CIS 5560

Cryptography
Lecture 11

Course website:
pratyushmishra.com/classes/cis-5560-s25/

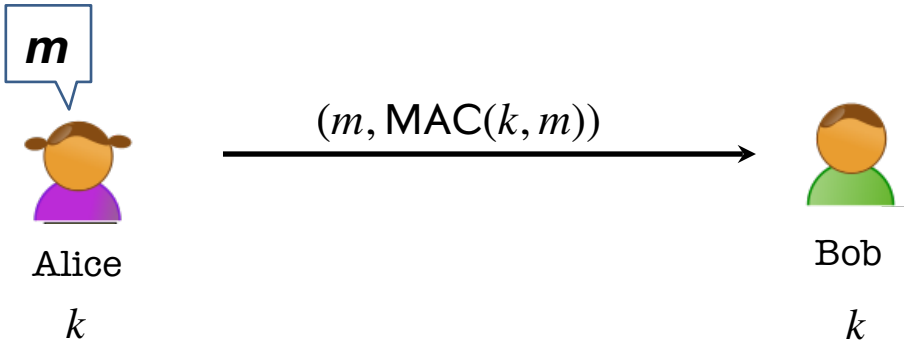
Slides adapted from Dan Boneh and Vinod Vaikuntanathan

Announcements

- **HW4 will be out today**
- **HW3 due tomorrow!**

Recap of last lecture

Constructing a MAC

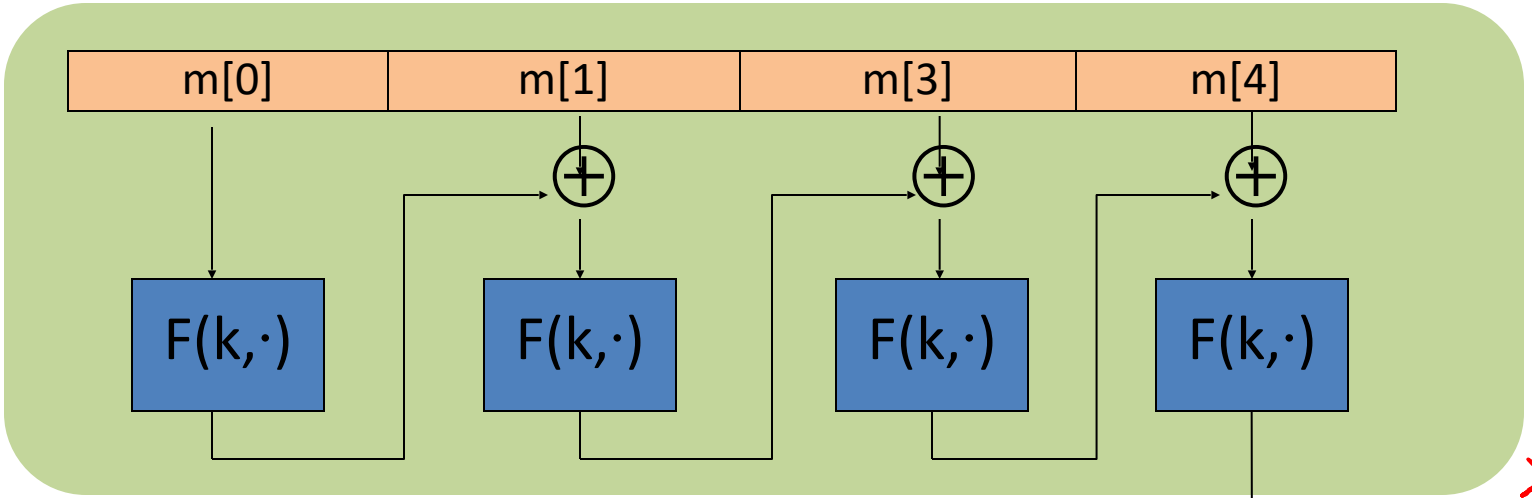


Gen(1^n): Produces a PRF key $k \leftarrow K$.
MAC$(k,\ m)$: Output $F_k(m)$.
Ver$(k,\ m,\ t)$: Accept if $F_k(m) = t$, reject otherwise.

Security: ??

Construction: encrypted CBC-MAC

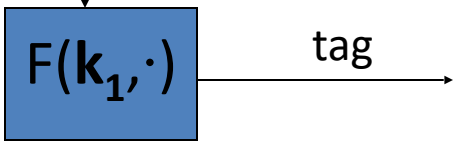
raw CBC



$X^{\leq L} = \bigcup_{i=1}^{L} X^i$

Let $F: K \times X \longrightarrow X$ be a PRP

Define new PRF $F_{ECBC}: K^2 \times X^{\leq L} \longrightarrow X$



Formal Definition: Collision-Resistant Hash Functions

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

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

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

MACs from Collision Resistance

Let MAC be a MAC for short messages over (K,M,T) (e.g. AES)

Let H: $M^{big} \rightarrow M$ be a hash function

Def: $MAC^{big} = (MAC^{big}, Ver^{big})$ over (K, M^{big}, T) as:

$MAC^{big}(k,m) = S(k,H(m))$; $Ver^{big}(k,m,t) = V(k,H(m),t)$

Thm: If MAC is a secure MAC and H is collision resistant

then MAC^{big} is a secure MAC.

Example: $MAC(k,m) = AES_{2\text{-block-cbc}}(k, SHA\text{-}256(m))$ is a secure MAC.

Generic attack on CRHFs

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

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

Algorithm:

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

How well will this work?

