CIS 5560

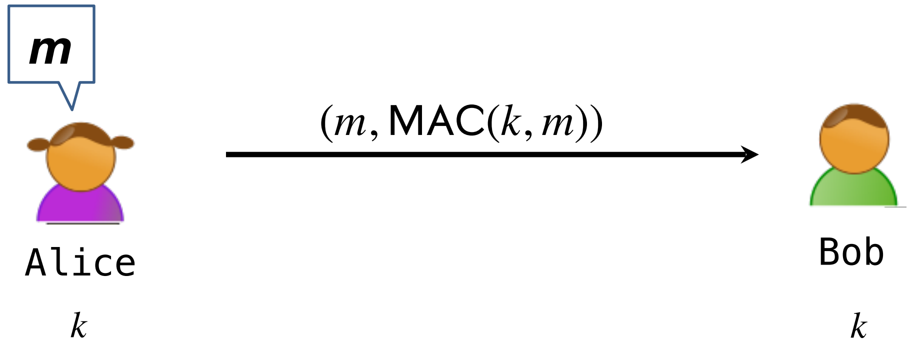
Cryptography
Lecture 11

Announcements

- **HW4 is out**
- **HW3 due tomorrow!**

Recap of last lecture

Constructing a MAC



Gen(1^n): Produces a PRF key $k \leftarrow K$.

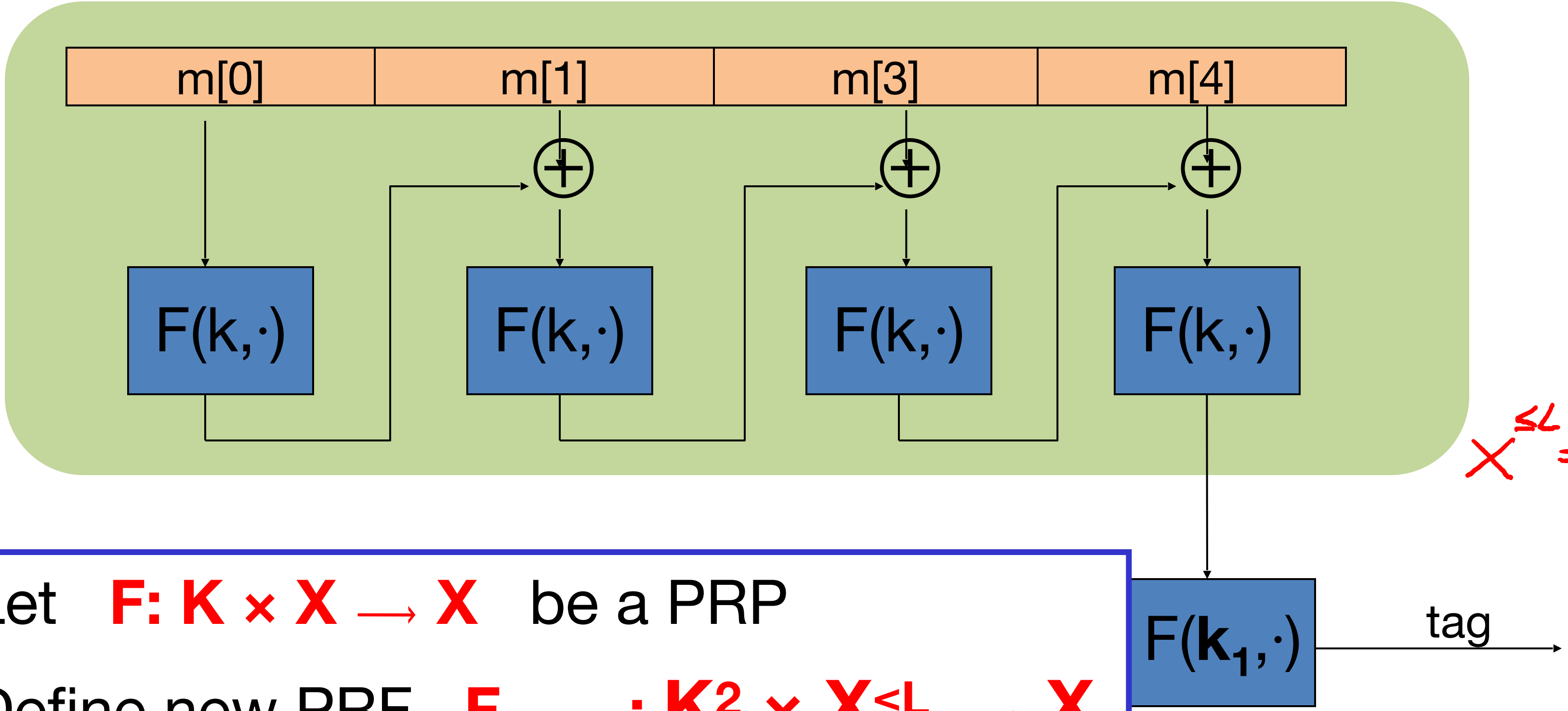
MAC(k, m): Output $F_k(m)$.

