CSE 539: Applied Cryptography
Lec 8: Crypto Hash Function

Ni Trieu (ASU)

Reading: https://joyofcryptography.com/pdf/chap11.pdf
https://en.wikipedia.org/wiki/Cryptographic_hash_function

Recap: Message Authentication Code (MAC)

- A MAC is like a signature that can be added to a piece of data, which certifies that someone who knows the secret key attests to this particular data

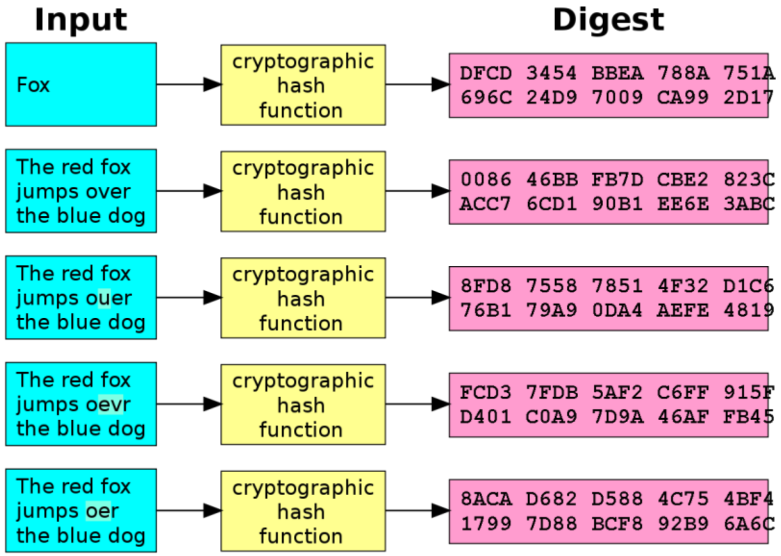
- A MAC scheme is a secure MAC if the adversary knows valid MACs corresponding to various messages, she cannot produce a valid MAC for a different message.

Hash Function

- A hash function maps a message of an arbitrary length to a n-bit output

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$

  - "Randomized" mapping of inputs to a n-bit output
- The output is known as the fingerprint or the message digest



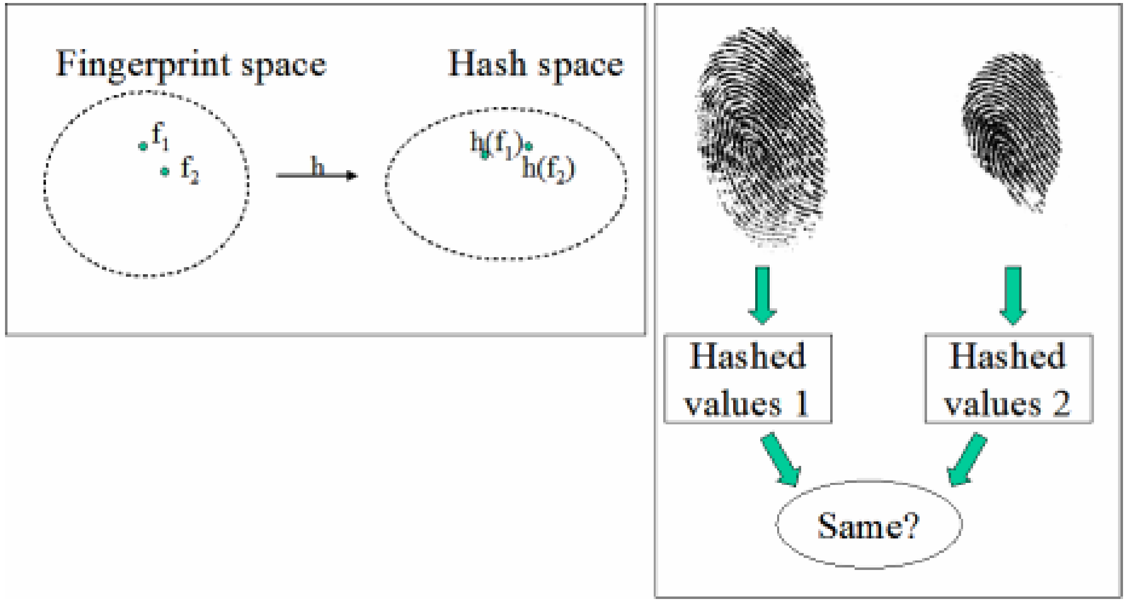
x	$h_1(x)$
000	0
001	0
010	0
011	0
100	1
101	1
110	1
111	1

