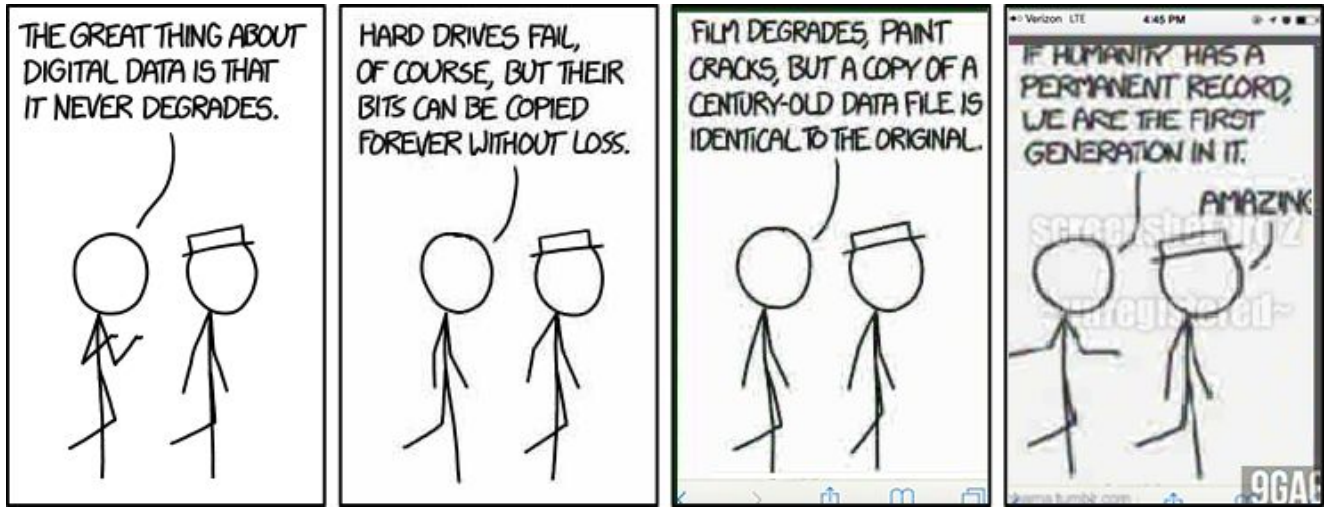
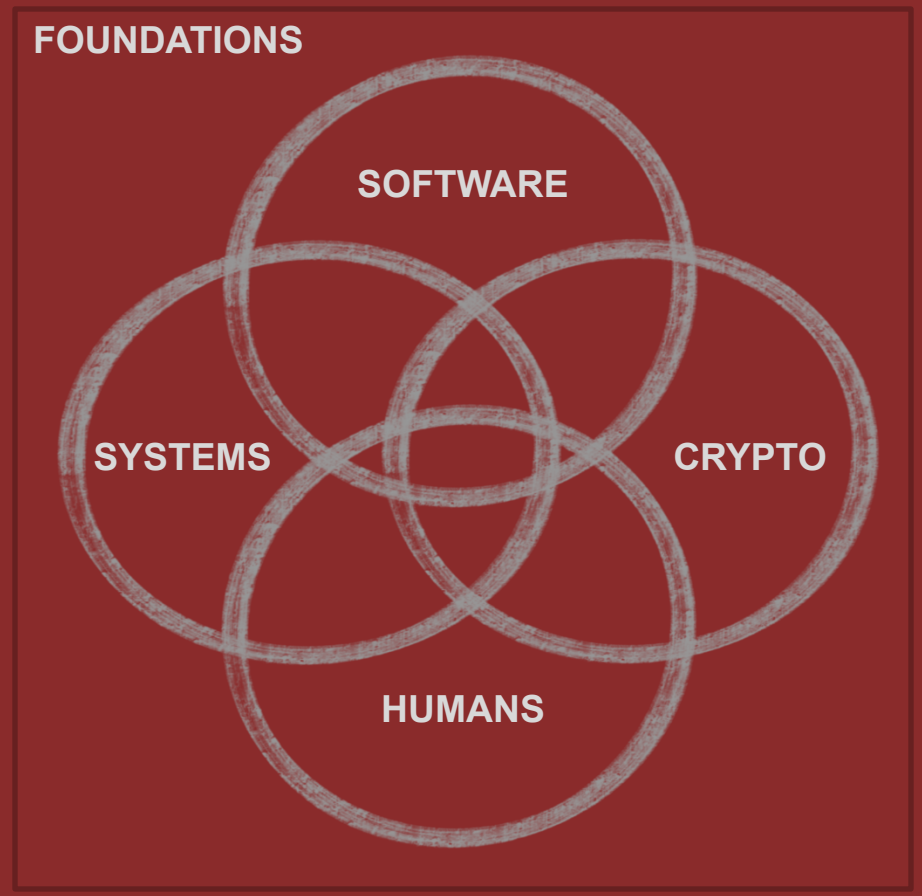
Διάλεξη #15 - Integrity

Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Εισαγωγή στην Ασφάλεια

Θανάσης Αυγερινός





Huge thank you to David Brumley from Carnegie Mellon University for the guidance and content input while developing this class (lots of slides from Dan Boneh @ Stanford!)