Ver(k, m, t): Accept if $F_k(m) = t$, reject otherwise.

Security: ??

Construction: encrypted CBC-MAC

raw CBC



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

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

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

Today

- Collision resistant hash functions
- Constructing CRHFs with long inputs
- HMAC
- Other properties of (hash) functions

Collision Resistance

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

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

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

A function H is **collision resistant** if for all efficient algs. A:

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

is negligible.

Example: SHA-256 (outputs 256 bits)

MACs from Collision Resistance

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

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

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

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

Thm: If MAC is a secure MAC and H is collision resistant
then MAC^{big} is a secure MAC.

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

MACs from Collision Resistance

$$MAC^{big}(k, m) = MAC(k, H(m)) \quad ;$$

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

Collision resistance is necessary for security:

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

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

step 1: adversary asks for $t \leftarrow MAC(k, m_0)$

step 2: output (m_1, t) as forgery

How easy is it to find collisions?

Generic attack on CRHFs

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

The birthday paradox

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

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

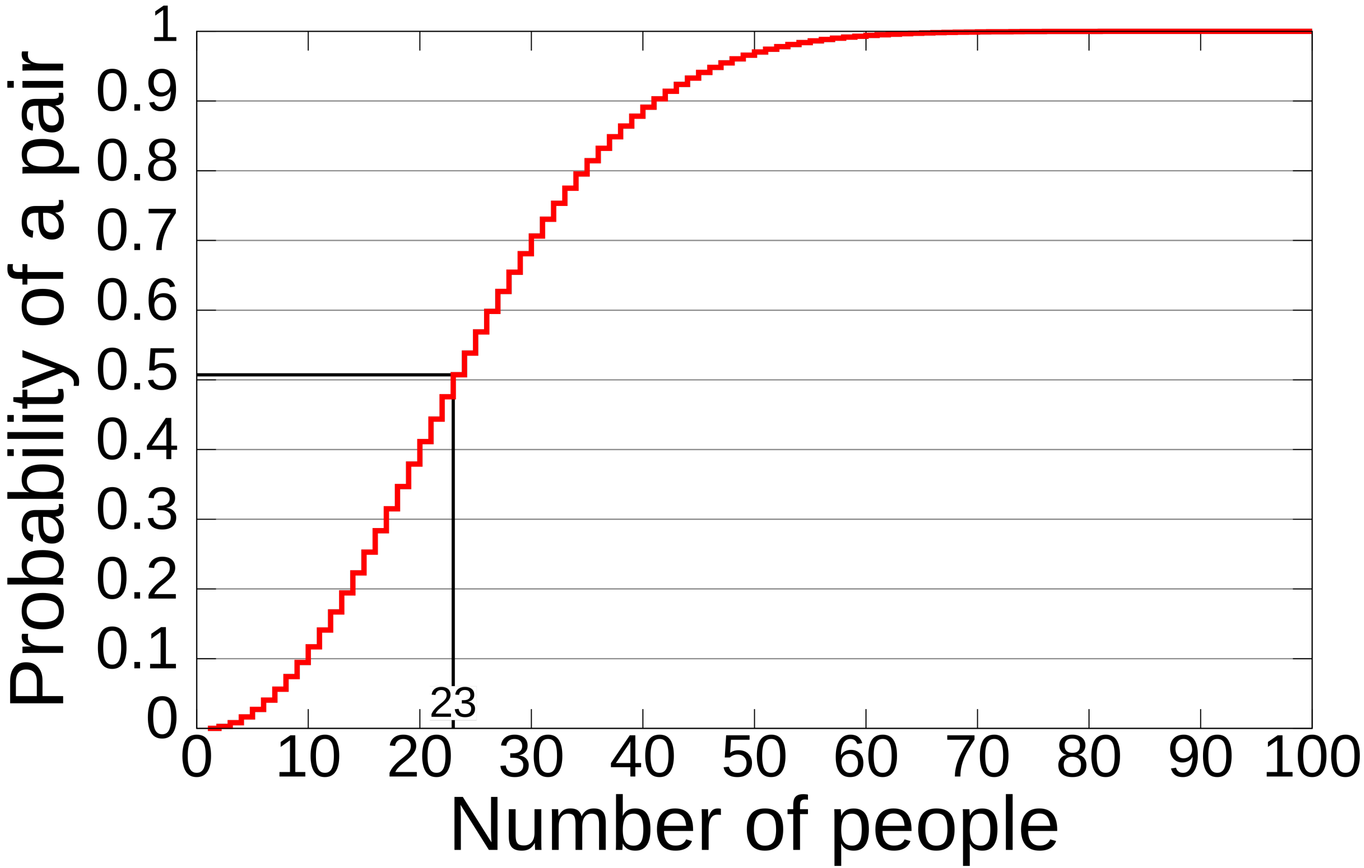
Proof: for <u>uniformly</u> independent $r_1, \ldots, r_n$,

