TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

PAGE:

Definition: A cryptographic hash function

$$h: \{0,1\}^* \longrightarrow \{0,1\}^d$$

maps bit-strings of arbitrary length

to bit strings of length d

in an efficient, deterministic, public,

"random" manner.

- Notes: "message digest", "fingerprint"

  no secret key

  Examples:

MD5	d=128
SHA-256	d=256
SHA3-256	d=256
SHAKE	d variable

- (Ideal) Random Oracle Model (ROM)

  h(x): if x already seen
  give same output as before
  else
  flip coin d times & give result as output

  (All parties have access to same random oracle.)

  Typically: show scheme works in ROM
  replace RO by SHA-256 & hope it works!

  "pseudorandomness" — PR

TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

PAGE:

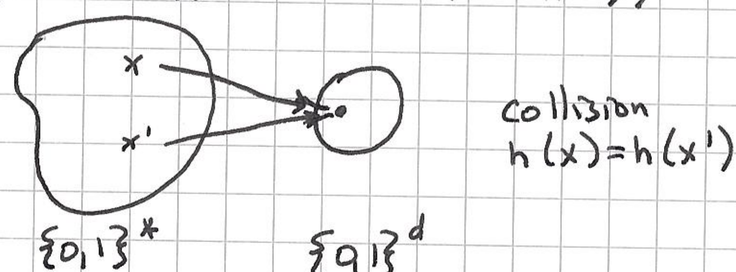
Properties

Definitional style:

1. Add "salt" or key, to make hash function family $\mathcal{H} = \{h_s : s \in \{0,1\}^k\}$
2. Let h be a single hash function, assume no violations of property occur in practice.

Collision-Resistance (CR)

"Infeasible" to find a "collision" (an x, x' s.t. $x \neq x'$ & $h(x) = h(x')$)



collisions exist; you just can't find one

Can find one in time $O(2^{d/2})$ – "birthday paradox" (ROM)
[& using $O(1)$ storage – Floyd's "two finger" algorithm]

∴ can use $h(x)$ as unique representative or proxy for x

Target collision-resistance (TCR)

"Infeasible", given x, to find $x' \neq x$ s.t. $h(x') = h(x)$.

Time is $O(2^d)$ in ROM.

Thm: CR ⇒ TCR

TOPIC: 6.857 - Hash Functions

DATE:

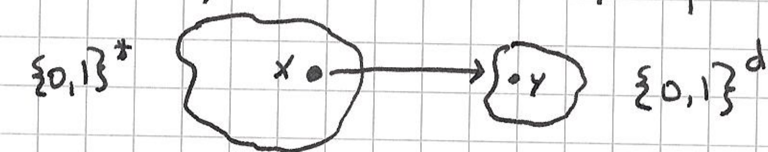
FILE UNDER:

PAGE:

Properties (cont.)

One-wayness (OW)

"Infeasible", given y (image under h of random x), to find any x' s.t. $h(x') = y$ (pre-image of y)



Brute-force in ROM takes $O(2^d)$ trials.

(OW different than CR!)

TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

PAGE:

Applications:

① Password "storage" (for login)

- Store h(PW), not PW, on computer
- At login, check hash of PW against stored h(PW)

- Disclosure of h(PW) should not reveal PW (or anything usable to log in)

- Need: OW

② File modification detector

- For each file F, store h(F) securely
- Before using F, recompute h(F) & check against stored value.

- Need: TCR

- Equivalent problem: hashes of downloadable software

- But: can't have machine "check itself"!

TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

PAGE:

Applications (cont.)

(3) Digital signatures ("hash & sign")

PK_A = Alice's public key (for signature verification)

SK_A = Alice's secret key (for signing)

Signing: $\sigma = \text{Sign}(SK_A, M)$ [Alice's sig on M]

Verify: $\text{Verify}(M, \sigma, PK_A) \in \{\text{True}, \text{False}\}$

Adversary wants to forge a signature that verifies.

- For large messages M, easier to sign h(M):

$\sigma = \text{Sign}(SK_A, h(M))$ ["hash & sign"]

Verifier recomputes h(M) from M, then verifies σ.

h(M) is a "proxy" for M.

- Need: CR [Else Alice finds collision x, x'

gets Bob to sign x (where $h(x) = h(x')$)

then claims Bob really signed x', not x.]

- Don't need OW: (e.g. using h = identity fn is OK.)

TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

PAGE:

(4)

MAC (Message Authentication Code)

MAC(K, M) needs shared key K

How to insert secret key K into hash function?

$h(K \| M)$ - does this work?

↑ concatenation

For <u>some</u> h, doesn't work (extension attacks!)

Fixes:
- $HMAC(K,M) = h(K \oplus pad_2, h(K \oplus pad_1, M))$

  <u>or</u>
- Use SHA3 for h (<u>sponge</u> construction!)

