

Admin:

Pset #1 due Mon 2/22 [submit separate pdf for each problem.]

Pset #2 out Mon 2/22.

Today:

Cryptographic Hash Functions

Definition

Random Oracle Model

Properties: OW, CR, TCR, PR, NM

Applications

Readings:

Katz/Lindell Chapter 5

Paar/Pelzl Chapter 11

Ferguson/Schneier/Kohno Chapter 5

Cryptographic hash functions

"Def": A cryptographic hash function maps
a domain D (bit strings of arbitrary length)
to a range R (bit strings of length d , or
bit strings of arbitrary length)

$$h: \underbrace{\{0,1\}^*}_{\text{all strings (of any length } \geq 0)} \rightarrow \underbrace{\{0,1\}^d}_{\text{all strings of length } d}$$

in an efficient, deterministic, public, "random" manner.

Sometimes called a "message digest function".

No secret key. Anyone can compute h from its published description.

Examples: MD5 $d=128$

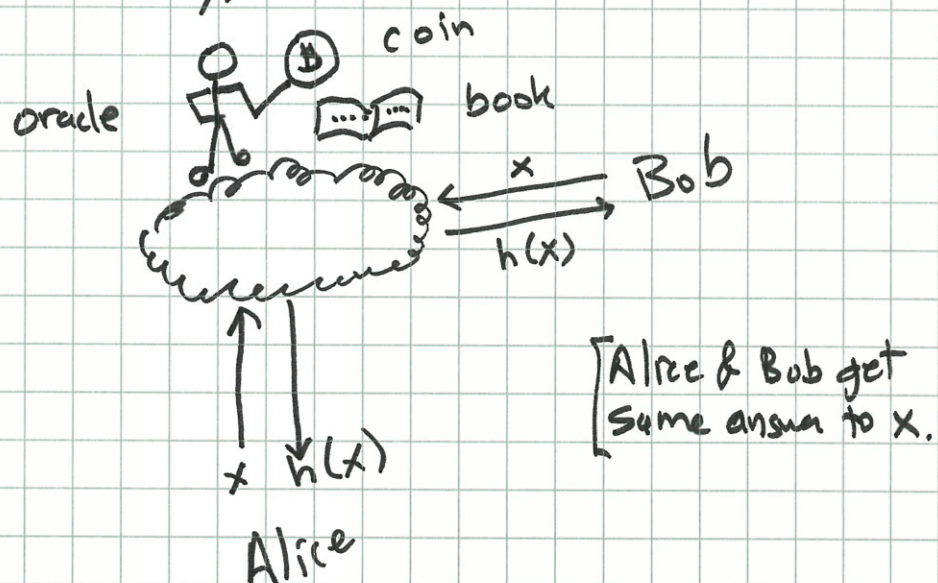
NIST $\left\{ \begin{array}{ll} \text{SHA-256} & d=256 \quad (\text{also SHA-512}) \\ \text{SHA3-256} & d=256 \quad (\text{Keccak}) \\ \text{SHAKE256(512)} & d=512 \quad (\text{Keccak}) \end{array} \right.$

$d=128, 160, 224, 256, 384, 512$ Common

VIL = Variable input length FIL = fixed input length
VOL = Variable output length FOL = fixed output length

An ideal hash function: a "Random Oracle" (RO)

- Theoretical model - good intuitive guidance, but not achievable in practice
- Oracle ("in the sky")
 - receives input x and returns $h(x)$
for any $x \in \{0,1\}^*$. $|h(x)| = d$ bits.
 - On input x :
 - if x not in book:
flip coin d times to determine $h(x)$
record pair $(x, h(x))$ in book
 - else return y where (x, y) in book
- Gives random answers, but use of book ensures consistency.



Random Oracle Model

Many crypto schemes proved secure in ROM ("Random Oracle Model") which assumes existence of RO.

Then RO is replaced by hash function (e.g. SHA-256) in practice, which is hopefully "pseudorandom enough" that ~~an~~ adversary can't exploit any flaws in SHA-256.

