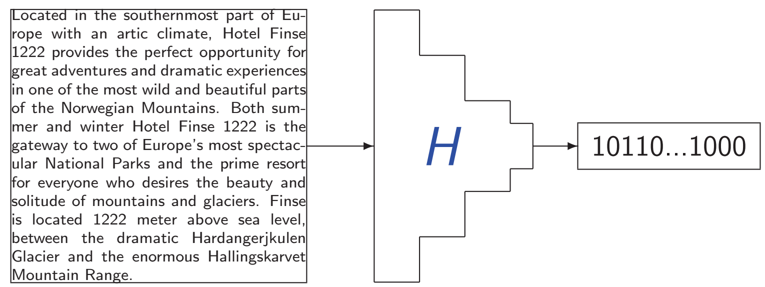
Cryptographic Hash Functions - Introduction

Lars R. Knudsen

April 21, 2008

Definition - hash function



$H : \{0,1\}^* \rightarrow \{0,1\}^n$, for fixed value of n

Cryptographic hash functions

- play a crucial role in cryptography
- many applications
- a many-to-one function
- should appear to be one-to-one in practice

Definition - more

- $H : \{0,1\}^* \to \{0,1\}^n$, for fixed value of n
- no secret parameters
- given x, easy to compute $H(x)$
- Often in practice:
  $H : \{0,1\}^M \to \{0,1\}^n$, for fixed value of n, big M

Hash Function Applications

- Digital signatures
- Password protection, Unix
- Message authentication codes, HMAC
- As random oracle in various protocols, RSA-OAEP, RSA-PSS, PKCS #1, v2.1
- Pseudo-random generator (key-derivation), DSS
- ...

Password protection

User id	H(password)
...	...
La, Shangri	09283409283977
Lan, Magel	01265743912917
Lang, Serge	02973477712981
Lange, Tanja	92837540921835
Langer, Bernhard	98240254444422
...	...

Problem: Parallel attack?!

Password protection, cont.

User id	Salt	H(password, salt)
...	...	...
La, Shangri	68678927431	09283409283977
Lan, Magel	00000000001	01265743912917
Lang, Serge	23092839482	02973477712981
Lange, Tanja	30092341218	92837540921835
Langer, Bernhard	86769872349	98240254444422
...	...	...

It should be “hard” to find preimage of H

Digital signatures (no hashing)

Alice | **Bob**

m message $\xrightarrow{sig(m),\, m}$ check signatur

Digital signatures with hashing

Sign $H(m)$ instead of m

Alice | **Bob**

m message

compute $H(m)$ $\xrightarrow{sig(H(m)),\, m}$ compute $H(m)$

check signatur on $H(m)$

Attack scenarios - inversion

Given m and $H(m)$, **Eve** finds m' such that $H(m) = H(m')$

Alice | **Eve** | **Bob**

$H(m)$ $\xrightarrow{sig(H(m)),\, m}$ $\xrightarrow{sig(H(m)),\, m'}$ compute $H(m')$

check sign. on $H(m')$

It must be "hard" to find 2nd preimages for H.

Attack scenarios - collisions

Given H, **Bob** finds m and m' such that $H(m) = H(m')$, tricks **Alice** into signing m

Alice | **Bob**

$H(m)$ $\xrightarrow{sig(H(m)),\, m}$ compute $H(m)$

check signatur on $H(m)$

..later..

you signed m' not m

It must be "hard" to find collisions for H.

Cryptographic Hash Functions

$H : \{0,1\}^* \rightarrow \{0,1\}^n$, for fixed value of n

Definitions

- **preimage**, given $H(x)$, find x' s.t. $H(x) = H(x')$
- **2nd preimage**, given x, find $x' \neq x$ s.t. $H(x) = H(x')$
- **collision**, $x \neq x'$, s.t. $H(x) = H(x')$

Random Oracle Model

Let $H : \{0,1\}^* \rightarrow \{0,1\}^n$ be a hash function.
Random Oracle Model:

- the values $H(x)$ are "random", that is, for any x and $y \in \{0,1\}^n$

$$\Pr(H(x) = y) = 2^{-n}$$

- let $\mathcal{X} = \{x_1, \ldots, x_t\}$,
  if $H(x_1), H(x_2), \ldots, H(x_t)$ known by attacker,
  then for any $x \notin \mathcal{X}$ and $y \in \{0,1\}^n$

$$\Pr(H(x) = y) = 2^{-n}$$

Trivial (brute-force) attacks

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

- given $y = H(x)$
- let $\mathcal{X} = \{x_1, \ldots, x_q\}$
- **for** $x' \in \mathcal{X}$ **if** $H(x') = y$ **then success**

