Message Authentication Codes (MACs) and Hashes

David Brumley
dbrumley@cmu.edu
Carnegie Mellon University

Credits:
Many slides from Dan Boneh's June 2012 Coursera crypto class, which is awesome!

Recap so far

- Information theoretically secure encryption: ciphertext reveals nothing about the plaintext

- Secure PRNG: Given first *k* output bits, adversary should do not better than guessing bit *k+1*
  - Principle: next bit is secure, not just "random looking" output

- Secure PRF: Adversary can't tell the difference between real random function and PRF
  - Principle: computationally indistinguishable functions

- Semantic security (computationally secure encryption): Adversary picks m_0,m_1, receives encryption of one of them, can't do better than guessing on which messages was encrypted.
  - Principle: ciphertext reveals no information about plaintext
  - Security is not just about keeping key private

Message Integrity

Goal: *integrity* (not secrecy)

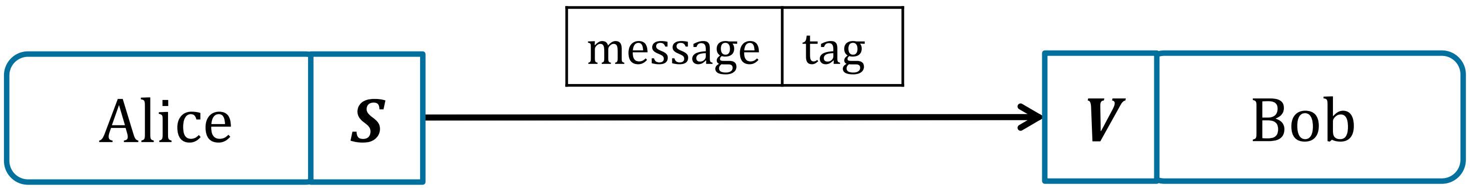
Examples:

- Protecting binaries on disk.
- Protecting banner ads on web pages

Security Principles:

- Integrity means no one can forge a signature

CRC



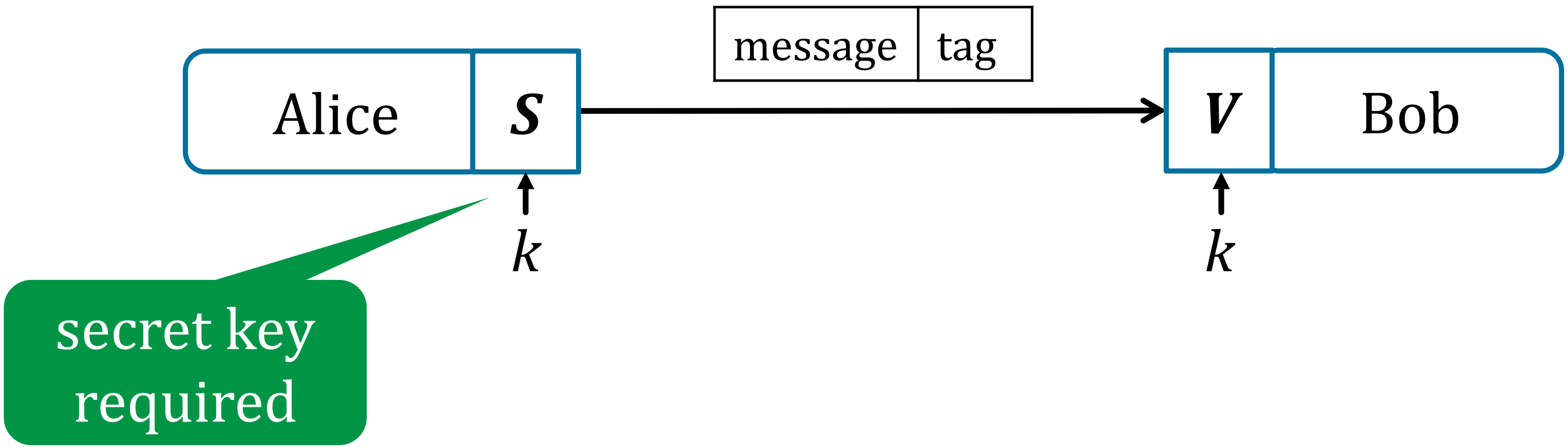
Generate tag:
tag $\leftarrow$ CRC(m)

Verify tag:
CRC(m, tag) ?= ‘yes’

Is this Secure?

- No! Attacker can easily modify message m and re-compute CRC.
- CRC designed to detect *random errors*, not malicious attacks.

Message Authentication Codes (MAC)



Defn: A *<u>Message Authentication Code (MAC)</u>* MAC = (S,V) defined over (K,M,T) is a pair of algorithms:

- S(k,m) outputs t in T
- V(k,m,t) outputs \`yes' or \`no'
- V(k, S(k,m), t) = 'yes' (consistency req.)

Example

Authorized Stock Ticker Publisher

1. k = KeyGen(l)
2. For each price update:
   t = S(stock||price,k)

Publish to the world

Adversary

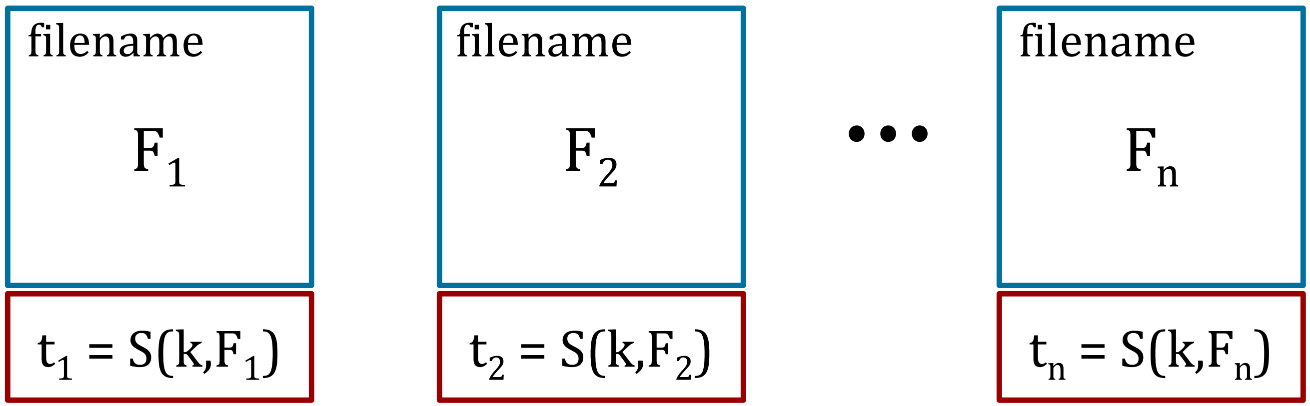
t = A(stock||price up)

A secure MAC should prevent this

e.g., to cause a buying frenzy

Example: Tripwire

At install time, generate a MAC on all files:

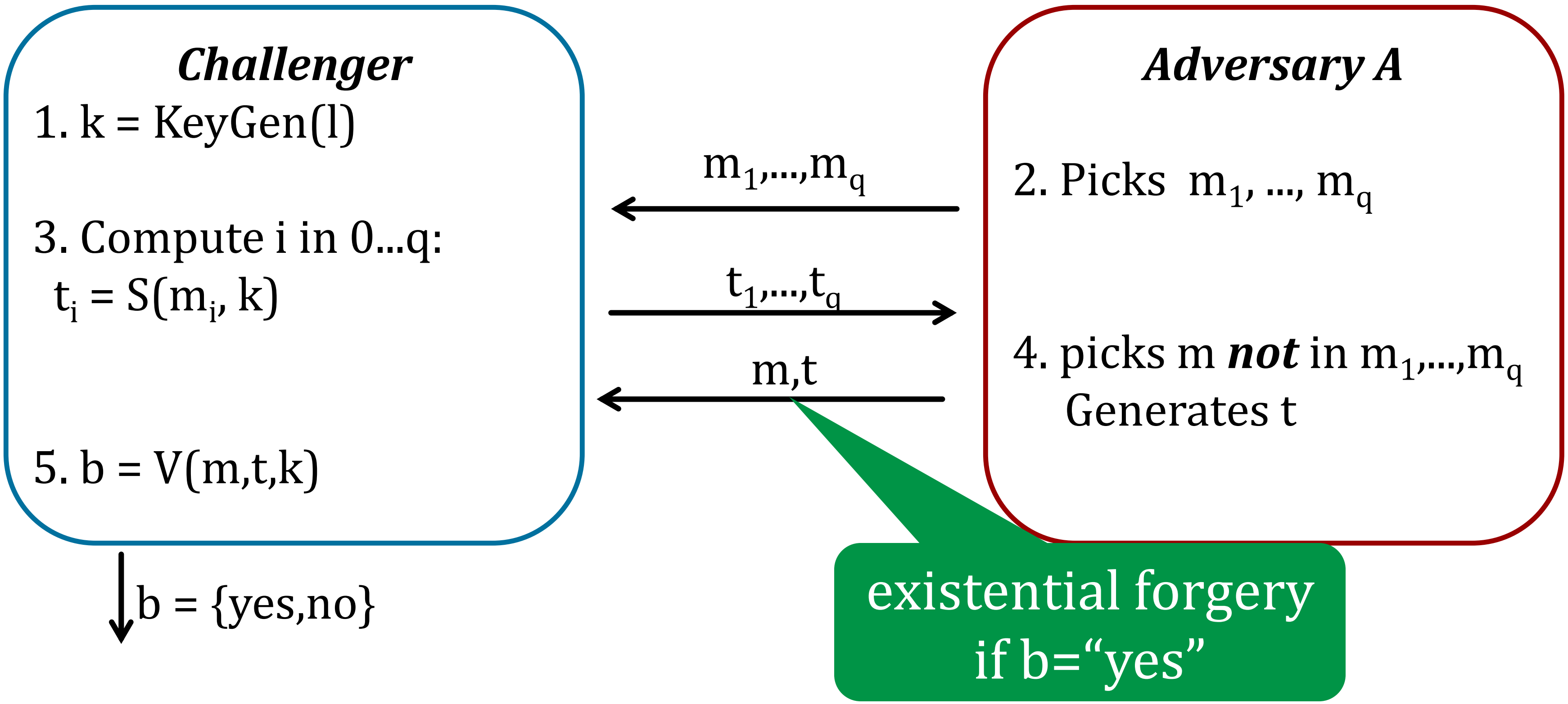


Later a virus infects system and modifies system files

User reboots into clean OS and supplies his password

- Then: secure MAC $\Rightarrow$ all modified files will be detected

Secure MAC Game



Security goal: ***A*** cannot produce a valid tag on a message

– Even if the message is gibberish

Secure MAC Game

Challenger

1. k = KeyGen(l)

3. Compute i in 0...q:
   $t_i = S(m_i, k)$

5. b = V(m,t,k)

$\downarrow$ b = {yes,no}

$\xleftarrow{m_1,\ldots,m_q}$

$\xrightarrow{t_1,\ldots,t_q}$

$\xleftarrow{m,t}$

Adversary A

2. Picks $m_1, \ldots, m_q$

4. picks m ***not*** in $m_1,\ldots,m_q$
   Generates t

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

$\mathrm{Adv}_{MAC}[A,I] = \Pr[\text{Chal. outputs 1}] < \varepsilon$

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

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

$$S(k, m_0) = S(k, m_1) \quad \text{for } ½ \text{ of the keys } k \text{ in } K$$

Can this MAC be secure?

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

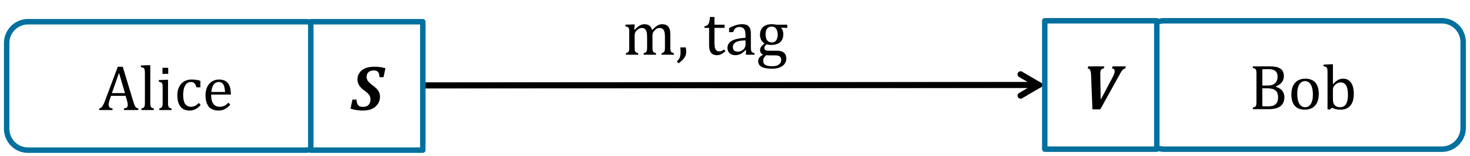
1. A sends m_0, receives (m_0, t_0)
2. A wins with (m_1, t_0)
3. $\text{Adv}[A,I] = ½$ since prob. of key is ½.