OW

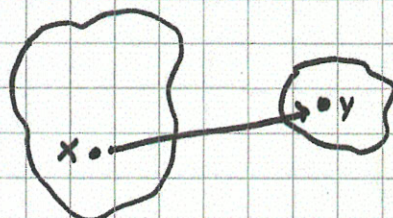
 y computed:

$$x \in_R \{0,1\}^*$$

$$y = h(x)$$

Hash function desirable properties:① "One-way" (pre-image resistance)

"Infeasible", given $y \in \{0,1\}^d$ to find
any x' s.t. $h(x') = y$ (x' is a "pre-image" of y)



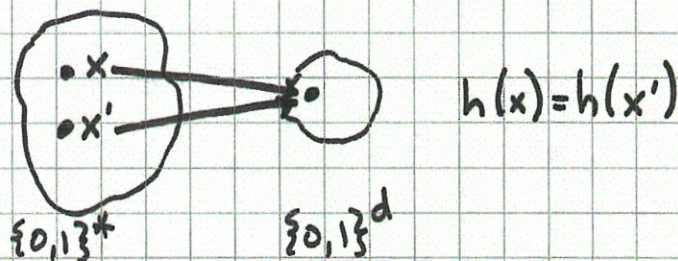
$$h: \{0,1\}^* \rightarrow \{0,1\}^d$$

(Note that a "brute-force" approach of trying
 x 's at random requires $\Theta(2^d)$ trials (in ROM).)

CR

② "Collision-resistance" (strong collision resistance)

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



(In ROM, requires trying about $2^{d/2}$ x 's
 $(x_1, x_2, \dots)$ before a pair x_i, x_j colliding is
 found. (This is the "birthday paradox".))

Note that collisions are unavoidable since

$$|\{0,1\}^*| = \infty$$

$$|\{0,1\}^d| = 2^d$$

Birthday paradox detail:

If we hash $x_1, x_2, \dots, x_n$ (distinct strings)

then

$$\begin{aligned} E(\# \text{ collisions}) &= \sum_{i \neq j} \Pr(h(x_i) = h(x_j)) \\ &= \binom{n}{2} \cdot 2^{-d} \quad [\text{if } h \text{ "uniform"}] \\ &\approx \frac{n^2 \cdot 2^{-d}}{2} \end{aligned}$$

This is ≥ 1 when $n \geq 2^{(d+1)/2} \approx 2^{d/2}$

The birthday paradox is the reason why hash function outputs are generally twice as big as you might naively expect; you only get 80 bits of security (w.r.t. CR) for a 160-bit output.

With some tricks, memory requirements can be dramatically reduced.

TCR

③ "Weak collision resistance" (target collision resistance, and pre-image resistance)

"Infeasible", given $x \in \{0,1\}^*$, to find $x' \neq x$ s.t. $h(x) = h(x')$.

Like CR, but one pre-image given & fixed.

(In ROM, can find x' in time $\Theta(2^d)$
(as for OW, since knowing x doesn't help in ROM to find x').

PRF

④ Pseudo-randomness

" h is indistinguishable under black-box access from a random oracle"

(To make this notion workable, really need a family of hash functions, one of which is chosen at random. A single, fixed, public hash function is easy to identify...

NM

⑤ Non-malleability

"Infeasible", given $h(x)$, to produce $h(x')$ where x and x' are "related" (e.g. $x' = x + 1$).

These are informal definitions...

Theorem: If h is CR, then h is TCR.
(But converse doesn't hold.)

Theorem: h is OW $\nleftrightarrow$ h is CR
(neither implication holds)
But if h "compresses", then $CR \Rightarrow OW$.

Hash function applications

- ① Password storage (for login)
 - Store $h(PW)$, not PW , on computer
 - When user logs in, check hash of his PW against table.
 - Disclosure of $h(PW)$ should not reveal PW (or any equivalent pre-image)
 - Need OW
- ② File modification detector
 - For each file F , store $h(F)$ securely (e.g. on off-line DVD)
 - Can check if F has been modified by recomputing $h(F)$
 - need WCR (aka TCR)
(Adversary wants to change F but not $h(F)$.)
 - Hashes of downloadable software = equivalent problem.

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 M , easier to sign $h(M)$:

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

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

In essence, $h(M)$ is a "proxy" for M .

- Need CR (Else Alice gets Bob to sign x , where $h(x) = h(x')$, then claims Bob really signed x' , not x .)

- Don't need OW (e.g. $h = \text{identity}$ is OK here.)

④ Commitments

- Alice has value x (e.g. auction bid)
- Alice computes $C(x)$ ("commitment to x ") & submits $C(x)$ as her "sealed bid"
- When bidding has closed, Alice should be able to "open" $C(x)$ to reveal x
- Binding property: Alice should not be able to open $C(x)$ in more than one way! (She is committed to just one x .)
- Secrecy (hiding): Auctioneer (or anyone else) seeing $C(x)$ should not learn anything about x .
- Non-malleability: Given $C(x)$, it shouldn't be possible to produce $C(x+1)$, say...

How:

$$C(x) = h(r \| x) \quad r \in_R \{0,1\}^{256}$$

To open: reveal r & x

- Note that this method is randomized (as it must be for secrecy).

• Need: OW, CR, NM

(really need more, for secrecy, as $C(x)$ should not reveal partial information about x , even.)