Intro to Cryptography : Hashing (Part I of II)

Hash functions
Random oracle model
Desirable properties
Applications to Security

Hash Functions

A hash function maps arbitrary strings of data to fixed length output in deterministic, public, "random" manner.

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

strings of arbitrary length ≥ 0 → strings of length d

Hash Functions

No secret key. All operations public.
Anyone can compute h, poly time computation

Examples:	MD4, MD5,	SHA-1,	SHA-256,	SHA-512
d :	128	160	256	512
broken (CR) :	2^6 ✓ 2^{37}	✓? 2^{69}		

Ideal: Random Oracle

(not achievable in practice)

Oracle: on input $x \in \{0,1\}^*$
if x not in book
flip coin d times to determine $h(x)$
record $(x, h(x))$ in book
else: return y where $(x,y) \in$ book

Gives random answer every time, except as required for consistency with previous answers. (h must be deterministic)

In practice, $\nexists$ RO so need something "pseudo random"

Desirable Properties

OW ① "one-way" (pre-image resistance)

Infeasible, given $y \in_R \{0,1\}^d$ to find any x s.t. $h(x) = y$

(x ← "pre-image" of y)

CR ② Collision-resistance (strong collision resistance)

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

TCR ③ Weak collision resistance (target CR, 2nd pre-image resistance)

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

PRF ④ Pseudo-randomness

Behavior indistinguishable from RO

NM ⑤ Non-malleability

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

Informal definitions. Formal requires family of hash functions

Facts

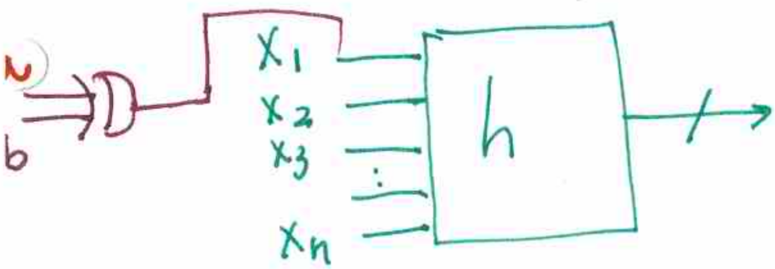
h is CR $\Rightarrow$ h is TCR (but not reverse)

h is OW $\not\Leftrightarrow$ h is CR, TCR (neither impl. holds)

Collisions can be found in $O(2^{d/2})$ – birthday attack

Inversion can be found in $O(2^d)$

Examples



h(x) is OW, CR

$h'(a, b, x_2, \ldots x_n)$ is still OW, but not TCR

OW $\not\Rightarrow$ TCR

$$h'(x) = \begin{cases} 0 \,\|\, x & \text{if } |x| \le n \\ 1 \,\|\, h(x) & \text{otherwise} \end{cases}$$

h is OW, CR, but h' is TCR, not OW

TCR $\not\Rightarrow$ OW

Applications

(1) Password storage

- Store $h(PW)$, not PW, on computer
- Use $h(PW)$ to compare against $h(PW')$ where PW' is the typed password

Disclosure of $h(PW)$ should not reveal PW

Need OW.

(2) File modification detector

- For each file F, store $h(F)$ securely (on DVD)
- check if F modified by recomputing $h(F)$
- need TCR (adversary wants to change F but not $h(F)$)

(3) Digital signatures

PK_A : Alice's Public key

SK_A = Alice's Private key

Signing: $\sigma = sign(SK_A, M)$

Verify: $verify(M, \sigma, PK_A) = true/false$

Adversary wants to forge a signature that verifies

For large M, easier to sign $h(M)$ $\sigma = sign(SK_A, h(M))$

Need CR, don't need OW.

Alice gets Bob to sign x, then claims he signed x', if ($h(x) = h(x')$)

Applications (contd.)

4 Commitments

Alice has value x (e.g., auction bid)
Alice then computes $C(x)$ "commitment to x" and submits it as her bid

$C(x)$ is her "sealed bid"
When bidding is over, Alice "opens" $C(x)$ to reveal x

Binding : Alice should not be able to open $C(x)$ in multiple ways.
Secrecy : Auctioneer seeing $C(x)$ should not learn anything about x
NM : Given $C(x)$ shouldn't be possible to produce $C(x+1)$

Need: NM, CR, OW (really need more for secrecy!)

$h'(x) = h(x) \| msb(x)$

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

to open reveal r & x

randomized

This could be OW but expose most significant bit and break secrecy!