MACs from PRFs

Secure PRF implies secure MAC

For a PRF $F: K \times X \longrightarrow Y$, define a MAC $I_F = (S,V)$ as:

- $S(k,m) = F(k,m)$
- $V(k,m,t)$: if $t = F(k,m)$, output 'yes' else 'no'



$tag \leftarrow F(k,m)$

accept msg if

$tag = F(k,m)$

Attacker who knows
$F(k,m_1), F(k,m_2), \ldots, F(k, m_q)$
has no better than $1/|Y|$ chance of
finding valid tag for new m

Security

<u>Thm</u>: If $F: K \times X \longrightarrow Y$ is a secure PRF and $1/|Y|$ is negligible (i.e., $|Y|$ is large), then I_F is a secure MAC.

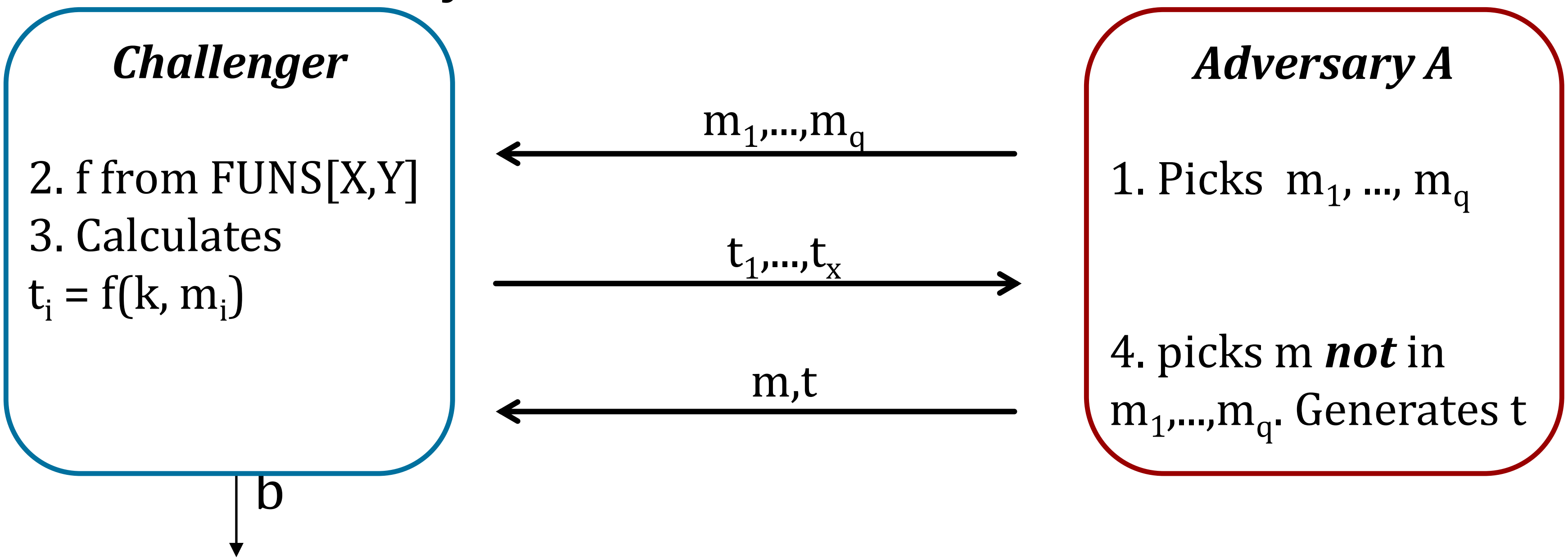
In particular, for every eff. MAC adversary **A** attacking I_F, there exists an eff. PRF adversary **B** attacking F s.t.:

$$\mathrm{Adv}_{\mathrm{MAC}}[\mathbf{A}, I_F] \leq \mathrm{Adv}_{\mathrm{PRF}}[\mathbf{B}, F] + 1/|Y|$$

A can't do better than brute forcing

Proof Sketch

Let f be a truly random function



A wins iff $t=f(k,m)$ and m not in $m_1,\ldots,m_q$

PR[***A*** wins] = Pr[***A*** guesses value of rand. function on new pt]

$= 1/|Y|$

Question

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

Is the derived MAC I_F a *practically* secure MAC system?

1. Yes, the MAC is secure because the PRF is secure
2. No tags are too short: guessing tags isn't hard
3. It depends on the function F

Adv[A,F] = 1/1024
(we need |Y| to be large)

Secure PRF *implies* secure MAC

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

Assuming output domain Y is large

So AES is already a secure MAC....

... but AES is only defined on 16-byte messages

Building Secure MACs

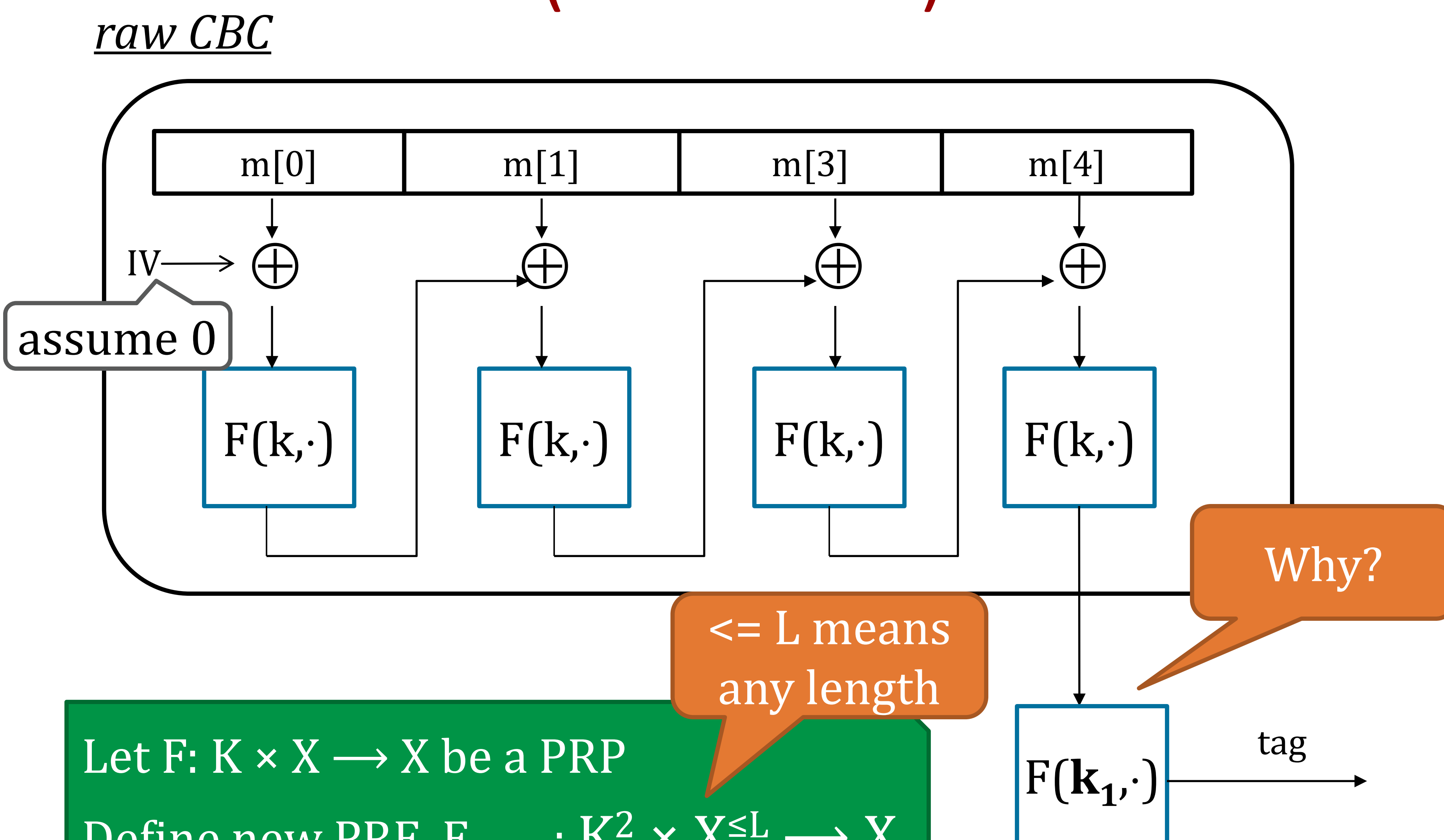
Given: a PRF for shorter messages (e.g., 16 bytes)

Goal: build a MAC for longer messages (e.g., gigabytes)

Construction examples:

- CBC-MAC: Turn small PRF into big PRF
- HMAC: Build from collision resistance

Construction 1: Encrypted CBC-MAC (ECBC-MAC)



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

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

Attack

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

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

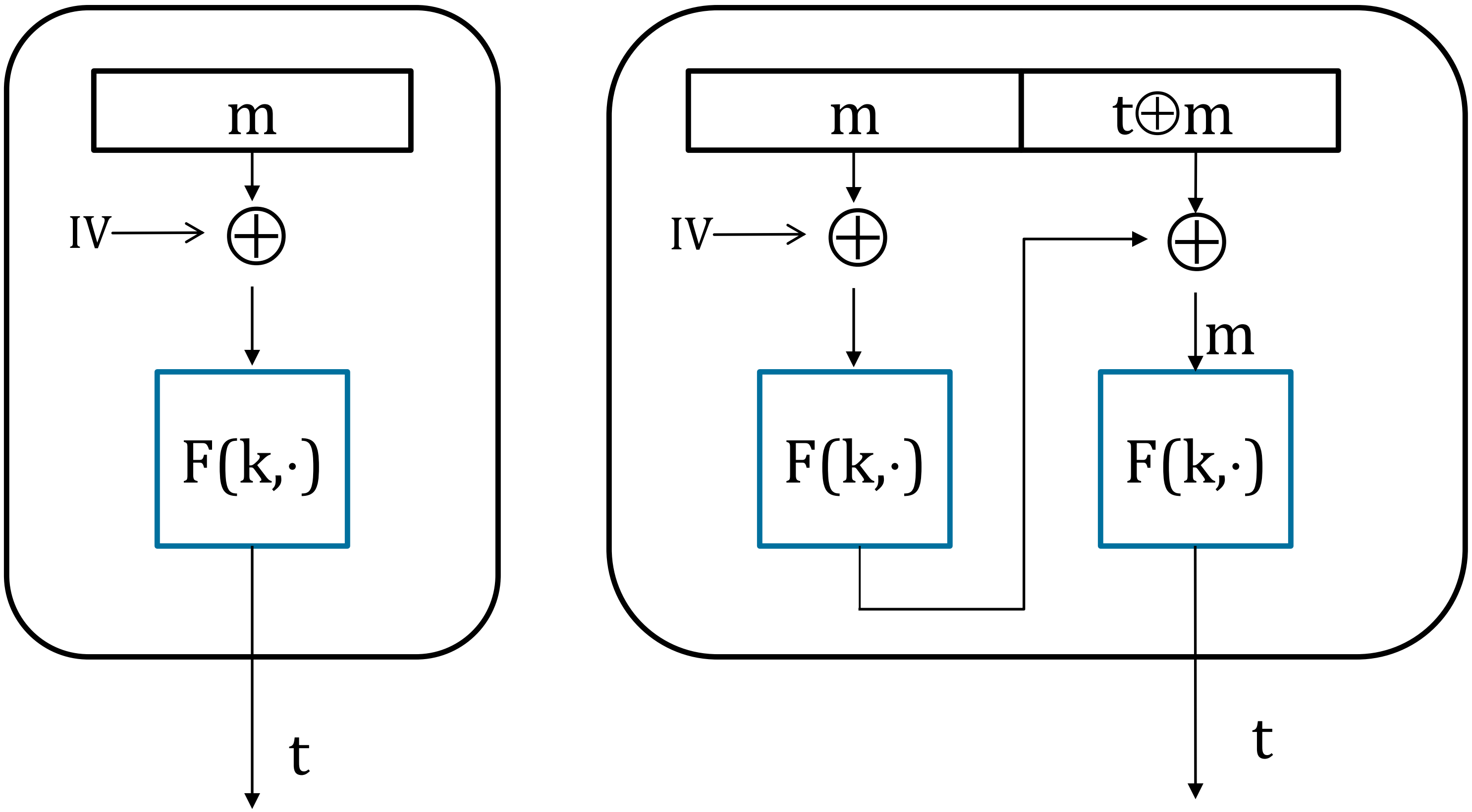
Then I_{RAW} is easily broken using a 1-chosen msg attack.

