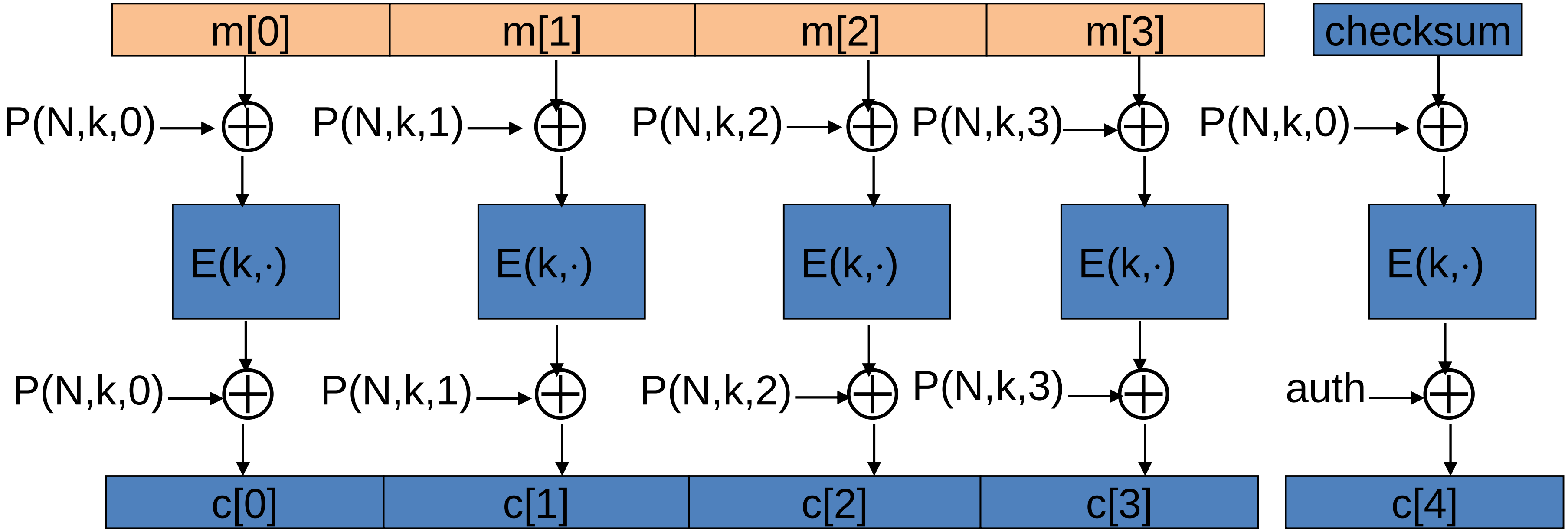
Why? Suppose not: $(m, t) \longrightarrow (m, t')$

Then Encrypt-then-MAC would not have Ciphertext Integrity !!



OCB: a direct construction from a PRP

More efficient authenticated encryption: one E() op. per block.



Performance:

Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

Cipher	code size	Speed (MB/sec)		
AES/GCM	large **	108	AES/CTR	139
AES/CCM	smaller	61	AES/CBC	109
AES/EAX	smaller	61		
			AES/CMAC	109
AES/OCB		129*	HMAC/SHA1	147

* extrapolated from Ted Kravitz's results ** non-Intel machines

End of Segment