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

$$= 1 - \prod_{i=1}^{n-1}\left(1 - \frac{i}{B}\right) \geq 1 - \prod_{i=1}^{n-1} e^{-i/B} \quad \text{(since } 1 - x \leq e^{-x}\text{)}$$

$$= 1 - e^{-\frac{1}{B}\sum_{i=1}^{n-1} i} \geq 1 - e^{-n^2/2B}$$

$$\geq 1 - e^{-0.72} = 0.53 > \frac{1}{2} \quad \text{(when } \frac{n^2}{2B} = 0.72\text{)}$$


Probability of a pair
Number of people
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0 10 20 30 40 50 60 70 80 90 100
23

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:

* SHA-1 is broken; do not use!

Collision-Resistant Hash Functions

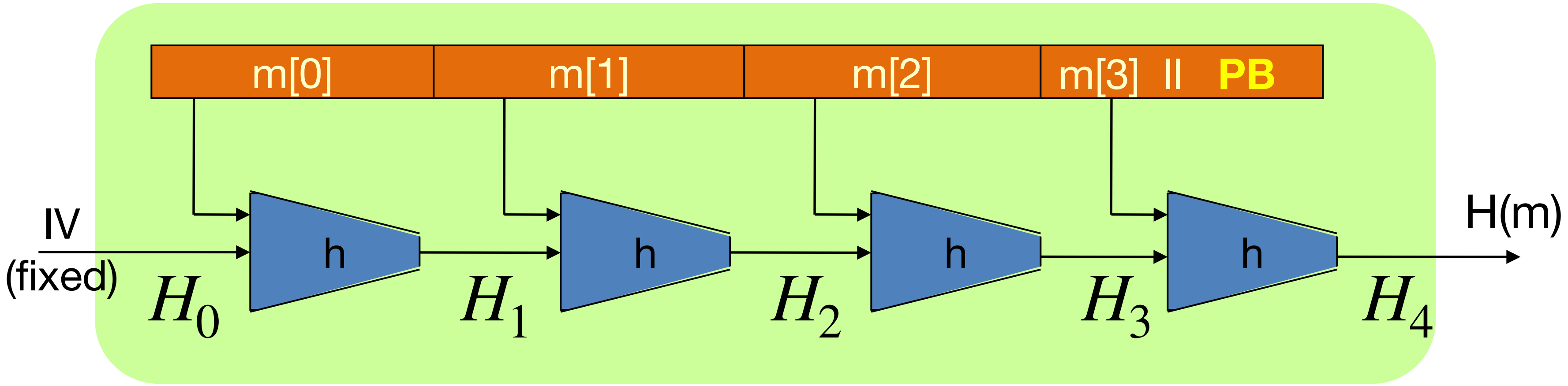
NIST

function	digest (bits)	speed (MB/s)	collision security	notes
SHA-1	160	1,350	2^{80}	broken — real collisions found (2^{63})
SHA-256	256	1,339 *	2^{128}	* SHA-NI hw accel; ~500 without
SHA-512	512	561	2^{256}	faster on 64-bit without SHA-NI
SHA3-256	256	330	2^{128}	Keccak sponge — no length extension
SHA3-512	512	183	2^{256}	also defines SHAKE XOFs
BLAKE2b	512	613	2^{256}	RFC 7693 — SHA-3 finalist successor
BLAKE3	256	~3,000 †	2^{128}	Merkle tree — scales with cores

Benchmark: OpenSSL 3.0.13, 8 KB blocks, single-threaded, x86-64 w/ SHA-NI. * SHA-NI hardware acceleration.
† BLAKE3 number from published benchmarks (AVX2, single-threaded); not in OpenSSL. Multi-threaded: ~15 GB/s on 16 cores.