Adversary works as follows:

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

Attack

Break in 1-chosen message attack



Problem: rawCBC(k, m|| t⊕m)

$= F(k, F(k,m) \oplus (t \oplus m)) = F(k, t \oplus (t \oplus m)) = F(k,m) = t$

ECBC-MAC analysis

<u>Recall</u>: We built ECBC-MAC from a PRP (e.g., block cipher)

F: K x X -> X

<u>Theorem</u>: 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.:

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

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

After signing $|X|^{1/2}$ messages, rekey

Implications

msgs MAC'ed with key

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

Suppose we want $\mathrm{AdvPRF}[A, F_{\mathrm{ECBC}}] \leq 1/2^{32}$

128-32 = 96

$q^2 = 2^{48*2} = 2^{96}$.

- then $(2q^2 /|X|) < 1/2^{32}$
- AES: $|X| = 2^{128} \Rightarrow q < 2^{48}$
- 3DES: $|X| = 2^{64} \Rightarrow q < 2^{16}$

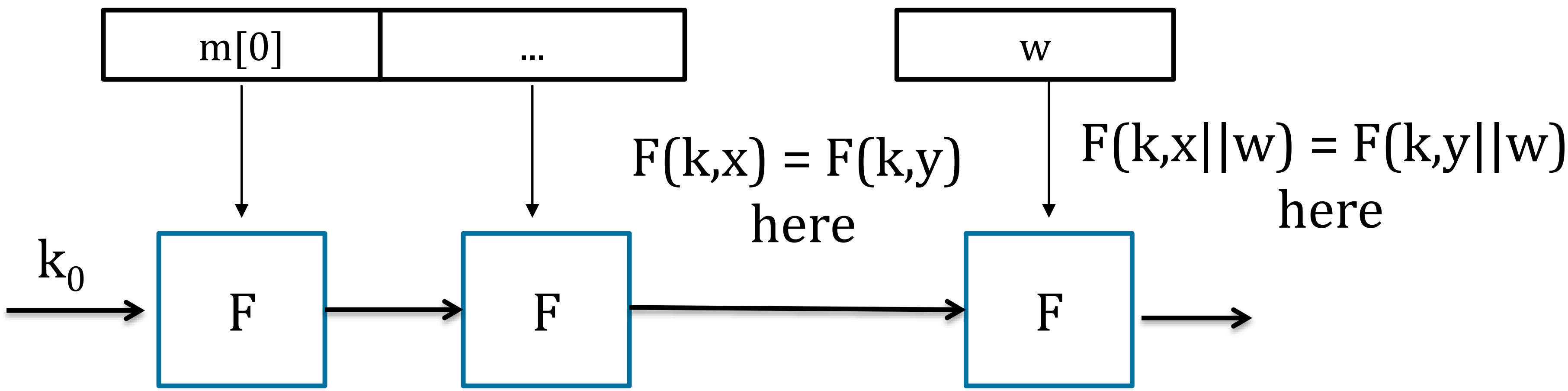
Must change key after 2^{48}, 2^{16} msgs

Extension Attack

Suppose the underlying PRF F is a PRP (e.g., AES).
Let F_{BIG} be ECBC. Then F_{BIG} has the following *extension property*:

$\forall x,y,w$:

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

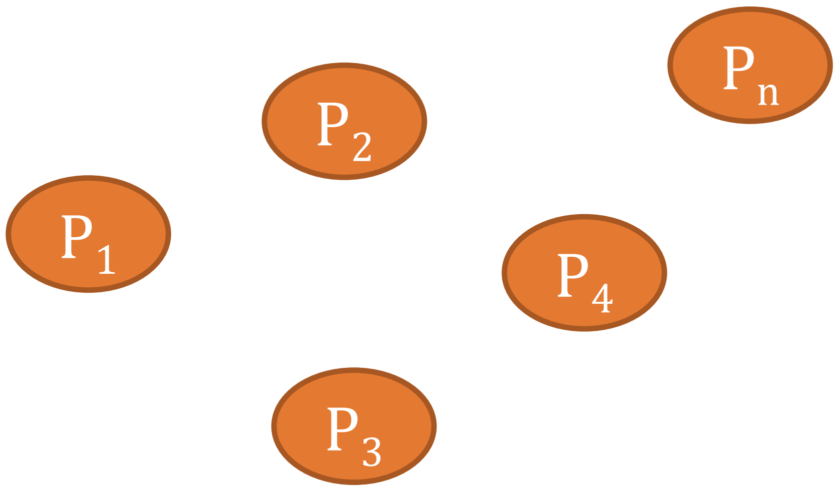


Attacker just needs to find such an x and y

Collisions and the Birthday Paradox

Birthday Paradox

Put n people in a room. What is the probability that 2 of them have the same birthday?



$PR[P_i = P_j] > .5$ with 23 people.
(Think: n^2 different pairs)

Birthday Paradox Rule of Thumb

Given N possibilities, and random samples x_1, …, x_j, $PR[x_i = x_j] \approx 50\%$ when $j = N^{1/2}$

Generic attack on hash functions

Let $H: M \to \{0,1\}^n$ 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?

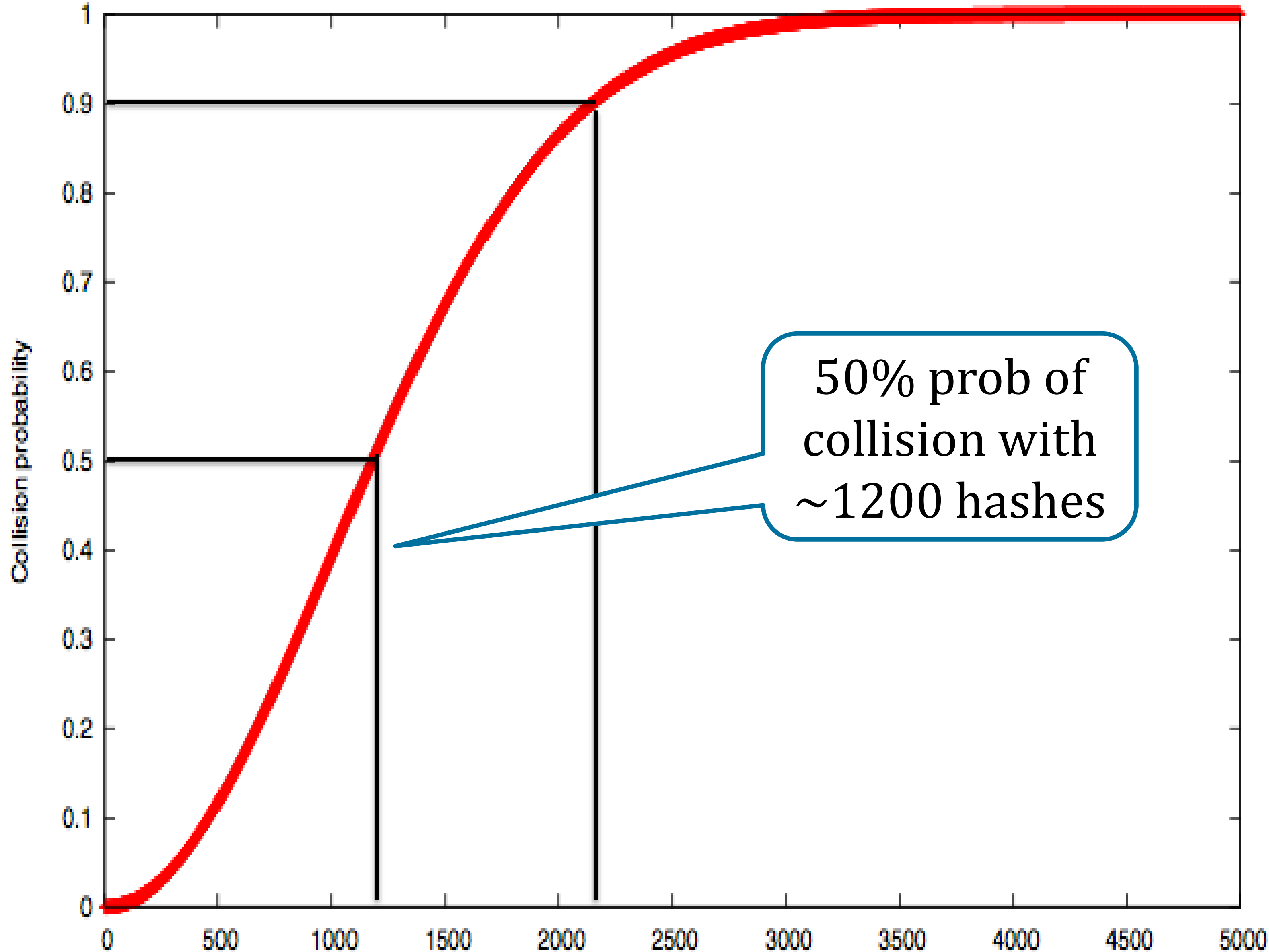
The birthday paradox

Let $r_1, \ldots, r_i \in \{1,\ldots,n\}$ be indep. identically distributed integers.

<u>Thm</u>:
when $\mathbf{i} = 1.2 \times \mathbf{n^{1/2}}$ then $\Pr[\, \exists i \neq j: \; r_i = r_j \,] \geq \frac{1}{2}$

If $H: M \to \{0,1\}^n$, then
$\Pr[\text{collision}] \sim \frac{1}{2}$
with $n^{1/2}$ hashes

B=10^6



samples n

Recall

msgs MAC'ed with key

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

Suppose we want $\mathrm{AdvPRF}[A, F_{\mathrm{ECBC}}] \le 1/2^{32}$

– then $(2q^2 / |X|) < 1/2^{32}$
– AES: $|X| = 2^{128} \Rightarrow q < 2^{47}$
– 3DES: $|X| = 2^{64} \Rightarrow q < 2^{15}$

Must change key after 2^{47}, 2^{15} msgs

Reason: the Birthday Paradox.

Generic attack

Let F_{BIG}: K x M $\rightarrow$ Y be a MAC with the extension property (e.g., CBC-MAC):

$$F_{BIG}(k, x) = F_{BIG}(k, y) \quad \Rightarrow \quad F_{BIG}(k, x||w) = F_{BIG}(k, y||w)$$

1. For i = 1, …, $2^{n/2}$ get $t_i = F(k, m_{i,})$
2. Look for a collision ($t_i = t_j$). (birthday paradox) If not found, got back to step 1.
3. Choose some w and for query $t = F_{BIG}(m_i \,||\, w)$
4. Output forgery ($m_j||w$, t)

