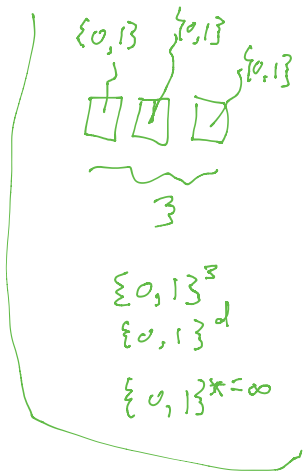
Lecture 2

$y = H(x)$

output $\{0,1\}^d$ ← H ← input/preimage $\{0,1\}^*$

hash function:
* deterministic
* open/public



(1) Pre-image resistance

Result: $d \geq 112$ bits for PR assuming hash is a random mapping

(2) Collision resistance

(3) Weak collision resistance.

An alternative conception

Think of a hash output as a series of coin flips.

heads = 0
tails = 1

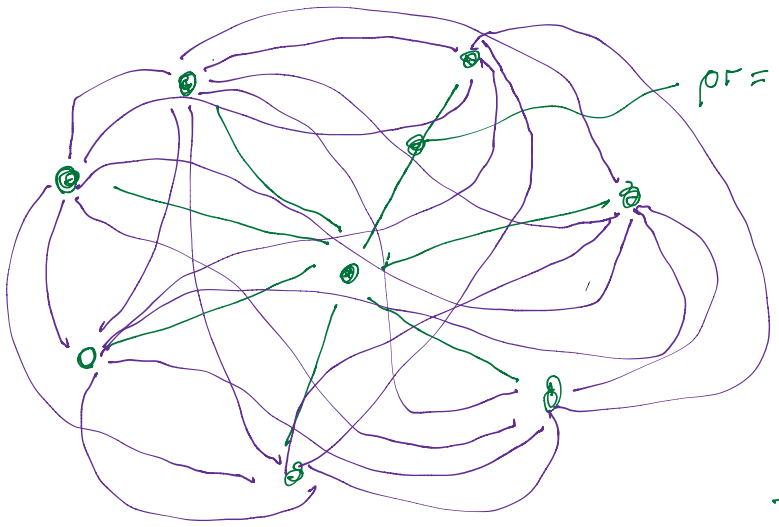
$H(x) =$ □ □ □ □ ... □

d

Collision Resistance (CR)

It is infeasible to find any x and x' $(x \neq x')$ such that $H(x) = H(x')$

Aside: Birthday Paradox

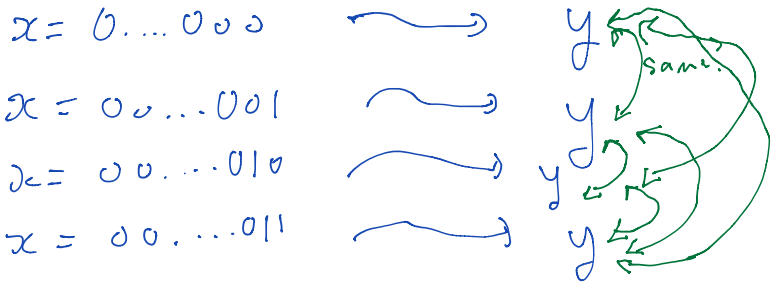


pr = 1/365.

Other People	
7	99
15	4849
22	4948

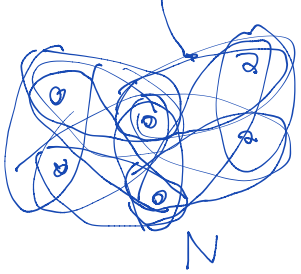
Back to hash functions

* Finding collisions should be much easier than finding pre-images
* Assume: hash is a random mapping
* Find collisions through exhaustive search.
* Pick an arbitrary input, hash it, compare to previous hashes.