Today

- Constructing CRHFs with long inputs
- HMAC
- Encryption schemes with confidentiality *and* integrity
- Authenticated Encryption
  - IND-CPA + Ciphertext integrity
  - IND-CCA
  -

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 uniformly independent $r_1, \ldots, r_n$,

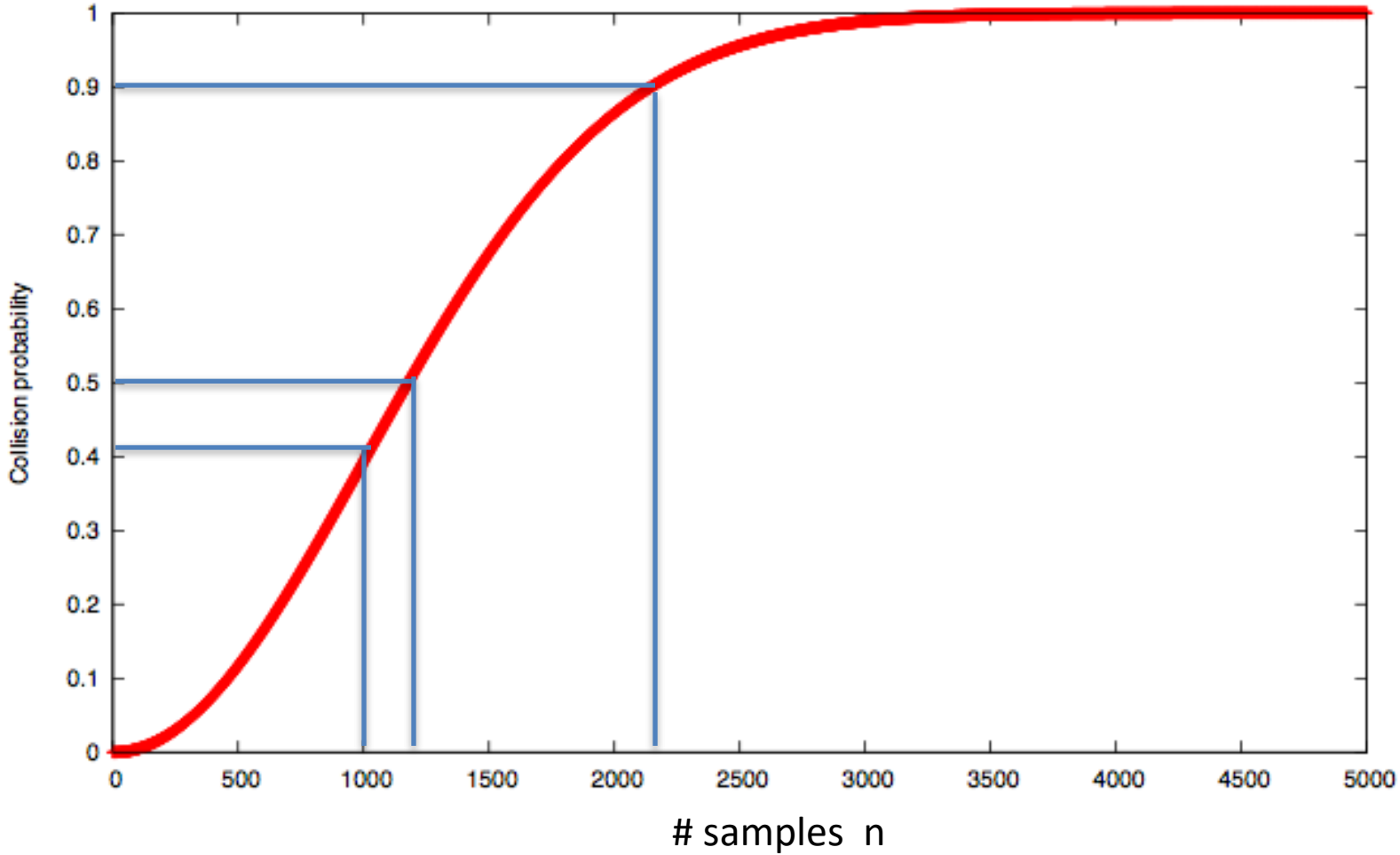
$$\Pr\left[\exists i \neq j : r_i = r_j\right] = 1 - \Pr\left[\forall i \neq j : r_i \neq r_j\right] = 1 - \left(\frac{B-1}{B}\right)\left(\frac{B-2}{B}\right) \cdots \left(\frac{B-n+1}{B}\right) =$$

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

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

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

$B=10^6$


Collision probability
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
samples n

Generic attack

Algorithm:

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

Expected number of iteration ≈ 2

Running time: **$O(2^{n/2})$** (space $O(2^{n/2})$)

Sample CRHFs: Crypto++ 5.6.0 [Wei Dai]

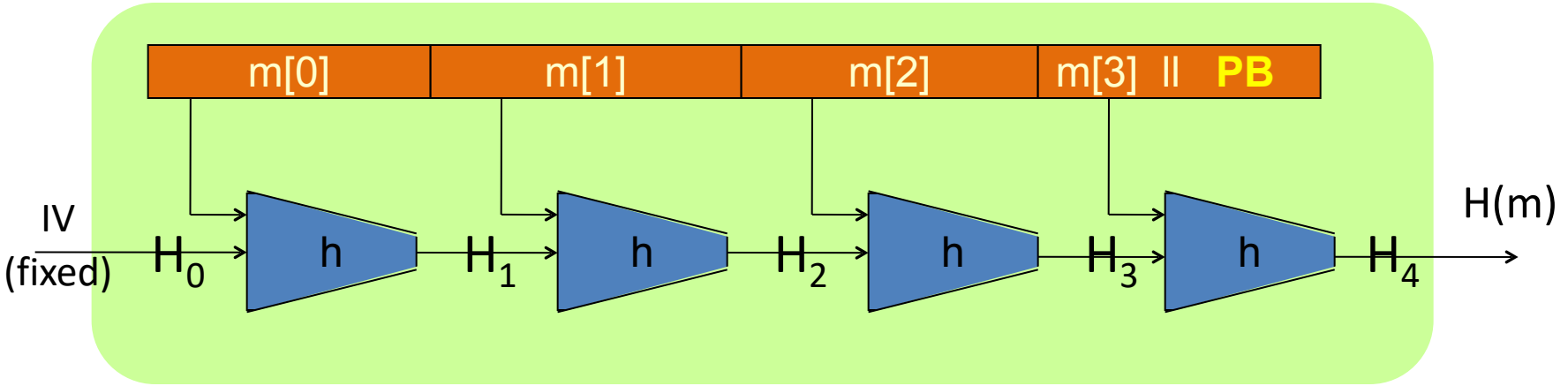
AMD Opteron, 2.2 GHz (Linux)

	function	digest size (bits)	Speed (MB/sec)	generic attack time
NIST standards	SHA-1	160	153	2^{80}
	SHA-256	256	111	2^{128}
	SHA-512	512	99	2^{256}
	Whirlpool	512	57	2^{256}

* SHA-1 is broken; do not use!

