lecture 16

Cryptographic Hash Functions

Def: A hash-function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called "hash-value".

\# prob that a randomly chosen string gets mapped to a particular n-bit hash-value (image) is ?

n bits → # poss 2^n → prob $= \frac{1}{2^n} = 2^{-n}$

2

Applications

1. **Digital Signature**: a long message is usually hashed (using a publicly available hash function) and only the hash-value is signed → time & space are saved

2. **Data Integrity**:

Message "m" → just protect your hash-value (crypto techniques) to be altered → Message "m′"

↓

its hash value "h"

protected hash-value "h"

generate new hash value from m′, say "h′"

if $h = h'$ → m′ also the same as m

Note $h: D \to R$ → hash codes, hash-value, hash-result, hash

$|D| > |R|$

in all cases, larger domains are mapped to smaller range

properties of hash functions

1. Compression: h maps an input x of arbitrary finite bitlength to an output $h(x)$ of fixed bitlength "n".
2. Ease of computation: given "h" and input "x" $h(x)$ is easy to compute.

Two classes of hash functions

1. Modification Detection Codes (MDCs)

   The purpose of an MDC is to provide a representative image or hash of a message $\xrightarrow{\text{goal}}$ data integrity

   $\xrightarrow{\text{Input}}$ message

2. Message Authentication Codes (MACs)

   The purpose of a MAC is to facilitate assurance regarding bothe the source of a message and its integrity. $\xrightarrow{\text{goal}}$ data integrity authentication

   $\xrightarrow{\text{Input}}$ message key

Hash Functions Security Requirments

1. **preimage resistance** it's computationally infeasible to find any preimage x' such that $h(x') = y$ when given any "y" for which a corresponding input is not known.

OWHF
one-way hash function

2. **2nd-preimage resistance** it's computationally infeasible to find any second input which has the same output as any specified input $\longrightarrow$ given x
find x' s.t $x' \neq x$
$h(x) = h(x')$

3. **Collision resistance** it's computationally infeasible to find <u>any two distinct</u> inputs x & x' which hash to the same output $\longrightarrow h(x) = h(x')$

Note: there is free choice of both inputs.

preimage resistant $\equiv$ one-way

2nd-preimage resistance $\equiv$ weak collision resistance

collision resistance $\equiv$ strong " "

- hash functions
  - unkeyed
    - MDC
    - ...
  - keyed
    - ...
    - MAC

OWHF $\left\{\begin{array}{l}\text{preimage Res} \\ \text{2nd-" "}\end{array}\right.$

$\left.\begin{array}{l}\text{2nd-" "} \\ \text{Collision "}\end{array}\right\} \to$ CRHF

Cheeksum (Mod value) $\longrightarrow$ Compression

DES $\longrightarrow$ preimage resistance

$\boxed{\text{Def.}}$ A message Authentication Code (MAC) is a family of functions h_k parameterized by a secret key "k" with the following properties:

1. $\boxed{\text{Ease of computation.}}$ h_k, k, x $\xrightarrow{\text{easy}}$ $\underbrace{h_k(x)}_{\text{MAC-value}}$

2. $\boxed{\text{Compression:}}$ h_k maps an input "x" of arbitrary finite bitlength to an output $h_k(x)$ of fixed bitlength.

3. $\boxed{\text{Computation-resistance}}$ given zero or more text-MAC pair $(x_i, h_k(x_i))$, it's computationally infeasible to compute any text-MAC pair $(x, h_k(x))$ for any new input where $x \neq x_i$

① objectives of adversaries vs MDCs

Adversary intends to attack an MDC

(a) To attack a OWHF: given a hash-value "y" find a preimage "x" s.t. $y = h(x)$ or given a pair $(x, h(x))$, find a 2nd preimage x' such that $h(x') = h(x)$.

(b) To attack a CRHF: find any two inputs x & x' such that $h(x') = h(x)$.

② objectives of adversaries vs MACs

(C) To attack a MAC: without any prior knowledge of a key "k", compute a new text-MAC pair $(x, h_k(x))$ for some text $x \neq x_i$ where one or more pairs $(x_i, h_k(x_i))$ are given.