TOPIC: 6.857 Hash Functions

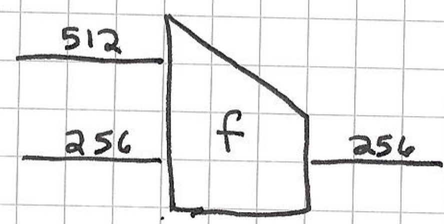
DATE:

FILE UNDER:

PAGE:

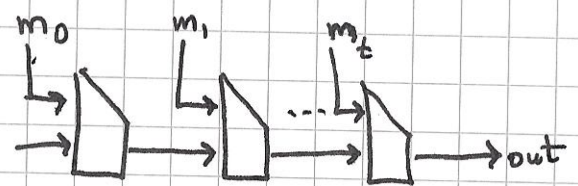
Constructions of hash functions

① Merkle-Damgard construction (e.g. SHA-256) (e.g. MD5)



f has 64 rounds on 8 32-bit words of state

OW, CR



m_t includes padding

10*L (L: 64 bit length of M)

(vulnerable to length-extension attacks if used as a MAC)

great as a hash function

Construction of hash functions

(2) "sponge" construction (e.g. Keccak, SHA3, SHAKE)

water ≈ randomness (entropy)

sponge can absorb (input keys or message)
" " be squeezed (output PR bit string)

Can be used as MAC: absorb key
absorb message
squeeze out MAC

Can be used as PRF: absorb key
Squeeze out PR stream (as long as wanted)

Can be used as hash fn: absorb message
squeeze out hash value

stateful (e.g. for SHA-3)

state = 25 words of 64 bits (1600 bits total)

8 word "capacity" (hidden state; adversary can't see or touch)
$c = 512$

17 word "rate" (visible state; adversary can see and/or modify)
$r = 1088$

Thm: Security (for CR & OW) is $c/2$ bits

(e.g. 256-bits for SHA-3)

TOPIC: 6.857 Hash Functions

DATE:

FILE UNDER:

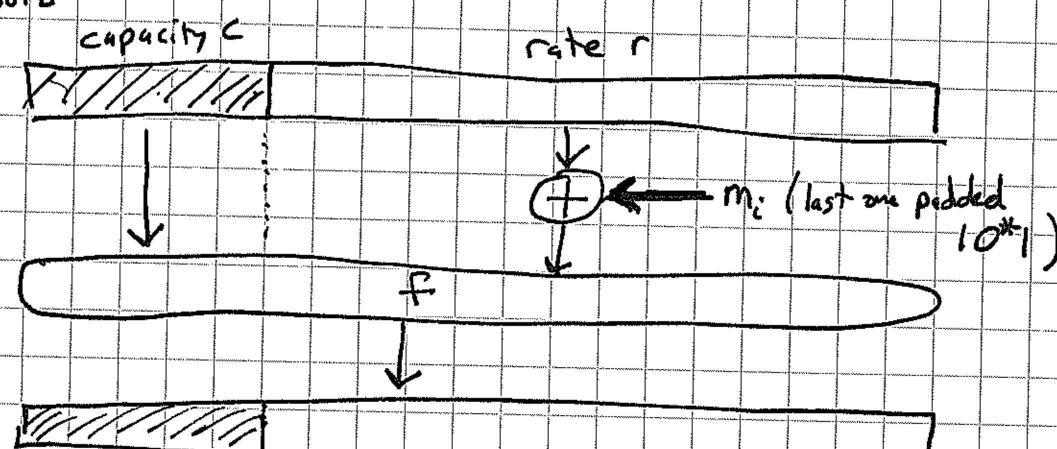
PAGE:

(2) (continued) sponge construction

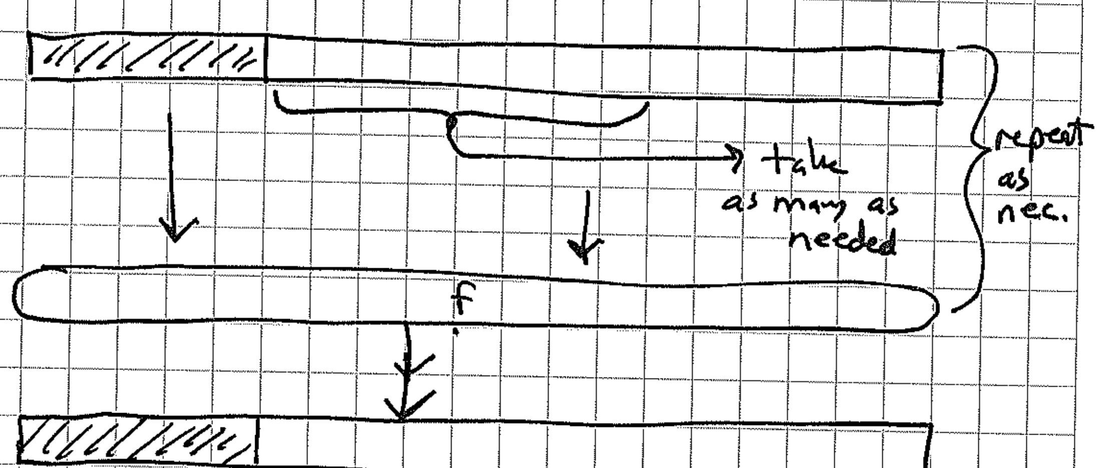
f = fixed, public function that maps state S 1-to-1 to a new value (permutation of $\{0,1\}^{1600}$) (f is keyless)

- scrambles 1600 bits of state so every bit of output depends in a complicated way on every bit of input
- AND, OR, NOT, rotate instructions
- 24 "rounds" - slightly different constant

absorb



Squeeze



TOPIC: 6.857 Applied Cryptography & Security

DATE: 2/22/2022

FILE UNDER:

PAGE:

Admin:
- Psets (#1 due, #2 out)
- Guest speaker Wednesday – Andy Sellars

Today: (Cryptographic) Hash Functions
- Definition & Random Oracle Model
- Properties: collision-resistance, one-wayness
- Applications
- Constructions

Readings:
Katz / Lindell (2nd ed) – Chapter 5
Paar / Pelzl – Chapter 11
Ferguson / Schneier / Kohno – Chapter 5