Constructing CRHFs for long messages:
The Merkle-Damgard Paradigm

The Merkle-Damgard iterated construction



Given **h: T × X ⟶ T** (compression function)

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

PB: padding block

1000…0 || msg len

64 bits

If no space for PB
add another block

MD collision resistance

Thm: if h is collision resistant then so is H.

Proof: collision on H ⇒ collision on h

Intermediate hashes

Suppose H(M) = H(M'). We build collision for h.

$$IV = H_0 \;,\; H_1 \;,\; \ldots \;,\; H_t \;,\; H_{t+1} = H(M)$$

$$IV = H_0' \;,\; H_1' \;,\; \ldots \;,\; H'_r,\; H'_{r+1} = H(M')$$

$$h(H_t, M_t \,\|\, PB) = H_{t+1} = H'_{r+1} = h(H'_r, M'_r \,\|\, PB')$$

There must be a r and t such that this holds

Otherwise,

Suppose $H_t = H'_r$ and $M_t = M'_r$ and $PB = PB'$

$\Rightarrow t = r$

Then: $h(H_{t-1}, M_{t-1}) = H_t = H'_t = h(H'_{t-1}, M'_{t-1})$

If $\begin{bmatrix} H_{t-1} \neq H'_{t-1} \\ \text{or} \\ M_{t-1} \neq M'_{t-1} \end{bmatrix}$ then we have a collision on h. STOP.

otherwise, $H_{t-1} = H'_{t-1}$ and $M_t = M'_t$ and $M_{t-1} = M'_{t-1}$.

Iterate all the way to beginning and either:

(1) find collision on h, or

(2) $\forall i: M_i = M'_i \Rightarrow M = M'$ ← cannot happen because M, M' are collision on H.

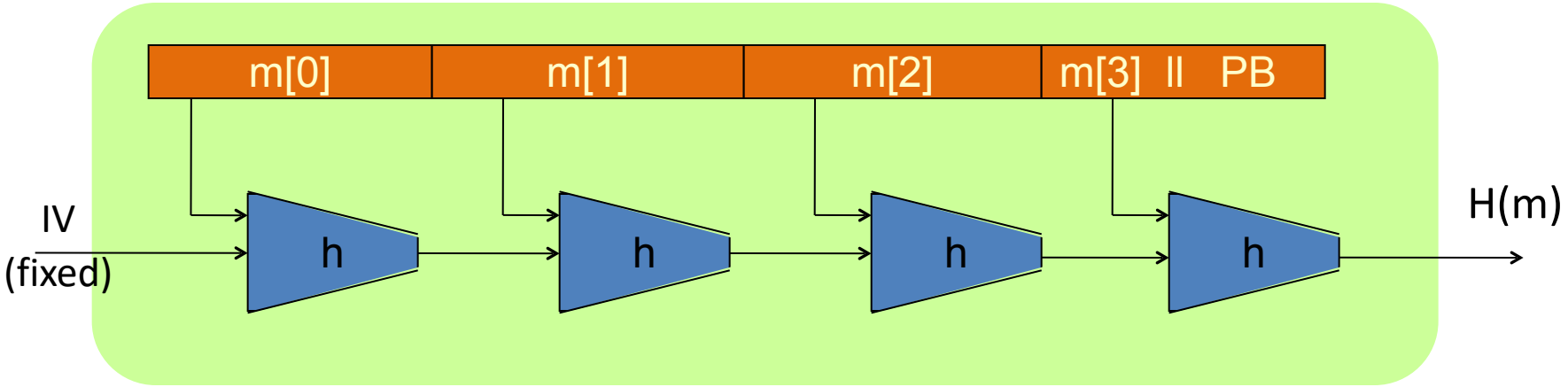
⇒ To construct C.R. function,

suffices to construct compression function

End of Segment

HMAC: a MAC from SHA-256

The Merkle-Damgard iterated construction



Thm: h collision resistant ⇒ H collision resistant

Can we use H to directly build a MAC?

MAC from a Merkle-Damgard Hash Function

H: $X^{\leq L} \longrightarrow T$ a C.R. Merkle-Damgard Hash Function

Attempt #1: **S(k, m) = H(k ll m)**

This MAC is insecure because:

- Given H(k ll m) can compute H(w ll k ll m ll PB) for any w.
- Given H(k ll m) can compute H(k ll m ll w) for any w.
- Given H(k ll m) can compute H(k ll m ll PB ll w) for any w.
- Anyone can compute H(k ll m) for any m.

