COS433/Math 473: Cryptography

Mark Zhandry

Princeton University

Fall 2020

Announcements/Reminders

HW3 due on Oct 20

HW4 will be released today, due Oct 27

Previously on COS 433…

Collision Resistant Hashing

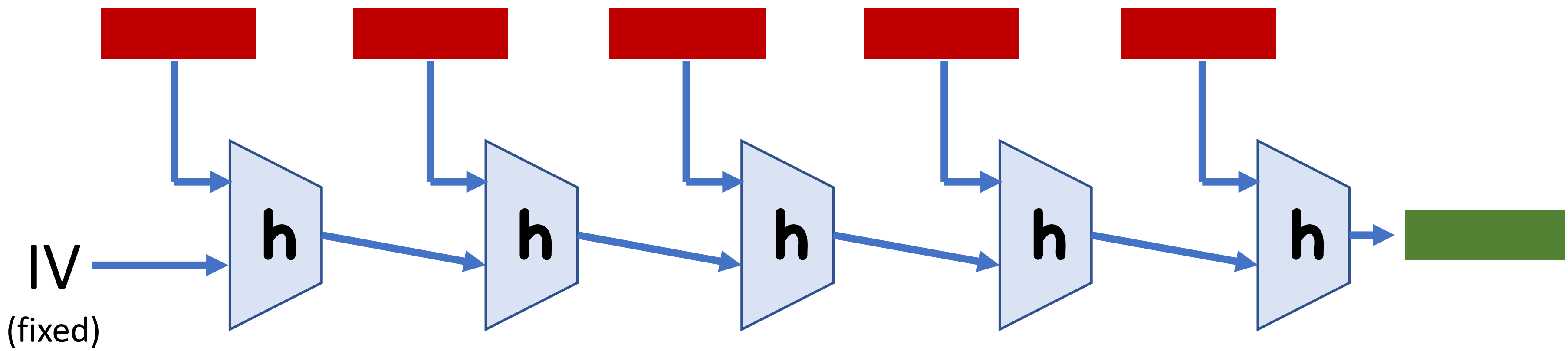
Collision Resistant Hashing

Syntax:

- Key space $\mathbf{K}$ (typically $\{0,1\}^\lambda$)
- Domain $\mathbf{D}$ (typically $\{0,1\}^l$ or $\{0,1\}^*$)
- Range $\mathbf{R}$ (typically $\{0,1\}^n$)
- Function $\mathbf{H}$: $\mathbf{K} \times \mathbf{D} \rightarrow \mathbf{R}$