Probability of success:

$$1 - (1 - 2^{-n})^q$$

With $q = 2^n$ probability of success $1 - (1 - 2^{-n})^{2^n} \approx 0.63$

Trivial (brute-force) attacks

n	$(1 - 2^{-n})^{2^n}$
5	0.6379
10	0.6323
15	0.6321
20	0.6321

q	$1 - (1 - 2^{-n})^q$
2^{n-1}	0.3935
2^n	0.6321
2^{n+1}	0.8647
2^{n+2}	0.9817

Trivial (brute-force) attacks

2nd preimage attack for $H : \{0,1\}^* \to \{0,1\}^n$

- given x and $y = H(x)$
- let $\mathcal{X} = \{x_1, \ldots, x_q\}$, s.t., $x \notin \mathcal{X}$
- **for** $x' \in \mathcal{X}$ **if** $H(x') = y$ **then success**

Probability of success:

$$1 - (1 - 2^{-n})^q$$

With $q = 2^n$ probability of success $1 - (1 - 2^{-n})^{2^n} \approx 0.63$

Introduction Definition Applications Properties Iterated hash functions

Trivial (brute-force) attacks

collision attack for $H : \{0,1\}^* \to \{0,1\}^n$

- let $\mathcal{X} = \{x_1, \ldots, x_q\}$,
- let $\mathcal{Y} = \{y_1, \ldots, y_q\}$, where $y_i = H(x_i)$
- **if** $y_i = y_j$ for some $i \neq j$ **then success**

Probability of success:

$$1 - e^{\frac{q(q-1)}{2 \cdot 2^n}}$$

With $q = \sqrt{2} \cdot 2^{n/2}$ one gets probability of success of $1 - e^{-1} \approx 0.63$

Introduction Definition Applications Properties Iterated hash functions

Birthday paradox

Choose q elements at random (with replacements) from set of S random elements, where $q \ll S$
Let p be probability of at least one collision

$$\begin{aligned} 1 - p &= 1 \cdot \frac{S-1}{S} \cdot \frac{S-2}{S} \cdots \frac{S-(q-1)}{S} \\ &= \prod_{k=1}^{q-1} \left(1 - \frac{k}{S}\right) \\ &\approx \prod_{k=1}^{q-1} \exp(-\frac{k}{S}) = \exp\left(-\frac{q(q-1)}{2S}\right) \end{aligned}$$

NB. $e^{-x} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \ldots$

Introduction Definition Applications Properties Iterated hash functions

Birthday paradox (2)

$$p \approx 1 - \exp\left(-\frac{q(q-1)}{2S}\right)$$

q	$\approx p$
$1.17\sqrt{S}$	50%
$1.41\sqrt{S}$	63%
$2\sqrt{S}$	86%
$4\sqrt{S}$	99.99%

birthday paradox: $(S, q) = (365, 23)$, $p \approx 1/2$

Birthday paradox used on hash functions

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

1. choose $q = 2^{(n+1)/2} = \sqrt{2} \cdot 2^{n/2}$ randomly chosen inputs each of at least $(n+1)/2$ bits
2. compute hash values for all k inputs

Prob(at least one collision) =

$$p \approx 1 - \exp\left(-\frac{q(q-1)}{2 \cdot 2^n}\right) \approx 1 - e^{-1} \simeq 0.63$$

Cryptographic hash functions - generic attacks

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

attack	rough complexity
collision	$\sqrt{2^n} = 2^{n/2}$
2nd preimage	2^n
preimage	2^n

Today: $n \geq 160$ is recommended

Aim: no better attacks than generic attacks

Reductions

$H : \{0,1\}^* \rightarrow \{0,1\}^n$, for fixed value of n

In random oracle model:

- 2nd preimage attack for $H \Rightarrow$ collision attack for H
- preimage attack for $H \Rightarrow$ collision attack for H

This lead to

- collisions hard $\Rightarrow$ 2nd preimages and preimages hard

Iterated hash functions

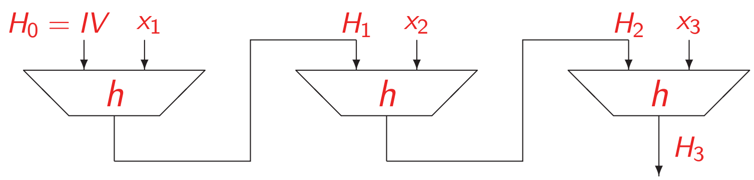
Let $h : \{0,1\}^N \rightarrow \{0,1\}^n, N > n$, compression function

Construct

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

where $M >> N$, such that collision for H implies collision for h

Iterated hash function (from h to H)