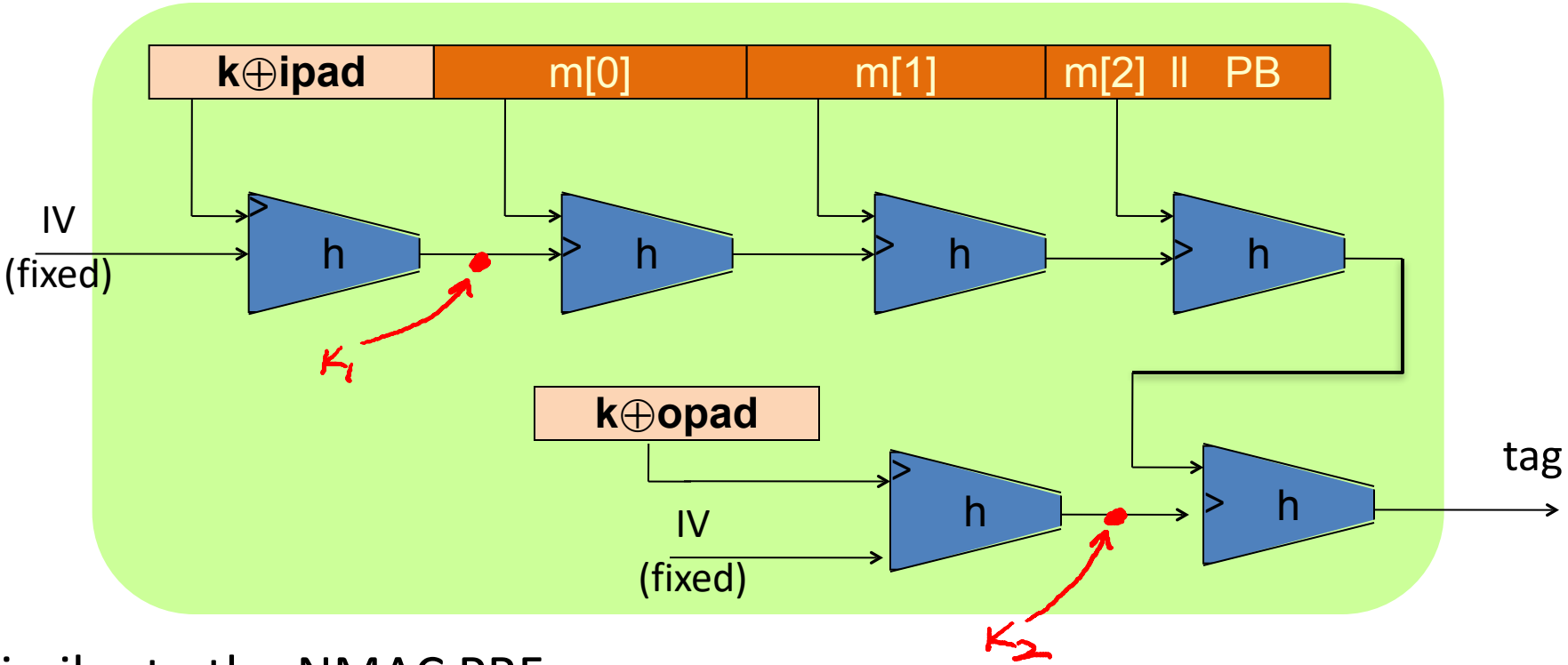
Standardized method: HMAC (Hash-MAC)

Most widely used MAC on the Internet.

Building a MAC out of a hash function H:

HMAC: $\mathsf{MAC}(k, m) = H(k \oplus \text{opad} \,||\, H(k \oplus \text{ipad} \,||\, m))$

HMAC in pictures



Similar to the NMAC PRF.

main difference: the two keys k_1, k_2 are dependent

HMAC properties

Built from a black-box implementation of SHA-256.

HMAC is assumed to be a secure PRF

- Can be proven under certain PRF assumptions about h(.,.)
- Security bounds similar to NMAC
  - Need $q^2/|T|$ to be negligible ($q << |T|^{½}$)

Story so far

Confidentiality: semantic security against a CPA attack

- Encryption secure against **eavesdropping only**

Integrity:

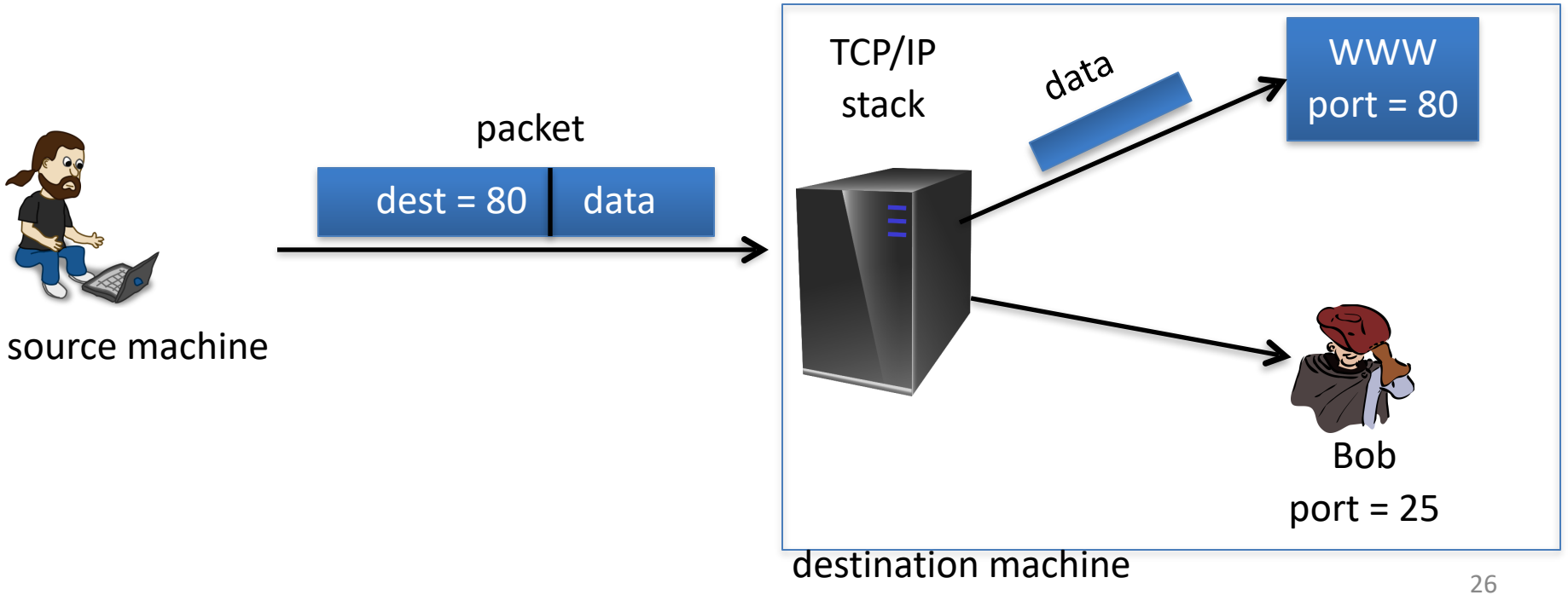
- Existential unforgeability under a chosen message attack
- CBC-MAC, HMAC, PMAC, CW-MAC