Ανακοινώσεις / Διευκρινίσεις

- Η εργασία #2 κλείνει στις 30, μην ξεχάσουμε το write up!
- Η επόμενη εργασία θα είναι σε web θέματα και θα ανοίξει από βδομάδα

Την προηγούμενη φορά

- Encryption Modes
  - Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
  - Counter Mode (CTR)
- Mistakes and Attacks

Σήμερα

- Message Integrity
  - Message Authentication Codes (MACs)
  - CBC-MAC, NMAC, CMAC
- Introduction to Hashing

Security in the News

Two brothers studying at MIT charged with stealing $25 million worth of cryptocurrency

Two brothers studying at the prestigious Massachusetts Institute of Technology (MIT) were arrested on Wednesday (May 15) and charged with stealing $25 million worth of cryptocurrency. Authorities said that the brothers- Anton Peraire-Bueno, 24, and James Peraire-Bueno, 28,- carried out a cutting-edge scheme to exploit the Ethereum blockchain's integrity and steal millions of dollars of cryptocurrency, the news agency Reuters reported.

The brothers executed their elaborate heist in April last year, stealing $25 million from traders in just 12 seconds by fraudulently gaining access to pending transactions and altering the movement of cryptocurrency, authorities said.

At MIT, the brothers studied computer science and mathematics and developed the skills and education they relied upon to carry out their fraud, prosecutors said on Wednesday.

An indictment charged them with conspiracy to commit wire fraud, wire fraud, and conspiracy to commit money laundering. The indictment alleged that for months, the brothers plotted to manipulate and tamper with the protocols used to validate transactions for inclusion on the Ethereum blockchain.

Message Integrity

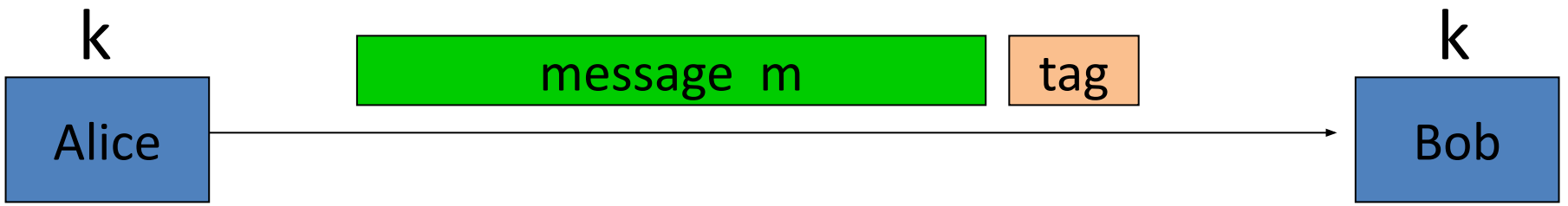
Message Integrity

Goal: **integrity**, no confidentiality.

Examples:

- Transaction data / ledger.
- Communications.
- Public binaries on disk.
- Banner ads on web pages.

Message integrity: MACs



Generate tag (Sign):

tag ← S(k, m)

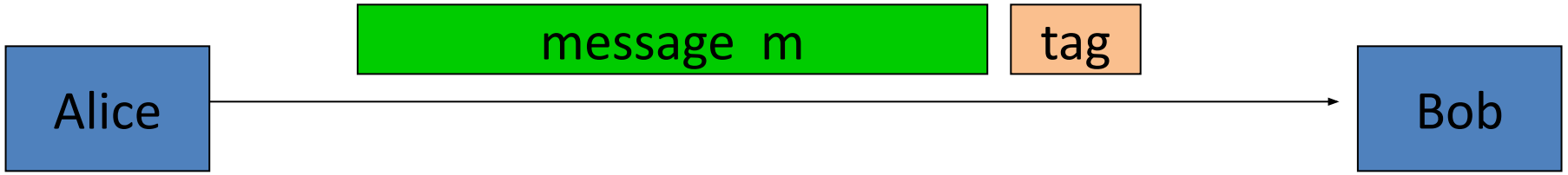
Verify tag:

V(k, m, tag) $\overset{?}{=}$ `yes'

Def: **MAC** I = (S,V) defined over (K,M,T) is a pair of algs:

- S(k,m) outputs t in T
- V(k,m,t) outputs `yes' or `no'

Integrity requires a secret key



Generate tag:

tag ← CRC(m)

Verify tag:

V(m, tag) $\stackrel{?}{=}$ `yes'

- Attacker can easily modify message m and re-compute CRC.
- CRC designed to detect **random**, not malicious errors.

Secure MACs

Attacker's power: **chosen message attack**

- for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k, m_i)$

Attacker's goal: **existential forgery**

- produce some **<u>new</u>** valid message/tag pair (m,t).

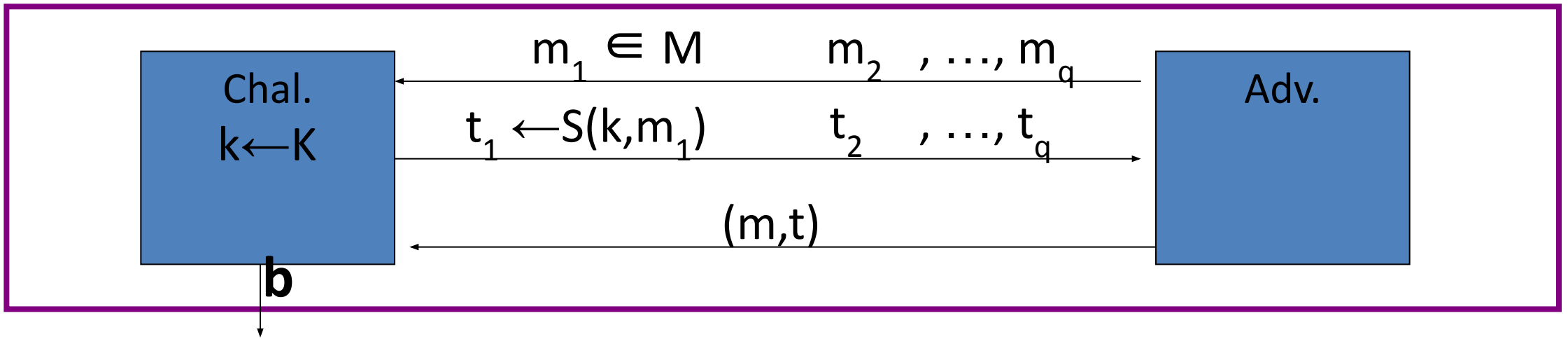
$$(m,t) \notin \{ (m_1,t_1) , \ldots , (m_q,t_q) \}$$