Implications

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

Suppose we want $\mathrm{AdvPRF}[A, F_{\mathrm{ECBC}}] \leq 1/2^{32}$

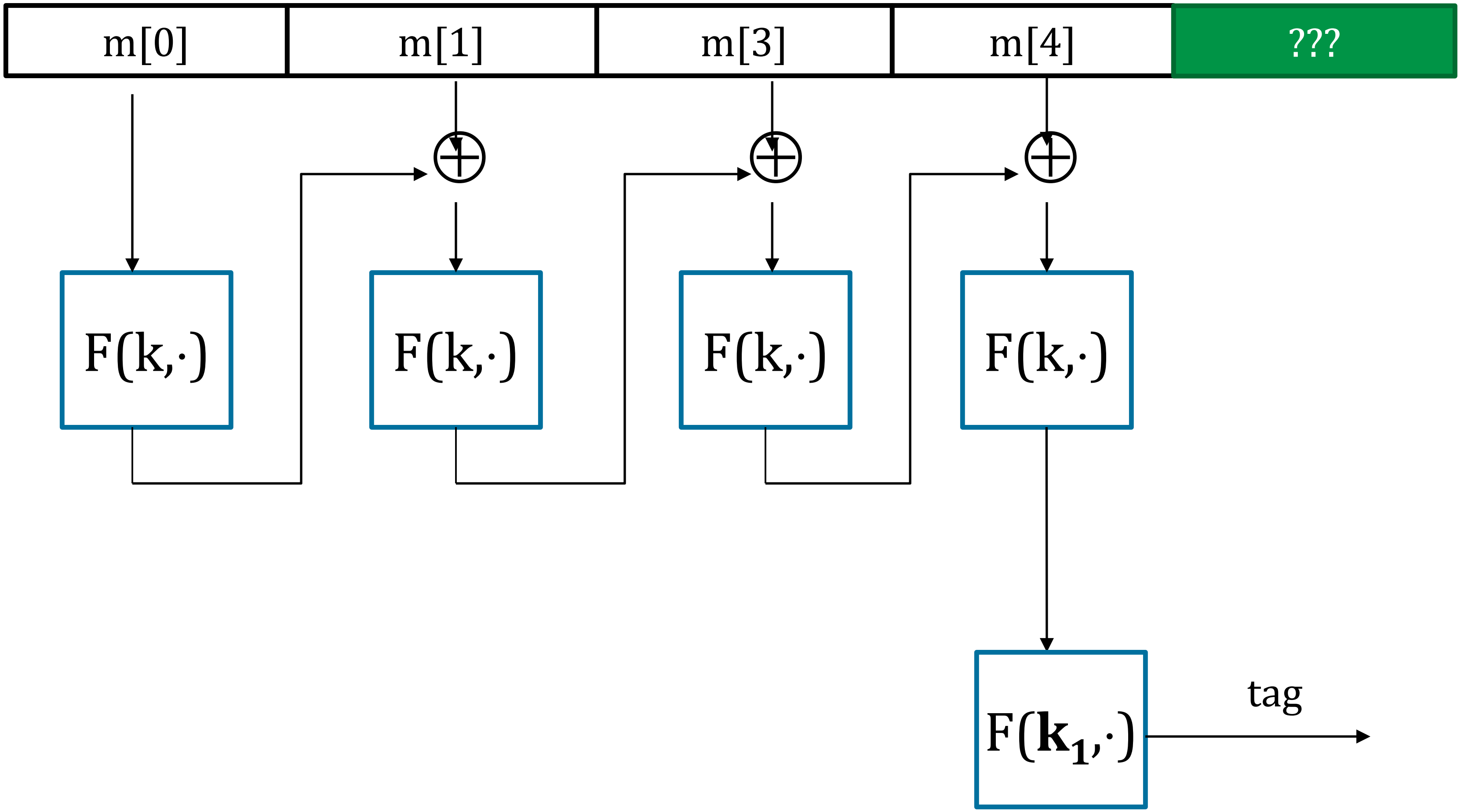
- then $(2q^2 / |X|) < 1/2^{32}$
- AES: $|X| = 2^{128} \Rightarrow q < 2^{47}$
- 3DES: $|X| = 2^{64} \Rightarrow q < 2^{15}$

Need PRF that can quickly change keys.

Padding

What is msg not a multiple of block size?

Recall CBC-MAC



CBC MAC padding

Idea: pad m with 0's

m[0]	m[1]		→	m[0]	m[1]	0000

Is the resulting MAC secure?

$100	00
$10	000

Yes, the MAC is secure

It depends on the underlying MAC

→ No

Problem: given tag on msg m attacker obtains tag on m||0 because pad(m) = pad(m'||0)

CBC MAC padding

For security, padding must be one-to-one (i.e., invertible)!

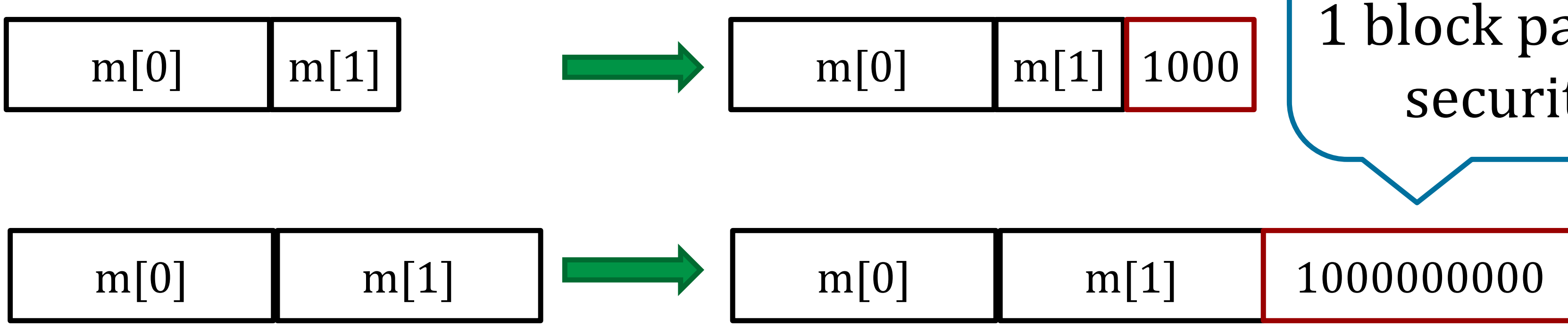
$$m_0 \neq m_1 \quad \Rightarrow \quad pad(m_0) \neq pad(m_1)$$

two distinct messages map to two distinct paddings

<u>ISO</u>: pad with "1000…00". Add new dummy block if needed.

– The "1" indicates beginning of pad.

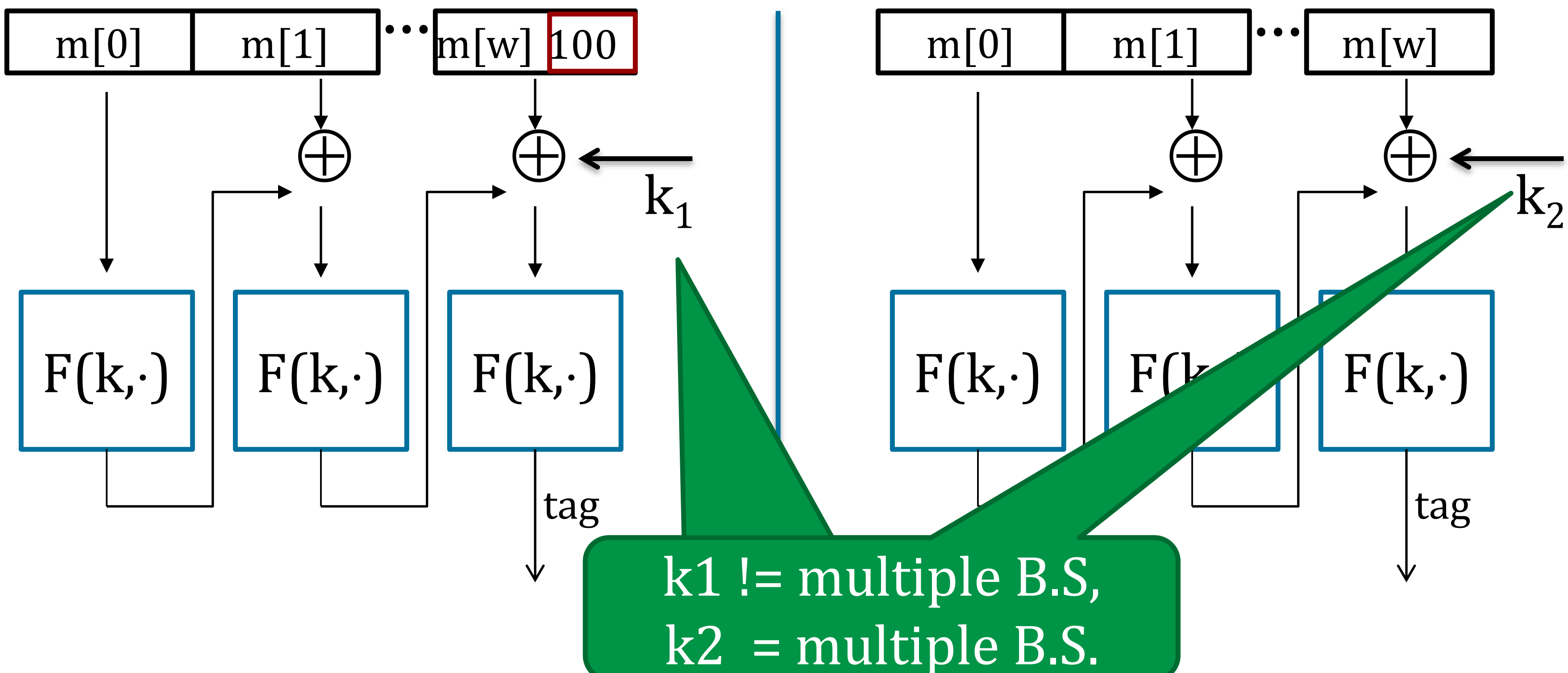
If m is same as block size, add 1 block pad for security



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)



HMAC (Hash-MAC)

Most widely used MAC on the Internet.

… but, we first we need to discuss hash function.

Hash Functions

Collision Resistance

Let $H: X \to Y$ be a hash function ($|X| >> |Y|$)

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

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

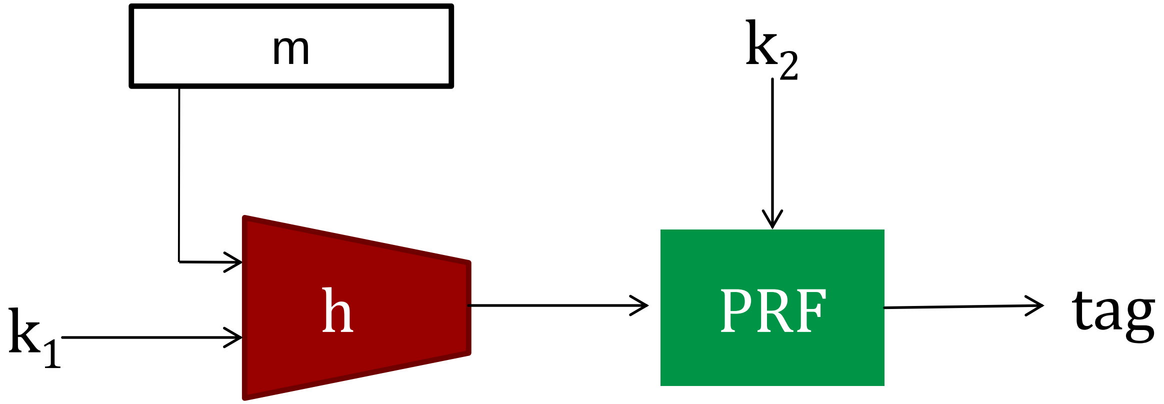
A function H is **collision resistant** if for all (explicit) "eff" algs. A:

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

is "negligible".

Example: SHA-256 (outputs 256 bits)

General Idea



Hash then PRF construction

MACs from Collision Resistance

Let $I = (S,V)$ be a MAC for short messages over (K,M,T) (e.g. AES)

Let $H: X \rightarrow Y$ and $S: K \times Y \rightarrow T$ $\quad (|X| >> |Y|)$

Def: $I^{big} = (S^{big}, V^{big})$ over (K, X^{big}, Y) as:

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

<u>**Thm**</u>: If I is a secure MAC and H is collision resistant, then I^{big} is a secure MAC.

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

MACs from Collision Resistance

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

Collision resistance is necessary for security:

Suppose: adversary can find $m_0 \neq m_1$ s.t. $H(m_0) = H(m_1)$.

Then: **S^{big}** is insecure under a 1-chosen msg attack

step 1: adversary asks for $t \longleftarrow S(k, m_0)$

step 2: output (m_1, t) as forgery

Sample Speeds Crypto++ 5.6.0 [Wei Dai]

AMD Opteron, 2.2 GHz (Linux)

NIST standards

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

* best known collision finder for SHA-1 requires 2^{51} hash evaluations

Collision Resistance and Passwords

Passwords

How do we save passwords on a system?

- Idea 1: Store in cleartext
- Idea 2: Hash

<u>Enrollment:</u> store h(password), where h is collision resistant

<u>Verification:</u> Check h(input) = stored passwd

Is this enough to be secure

Brute Force

Online Brute Force Attack:

input: *hp* = hash(password) to crack

for each i in dictionary file

  if(h(i) == hp)

    output success;

Time Space Tradeoff Attack:

precompute: h(i) for each i in dict file in hash tbl

input: *hp* = hash(password)

check if *hp* is in hash tbl

"rainbow tables"

Salts

Enrollment:

1. compute hp=h(password + salt)
2. store salt || hp

Verification:

1. Look up salt in password file
2. Check h(input||salt) == hp

What is this good for security, given that the salt is public?