* How many inputs (N) do we have to try to see 1 collision (on expectation)

$$E[\#\text{ of collisions}] = (\#\text{ of chances})(\Pr[\text{that one chance is a match}])$$



$$1 = \binom{N}{2}\left(1/2^d\right)$$

$$1 = \frac{N(N-1)}{2}\left(1/2^d\right)$$

$$1 = \frac{N^2-N}{2}\left(2^{-d}\right)$$

$$1 \overset{\text{approx}}{\simeq} N^2\left(2^{-d}\right)$$

$$2^d = N^2$$

$$N = \sqrt{2^d} = 2^{d/2}$$

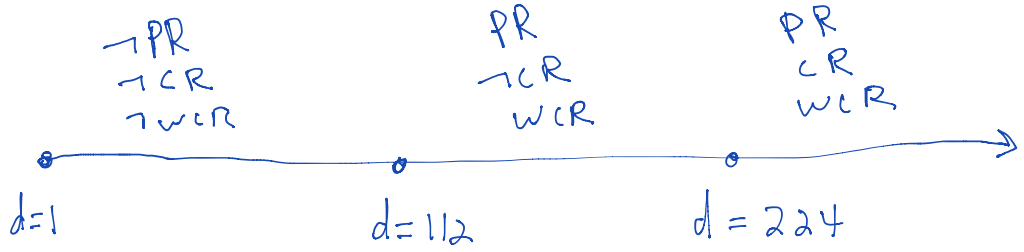
↳ $n = 2^{d/2}$

Secure (infeasible to find a collision) $2^{112} = 2^{d/2}$

$112 = d/2$

$d = 2 \cdot 112$

$= 224$



(3) Weak Collision Resistance (WCR)

↳ Second Pre-image Resistance

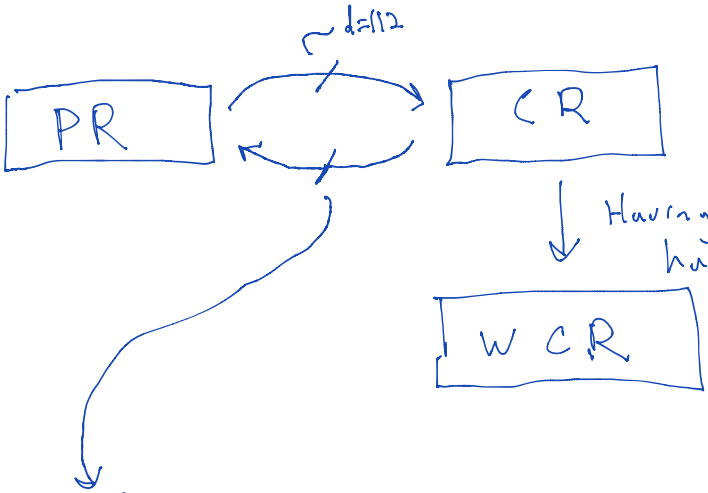
Infeasible, given $x \in \{0,1\}^*$, to find $x' \neq x$ such that $H(x) = H(x')$

* Analysis omitted: $d \geq 112$ bits

Difference between PR and WCR

* In PR: given y, find x
* In WCR: given (x', y), find x $(\neq x')$

Relationship Between Properties



Pathological design of a hash function

* Assume H is a secure hash function (CR & PR)
* Design a new hash function H' that uses H ~PR, CR

$\{0,1\}^{d+1} \rightarrow$ $y = h'(x) = \begin{cases} 0 \,||\, x, & |x| = d \\ 1 \,||\, h(x) & \sim PR \;\&\; CR \end{cases}$

Real Hashes

(92) MD5

(95) SHA 1

(2001) SHA 2 → SHA 256
↳ SHA 512

(2015) SHA 3 → SHA3-256
↳ SHA3-514

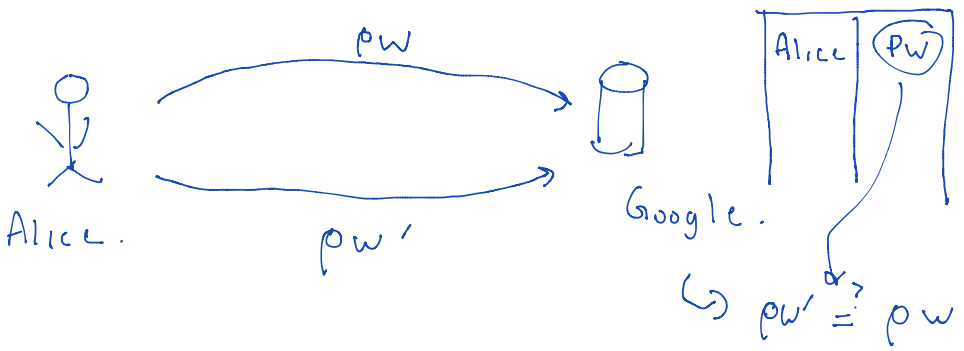
d	PR?	CR?
128	✓	X
160	✓	X
256	✓	✓
512	✓	✓
256	✓	✓
512	✓	✓