Hash Function

- A hash function maps a message of an arbitrary length to a n-bit output

$$\boldsymbol{H}: \{\mathbf{0}, \mathbf{1}\}^* \rightarrow \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{n}}$$

- What is an example of hash functions?
  - Give a hash function that maps Strings to integers in [0,2^{32}-1]



Example: Usages of Hash Functions

- In data-structures: for efficiency

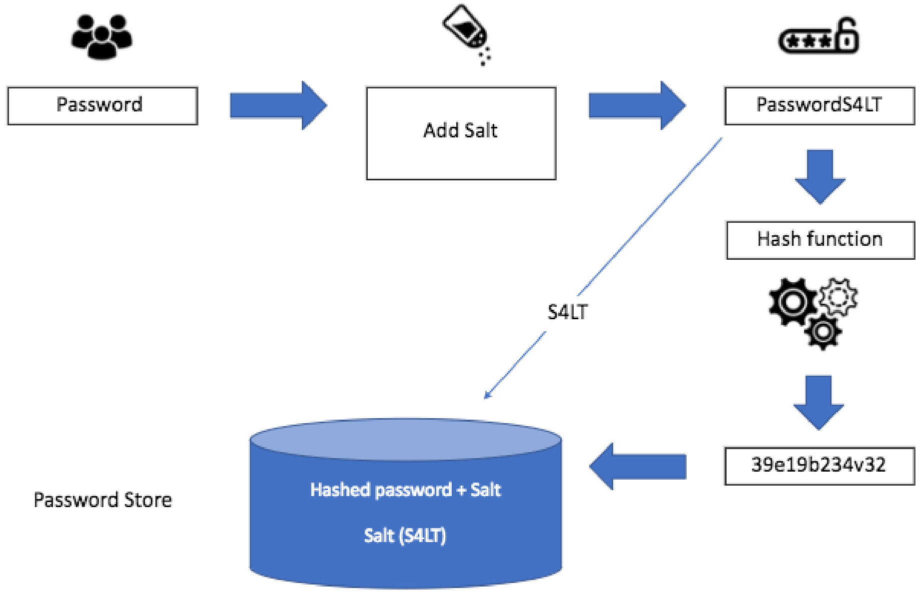
Example: Usages of Hash Functions

- In cryptography: MAC (previous lecture)

Example: Usages of Hash Functions

- In cryptography: Timestamping
  - How to prove that you have discovered a secret on an earlier date without disclosing it?

- In cryptography: Storing Password



Example: Usages of Hash Functions

- Primary use: Domain extension (compress long inputs, and feed them into boxes that can take only short inputs)

=> Typical security requirement: “collision resistance”

Security Properties for Hash Functions

Given a function $h: X \rightarrow Y$,

- Collision resistance:
  - It should be hard to compute any collision $x \neq x'$ such that $H(x) = H(x')$

- Second-preimage resistance (weak collision resistant):
  - Given x, it should be hard to compute any collision involving x. In other words, it should be hard to compute $x' \neq x$ such that $H(x) = H(x')$

Brute force Attacks on Hash Functions

- Attacking collision resistance
  - Goal: given h, find x, x' such that h(x)=h(x')
  - Algorithm: pick a random set X' of q values in X
    - For each $x \in X'$, computes $y_x = h(x)$
      - if $y_x = y_{x'}$ for some $x' \neq x$ then return (x,x') else fail

Collision brute force:

$\mathcal{A}_{cr}()$:
  for $i = 1, \ldots$:
    $x_i \leftarrow \{0,1\}^m$
    $y_i := H(x_i)$
    if there is some $j < i$ with $x_i \neq x_j$ but $y_i = y_j$:
      return (x_i, x_j)

