Cryptographic Hash Functions - Introduction

Lars R. Knudsen

April 21, 2008

Introduction Definition Applications Properties Iterated hash functions

Definition - hash function

Located in the southernmost part of Europe with an artic climate, Hotel Finse 1222 provides the perfect opportunity for great adventures and dramatic experiences in one of the most wild and beautiful parts of the Norwegian Mountains. Both summer and winter Hotel Finse 1222 is the gateway to two of Europe’s most spectacular National Parks and the prime resort for everyone who desires the beauty and solitude of mountains and glaciers. Finse is located 1222 meter above sea level between the dramatic Hardangerjukel Glacier and the enormous Hallingskarvet Mountain Range.

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

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
Introduction Definition Applications Properties Iterated hash functions

Definition - more

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

<table>
<thead>
<tr>
<th>User id</th>
<th>H(password)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>La, Shangri</td>
<td>09283409283977</td>
</tr>
<tr>
<td>Lan, Magel</td>
<td>01265743912917</td>
</tr>
<tr>
<td>Lang, Serge</td>
<td>02973477712981</td>
</tr>
<tr>
<td>Lange, Tanja</td>
<td>92837540921835</td>
</tr>
<tr>
<td>Langer, Bernhard</td>
<td>98240254444422</td>
</tr>
</tbody>
</table>

Problem: Parallel attack?!

Password protection, cont.

<table>
<thead>
<tr>
<th>User id</th>
<th>Salt</th>
<th>H(password, salt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>La, Shangri</td>
<td>68678927431</td>
<td>09283409283977</td>
</tr>
<tr>
<td>Lan, Magel</td>
<td>0000000000001</td>
<td>01265743912917</td>
</tr>
<tr>
<td>Lang, Serge</td>
<td>23092839482</td>
<td>02973477712981</td>
</tr>
<tr>
<td>Lange, Tanja</td>
<td>30092341218</td>
<td>92837540921835</td>
</tr>
<tr>
<td>Langer, Bernhard</td>
<td>86769872349</td>
<td>98240254444422</td>
</tr>
</tbody>
</table>

It should be “hard” to find preimage of $H$
Digital signatures (no hashing)

Alice Bob

\[ m \rightarrow \text{sig}(m), m \rightarrow \text{check signature} \]

Digital signatures with hashing

Sign \( H(m) \) instead of \( m \)

Alice Bob

\[ m \rightarrow \text{compute } H(m), \text{sig}(H(m)), m \rightarrow \text{compute } H(m), \text{check signature on } H(m) \]

Attack scenarios - inversion

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

Alice Eve Bob

\[ H(m) \rightarrow \text{sig}(H(m)), m \rightarrow \text{sig}(H(m)), m' \rightarrow \text{compute } H(m'), \text{check signature on } H(m') \]

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

\[ H(m) \rightarrow \text{sig}(H(m)), m \rightarrow \text{compute } H(m), \text{check signature on } H(m), \ldots \rightarrow \text{you signed } m' \text{ not } m \]

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

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 $1 - (1 - 2^{-n})^{2^n} \approx 0.63$

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 $x \not\in \mathcal{X}$ and $y \in \{0,1\}^n$
  $$Pr(H(x) = y) = 2^{-n}$$

Trivial (brute-force) attacks

<table>
<thead>
<tr>
<th>$n$</th>
<th>$(1 - 2^{-n})^{2^n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.6379</td>
</tr>
<tr>
<td>10</td>
<td>0.6323</td>
</tr>
<tr>
<td>15</td>
<td>0.6321</td>
</tr>
<tr>
<td>20</td>
<td>0.6321</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$q$</th>
<th>$1 - (1 - 2^{-n})^q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^{n-1}$</td>
<td>0.3935</td>
</tr>
<tr>
<td>$2^n$</td>
<td>0.6321</td>
</tr>
<tr>
<td>$2^{n+1}$</td>
<td>0.8647</td>
</tr>
<tr>
<td>$2^{n+2}$</td>
<td>0.9817</td>
</tr>
</tbody>
</table>
Introduction Definition Applications Properties Iterated hash functions

Trivial (brute-force) attacks

2nd preimage attack for $H : \{0, 1\}^* \to \{0, 1\}^n$
- given $x$ and $y = H(x)$
- let $X = \{x_1, \ldots, x_q\}$, s.t., $x \not\in X$
- for $x' \in X$ if $H(x') = y$ then success
Probability of success:

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

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

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

$$1 - p = 1 - \frac{S - 1}{S} \cdot \frac{S - 2}{S} \cdots \frac{S - (q - 1)}{S}$$

$$= \prod_{k=1}^{q-1} \left(1 - \frac{k}{S}\right)$$

$$\approx \prod_{k=1}^{q-1} \exp\left(-\frac{k}{S}\right) = \exp\left(-\frac{q(q-1)}{2S}\right)$$

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

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

<table>
<thead>
<tr>
<th>$q$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1.17\sqrt{S}$</td>
<td>50%</td>
</tr>
<tr>
<td>$1.41\sqrt{S}$</td>
<td>63%</td>
</tr>
<tr>
<td>$2\sqrt{S}$</td>
<td>86%</td>
</tr>
<tr>
<td>$4\sqrt{S}$</td>
<td>99.99%</td>
</tr>
</tbody>
</table>

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

$$\text{Prob( at least one collision)} = p \approx 1 - \exp \left( -\frac{q(q - 1)}{2 \cdot 2^n} \right) \approx 1 - e^{-1} \approx 0.63$$

Cryptographic hash functions - generic attacks

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

<table>
<thead>
<tr>
<th>attack</th>
<th>rough complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>collision</td>
<td>$\sqrt{2^n} = 2^{n/2}$</td>
</tr>
<tr>
<td>2nd preimage</td>
<td>$2^n$</td>
</tr>
<tr>
<td>preimage</td>
<td>$2^n$</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>Rough Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision</td>
<td>( \sqrt{2^n} = 2^{n/2} )</td>
</tr>
<tr>
<td>2nd preimage</td>
<td>( k2^{n/2} + 2^{n-k} )</td>
</tr>
</tbody>
</table>

### 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 \( x \in \{0,1\}^n \) be message
  2. Apply padding rule such that
     \[ x = x_1 | x_2 | \ldots | x_t \]
     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 \( h \)

### 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
  \[ z = \text{pad}(x) | y \quad \text{and} \quad z' = \text{pad}(x') | y, \]
  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 ?
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
  \[ F(m) = G(m) \| H(m), \]
  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

- \( F(m) = G(m) \| H(m) \), 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 criticism he shows meaningful collisions on MD4
- Often it requires only little extra effort to make collisions meaningful
- Daum-Lucks 05 on PostScript

Collision in Postscript (Daum-Lucks 2005)

- Notation: \((S_1)(S_2)\text{eqT1T2ifelse}\)
- Meaning: If \(S_1 = S_2\) then \(T_1\) else \(T_2\)
- Find random messages \(S_1\) and \(S_2\) which collide under hash function
- Construct PS1 and PS2 for arbitrary \(T_1\) and \(T_2\)
  - PS1: \(...(S_1)(S_2)\text{eqT1T2ifelse}...\)
  - PS2: \(...(S_2)(S_2)\text{eqT1T2ifelse}...\)