Correctness: $n \ll l$

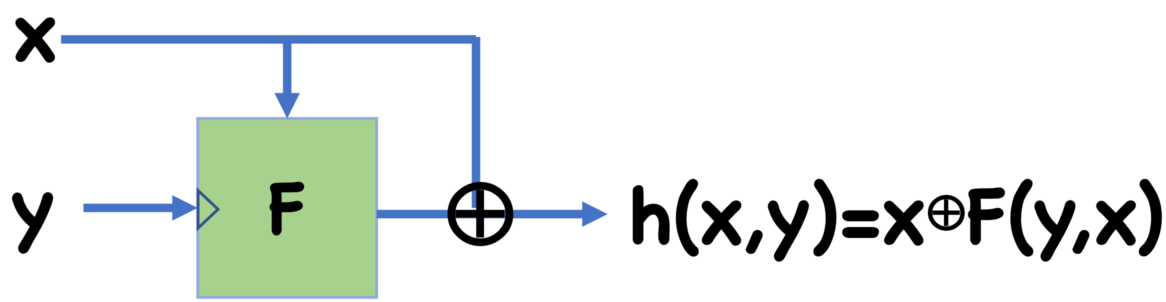
Merkle-Damgard



Constructing **h**

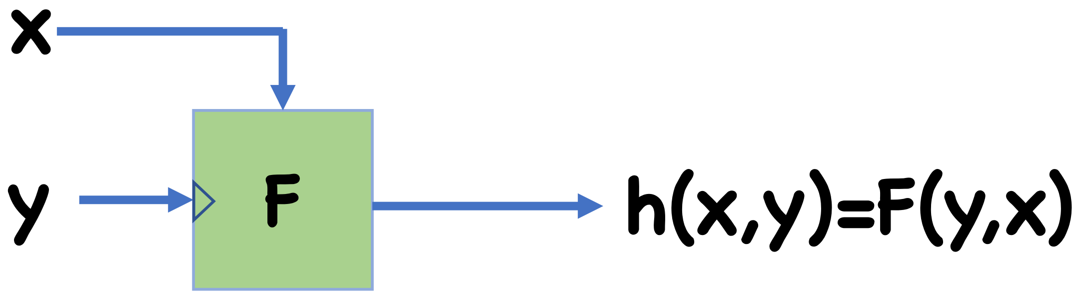
Common approach: use block cipher

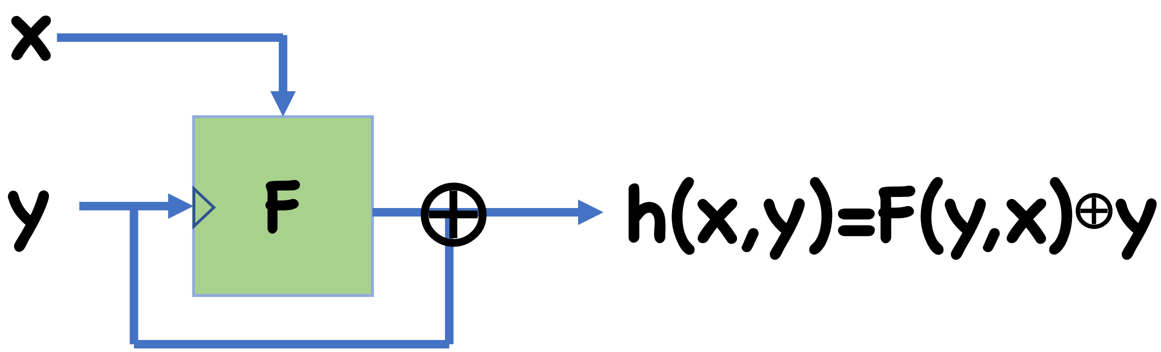
Davies-Meyer



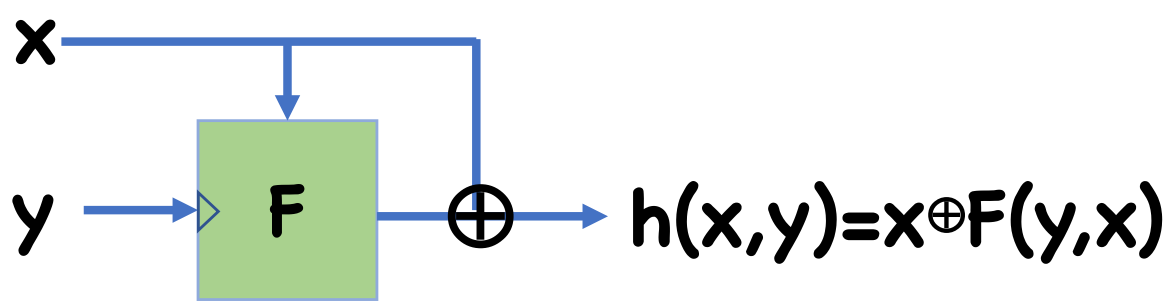
Constructing **h**

Some other possibilities are insecure





Constructing **h**



Why do we think Davies-Meyer is reasonable?

- Cannot prove collision resistance just based on **F** being a secure PRP

Instead, can argue security in "ideal cipher" model

- Pretend **F**, for each key **y**, is a uniform random permutation

Today

- Collision resistance cont.
- Random Oracle Model
- Commitments

We said 128 bit security is usually enough

However, 128-bit blocks insufficient for compression function. Why?

Birthday Attack

If the range of a hash function is $\mathbf{R}$, a collision can be found in time $\mathbf{T=O(|R|^{1/2})}$

Attack:

- Given key $\mathbf{k}$ for $\mathbf{H}$
- For $\mathbf{i=1,\ldots, T}$,
  - Choose random $\mathbf{x_i}$ in $\mathbf{D}$
  - Let $\mathbf{t_i \leftarrow H(k,x_i)}$
  - Store pair $\mathbf{(x_i,\ t_i)}$
- Look for collision amongst stored pairs

Birthday Attack

Analysis:

Expected number of collisions

= Number of pairs × Prob each pair is collision

$\approx$ **(T choose 2) × 1/|R|**

By setting **$T=O(|R|^{1/2})$,** expectend number of collisions found is at least **1**