- h compression function: $\{0,1\}^{N-n} \times \{0,1\}^n \rightarrow \{0,1\}^n$, where $N > n$
- apply padding of input to multiple of $N - n$
- divide input into blocks $x_1, x_2,, \ldots, x_t$, where $|x_i| = N - n$
- define output of $H : \{0,1\}^* \rightarrow \{0,1\}^n$ as final h output

Extending hash functions - Merkle-Damgård

Padding-rule: $x \neq y \Rightarrow \text{pad}(x) \neq \text{pad}(y)$
Construct H from h:

1. let $IV \in \{0,1\}^n$ be fixed, let $x \in \{0,1\}^v$ be message
2. apply padding rule such that

$$x = x_1 \mid x_2 \mid \ldots \mid x_t$$

   where x_t full block which contains the integer v as a string
3. define $h_0 = IV$ and $h_i = h(x_i, h_{i-1})$ for $1 \leq i \leq t$
4. define $H(x) = h_t$

Theorem: collision for $H \Rightarrow$ collision for h

Properties of iterated hash functions

Compression function $h : \{0,1\}^N \rightarrow \{0,1\}^n$

- Given 2^k hashed messages, effort to find 2nd preimage of ≥ 1 of them is 2^{n-k} (Merkle)
- Given hashed messages with 2^k message blocks, effort to find 2nd preimage is $\simeq k2^{n/2} + 2^{n-k}$ (Dean, Kelsey-Schneier)

attack	rough complexity
collision	$\sqrt{2^n} = 2^{n/2}$
2nd preimage	$k2^{n/2} + 2^{n-k}$
preimage	2^n

The extension attack for iterated hash functions

- Let $\text{pad}(x)$ and $\text{pad}(x')$ be result of padding strings x and x'.
- Assume $\text{pad}(x)$ and $\text{pad}(x')$ of same lengths and that

$$H(x) = H(x')$$

- Let y be non-empty string and let

$$z = \text{pad}(x) \mid y \quad \text{and} \quad z' = \text{pad}(x') \mid y,$$

  where '|' denotes concatenation of strings.
- Then

$$H(z) = H(z')$$

Properties of iterated hash functions

1. h is collision-resistant $\Rightarrow$ H is collision-resistant (MD)
2. h is random oracle $\Rightarrow$ H is random oracle ?
3. Coron et al 05: construction satisfying 2.
4. Bellare-Ristenpart 06: construction satisfying both 1. and 2.

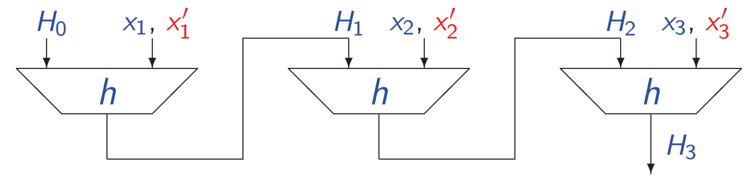
How to beat collision resistance

- Make output of hash function sufficiently large (s.t. $2^{n/2}$ is huge)
- Family of (strong) hash functions. Choose member of family at random, then hash.
- How not to do it.
  Assume $2^{n/2}$ operations are in range of attacker. Define hash as

$$F(m) = G(m) \mid H(m),$$

where
  - G hash function of n bits
  - H iterated hash function of n bits

Iterated hash functions - multi-collisions



- assume x_1 and x_1' is a collision for h using H_0
- assume x_2 and x_2' is a collision for h using H_1
- assume x_3 and x_3' is a collision for h using H_2
- leads to eight collisions for H
- Coppersmith 85, Joux 04

Concatenated hash function - collision

- $F(m) = G(m) \mid H(m)$, where
  - G hash function of n bits
  - H iterated hash function of n bits
- Find $2^{n/2}$-collision on H in multi-collision attack.
- One of these gives collision also for $G \Rightarrow$
  Collision for F with effort $(n/2)2^{n/2}$.

Hash function collisions irrelevant ?

- Often heard criticism, collisions are on “random” messages, so not important
- Dobbertin breaks MD4 in 94, after criticsm he shows meaningful collisions on MD4
- Often it requrires only little extra effort to make collisions meaningful
- Daum-Lucks 05 on PostScript

Collision in Postscript (Daum-Lucks 2005)

- Notation: (S1)(S2)eqT1T2ifelse
- Meaning: If $S1 = S2$ then $T1$ else $T2$
- Find random messages S1 and S2 which collide under hash function
- Construct PS1 and PS2 for arbitrary T1 and T2
- PS1: ...(S1)(S2)eqT1T2ifelse...
- PS2: ...(S2)(S2)eqT1T2ifelse...