Constructing CRHFs for long messages: Merkle-Damgard

The Merkle-Damgard iterated construction



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

we obtain $H : X_{\leq L} \to T$. H_i - chaining variables

PB: padding block

If no space for PB add another block

MD collision resistance

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

Proof: collision on H $\Rightarrow$ collision on h

Intermediate hashes

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

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

$$\text{IV} = H'_0 \;,\quad H'_1 \;,\; \ldots \;,\; H'_r \;,\quad H'_{r+1} = H(M')$$

There must be a r and t such that this holds

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

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 $\left[H_{t-1} \neq H'_{t-1} \text{ or } M_{t-1} \neq M'_{t-1} \right]$ 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:

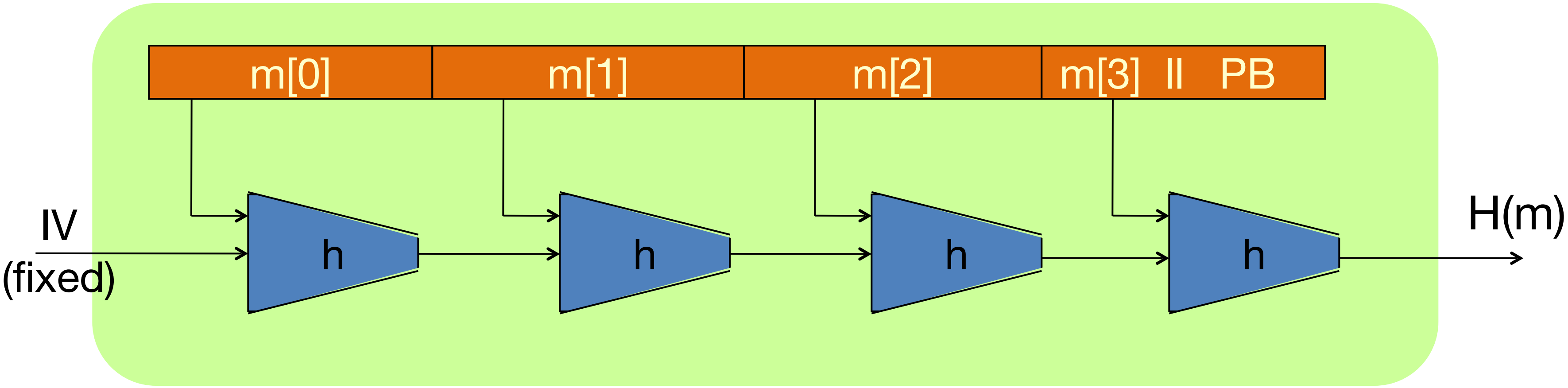
(1) find collision on h, or

(2) $\forall i: M_i = M'_i \Rightarrow M = M'$

cannot happen because M, M' are collision on H.

HMAC: a MAC from SHA-256

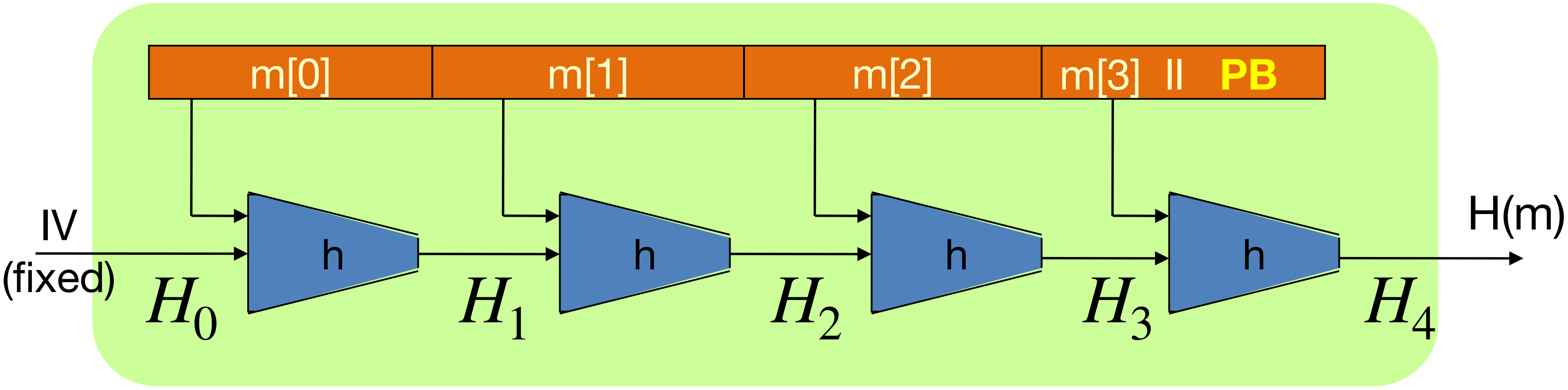
The Merkle-Damgard iterated construction



Thm: h collision resistant ⇒ H collision resistant

Can we use H to directly build a MAC?