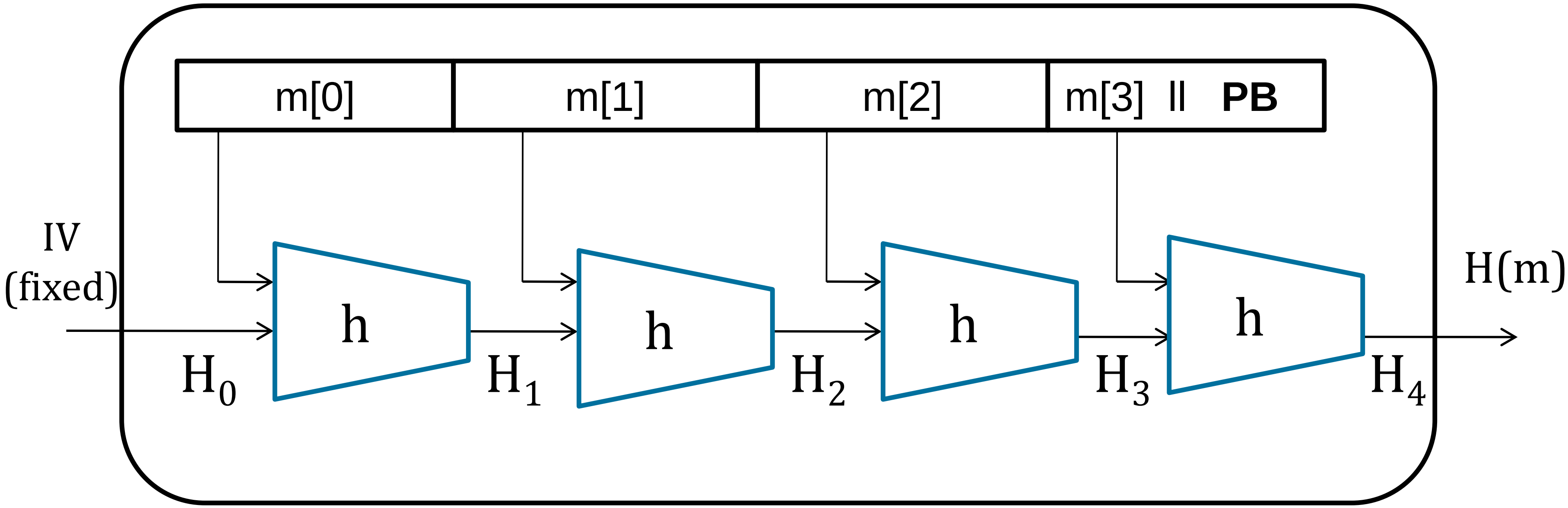
Salt doesn't increase security against online attack, but does make tables much bigger.

Merkle-Damgard

How to construct collision resistant hash functions

http://www.merkle.com/

The Merkle-Damgard iterated construction



Given $\mathbf{h: T \times X \longrightarrow T}$ (compression function)

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

PB: padding block

$$\underbrace{\boxed{1000\ldots0 \;\|\; \text{msg len}}}_{\text{64 bits}}$$

If no space for PB add another block

Security of Merkle-Damgard

<u>Thm</u>: if h is collision resistant then so is H.

Proof Idea:

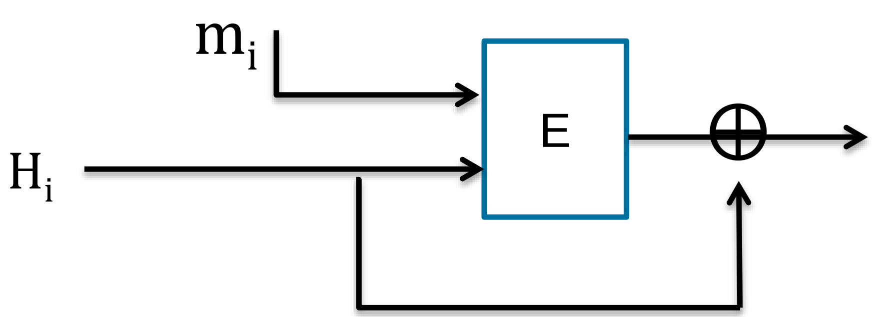
via contrapositive. Collisions on $H \Rightarrow$ collision on h

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

Compr. func. from a block cipher

E: K× $\{0,1\}^n \longrightarrow \{0,1\}^n$ a block cipher.

The **Davies-Meyer** compression function
$h(H, m) = E(m, H) \oplus H$



<u>Thm</u>: Suppose E is an ideal cipher (collection of |K| random perms.).

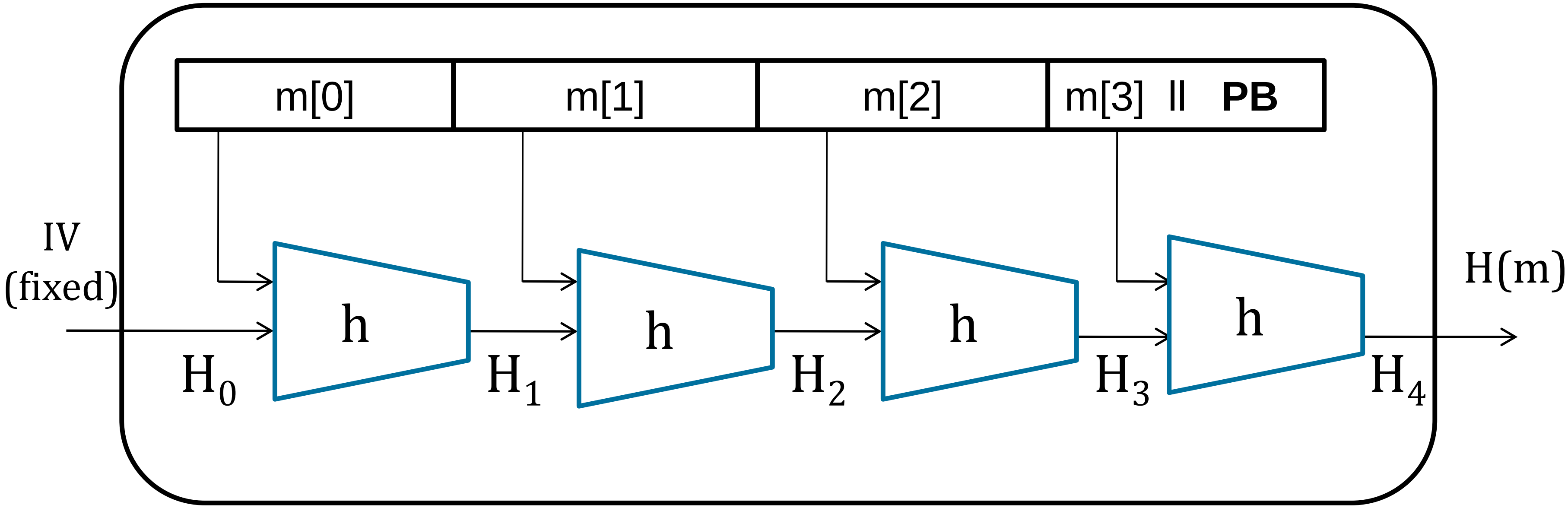
Best possible !!

Finding a collision *h(H,m)=h(H',m')* takes $O(2^{n/2})$ evaluations of (E,D).

Hash MAC (HMAC)

Most widely used approach on the internet, e.g., SSL, SSH, TLS, etc.

Recall Merkel-Damgard



<u>Thm</u>:

h collision resistant implies H collision resistant

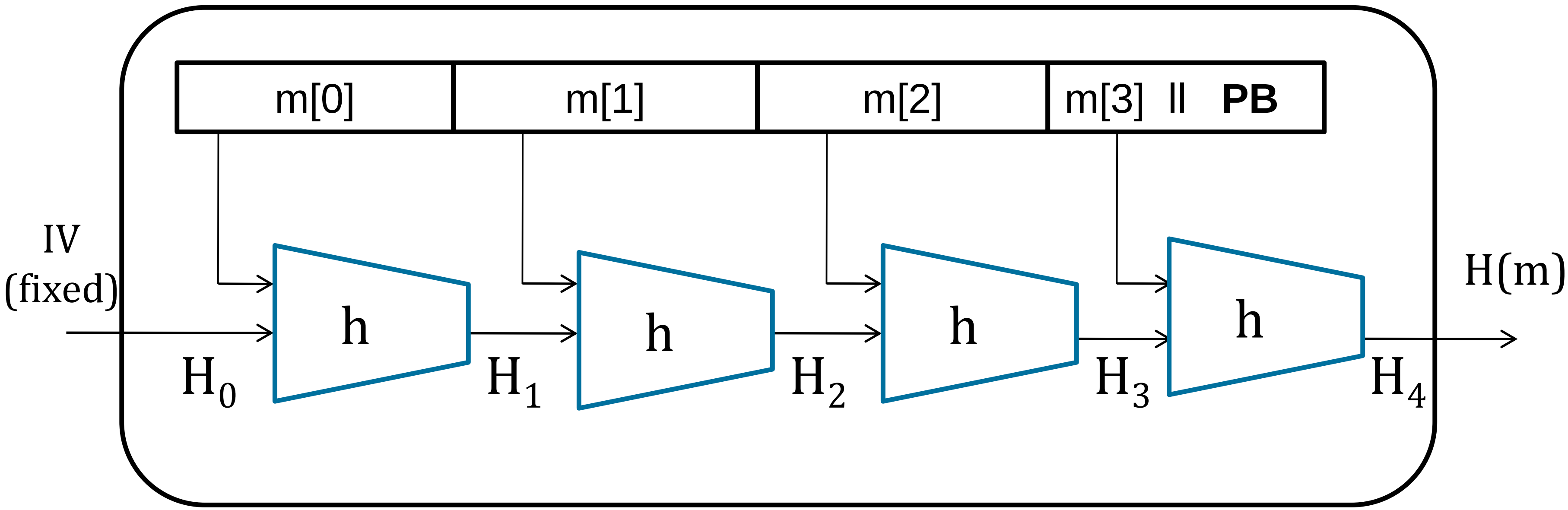
Can we build a MAC out of H?

Attempt 1

Let H: $X^{\leq L} \longrightarrow T$ be a Merkle-Damgard hash, and:

S(k,m) = H(k||m)

is this secure? no! why?