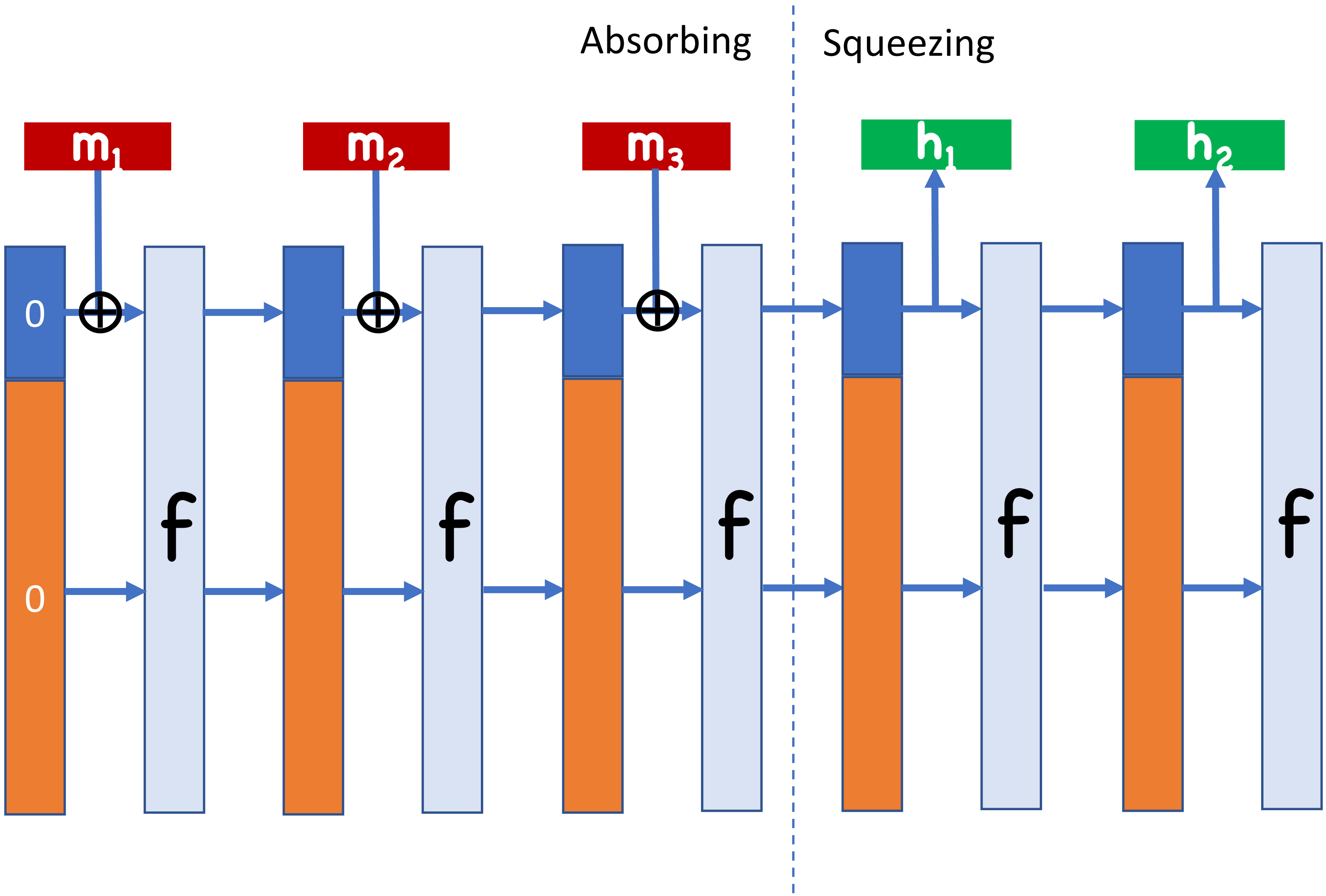
$\Rightarrow$ likely to find a collision

Birthday Attack

Space?

Possible to reduce memory requirements to **$O(1)$**

Sponge Construction



Sponge Construction

Advantages:

- Round function **f** can be public invertible function
  (i.e. unkeyed SPN network)

- Easily get different input/output lengths

SHA-1,2,3

SHA-1,2 are hash functions built as follows:

- Build block cipher (SHACAL-1, SHACAL-2)
- Convert into compression function using Davies-Meyer
- Extend to arbitrary lengths using Merkle-Damgard

SHA-3 is based on sponge construction

SHA-1,2,3

SHA-1 (1995) is no longer considered secure

- 160-bit outputs, so collisions in time 2^{80}
- 2017: using some improvements over birthday attack, able to find a collision

SHA-2 (2001)

- Longer output lengths (256-bit, 512-bit)
- Few theoretical weaknesses known

SHA-3 (2015)

- NIST wanted hash function built on different principles

Basing MACs on Hash Functions

Idea: $\mathbf{MAC(k,m) = H(k \,||\, m)}$

Thought: if $\mathbf{H}$ is a "good" hash function and $\mathbf{k}$ is random, should be hard to predict $\mathbf{H(k \,||\, m)}$ without knowing $\mathbf{k}$

Unfortunately, cannot prove secure based on just collision resistance of $\mathbf{H}$

Random Oracle Model

Pretend **H** is a truly random function

Everyone can query **H** on inputs of their choice
- Any protocol using **H**
- The adversary (since he knows the key)