This module: encryption secure against **tampering**

- Ensuring both confidentiality and integrity

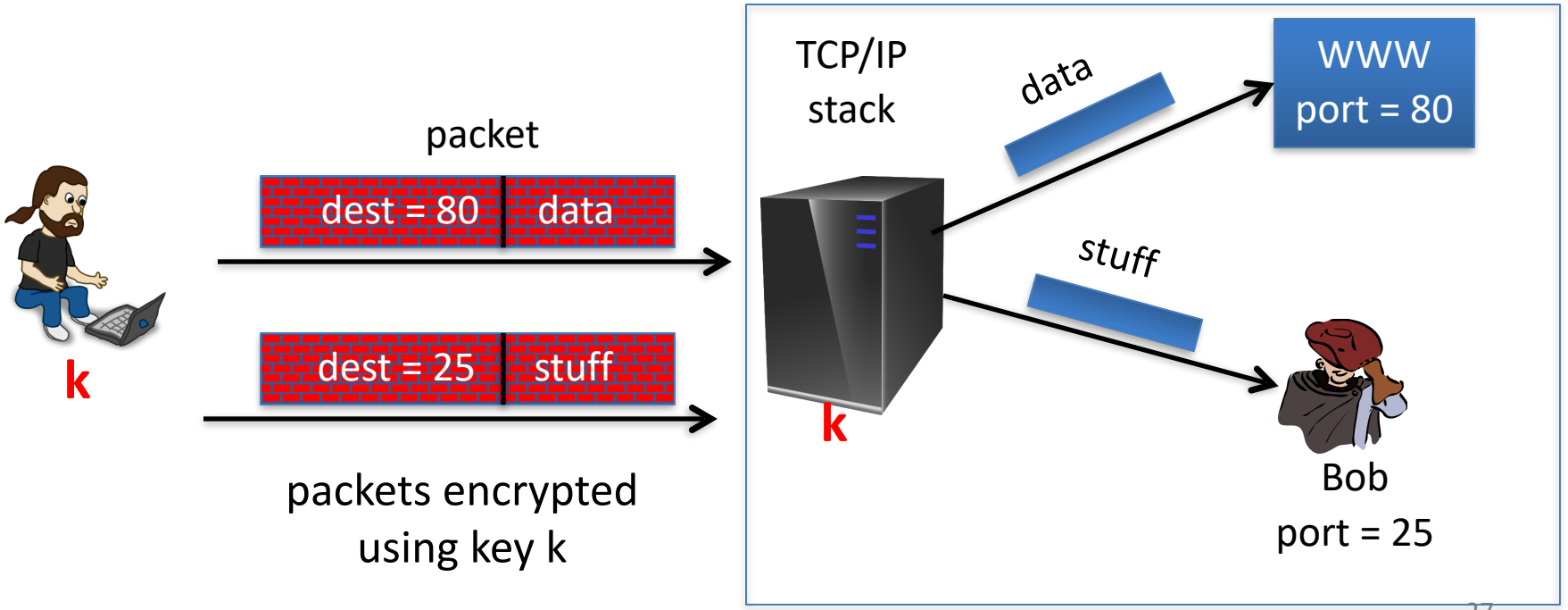
Sample tampering attacks

TCP/IP: (highly abstracted)