Naive collisions take 2^{64}

Only 80-bits of CR

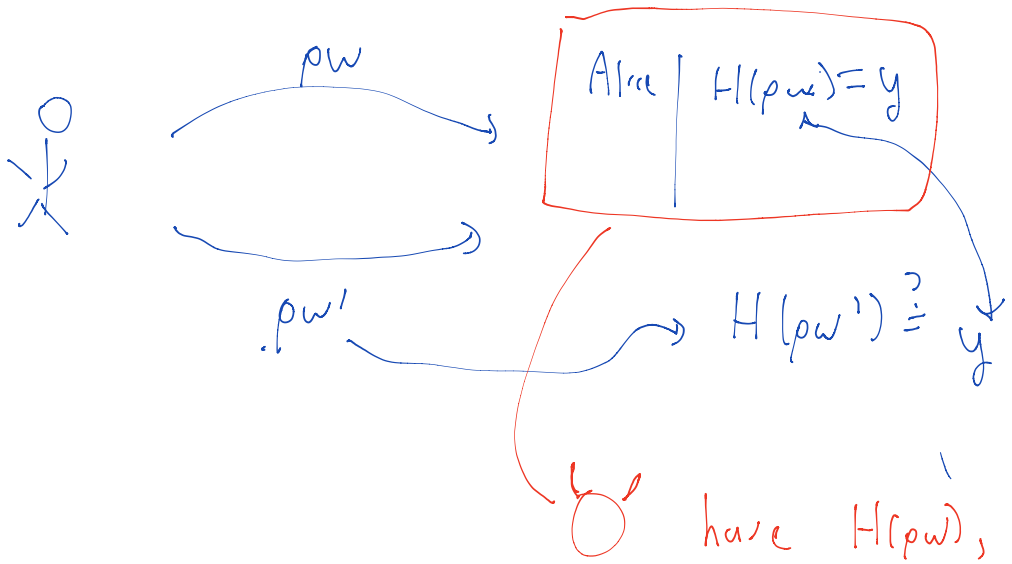
Applications

① Password Storage.



Issue: adversary may steal the password table.

Idea: store H(pw) instead of pw in password table.



have H(pw),
Can they guess pw?
(1) Not directly b/c H is PR

PR
(&WCR)

(2) If you can guess pw, you can confirm if pw is correct by seeing if it matches H(pw)

* You can download the hashes of common passwords in a file (rainbow table)

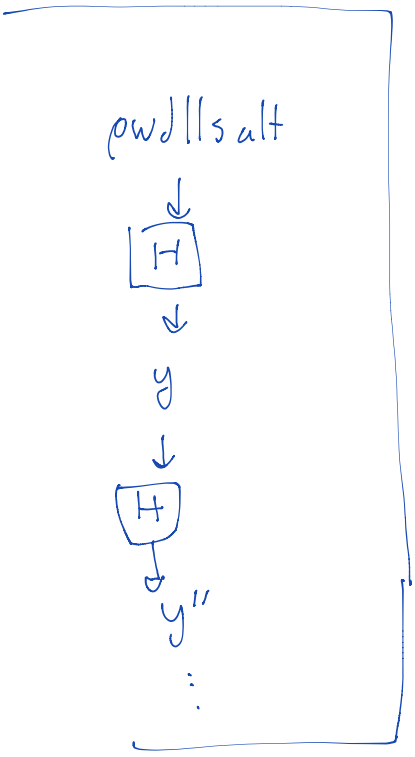
* Make hash function unique per user:
store: $\langle H(pwd \| salt), salt \rangle$ ← not a secret
salt ← random number $\in \{0,1\}^{128}$

* Make hash slower:

Store: $< H^{1000}(pwd||salt), salt >$

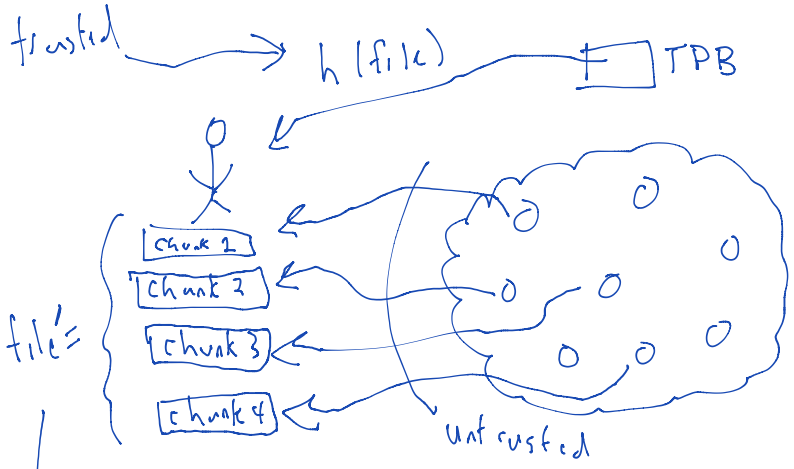
↳ PBKDF2

↳ password key derivation function v2



Example 2:

(Filesharing ~ Bittorrent)



$h(file') \stackrel{?}{=} h(file) = h(chunk1||chunk2||c3||c4)$

Infeasible to send a different (malicious) chunk and have the hash match:

WCR & CR

meaningful collision

Example 3: Hash & Sign Paradigm

↳ see later

↳ C.R.

Note: a secure hash function means PR & CR and technically only that

Other Properties distinct from PR and CR we would expect:

* Non-malleability.: given y where $y = h(x)$ for some unknown x, it should be infeasible to compute $y' = h(f(x))$ for some function f

e.g., $h(x+1)$, $h(2x)$, $h(x||a)$ ← append of your choice

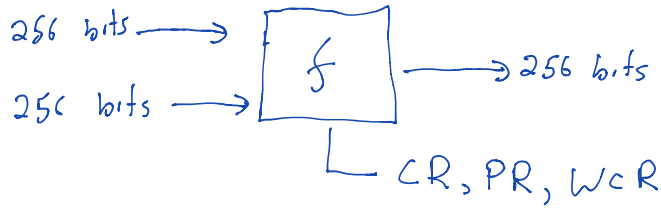
Length extension attacks

↳ MD5, SHA1, SHA2 → Merkle-Damgaard

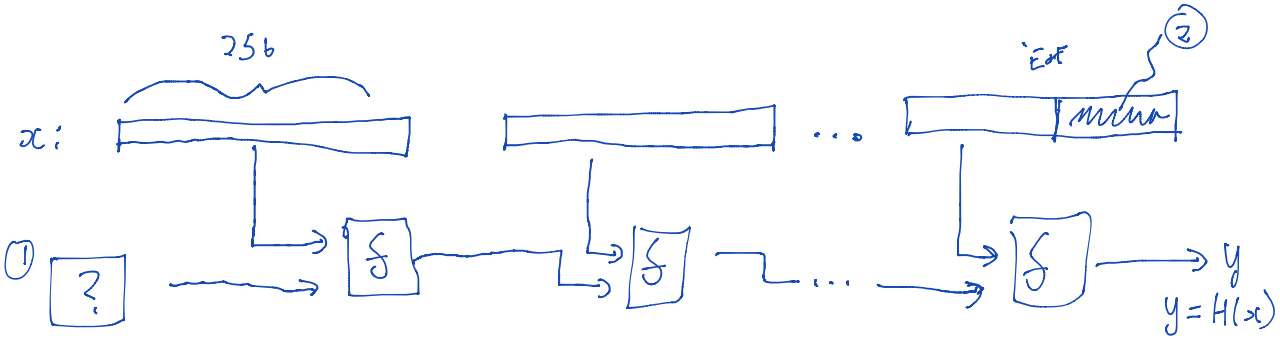
↳ Not to SHA3 → Sponge

Look inside a hash function

① Compression function.



② How do hashes deal with infinite size inputs? They operate chunk-by-chunck.



① Pick a number and fix for the hash function: Initialization Vector
↑ NIST

③ Padding