$\Rightarrow$ attacker cannot produce a valid tag for a new message

$\Rightarrow$ given (m,t) attacker cannot even produce (m,t') for $t' \neq t$

Secure MACs

- For a MAC I=(S,V) and adv. A define a MAC game as:



b=1 if V(k,m,t) = `yes' and (m,t) $\notin$ { (m_1,t_1) , … , (m_q,t_q) }

b=0 otherwise

Def: I=(S,V) is a **secure MAC** if for all "efficient" A:

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

Let I = (S,V) be a MAC.

Suppose an attacker is able to find $m_0 \neq m_1$ such that

$S(k, m_0) = S(k, m_1)$ for ½ of the keys k in K

Can this MAC be secure?

- Yes, the attacker cannot generate a valid tag for m_0 or m_1
- No, this MAC can be broken using a chosen msg attack
- It depends on the details of the MAC
-

Let I = (S,V) be a MAC.

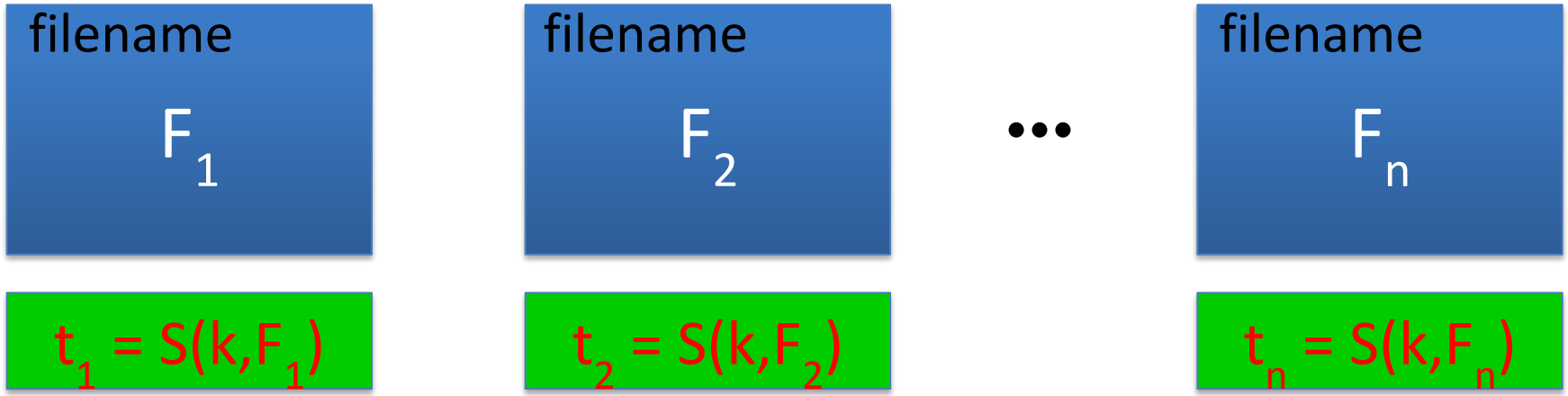
Suppose S(k,m) is 5 bits long

Can this MAC be secure?

- ○ No, an attacker can simply guess the tag for messages
- ○ It depends on the details of the MAC
- ○ Yes, the attacker cannot generate a valid tag for any message
- ○

Example: protecting system files

Suppose at install time the system computes:



k derived from user's password

Later a virus infects system and modifies system files

User reboots into clean OS and supplies his password

– Then: secure MAC ⇒ all modified files will be detected

Using PRFs to build MACs

Review: Secure MACs

MAC: signing alg. $S(k,m) \rightarrow t$ and verification alg. $V(k,m,t) \rightarrow 0,1$

Attacker's power: **chosen message attack**

- for $m_1, m_2, \ldots, m_q$ attacker is given $t_i \leftarrow S(k,m_i)$

Attacker's goal: **existential forgery**

- produce some **new** valid message/tag pair (m,t).

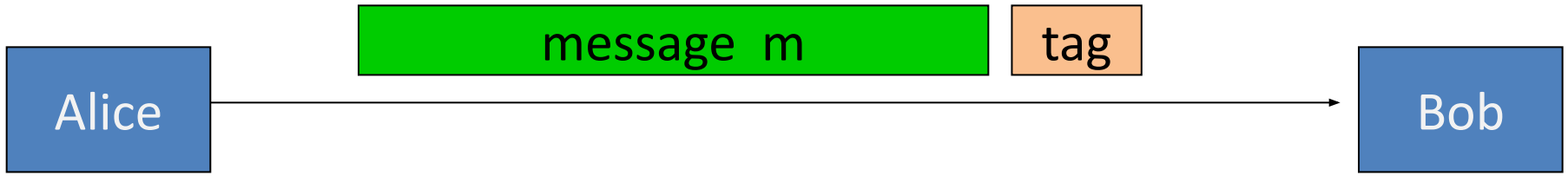
$$(m,t) \notin \{ (m_1,t_1), \ldots, (m_q,t_q) \}$$

$\Rightarrow$ attacker cannot produce a valid tag for a new message

Secure PRF ⇒ Secure MAC

For a PRF **F: K × X → Y** define a MAC $I_F = (S,V)$ as:

- S(k,m) := F(k,m)
- V(k,m,t): output `yes' if t = F(k,m) and `no' otherwise.



tag ← F(k,m)

accept msg if

tag = F(k,m)

A bad example

Suppose $F: K \times X \longrightarrow Y$ is a secure PRF with $Y = \{0,1\}^{10}$

Is the derived MAC I_F a secure MAC system?