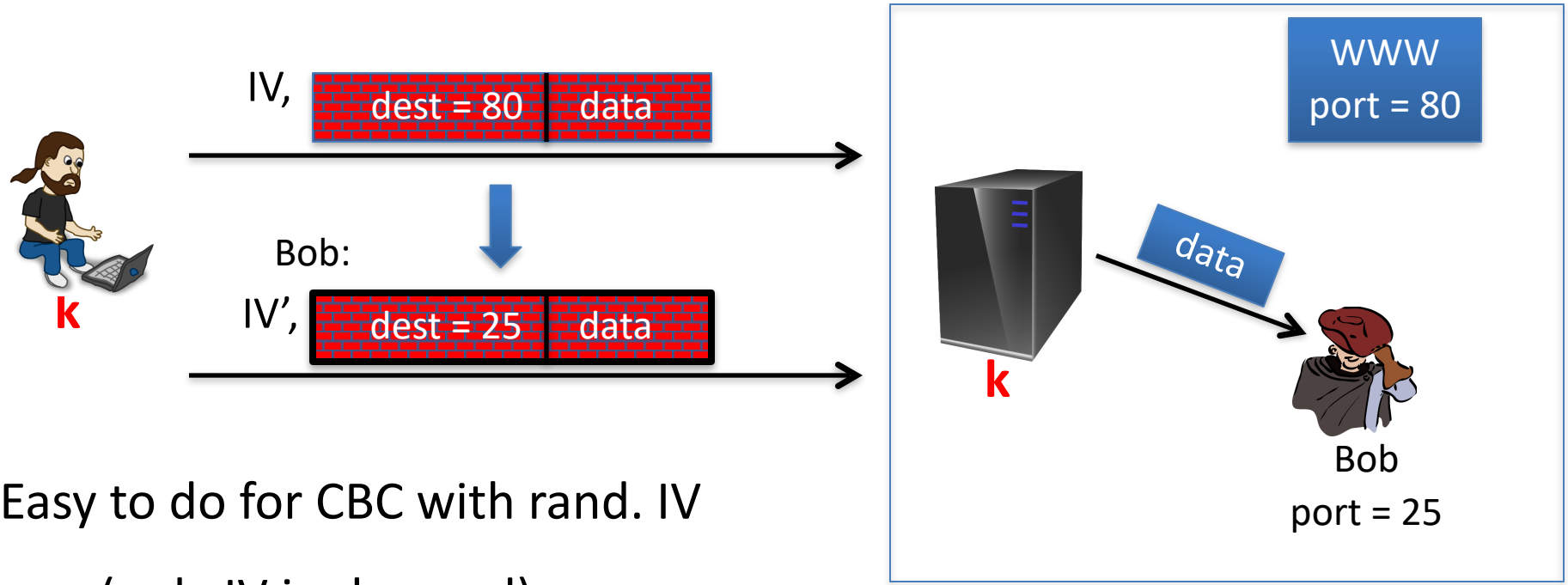
Sample tampering attacks

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?

$m[0] = D(k, c[0]) \oplus IV$ = "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: $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$

but $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

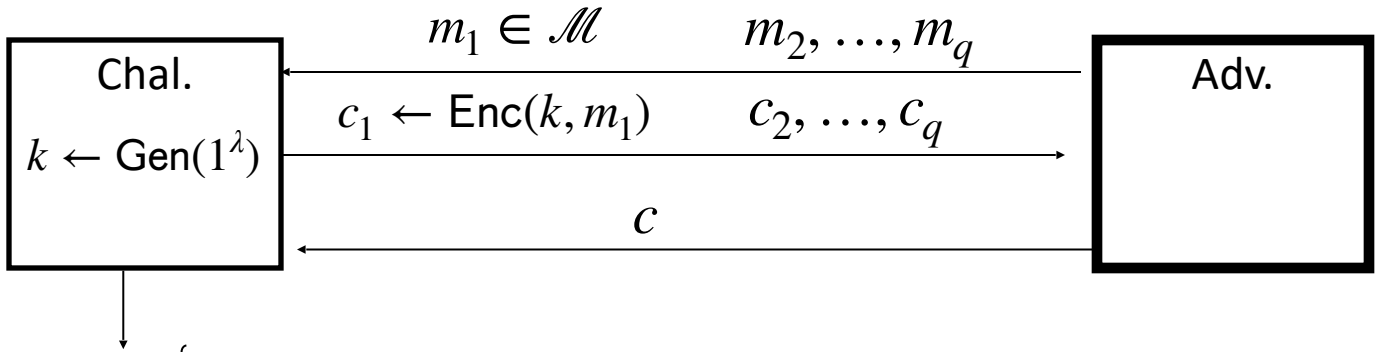
ciphertext is rejected

Security: 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 $\mathcal{M}$.



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

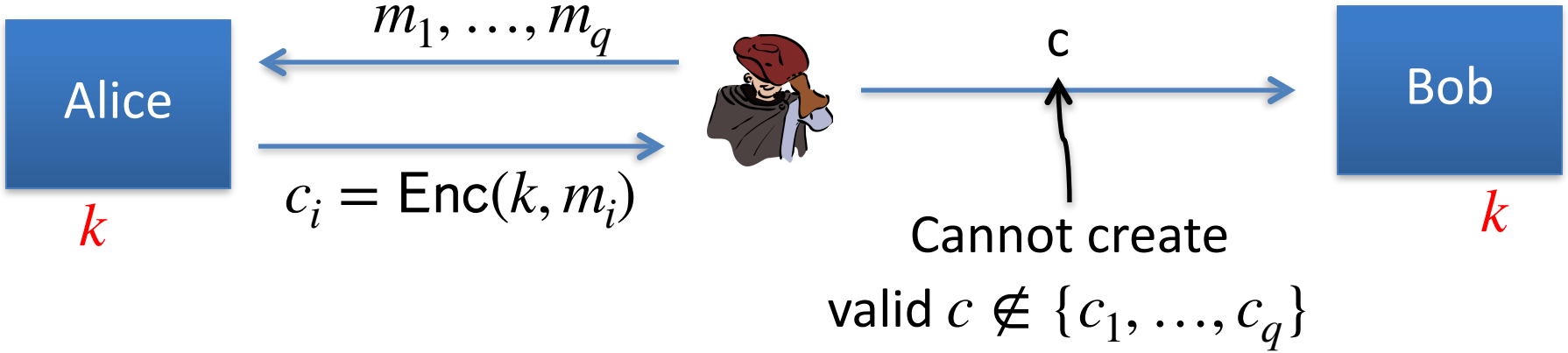
(2) has ciphertext integrity

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

(but message could be a replay)

Implication 2

Authenticated encryption

↓

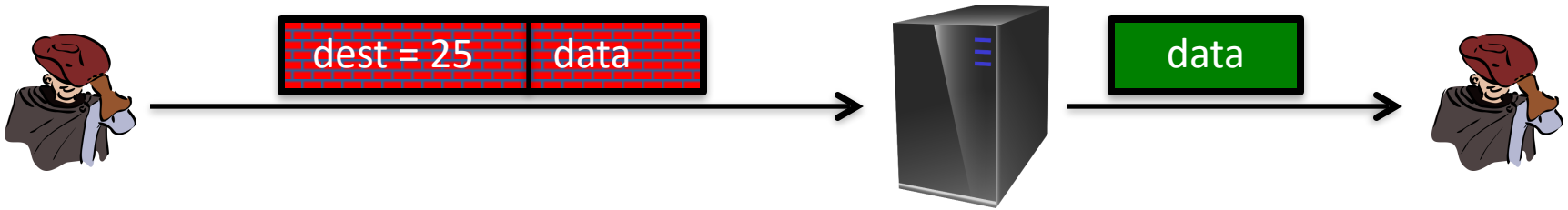
Security against **chosen ciphertext attacks**