A query to **H** has a time cost of 1

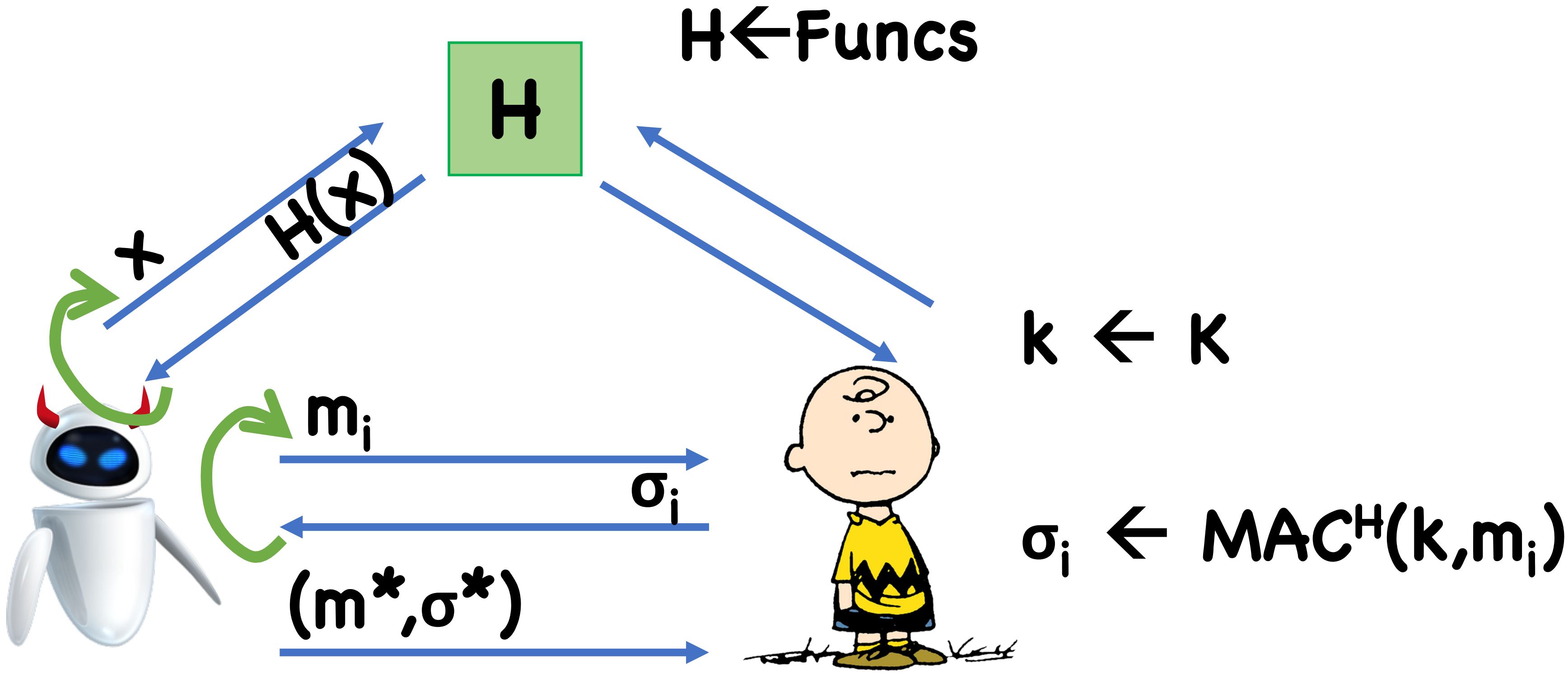
Intuitively captures adversaries that simple query **H**, but don't take advantage of any structure

MAC in ROM

$\mathsf{MAC}^H(k,m) = H(k||m)$
$\mathsf{Ver}^H(k,m,\sigma) = (H(k||m) == \sigma)$