- Yes, the MAC is secure because the PRF is secure
- No tags are too short: anyone can guess the tag for any msg
- It depends on the function F

Security

Thm: If **F: K×X⟶Y** is a secure PRF and $1/|Y|$ is negligible (i.e. $|Y|$ is large) then I_F is a secure MAC.

In particular, for every eff. MAC adversary A attacking I_F there exists an eff. PRF adversary B attacking F s.t.:

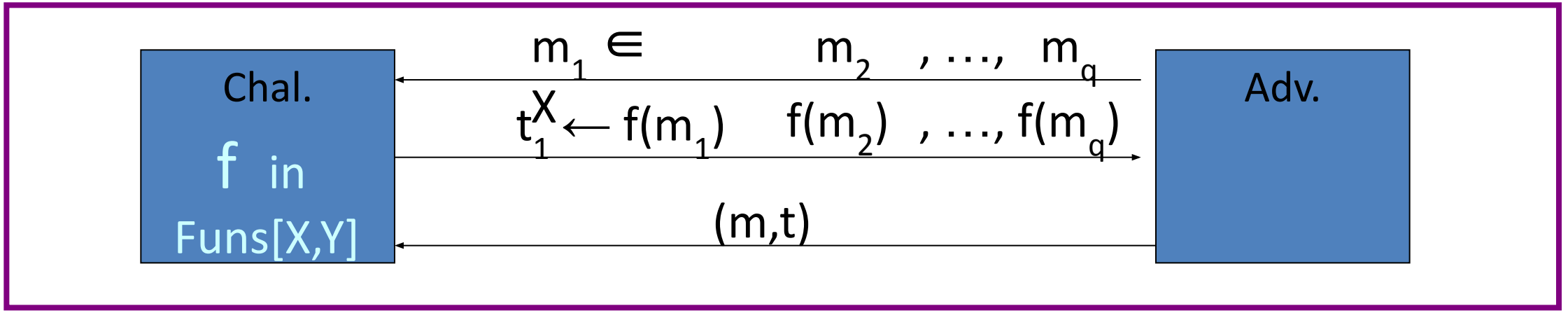
$$Adv_{MAC}[A, I_F] \leq Adv_{PRF}[B, F] + 1/|Y|$$

$\Rightarrow$ I_F is secure as long as $|Y|$ is large, say $|Y| = 2^{128}$.

Proof Sketch

Suppose $f: X \rightarrow Y$ is a truly random function

Then MAC adversary A must win the following game:



A wins if $t = f(m)$ and $m \notin \{ m_1, \ldots, m_q \}$

$\Rightarrow$ Pr[A wins] = 1/|Y|

same must hold for F(k,x)

Examples

- AES: a MAC for 16-byte messages.
- Main question: how to convert Small-MAC into a Big-MAC ?
- Two main constructions used in practice:
  - **CBC-MAC** (banking – ANSI X9.9, X9.19, FIPS 186-3)
  - **HMAC** (Internet protocols: SSL, IPsec, SSH, …)
- Both convert a small-PRF into a big-PRF.

Truncating MACs based on PRFs

Easy lemma: suppose $F: K \times X \longrightarrow \{0,1\}^n$ is a secure PRF.

Then so is $F_t(k,m) = F(k,m)[1 \ldots t]$ for all $1 \le t \le n$

$\Rightarrow$ if (S,V) is a MAC is based on a secure PRF outputting n-bit tags

the truncated MAC outputting w bits is secure

… as long as $1/2^w$ is still negligible (say $w \ge 64$)

CBC-MAC and NMAC

MACs and PRFs

Recall: secure PRF **F** ⇒ secure MAC, as long as |Y| is large

S(k, m) = F(k, m)

Our goal:

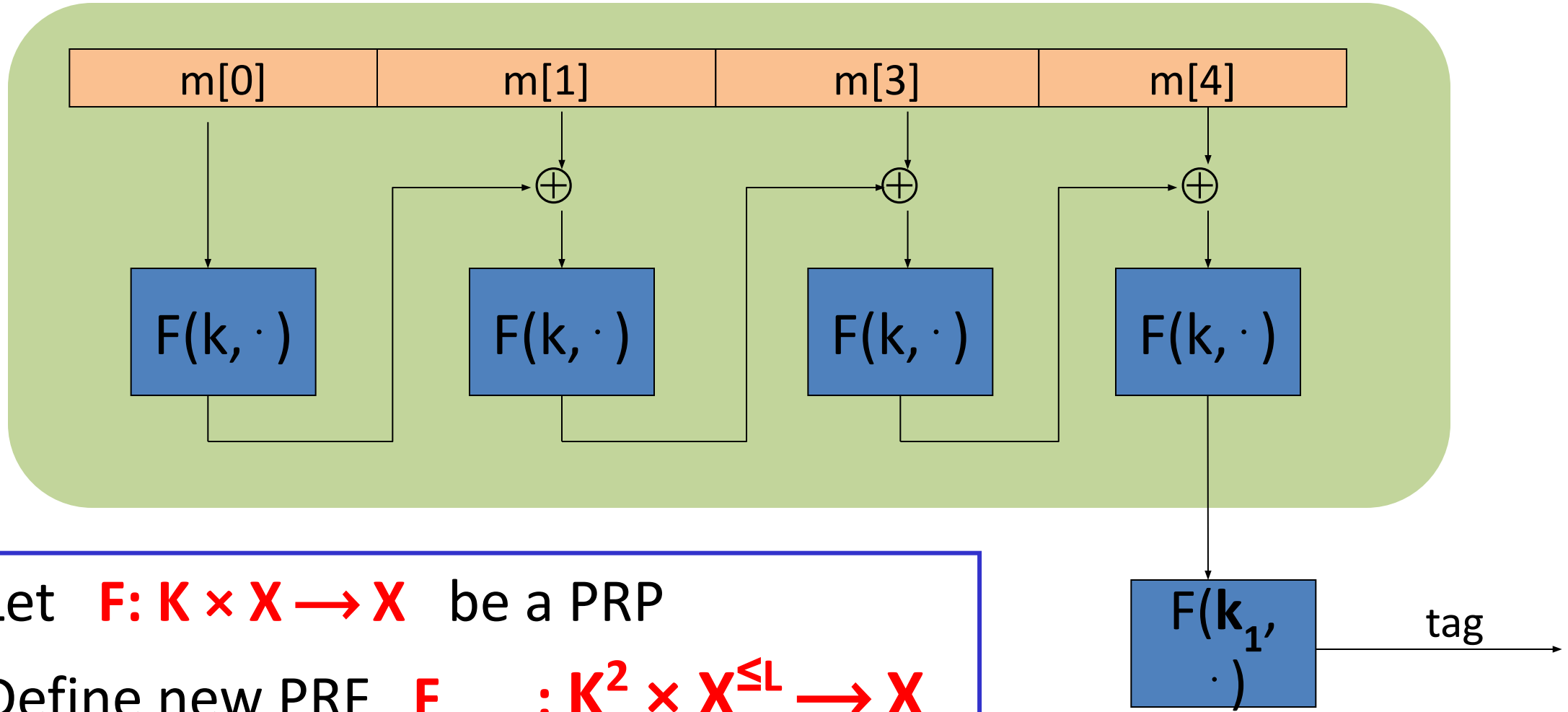
given a PRF for short messages (AES)

construct a PRF for long messages

From here on let $X = \{0,1\}^n$ (e.g. n=128)

Construction 1: encrypted CBC-MAC

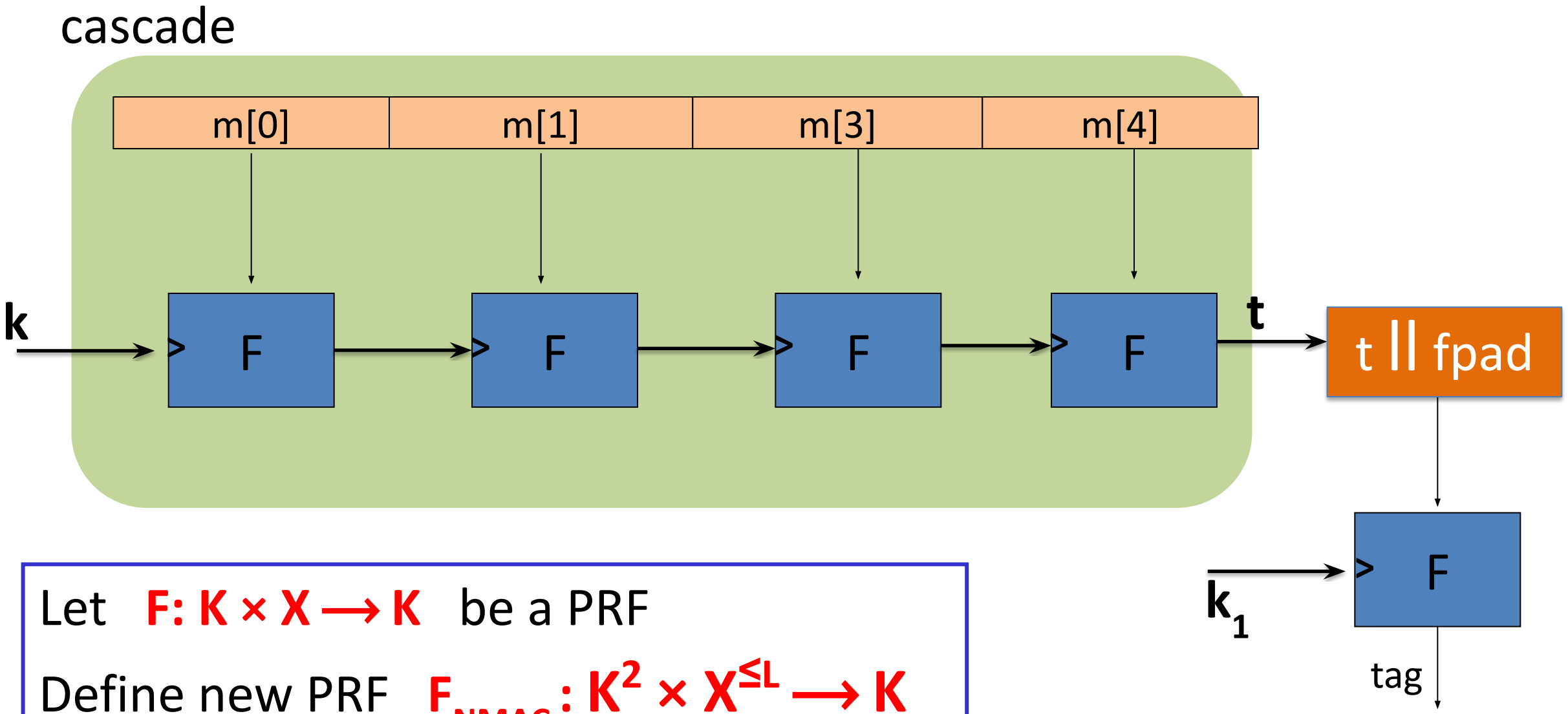
raw CBC



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

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

Construction 2: NMAC (nested MAC)



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

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

Why the last encryption step in ECBC-MAC and NMAC?