Brute force Attacks on Hash Functions

- Attacking second-preimage resistance
  - Goal: given $h: X \rightarrow Y$, $y \in Y$, find x such that $h(x)=y$
  - Algorithm: pick a random value x in X,
    - check if h(x)=y, if h(x)=y, returns x; otherwise iterate
    - after failing q iterations, return fail

Second preimage brute force:

$\underline{\mathcal{A}_{2\text{pi}}(x)}$:

while true:

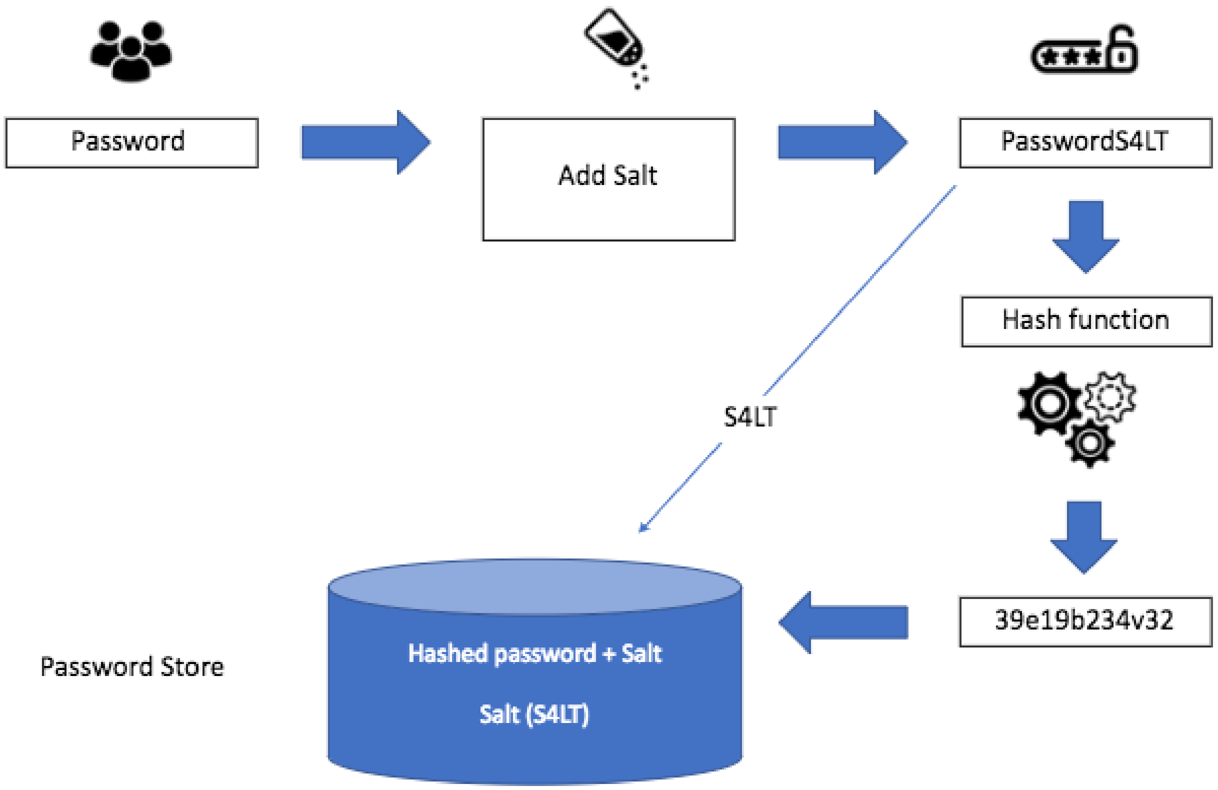
$\quad x' \leftarrow \{0,1\}^m$

$\quad y' := H(x')$

$\quad$ if $y' = H(x)$: return x'

Hash Function in Practice (Salted Hash)