The Merkle-Damgard iterated construction



Thm: h collision resistant $\Rightarrow$ 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

<u>Attempt #1</u>: $\mathrm{MAC}(k, m) := H(k \,||\, m)$

This MAC is insecure because:

- Given $H(k \,||\, m)$ can compute $H(k \,||\, m \,||\, PB \,||\, w)$ for any w.
- Given $H(k \,||\, m)$ can compute $H(k \,||\, m \,||\, w)$ for any w.
- Given $H(k \,||\, m)$ can compute $H(w \,||\, k \,||\, m \,||\, PB)$ for any w.
- Anyone can compute $H(k \,||\, m)$ for any m.

Standardized method: HMAC (Hash-MAC)

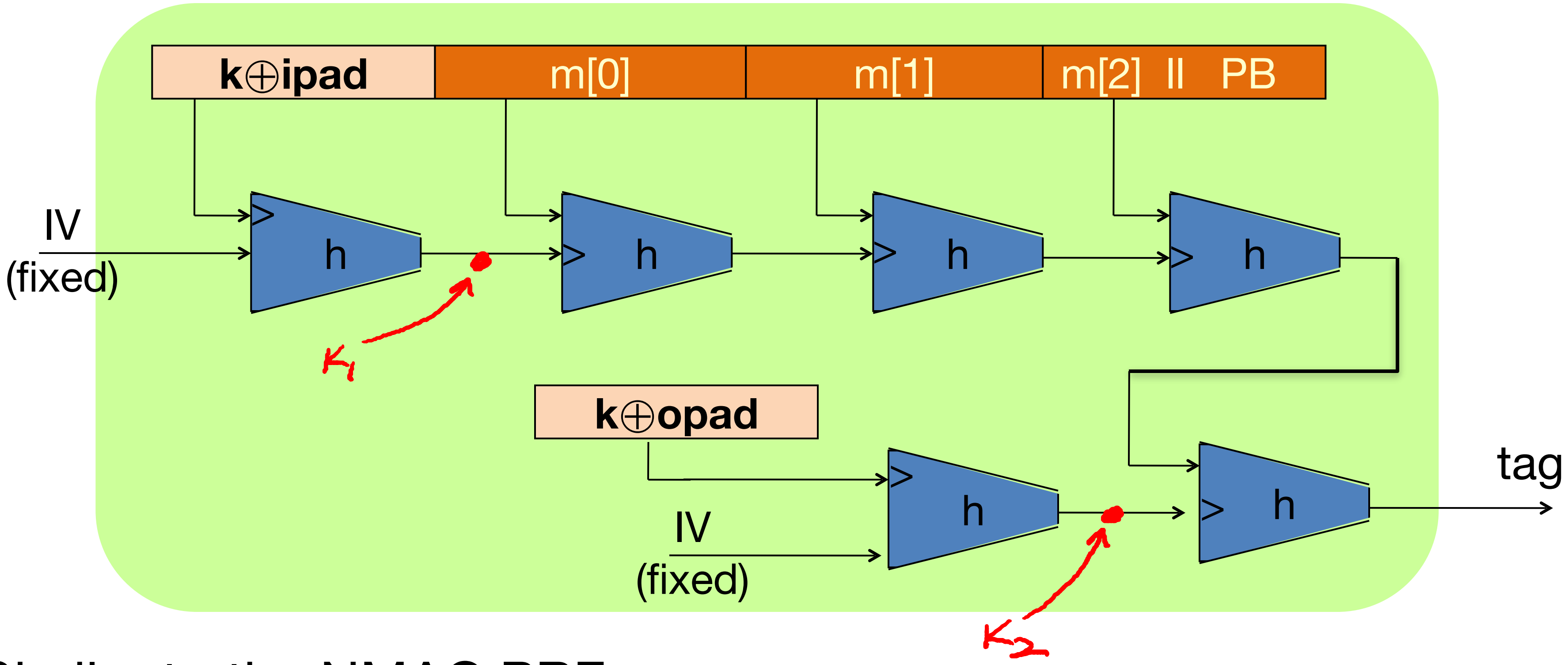
Most widely used MAC on the Internet.