Theorem: $H(k \; || \; m)$ is a CMA-secure MAC in the random oracle model

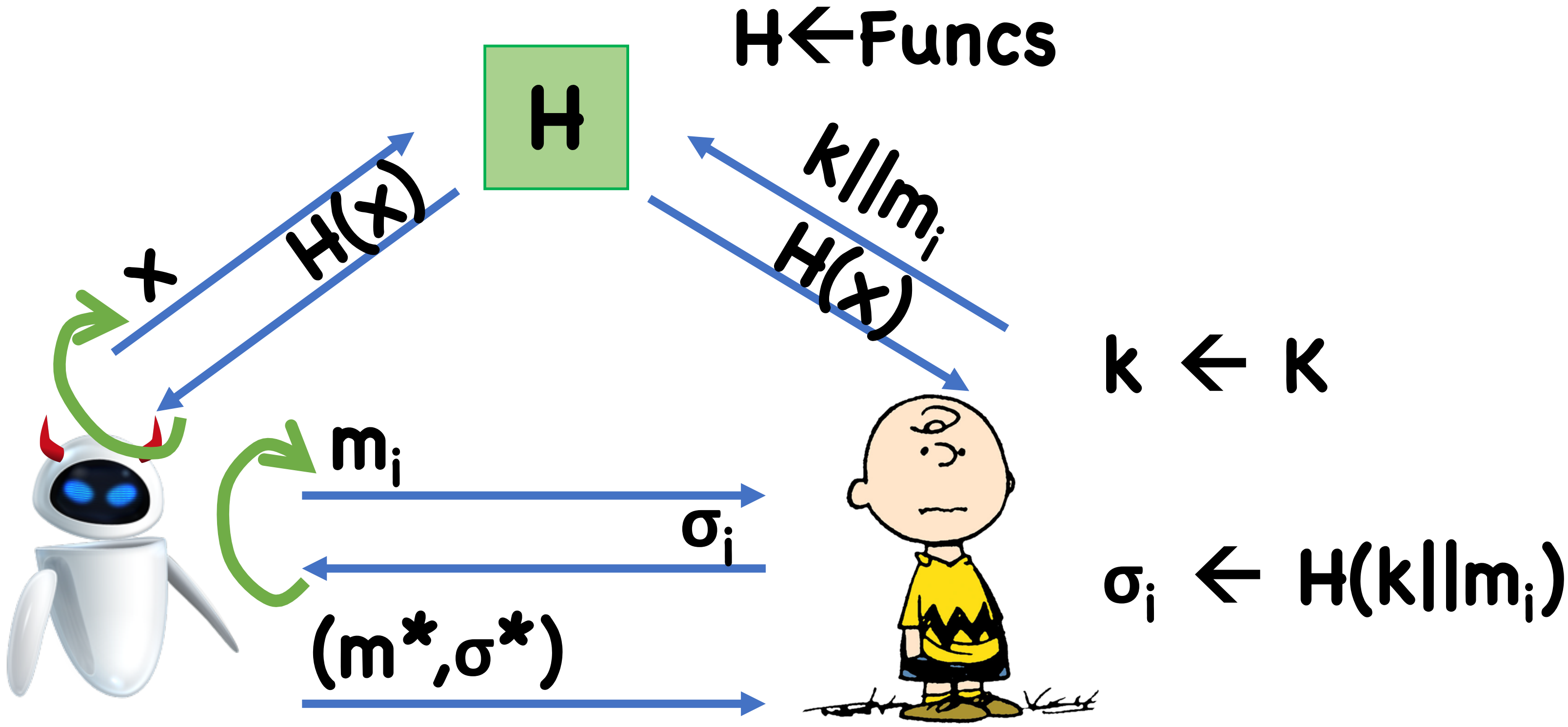
Meaning



Output 1 iff:

- $m^* \notin \{m_1,\ldots\}$
- $Ver^H(k,m^*,\sigma^*)=1$

Meaning



Output 1 iff:

- $m^* \notin \{m_1, \ldots\}$
- $H(k||m^*) == \sigma^*$

The ROM

A random oracle is a good

- PRF: $F(k,x) = H(k||x)$

- PRG (assuming H is expanding):
  - Given a random x, $H(x)$ is pseudorandom since adv is unlikely to query H on x

- CRHF:
  - Given poly-many queries, unlikely for find two that map to same output

The ROM

The ROM is very different from security properties like collision resistant

What does it mean that “Sha-1 behaves like a random oracle”?

- No satisfactory definition

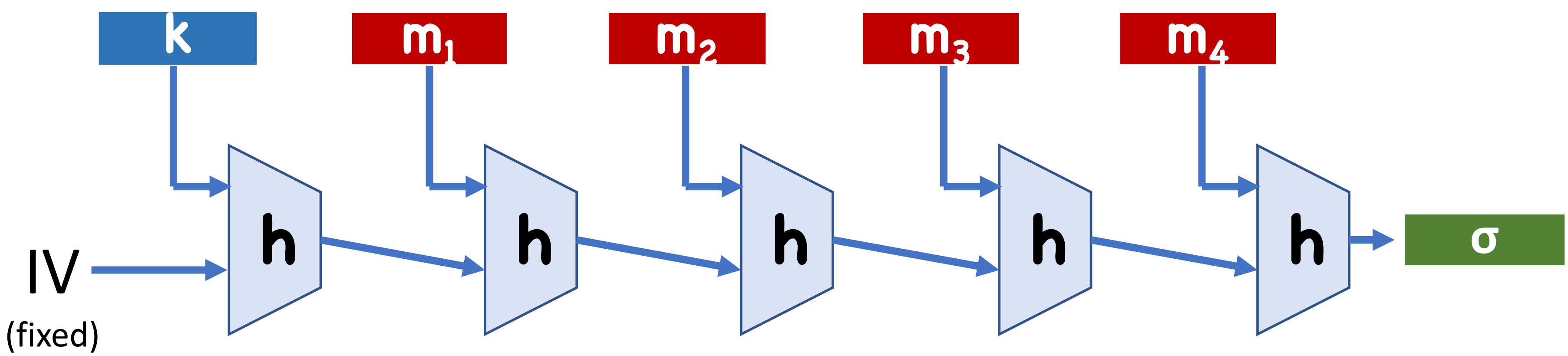
Therefore, a ROM proof is a heuristic argument for security

- If insecure, adversary must be taking advantage of structural weaknesses in H

When the ROM Fails

$\mathrm{MAC}^H(k,m) = H(k||m)$
$\mathrm{Ver}^H(k,m,\sigma) = (H(k||m) == \sigma)$

Instantiate with Merkle-Damgard (variable length)?