Existential forgery:
H(k||m) = H(k||m||PB||w)
(just one more h)

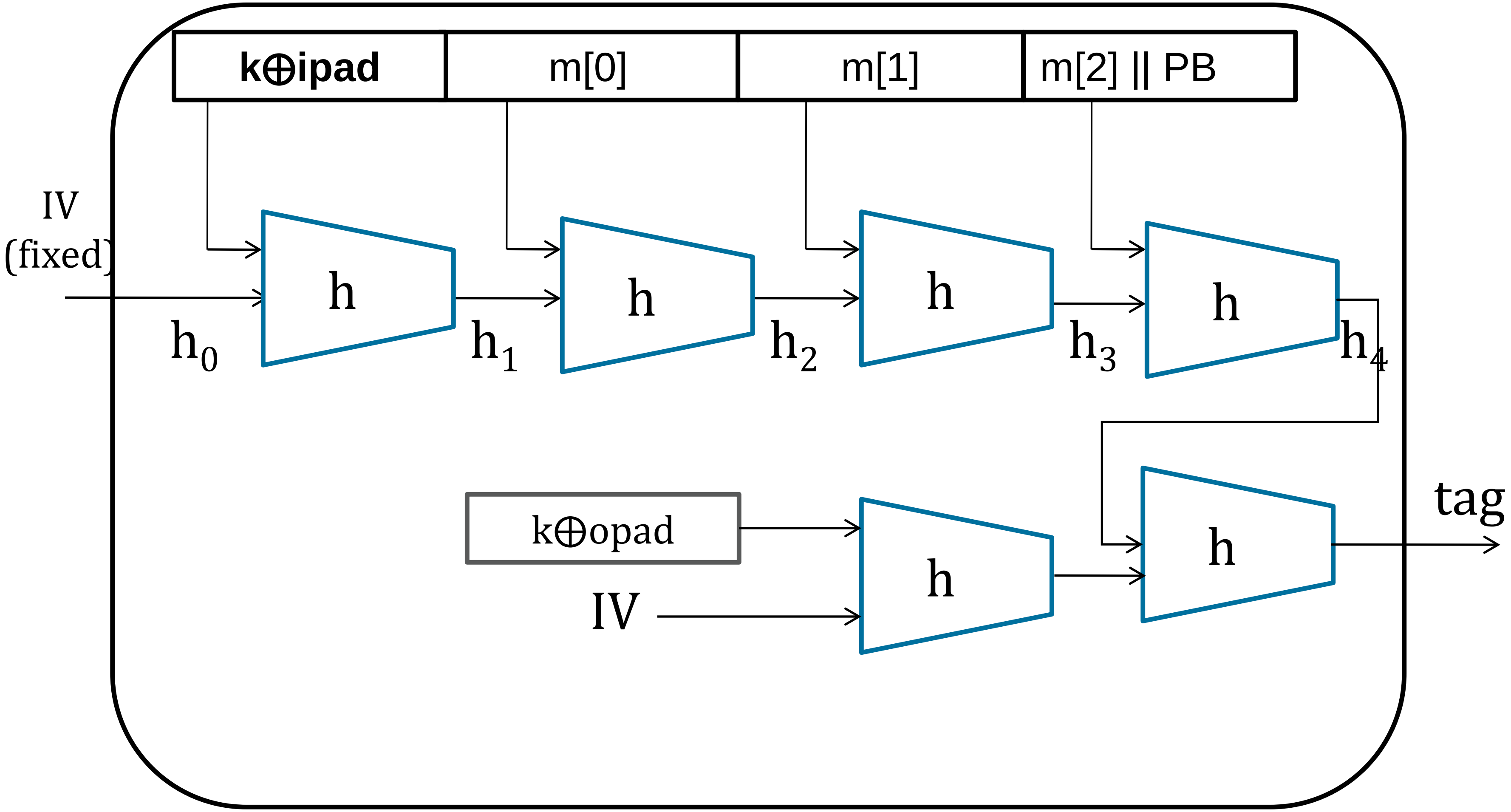
Hash Mac (HMAC)

Build MAC out of a hash

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

- Example: H = SHA-256

HMAC



PB: Padding Block

Recap

- MAC's from PRF
  - NMAC
  - CBC-MAC
  - PMAC
- MAC's from collision resistant hash functions
  - Make CRF with merkle-damgard from PRF
- Attackers goal: existential forgery

Further reading

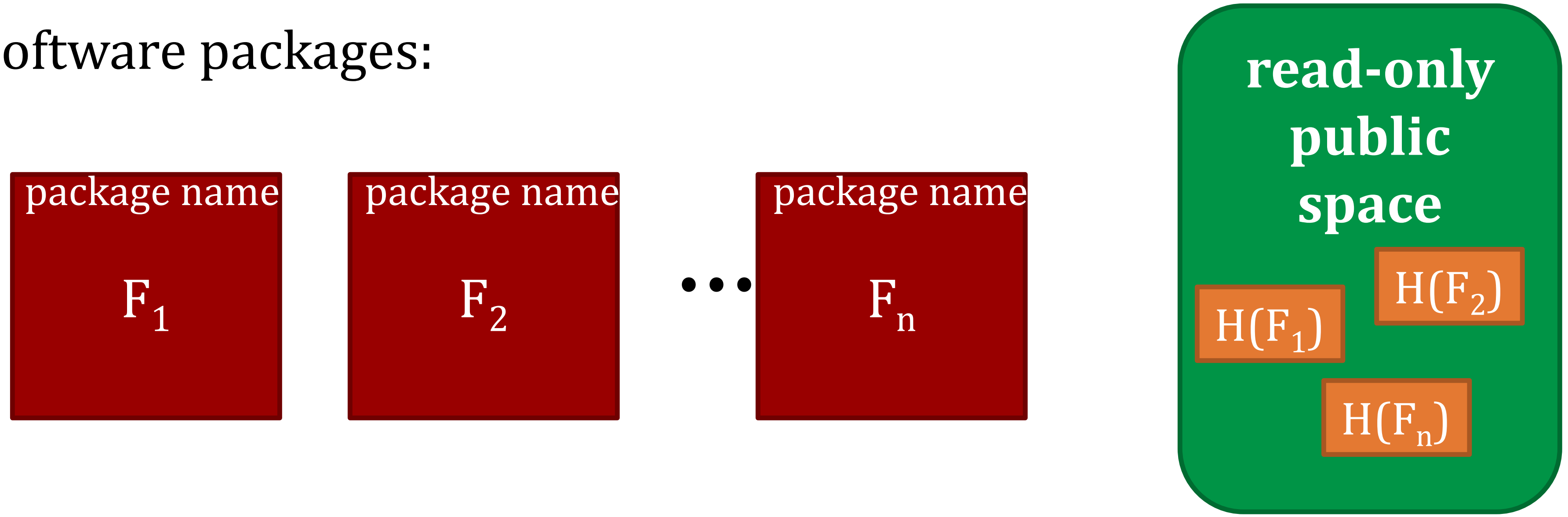
- J. Black, P. Rogaway: CBC MACs for Arbitrary-Length Messages: The Three-Key Constructions. J. Cryptology 18(2): 111-131 (2005)
- K. Pietrzak: A Tight Bound for EMAC. ICALP (2) 2006: 168-179
- J. Black, P. Rogaway: A Block-Cipher Mode of Operation for Parallelizable Message Authentication. EUROCRYPT 2002: 384-397
- M. Bellare: New Proofs for NMAC and HMAC: Security Without Collision-Resistance. CRYPTO 2006: 602-619
- Y. Dodis, K. Pietrzak, P. Puniya: A New Mode of Operation for Block Ciphers and Length-Preserving MACs. EUROCRYPT 2008: 198-219

Questions?

END

Protecting file integrity using C.R. hash

Software packages:



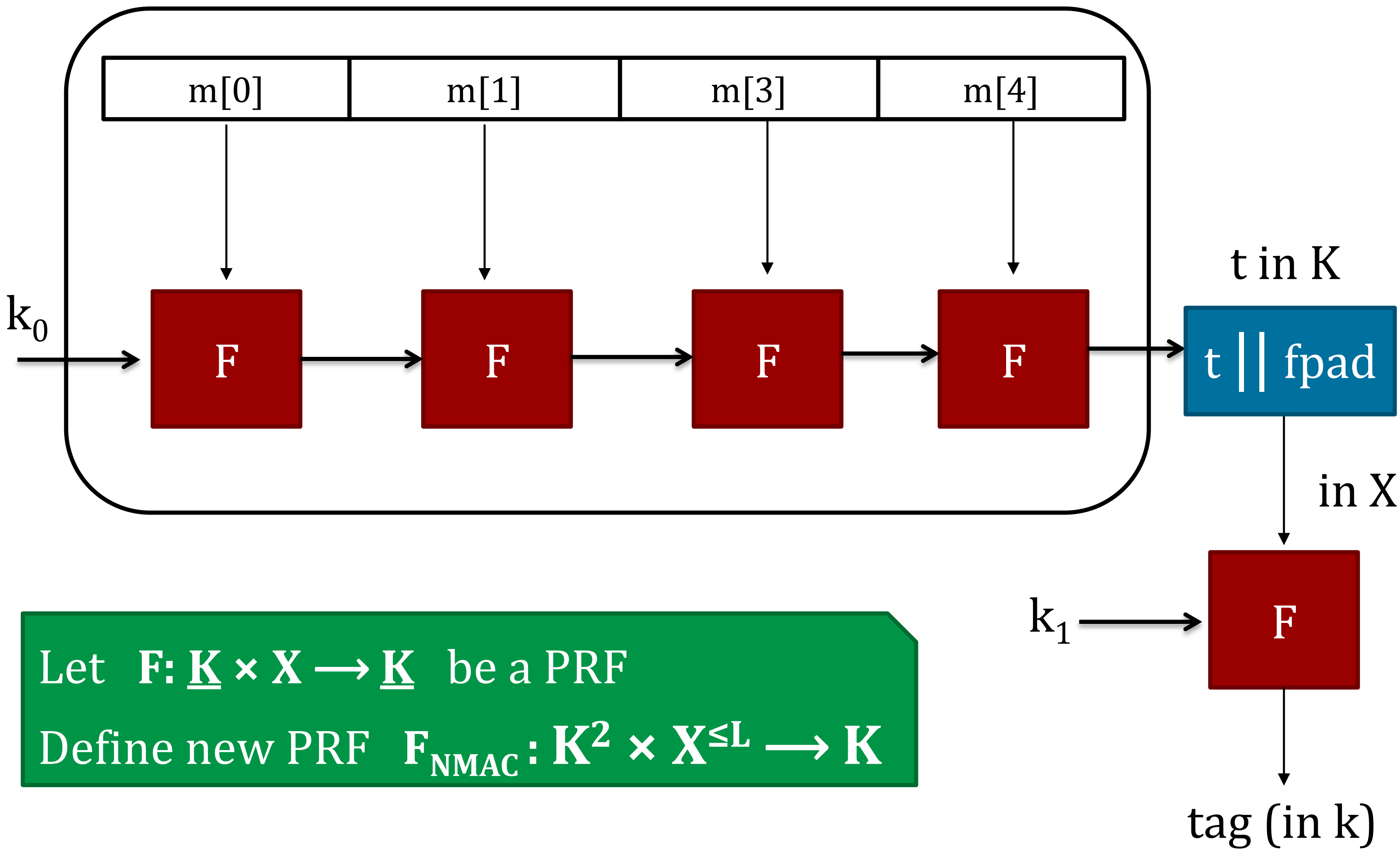
When user downloads package, can verify that contents are valid

H collision resistant $\Rightarrow$
attacker cannot modify package without detection

no key needed (public verifiability), but requires read-only space

Construction 2: Nested MAC (NMAC)

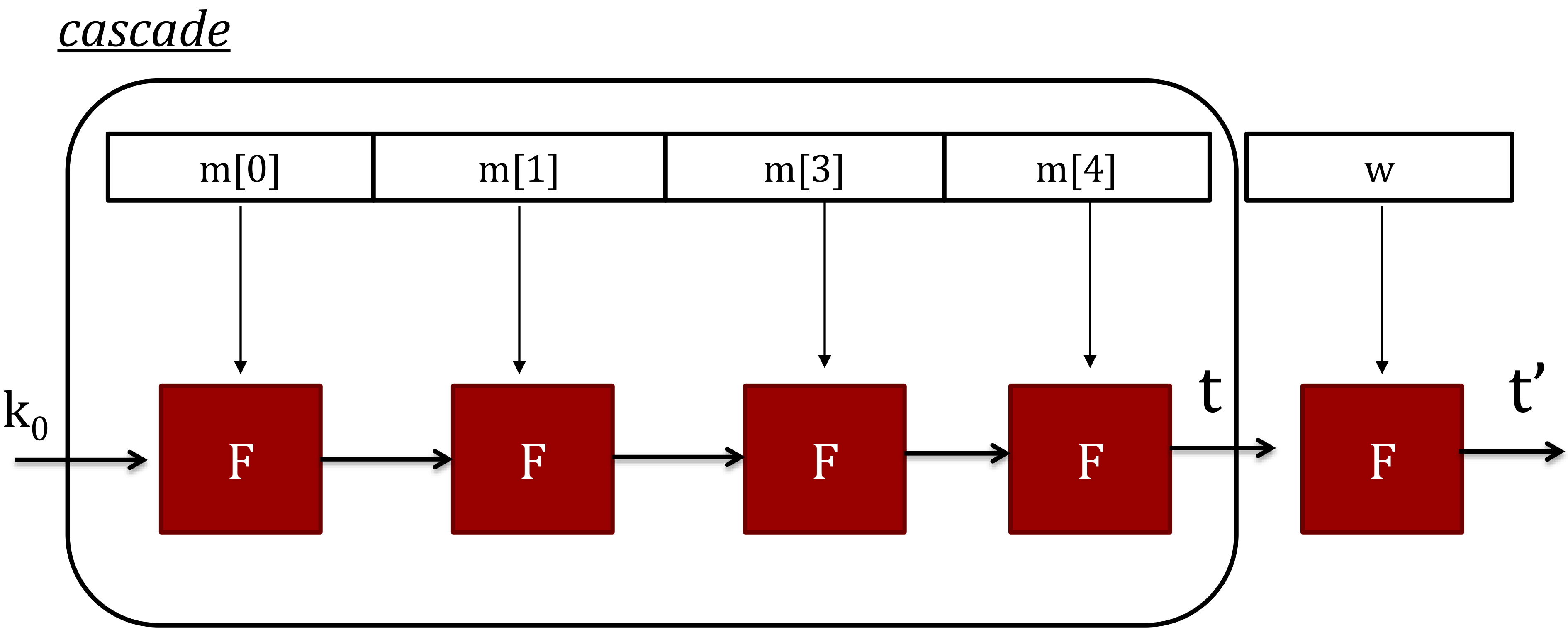
<u>*cascade*</u>



Let **F**: $\underline{\mathbf{K}} \times \mathbf{X} \longrightarrow \underline{\mathbf{K}}$ be a PRF

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

Cascade is insecure



1-query attack: given cascade(m,k_0) = t,
can derive cascade(m||w, t)= t'

- ECBC and NMAC are sequential.