Building a MAC out of a hash function H:

HMAC:

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

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 ($q << |T|^{1/2}$)

Other properties of (hash) functions

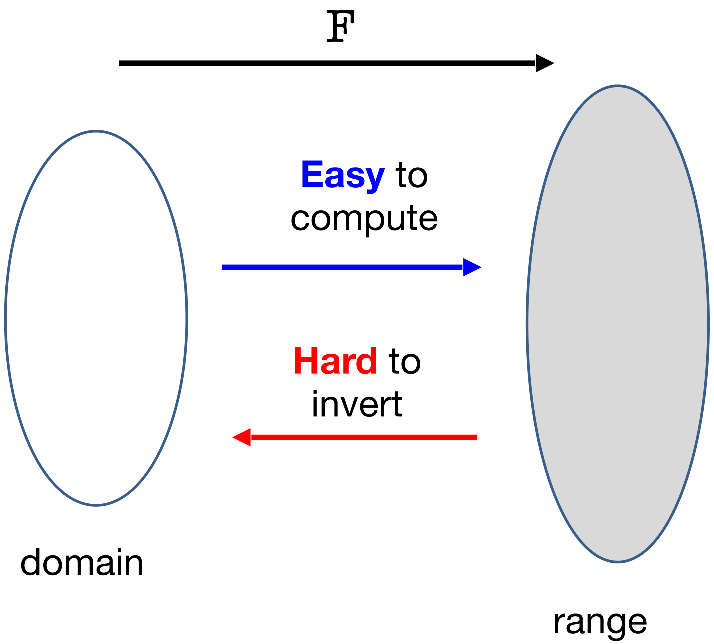
Other properties of (hash) functions

- Collision resistance:
  - Can't find two inputs with same output
  - That is, can't find $x \neq x'$ such that $h(x) = h(x')$
- One-wayness/Preimage resistance:
  - Difficult to find input given an output
  - That is, given $y \in \mathsf{Range}(h)$, can't find x s.t. $h(x) = y$
- 2nd-preimage resistance:
  - Given input x, can't find another input with same output
  - That is, given x, can't find x' s.t. $h(x) = h(x')$

How are these properties related?

- Q1: If h is collision-resistant, is it also 2nd-preimage resistant?
  - Yes! If you can't find *any* collisions, you also can't find a *specific* collision
- Q2: If h is one-way, is it also collision-resistant?
  - No. E.g.: h outputs 0^n on two inputs.
- Q2: If h is collision-resistant, is it also one-way?
  - Not necessarily! E.g.: let h be CRH. Then construct f such that if first bit of input x is 0, then output rest of input, otherwise, output $h(x)$.

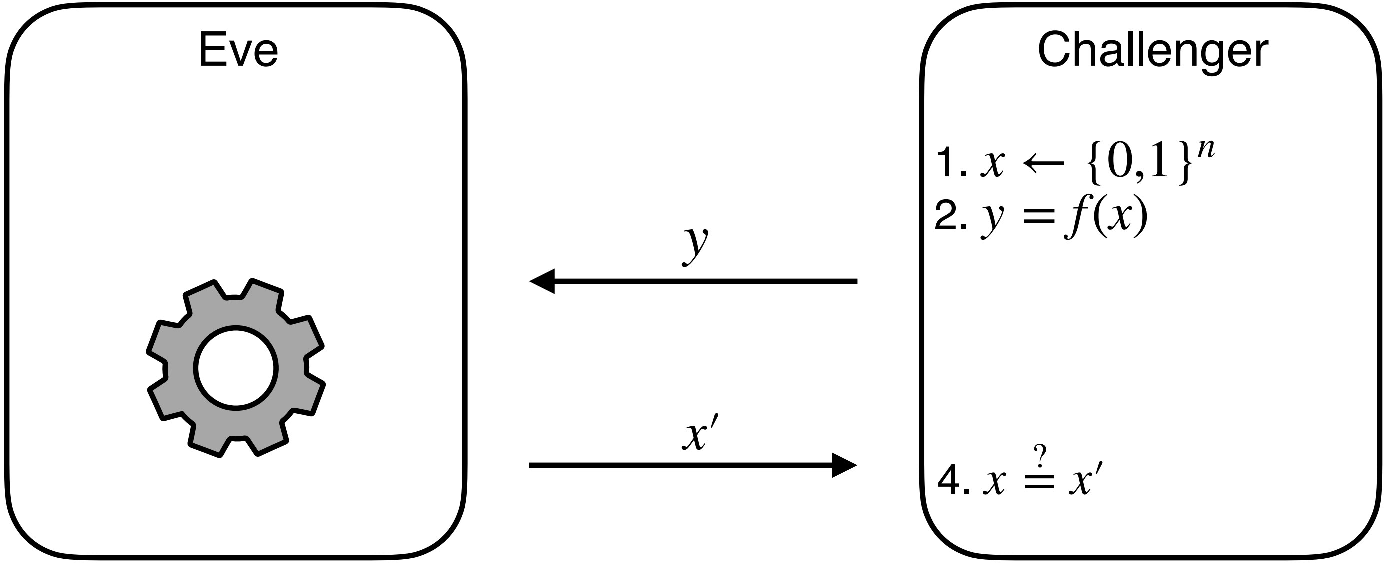
One-way Functions (Informally)