When the ROM Fails

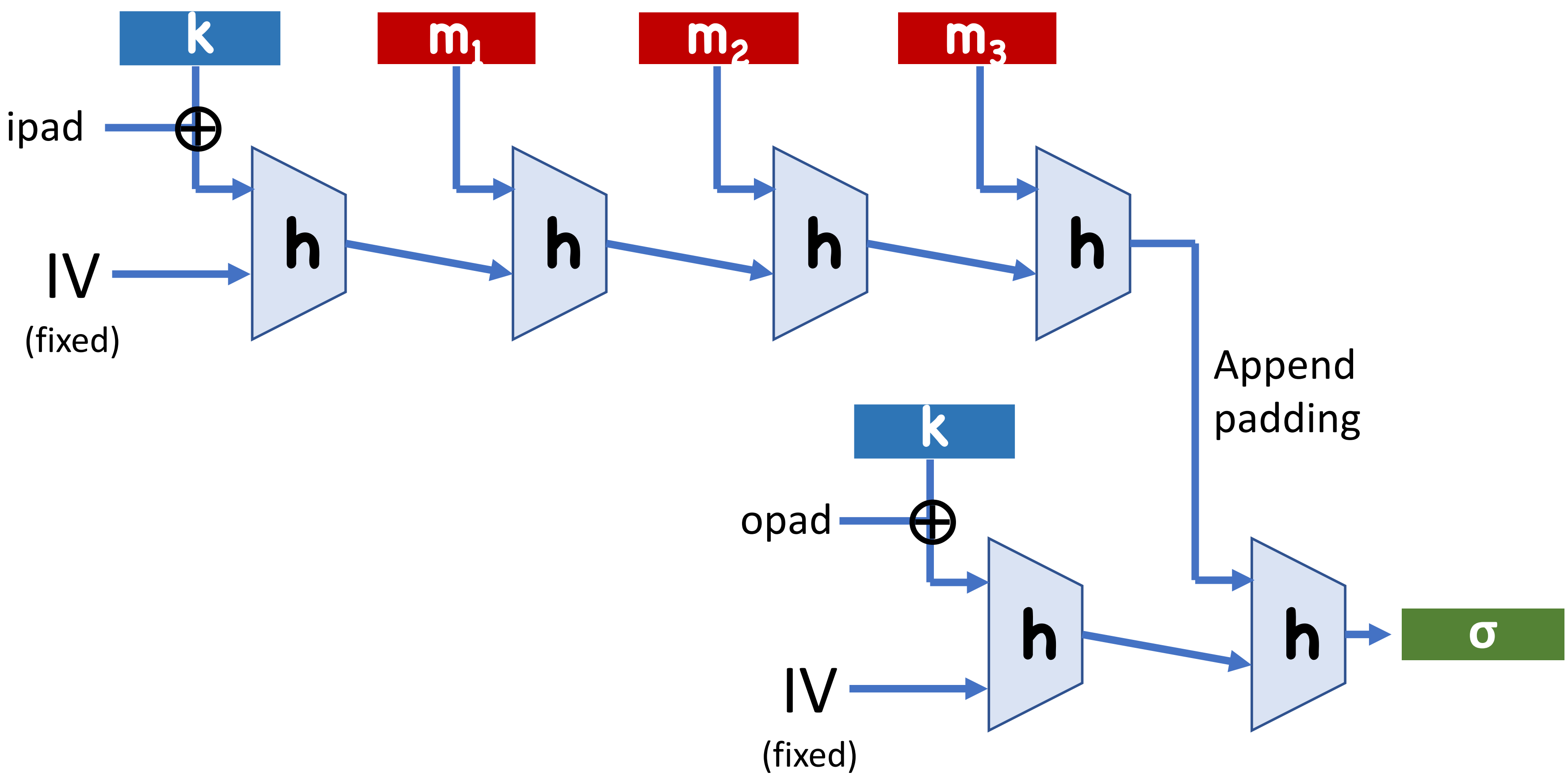
ROM does not apply to regular Merkle-Damgard