Chosen ciphertext attacks

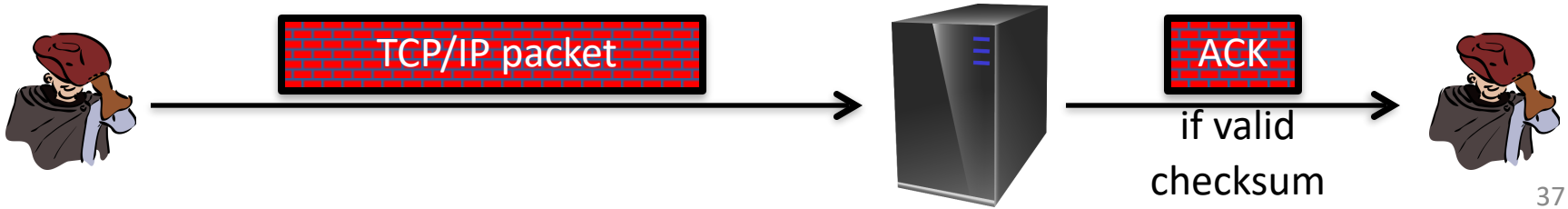
Example chosen ciphertext attacks

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

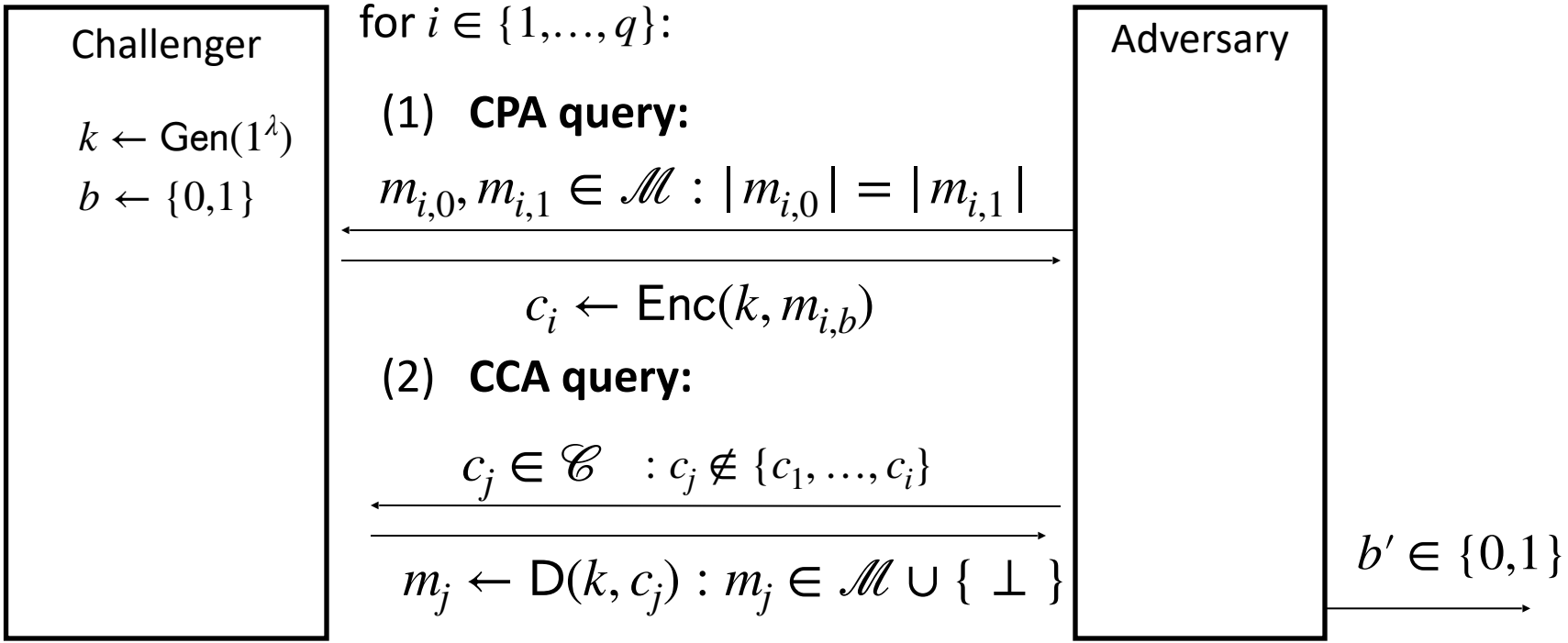
(conservative modeling of real life)

Adversary's goal:

Learn partial information about challenge plaintext

Chosen ciphertext security: definition

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



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

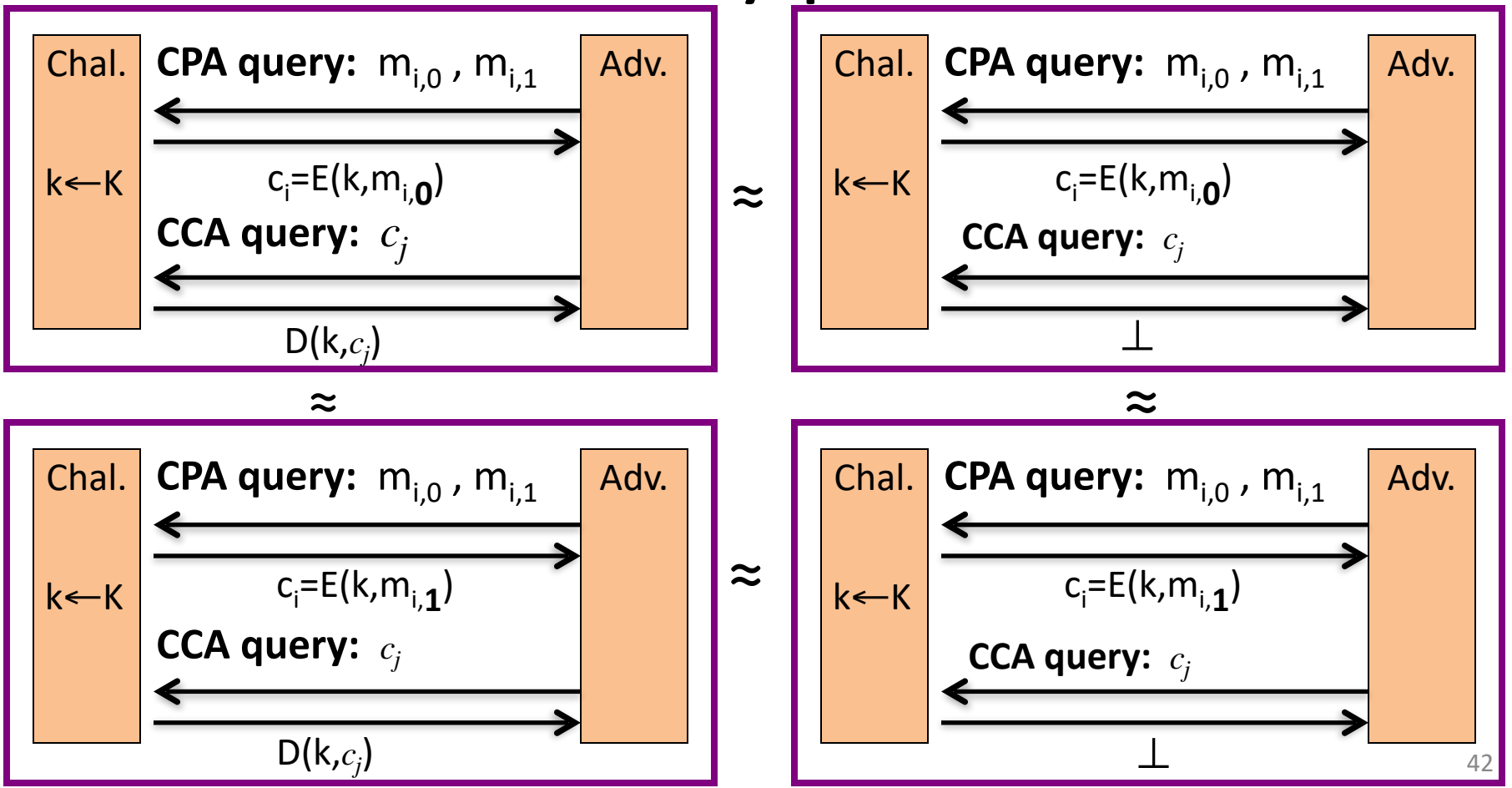
Thm: Let (E,D) be a cipher that provides AE.

Then (E,D) is CCA secure !

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

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

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:

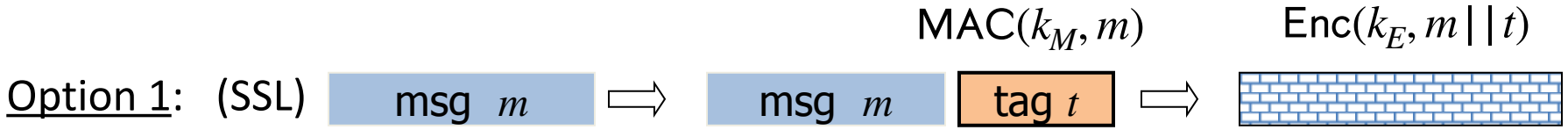
- Provide API for CPA-secure encryption (e.g. CBC with rand. IV)
- Provide API for MAC (e.g. HMAC)

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

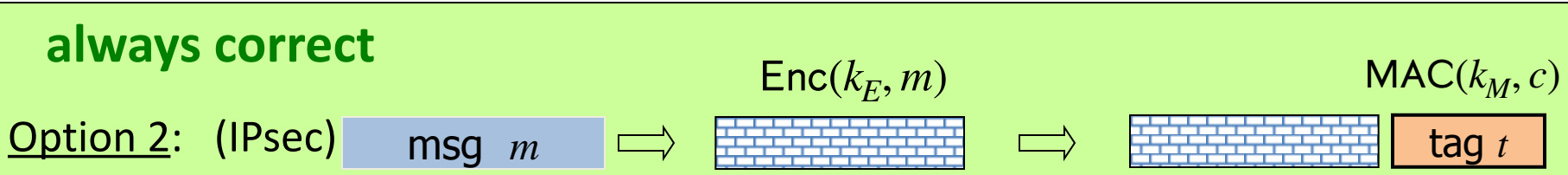
- Not all combinations provide AE …

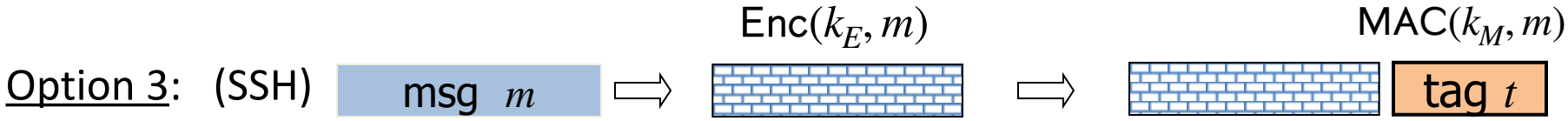
Combining MAC and ENC (CCA)

Encryption key k_E. MAC key = k_M



always correct





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

Security of Encrypt-then-MAC

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 !!