Source of all hard problems in cryptography!

What is a good definition?

OWF Security Attempt #1



One-way Functions (Take 1)

A function (family) $\{F_n\}_{n\in\mathbb{N}}$ where $F(\cdot):\{0,1\}^n \rightarrow \{0,1\}^{m(n)}$ is **one-way** if for every p.p.t. adversary A, the following holds:

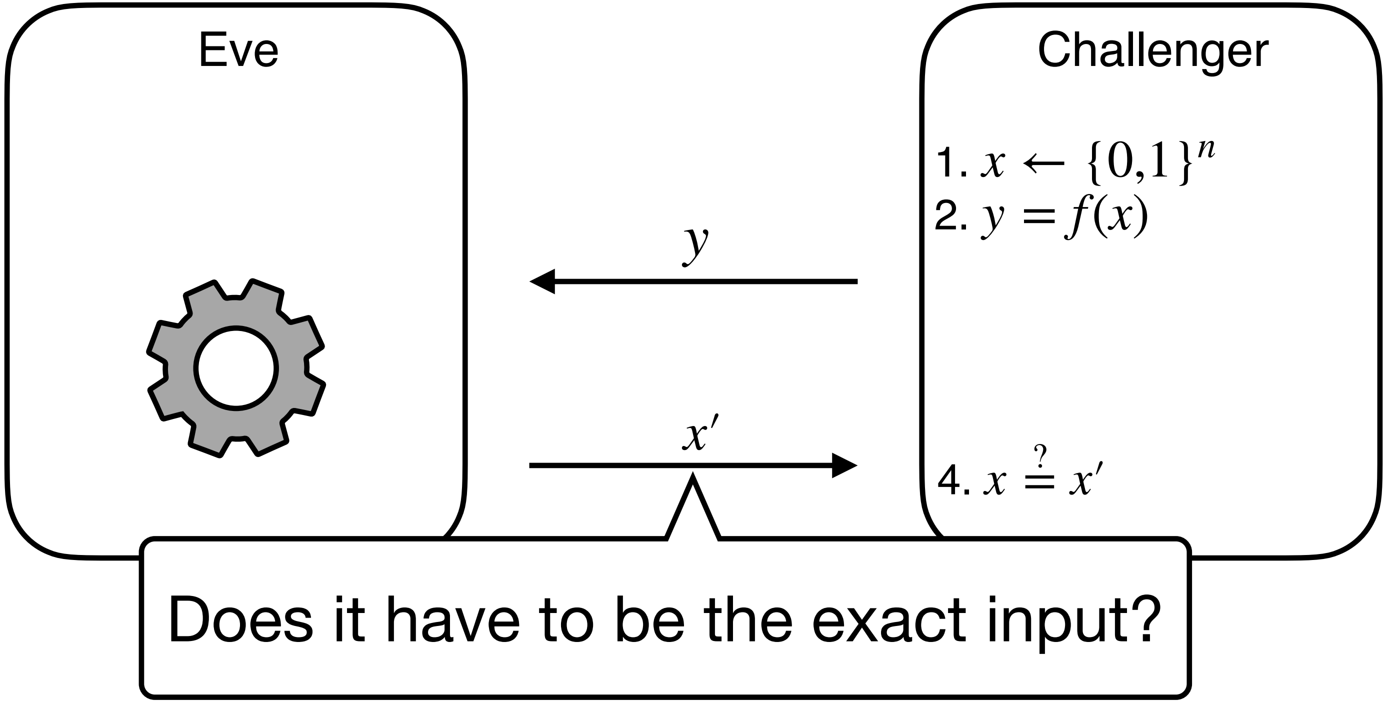
$$\Pr\left[A(1^n, y) = x \;\middle|\; \begin{array}{c} x \leftarrow \{0,1\}^n \\ y := F_n(x) \end{array}\right] = \mathsf{negl}(n)$$

Consider $\mathbf{F_n(x) = 0}$ for all x.