- Even if **h** is an ideal hash function

Takeaway: be careful about using ROM for non-"monolithic" hash functions

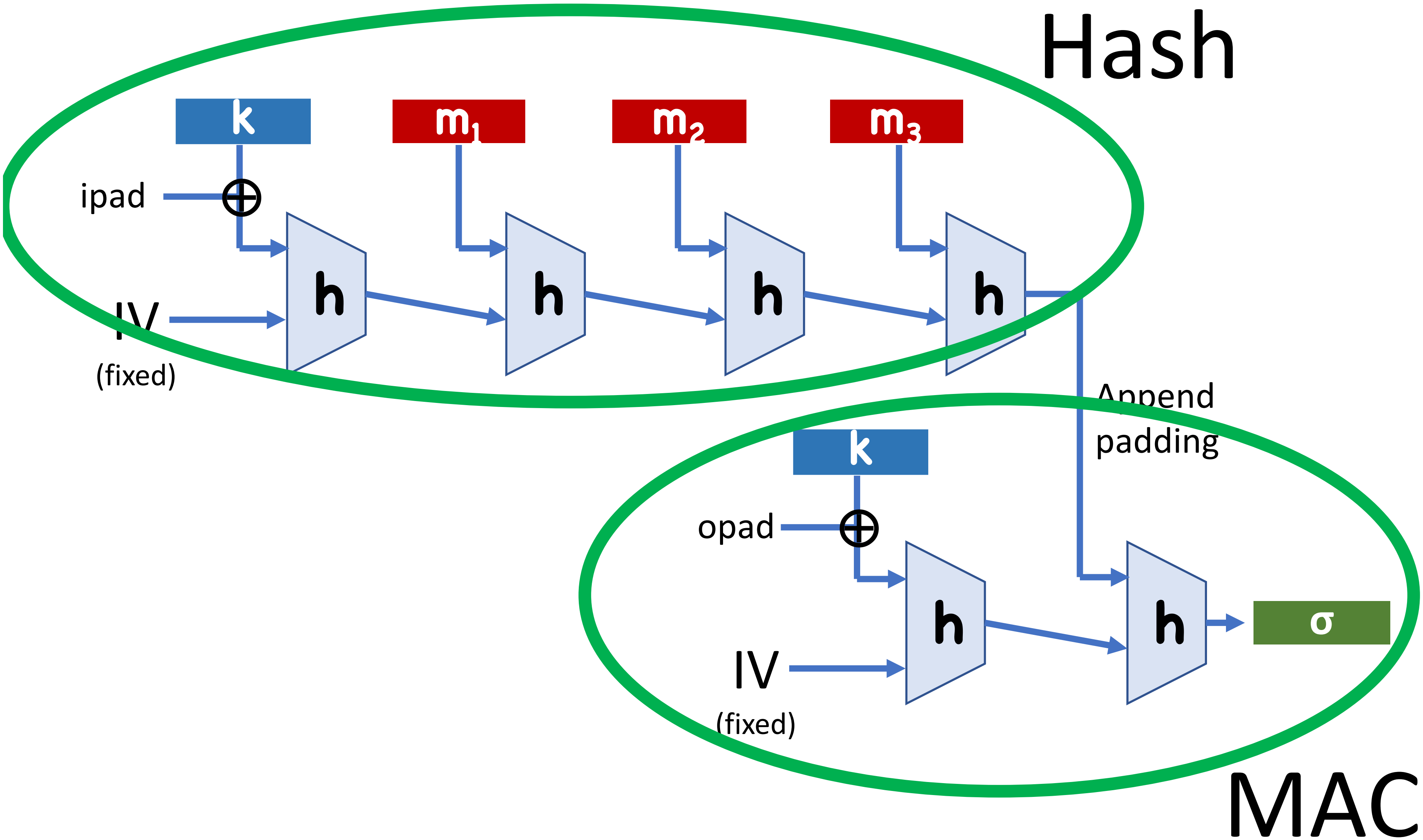
- Though still possible to pad MD in a way that makes it an ideal hash function if **h** is ideal

HMAC



k
m_1
m_2
m_3
ipad
IV
(fixed)
h
h
h
h
Append
padding
k
opad
h
h
σ
IV
(fixed)

HMAC



Hash
k
m1
m2
m3
ipad
h
h
h
h
IV
(fixed)
Append
padding
k
opad
h
h
σ
IV
(fixed)
MAC

HMAC

ipad,opad?

- Two different (but related) keys for hash and MAC
- ipad makes hash a “secret key” hash function
- Even if not collision resistant, maybe still impossible to find collisions when hash key is secret
- Turned out to be useful after collisions found in MD5

Committments

Anagrams and Astronomy

<u>Galileo and the Rings of Saturn</u>

- 1610: Galileo observed the rings of Saturn, but mistook them for two moons

- Galileo wanted extra time for verification, but not to get scooped

- Circulates anagram
  SMAISMRMILMEPOETALEUMIBUNENUGTTAUIRAS

- When ready, tell everyone the solution:
  altissimum planetam tergeminum observavi
  (“I have observed the highest planet tri-form”)

Anagrams and Astronomy

<u>Enter Huygens</u>

- 1656: Realizes Galileo actually saw rings
- Circulates

AAAAAAA CCCCC D EEEEE G H IIIIIII LLLL MM NNNNNNNNN OOOO PP Q RR S TTTTT UUUUU

- Solution:

annulo cingitur, tenui, plano, nusquam cohaerente, ad eclipticam inclinato

("it is surrounded by a thin flat ring, nowhere touching, and inclined to the ecliptic")

Commitment Scheme

Different than encryption

- No need for a decryption procedure
- No secret key
- But still need secrecy (“hiding”)
- Should only be one possible opening (“binding”)
- (Sometimes other properties needed as well)

Anagrams are Bad Commitments

If too short (e.g. one, two, three words), possible to reconstruct answer

- Even easier if have reasonable guess for answer

If too long, multiple possible solutions

- Kepler tries to solve Galileo's anagram as

`salue umbistineum geminatum martia proles`

(hail, twin companionship, children of Mars)

(Non-interactive) Commitment Syntax

Message space $\mathcal{M}$
Ciphertext Space $\mathcal{C}$
(suppressing security parameter)

Com(m; r): outputs a commitment **c** to **m**

- Why have **r**?