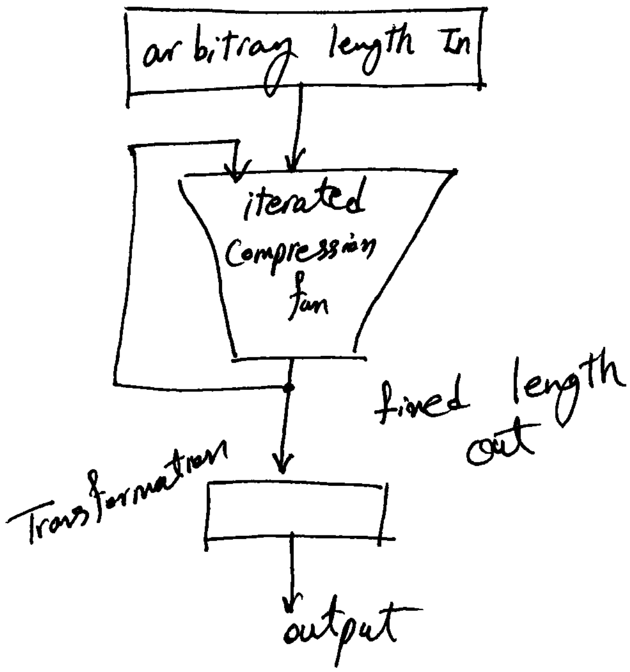
(C.1) $\boxed{\text{known-text attack}}$: one or more text-MAC pairs $(x_i, h_k(x_i))$ are available

(C.2) $\boxed{\text{Chosen-text attack}}$: one or more text-MAC pairs $(x_i, h_k(x_i))$ are available for x_i chosen by the adversary.

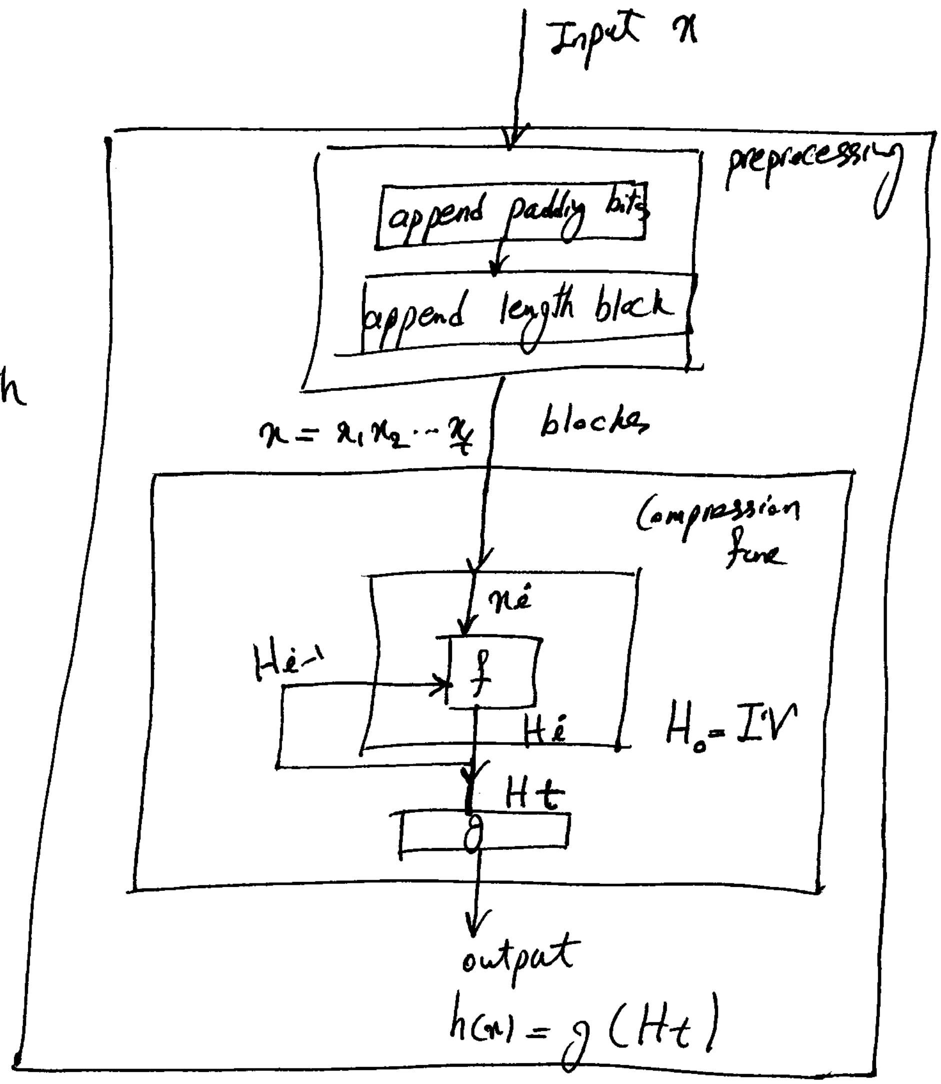
(C.3) $\boxed{\text{Adaptive chosen-text attack}}$: The x_i may be chosen by the adv. as above, now allowing successive choices to be based on the results of prior queries.

Hash Functions Architecture

High-level view



detailed view



Example

$$\begin{cases} x = x_1 x_2 \dots x_t \\ H_0 = IV, \quad H_i(H_{i-1}, x_i) \\ \qquad 1 \leqslant i \leqslant t \end{cases} \qquad h(x) = g(H_t)$$

$$H_0 = 0^n, \quad H_i = f(H_{i-1} \,\|\, x_i)$$