This is one-way according to the above definition.
In fact, impossible to find *the* inverse even if A has unbounded time.

Conclusion: not a useful/meaningful definition.

OWF Security Attempt #2



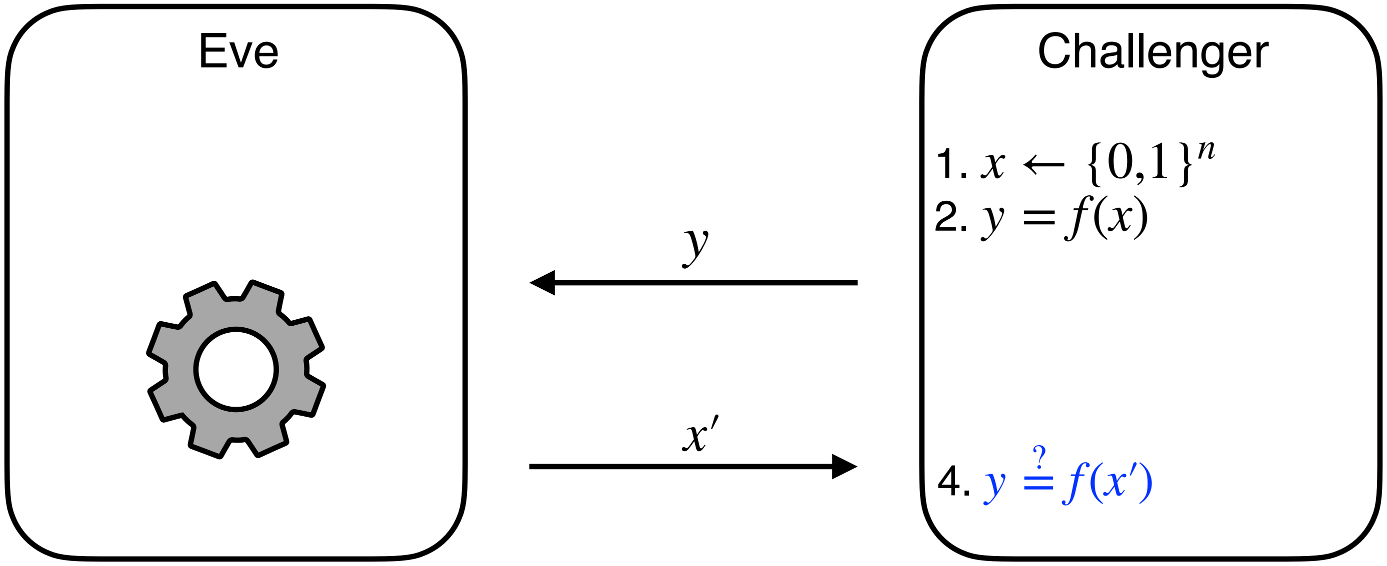
One-way Functions (Take 1)

A function (family) $\{F_n\}_{n\in\mathbb{N}}$ where $F(\cdot) : \{0,1\}^n \to \{0,1\}^{m(n)}$ is **one-way** if for every p.p.t. adversary A, the following holds:

$$\Pr\left[A(1^n, y) = x \,\middle|\, \begin{array}{c} x \leftarrow \{0,1\}^n \\ y := F_n(x) \end{array}\right] = \text{negl}(n)$$

The Right Definition: Impossible to find ***an*** inverse efficiently.

OWF Security Attempt #2



One-way Functions: The Definition

A function (family) $\{F_n\}_{n\in\mathbb{N}}$ where $F(\cdot):\{0,1\}^n \to \{0,1\}^{m(n)}$ is **one-way** if for every p.p.t. adversary A, the following holds:

$$\Pr\left[F_n(x') = y \;\middle|\; \begin{array}{r} x \leftarrow \{0,1\}^n \\ y := F_n(x) \\ x' \leftarrow A(1^n, y) \end{array}\right] = \text{negl}(n)$$

- Can always find *an* inverse with unbounded time
- … but should be hard with probabilistic polynomial time

One-way Permutations:

One-to-one one-way functions with $m(n) = n$.