Commitments with Setup

Message space **M**
Ciphertext Space **C**
(suppressing security parameter)

Setup(): Outputs a key **k**
Com(k, m; r): outputs a commitment **c** to **m**

Using Commitments

Commit Stage

$r \leftarrow R$

$c \leftarrow Com(m;r)$

m

Reveal Stage

m, r

Check that

$c = Com(m;r)$

Using Commitments (with setup)

k←Setup()

Commit Stage

r←R

c←Com(k,m;r)

m

Reveal Stage

m,r

Check that

c = Com(k,m;r)

Security Properties

Hiding: **c** should hide **m**

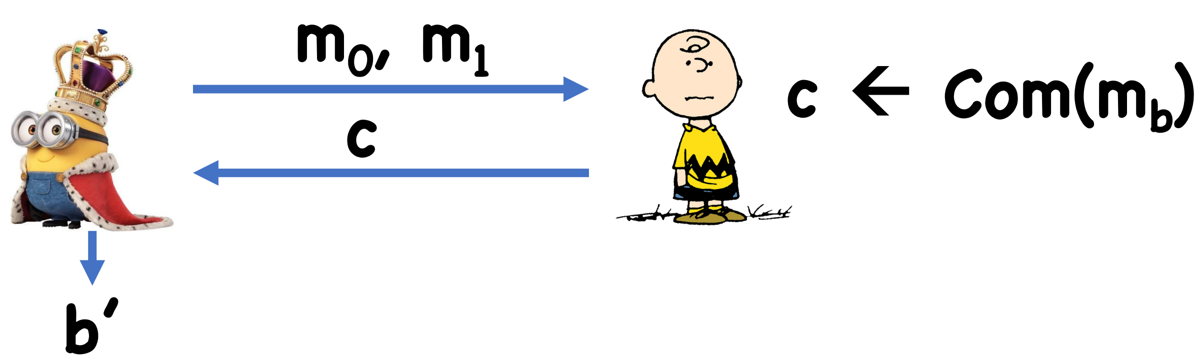
- Perfect hiding: for any $\mathbf{m_0}$, $\mathbf{m_1}$,

$$\mathrm{Com}(m_0) \stackrel{d}{=} \mathrm{Com}(m_1)$$

- Statistical hiding: for any $\mathbf{m_0}$, $\mathbf{m_1}$,

$$\Delta(\ \mathrm{Com}(m_0),\ \mathrm{Com}(m_1)\ ) < \mathrm{negl}$$

- Computational hiding:



Security Properties (with Setup)

Hiding: **c** should hide **m**

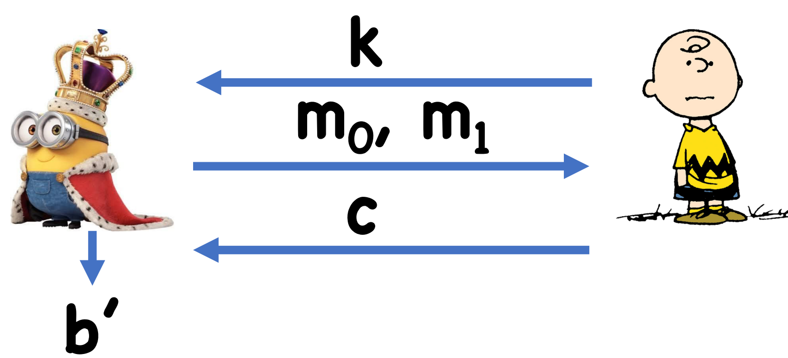
- Perfect hiding: for any m_0, m_1,

$$k, \mathrm{Com}(k, m_0) \stackrel{d}{=} k, \mathrm{Com}(k, m_1)$$

- Statistical hiding: for any m_0, m_1,

$$\Delta(\ [k, \mathrm{Com}(k, m_0)],\ [k, \mathrm{Com}(k, m_1)]\ ) < \mathrm{negl}$$

- Computational hiding:



$c \leftarrow \mathrm{Com}(k, m_b)$

Security Properties

Binding: Impossible to change committed value

- Perfect binding: For any $\mathbf{c}$, $\exists$ at most a single $\mathbf{m}$ such that $\mathbf{c = Com(m;r)}$ for some $\mathbf{r}$

- Computational binding: no efficient adversary can find $\mathbf{(m_0,r_0),(m_1,r_1)}$ such that:

  $\mathbf{Com(m_0;r_0)=Com(m_1;r_1)}$

  $\mathbf{m_0 \neq m_1}$

Security Properties (with Setup)

Binding: Impossible to change committed value

- Perfect binding: For any **k,c**, $\exists$ at most a single **m** such that **c = Com(k,m;r)** for some **r**

- Statistical binding: except with negligible prob over **k**, for any **c**, $\exists$ at most a single **m** such that **c = Com(k,m;r)** for some **r**

- Computational binding: no PPT adversary, given **k←Setup()**, can find $(m_0,r_0),(m_1,r_1)$ such that

  $\mathrm{Com}(k,m_0;r_0)=\mathrm{Com}(k,m_1;r_1)$

  $m_0 \neq m_1$

Who Runs **Setup()**

Alice?

- Must ensure that Alice cannot devise **k** for which she can break binding

Bob?

- Must ensure Bob cannot devise **k** for which he can break hiding

Solution: Trusted third party (TTP)

Announcements/Reminders

HW3 due on Oct 20

HW4 will be released today, due Oct 27