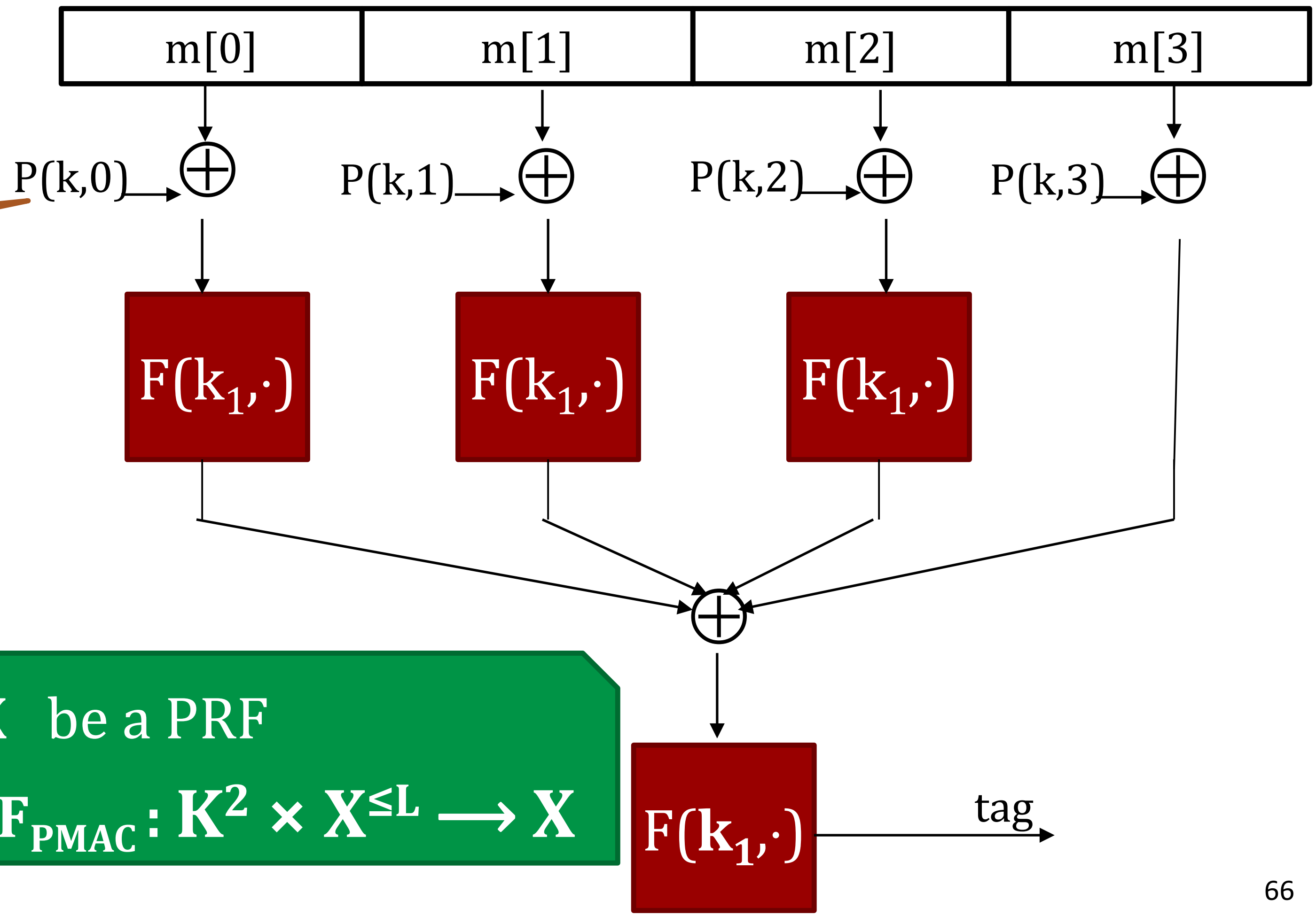
- Can we build a parallel MAC from a small PRF ??

Construction 3: PMAC – Parallel MAC

P(k, i): an easy to compute function

key = (k, k_1)

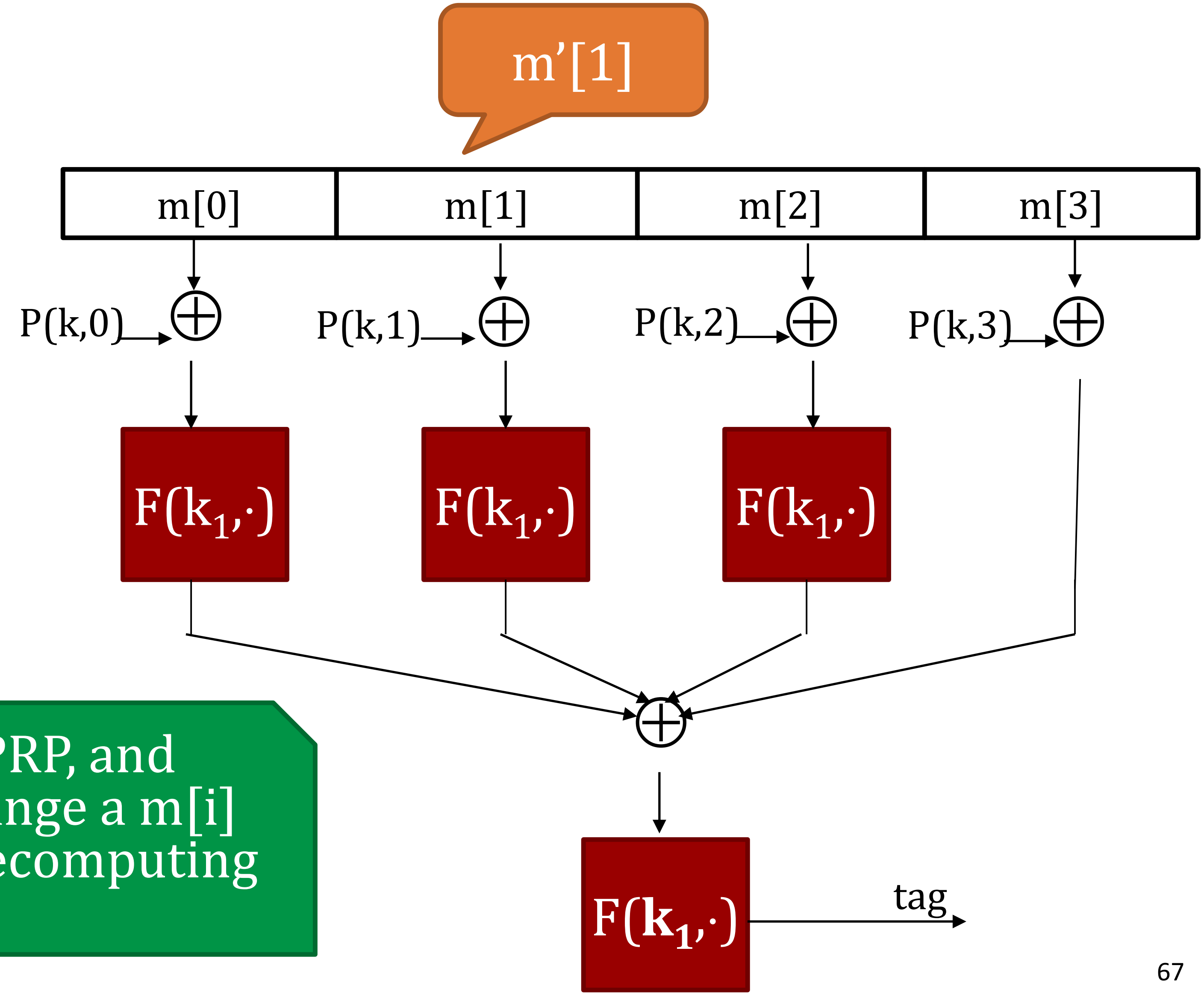
P(k,#) prevents block swapping attack



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

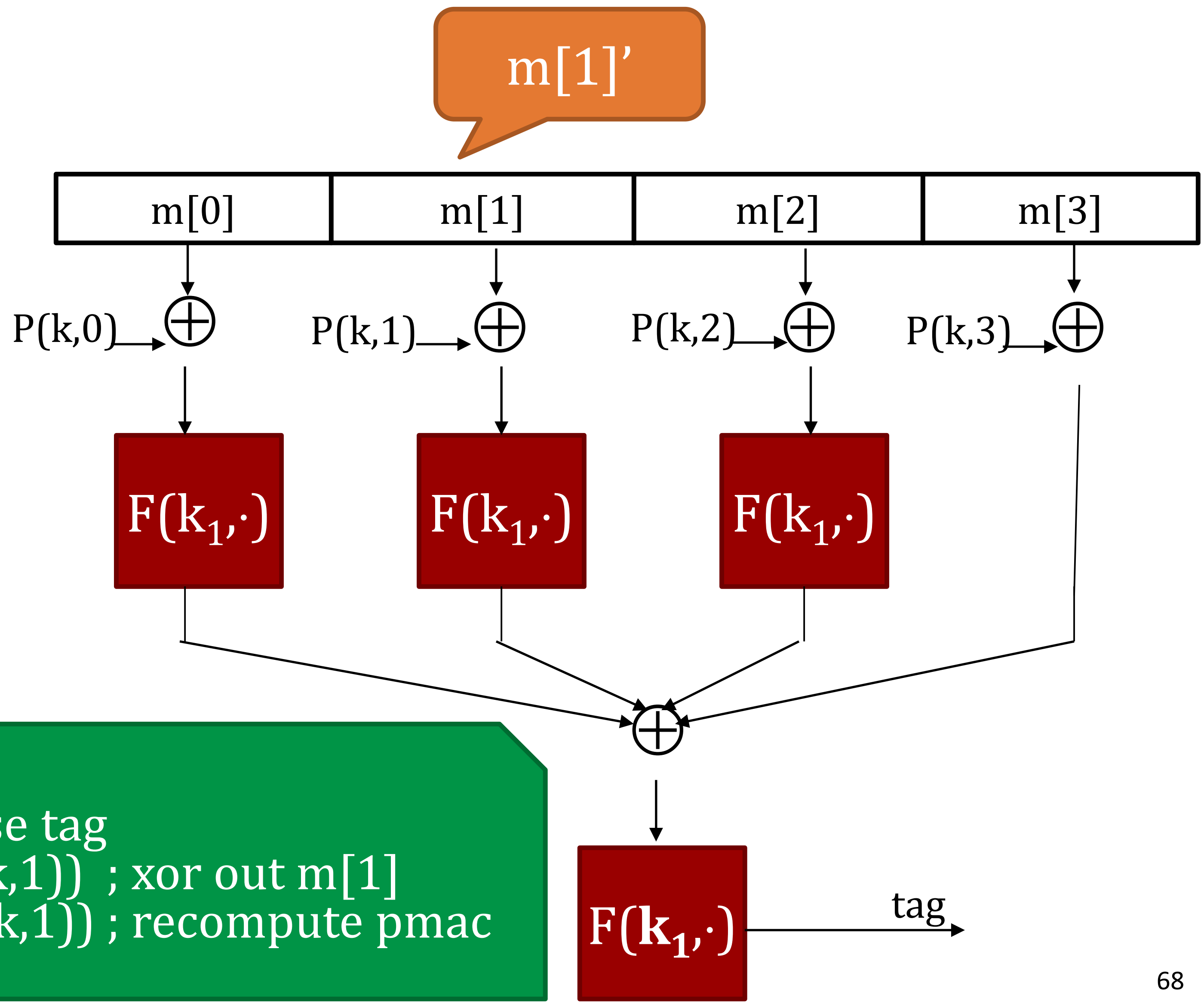
Define new PRF $\mathbf{F_{PMAC} : K^2 \times X^{\leq L} \longrightarrow X}$

Cool Feature: Incremental Updates



Suppose F is a PRP, and suppose we change a m[i] to m'[i]. Then recomputing the tag is easy.