NMAC: suppose we define a MAC $I = (S,V)$ where

$S(k,m) = \text{cascade}(k, m)$

- This MAC is secure
- This MAC can be forged without any chosen msg queries
- This MAC can be forged with one chosen msg query
- This MAC can be forged, but only with two msg queries

Why the last encryption step in ECBC-MAC?

Suppose we define a MAC $I_{RAW} = (S,V)$ where

$S(k,m) = \text{rawCBC}(k,m)$

Then I_{RAW} is easily broken using a 1-chosen msg attack.

Adversary works as follows:

- Choose an arbitrary one-block message $m \in X$
- Request tag for m. Get $t = F(k,m)$
- Output t as MAC forgery for the 2-block message $(m, t \oplus m)$

Indeed: $\text{rawCBC}\big(k, (m, t\oplus m)\big) = F\big(k, F(k,m)\oplus(t\oplus m)\big) = F(k, t\oplus(t\oplus m)) = t$

ECBC-MAC and NMAC analysis

Theorem: For any L>0,

For every eff. q-query PRF adv. A attacking F_{ECBC} or F_{NMAC}

there exists an eff. adversary B s.t.:

$$Adv_{PRF}[A, F_{ECBC}] \le Adv_{PRP}[B, F] + 2\, q^2 / |X|$$

$$Adv_{PRF}[A, F_{NMAC}] \le q \cdot L \cdot Adv_{PRF}[B, F] + q^2 / 2|K|$$

CBC-MAC is secure as long as $q << |X|^{1/2}$

NMAC is secure as long as $q << |K|^{1/2}$ (2^{64} for AES-128)

An example

$$Adv_{PRF}[A, F_{ECBC}] \leq Adv_{PRP}[B, F] + \mathbf{2\, q^2 / |X|}$$

q = # messages MAC-ed with k

Suppose we want $Adv_{PRF}[A, F_{ECBC}] \leq 1/2^{32}$ $\Leftarrow$ $q^2 / |X| < 1/ 2^{32}$

- AES: $|X| = 2^{128} \Rightarrow q < 2^{48}$

  So, after 2^{48} messages must, must change key

- 3DES: $|X| = 2^{64} \Rightarrow q < 2^{16}$

The security bounds are tight: an attack

After signing $|X|^{1/2}$ messages with ECBC-MAC or

$|K|^{1/2}$ messages with NMAC

the MACs become insecure

Suppose the underlying PRF F is a PRP (e.g. AES)

- Then both PRFs (ECBC and NMAC) have the following extension property:

$$\forall x,y,w: \quad F_{BIG}(k, x) = F_{BIG}(k, y) \quad \Rightarrow \quad F_{BIG}(k, \mathbf{xllw}) = F_{BIG}(k, \mathbf{yllw})$$

The security bounds are tight: an attack

Let $F_{BIG}: \mathbf{K \times X \longrightarrow Y}$ be a PRF that has the extension property

$$F_{BIG}(k, x) = F_{BIG}(k, y) \quad \Rightarrow \quad F_{BIG}(k, \mathbf{xllw}) = F_{BIG}(k, \mathbf{yllw})$$

Generic attack on the derived MAC:

step 1: issue $|Y|^{1/2}$ message queries for rand. messages in X.

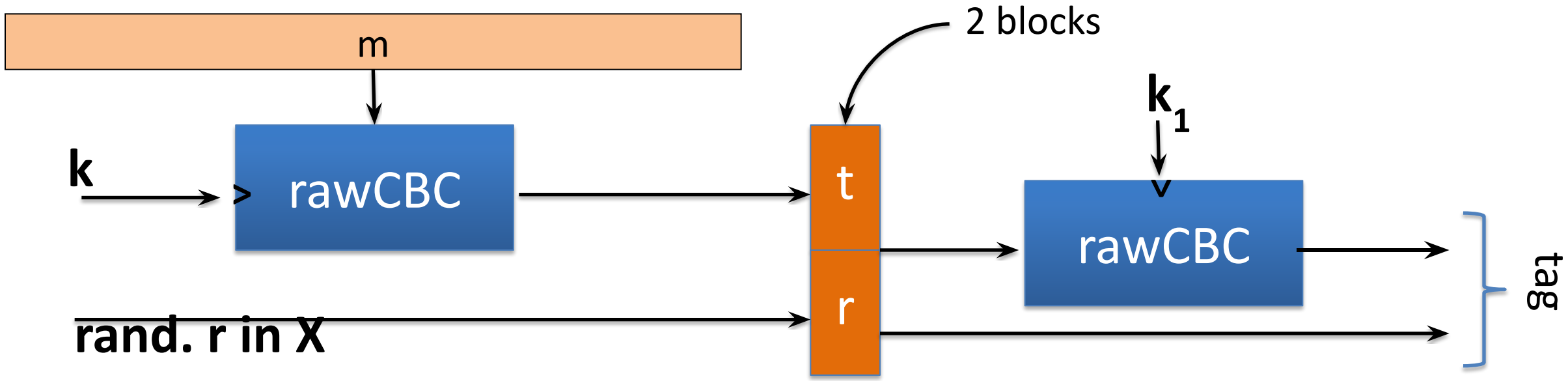
obtain (m_i, t_i) for $i = 1, \ldots, |Y|^{1/2}$

step 2: find a collision $t_u = t_v$ for $u \neq v$ (one exists w.h.p by b-day paradox)

step 3: choose some w and query for $t := F_{BIG}(k, \mathbf{m_u llw})$

step 4: output forgery $\mathbf{(m_v llw, t)}$. Indeed $t := F_{BIG}(k, \mathbf{m_v llw})$

Better security: a rand. construction



Let **F: K × X ⟶ X** be a PRF. Result: MAC with tags in X^2.