- Salted Hash $H(s||x)$ for a salt s
- Without salts,
  - Two users might have the same password
  - An attacker can compute a dictionary of (p,H(p)) for common passwords
  - => this dictionary makes it easy to attack all users at once, since all users are using the same hash function



Hash Algorithm

- NIST standards (https://www.nist.gov/)
  - Mandatory in US Government
  - Adopted globally

- MD5, SHA (SHA-0) is no good anymore
- SHA-1 has attacks and is not recommended
- SHA-2 looks good for now
  - What happens when there’s an attack?
  - It takes years to create and analyze functions

Merkle-Damgård Construction

- Building a hash function, especially one that accepts inputs of arbitrary length, seems like a challenging task => Merkle-Damgård construction.
- **Merkle–Damgård construction** or **Merkle–Damgård hash function** is a method of building collision-resistant cryptographic hash functions from collision-resistant one-way compression functions.[1]: 145 This construction was used in the design of many popular hash algorithms such as MD5, SHA-1 and SHA-2.
  - https://en.wikipedia.org/wiki/Merkle%E2%80%93Damg%C3%A5rd_construction

Merkle-Damgård Construction

Construction 11.2 (Merkle-Damgård)

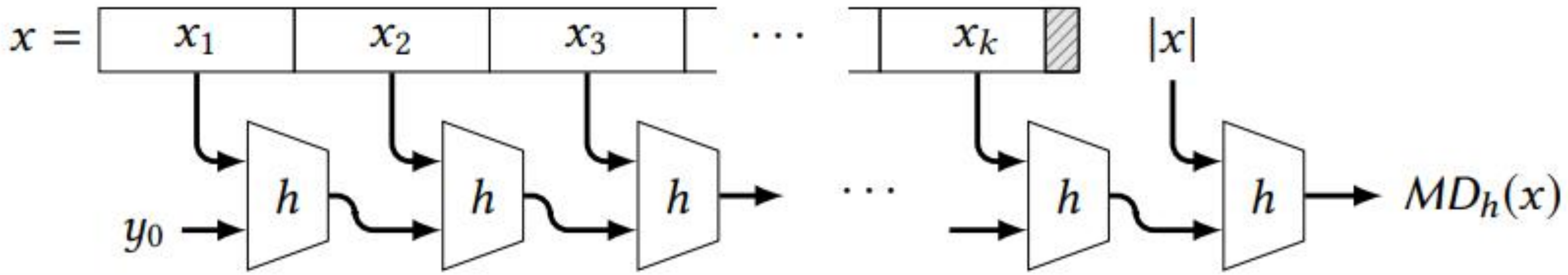
Let $h : \{0,1\}^{n+t} \to \{0,1\}^n$ be a compression function. Then the ***Merkle-Damgård transformation*** *of h is $MD_h : \{0,1\}^* \to \{0,1\}^n$, where:*

$\underline{\text{MDPAD}_t(x)}$

$\ell := |x|$, as length-t binary number
while $|x|$ not a multiple of t:
  $x := x \| 0$
return $x \| \ell$

$\underline{\text{MD}_h(x):}$

$x_1 \| \cdots \| x_{k+1} := \text{MDPAD}_t(x)$
// *each x_i is t bits*
$y_0 := 0^n$
for $i = 1$ to $k+1$:
  $y_i := h(y_{i-1} \| x_i)$
output y_{k+1}



Merkle-Damgård Construction

- Quiz Sample:

Suppose we have a compression function $h : \{0,1\}^{48} \rightarrow: \{0,1\}^{32}$. We build a Merkle-Damgård hash function out of this compression function and wish to compute the hash of the following 5-byte (40-bit) string:

$$x = 01100011 \;\; 11001101 \;\; 01000011 \;\; 1001011101010000$$

What are the value of block size t, and the output of $\text{MDPAD}_t(x)$

Hash Function

- Quiz Sample:

Given a non-collision resistant hash function H and a fixed constant k, consider the following hash function H' that applies the hash function H k times:

$$H'(x) = \underbrace{H(H(\ldots H(H(x))\ldots))}_{k \text{ times}}$$

Is the H' not collision resistant?