Cool Feature: Incremental Updates



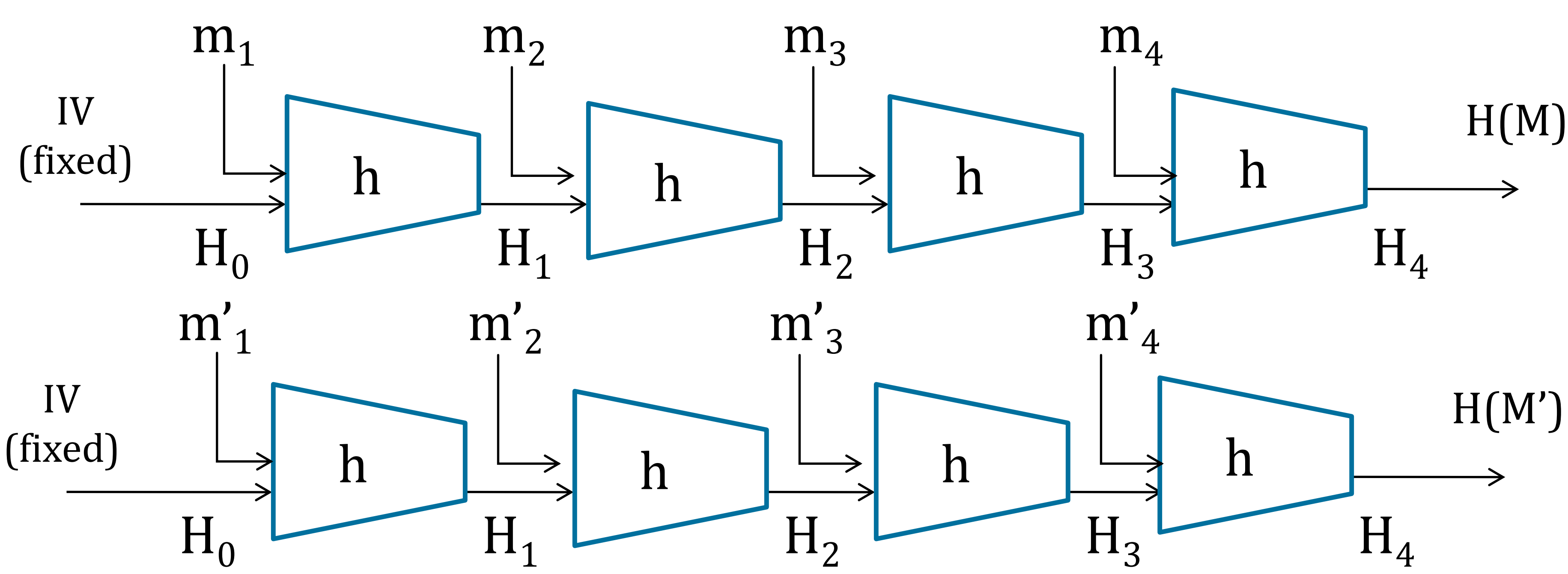
tag' =
$F^{-1}(k_1, \text{tag})$; reverse tag
$\oplus\ F(k_1, m[1] \oplus P(k,1))$; xor out m[1]
$\oplus\ F(k_1, m[1]' \oplus P(k,1))$; recompute pmac

HMAC (Hash-MAC)

Most widely used MAC on the Internet.

… but, we first we need to discuss hash function.

Proof: collision on H → collision on h

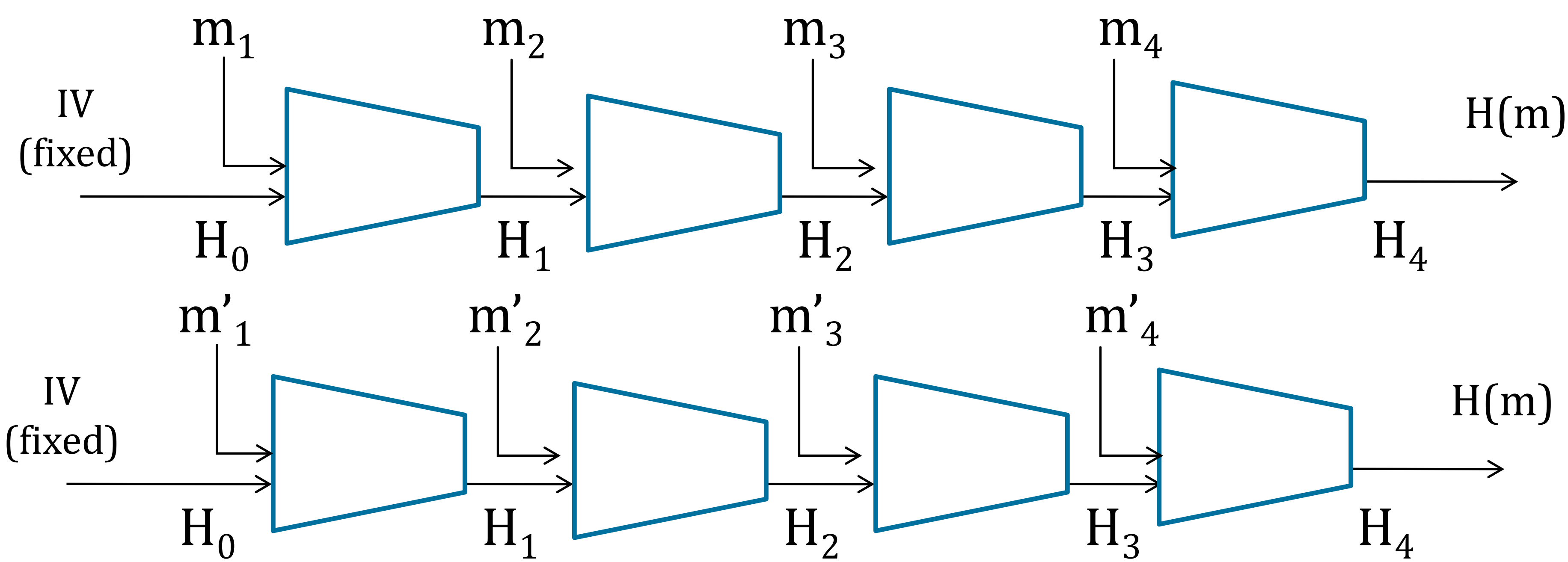


Let |M| = |M'|, M≠M', and H(M) = $h_i(m_i, H_{i-1})$ with m_0 = IV

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

<u>Case 1</u>: $m_i \neq m'_i$ *or* $H_{i-1} \neq H'_{i-1}$. But since $f(m_i, H_{i-1}) = f(m'_i, H'_{i-1})$ there is a collision in h and we are done. Else recurse

Proof: collision on H → collision on h



Let $|M| = |M'|$, $M \neq M'$, and $H(M) = h_i(m_i, H_{i-1})$ with $m_0 = IV$

Suppose $H(M) = H(M')$. We build a collision on h.

<u>Case 2</u>: $m_i = m'_i$ *and* $H_{i-1} \neq H'_{i-1}$ for all i. But then $M = M'$, violating our assumption.

Question

Suppose we define **$h(H, m) = E(m, H)$**

Then the resulting h(.,.) is not collision resistant.

To build a collision (H,m) and (H',m'), i.e.,

$$E(m,H) = E(m', H')$$

choose random (H,m,m') and construct H' as follows:

1. $H'=D(m', E(m,H))$

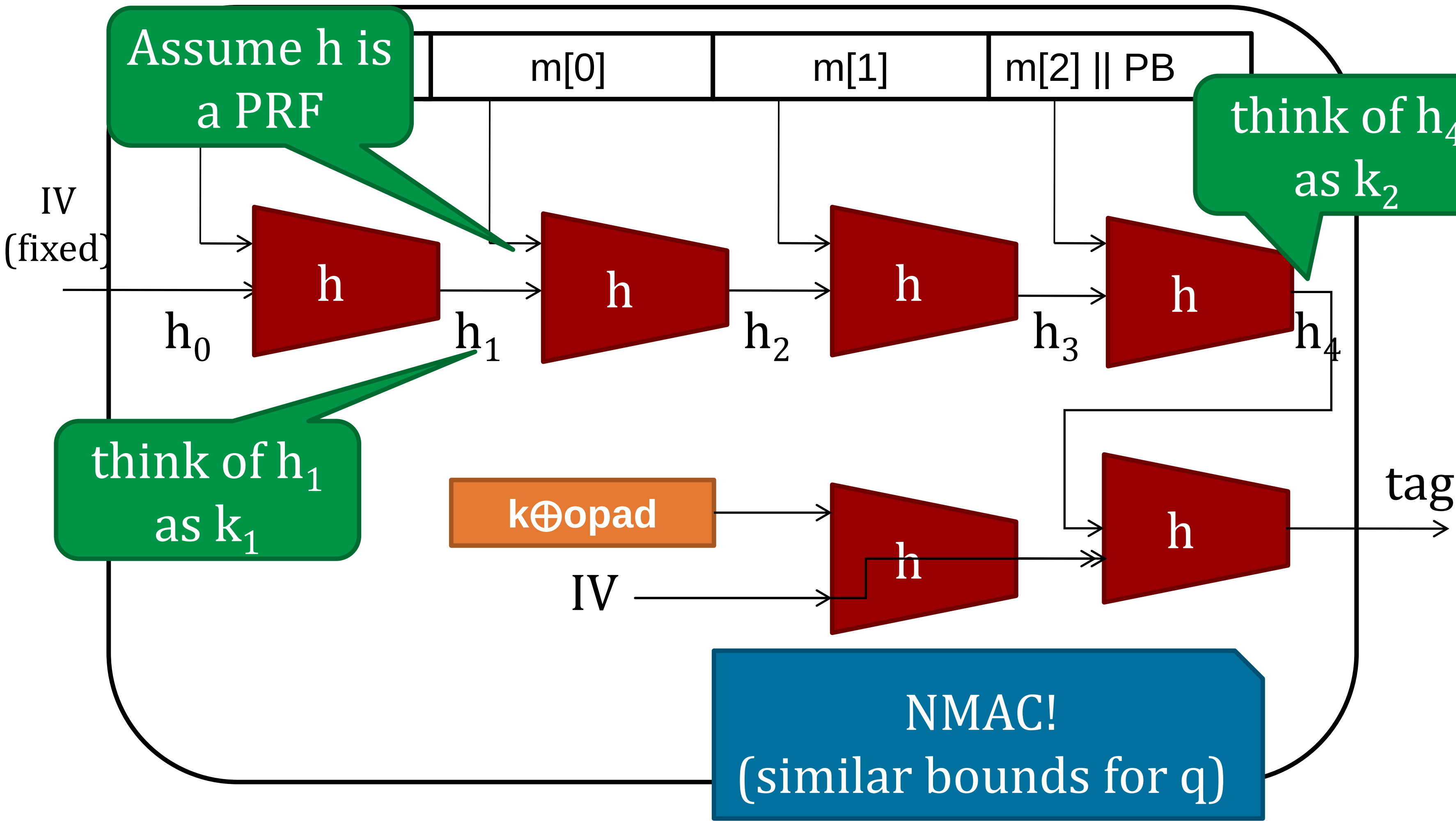
2. $H'=E(m', D(m,H))$

3. $H'=E(m', E(m,H))$

4. $H'=D(m', D(m,H))$

$E(m', H')$
$= E(\underline{m'}, D(\underline{m'}, E(m,H)))$
$= E(m,H)$

HMAC



PB: Padding Block