Security: $Adv_{MAC}[A, I_{RCBC}] \le Adv_{PRP}[B, F] \cdot (1 + 2\,q^2 / |X|)$

⇒ For 3DES: can sign **$q=2^{32}$** msgs with one key

Comparison

ECBC-MAC is commonly used as an AES-based MAC

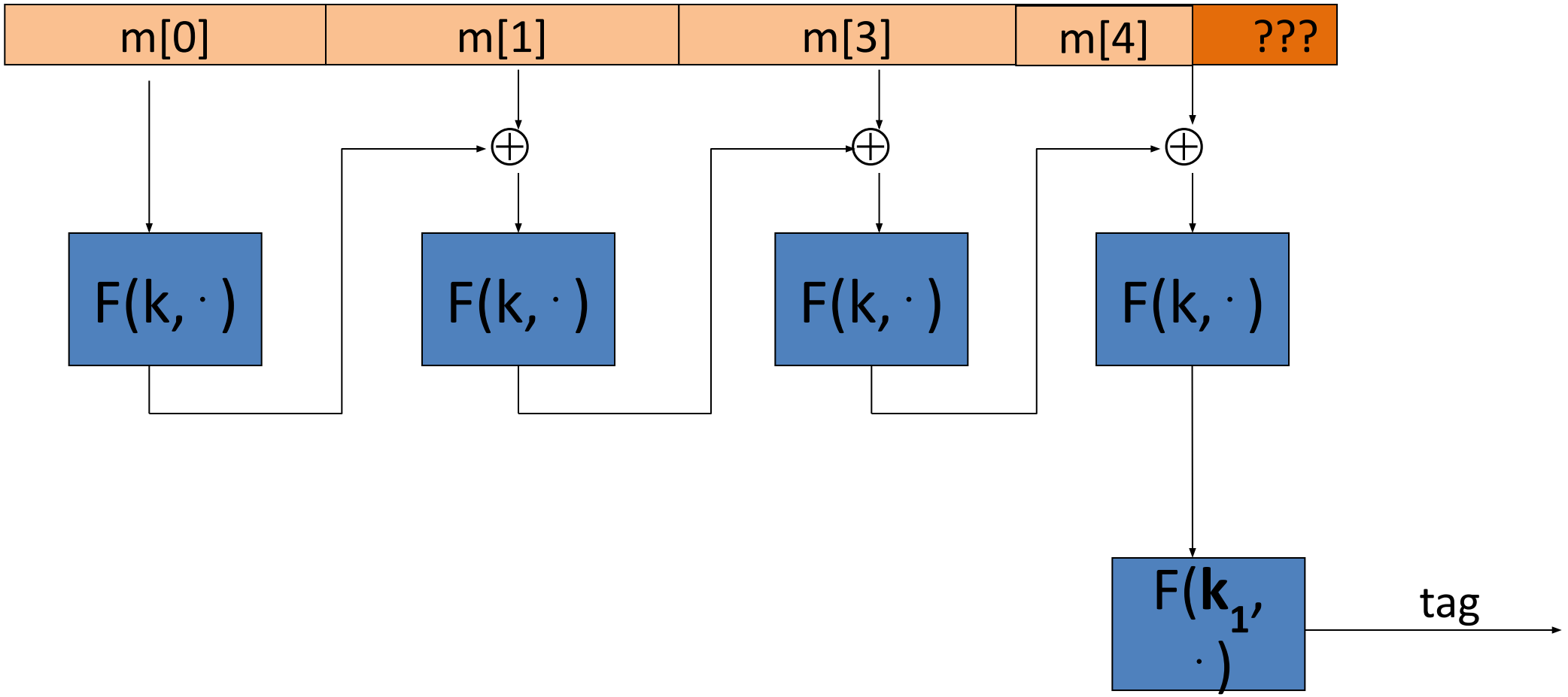
- CCM encryption mode (used in 802.11i)
- NIST standard called CMAC

NMAC not usually used with AES or 3DES

- Main reason: need to change AES key on every block
  requires re-computing AES key expansion
- But NMAC is the basis for a popular MAC called HMAC (next)

What about padding?

What if msg. len. is not multiple of block-size?



CBC MAC padding

Bad idea: pad m with 0's

Is the resulting MAC secure?

- Yes, the MAC is secure
- It depends on the underlying MAC
- No, given tag on msg **m** attacker obtains tag on **mll0**
-

Problem: pad(m) = pad(mll0)

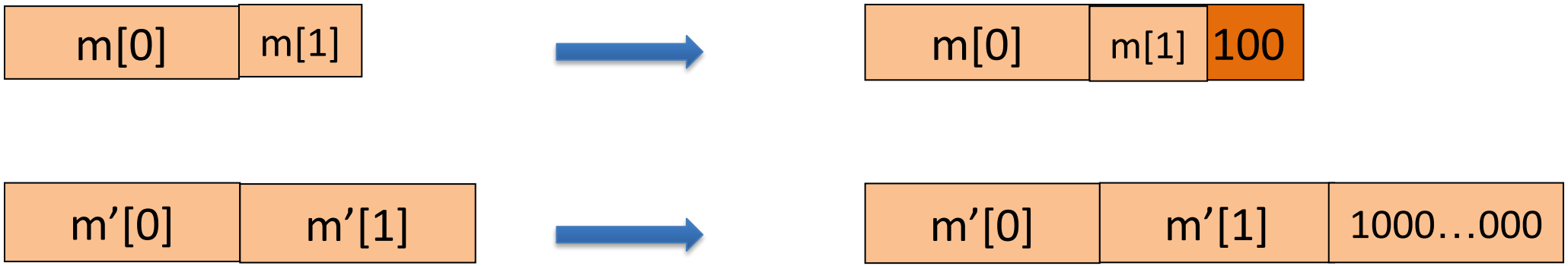
CBC MAC padding

For security, padding must be invertible !

$$\text{len}(m_0) \neq \text{len}(m_1) \quad \Rightarrow \quad \text{pad}(m_0) \neq \text{pad}(m_1)$$

ISO: pad with "1000…00". Add new dummy block if needed.

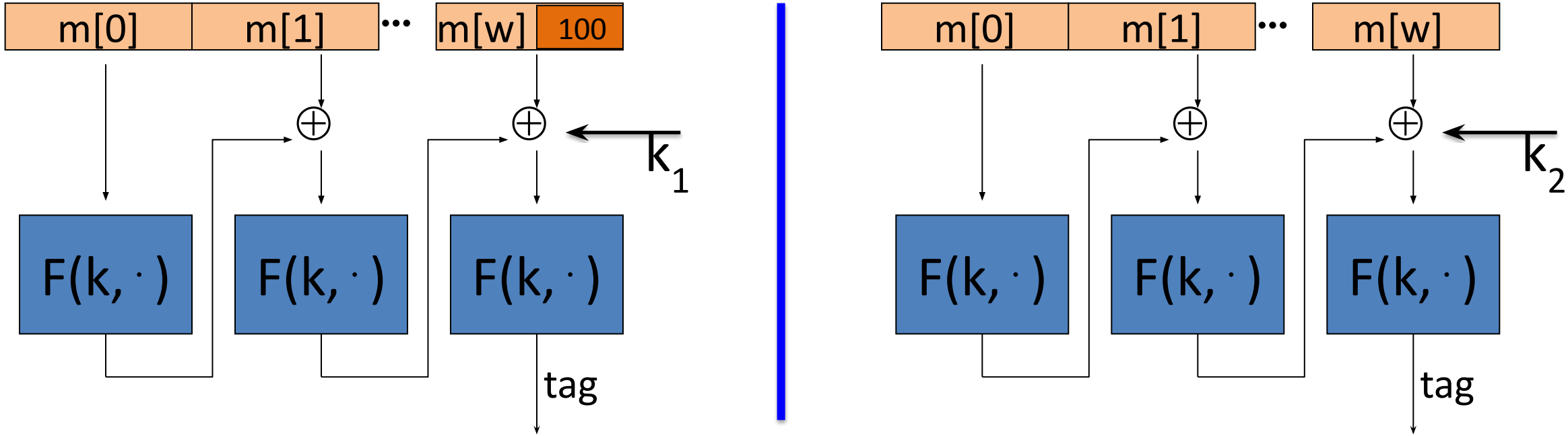
– The "1" indicates beginning of pad.



CMAC (NIST standard)

Variant of CBC-MAC where key = (k, k_1, k_2)

- No final encryption step (extension attack thwarted by last keyed xor)
- No dummy block (ambiguity resolved by use of k_1 or k_2)



More MACs - More Fun!

- PMAC - parallel MAC computation!
- One-time MAC / Many-time MAC
- Carter-Wegman MAC
- … and many more

but we still didn't talk about the extremely common HMAC (Hash MAC)

Hashes and Resistance

Cryptographic Hash Functions

A Cryptographic Hash Function (CHF) is an algorithm that maps an arbitrary binary string to a string of n bits. $H : \{0, 1\}^* \rightarrow \{0,1\}^n$

- Message space much larger than output space

  $H: M \rightarrow T, |M| >> |T|$

- Given the output, we want the input to remain secret and also make it hard for other inputs to get the same output (collision).
- Applications: everywhere (from storing passwords,

Hash Function Properties

Let H: M -> T, |M| >> |T|

- Pre-image resistance. H is pre-image resistant if given a hash value h, it should be difficult to find any message m such that H(m) = h. In other words, P[H(random m) = h] = 1/|T|.
- Second pre-image resistance (weak collision resistance). H is second-preimage resistant if given a message m_1, it should be difficult to find a different m_2 such that $H(m_1) = H(m_2)$.
- (Strong) Collision resistance. H is collision resistant if it is difficult to find any two different messages m_1 and m_2 such that $H(m_1) = H(m_2)$.

Collision Resistance => Second-preimage Resistance

Second-preimage Resistance => Preimage Resistance?

*only true under certain conditions (|M| >> |T|)

Collision Resistance Definition

Let $H: M \rightarrow T$ be a hash function ($|M| >> |T|$)

A **collision** for H is a pair $m_0, m_1 \in M$ such that:

$$H(m_0) = H(m_1) \quad \text{and} \quad m_0 \neq m_1$$

A function H is **collision resistant** if for all (explicit) "eff" algs. A:

$$\mathbf{Adv_{CR}[A,H] = Pr[\ A \text{ outputs collision for } H]}$$

is "neg".

Example: SHA-256 (outputs 256 bits)

Ευχαριστώ και καλή μέρα εύχομαι!

Keep hacking!