Hash Functions and SHA-3

Lars R. Knudsen

February 13, 2008

### Definition - hash function

Aboriginal settlers arrived on the continent from Southeast Asia about 40,000 years before the first Europeans began exploration in the 17th century. No formal territorial claims were made until 1770, when Capt. James Cook took possession in the name of Great Britain. Six colonies were created in the late 18th and 19th centuries; they federated and became the Commonwealth of Australia in 1901. The new country took advantage of its natural resources to rapidly develop agricultural and manufacturing industries and to make a major contribution to the British effort in World Wars I and II.

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

### Generic attacks

For \( H : \{0,1\}^* \rightarrow \{0,1\}^n \)

<table>
<thead>
<tr>
<th>Attack</th>
<th>Rough complexities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collisions</td>
<td>( \sqrt{2^n} = 2^{n/2} )</td>
</tr>
<tr>
<td>2nd preimages</td>
<td>( 2^n )</td>
</tr>
<tr>
<td>Preimage</td>
<td>( 2^n )</td>
</tr>
</tbody>
</table>

Goal: generic attacks are best (known) attacks
Further properties

- “Behave like” a random oracle
- Indifferentiable from random oracle
- Variants of (second)-preimage resistance
  - aPre, ePre, aSec, and eSec
- Security against
  - Extension attack
  - Multi-collisions

Iterated hash functions - (Merkle-Damgård schemes)

For \( H : \{0,1\}^* \rightarrow \{0,1\}^n \)

<table>
<thead>
<tr>
<th>Attack</th>
<th>Rough Complexities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collisions</td>
<td>(2^{n/2})</td>
</tr>
<tr>
<td>2nd preimages</td>
<td>(k2^{n/2} + 2^{n-k}) with (2^k) blocks</td>
</tr>
<tr>
<td>Preimage</td>
<td>(2^n)</td>
</tr>
</tbody>
</table>
Merkle (1989)

$h : \{0,1\}^m \to \{0,1\}^t$, assume $m > t$

- Split message, $x$, into blocks of $m - t$ bits.
- If last block incomplete, pad with zeros.
- Append extra block containing length of $x$ (bits)
- Define $h_{i+1} = h(h_i, x_{i+1})$, $H(x) = h_0$.
- Collision for $H$ means collision for $h$

Damgård (1989)

$h : \{0,1\}^m \to \{0,1\}^t$, assume $m > t + 1$

- Split message, $x$, into blocks of $m - t - 1$ bits.
- If last block incomplete, pad with $d$ zeros.
- Append extra block containing bin. repr. of $d$ (fixed length)
- Then define $h_1 = h(\text{iv} \mid 0 \mid x_1)$ $h_{i+1} = h(h_i \mid 1 \mid x_{i+1})$
- $H(x) = h_s$.

Merkle-Damgård Strengthening, Lai-Massey (1992)

Parallelizable hash: $h : \{0,1\}^{2t} \to \{0,1\}^t$

- Message $x$ of $j$ bits.
- Pad message with 0s until length is $2^j \cdot t$ for some $j$.
- Let $h_0$ be padded message of $2^j t$ bits.
- Hash $h_0$ to $h_1$ of $2^{j-1} t$ bits using $h$
- Hash $h_1$ to $h_2$ of $2^{j-2} t$ bits using $h$
- Gives $h_j$ of $t$ bits
- $H(x) = h(h_j \mid \text{length}(x))$

- Digital signatures .... for efficiency:
- “Let $g$ be a one-way mapping from binary $N$-space to binary $n$-space...”. “Take the $N$ bit message $m$ and operate on it with $g$ to obtain the $n$ bit vector $m'$.”
- “It must be hard even given $m$ to find a different inverse image of $m'$”
- “Finding such functions appears to offer little trouble”

Hash function using a block cipher

Why build on a block cipher?

- it’s natural!
- use existing technology
- transfer security (trust?!) to hash construction
- schemes "slow" (partly due to key-schedules)
- weaknesses of block cipher not relevant for encryption

Diffie-Hellman, $\kappa > n$

$e : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^n$

$(m_i | h_{i-1})$

- $x_0$ fixed block
- 2nd preimages hard if $e$ secure against known-plaintext attack

Block cipher based hash functions

- “Diffusion is more important than confusion in hash functions”
- “Confusion is more important than diffusion in block ciphers”
- Why? Why not have S-boxes in hash functions?
- How fast should/can a hash function be?
Additive stream cipher

\[ k \]
\[ j \]
\[ z_1, z_2, z_3, z_4, \ldots \]

Hash function

\[ k \]
\[ p_i \rightarrow c_i \]

\[ m_j \]
\[ h_{i-1} \rightarrow f \rightarrow h_i \]
### Speed

- Additive stream cipher, known/chosen plaintext attack
- Block cipher, chosen plaintext attack
- Hash function, known/chosen-key attack

<table>
<thead>
<tr>
<th>Stream</th>
<th>4-8 cycles/byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>20 cycles/byte</td>
</tr>
<tr>
<td>SHA-1</td>
<td>11 cycles/byte</td>
</tr>
<tr>
<td>SHA-512</td>
<td>18 cycles/byte</td>
</tr>
</tbody>
</table>

### Hash rate

- Given hash function built from block cipher
  \[ e : \{0, 1\}^\kappa \times \{0, 1\}^n \rightarrow \{0, 1\}^n \]
- Rate usually is defined as
  \[ \frac{\# \text{ n-bit blocks hashed}}{\# \text{ invocations of } e} \]
- Ought perhaps be defined as
  \[ \frac{\# \text{ n-bit blocks hashed}}{\# \text{ invocations of } e + \# \text{ key-schedules}} \]
Davies-Price variant of Rabin’s scheme 1980

\[ h_0 \rightarrow m_1 \oplus e \rightarrow m_2 \rightarrow \cdots \rightarrow m_t \rightarrow h_t \]

\[ h_t \rightarrow m_1 \oplus e \rightarrow m_2 \rightarrow \cdots \rightarrow m_t \rightarrow h_{2t} \]

- Coppersmith 1985:
  - preimage attack on one-chain Rabin \( \approx 2^{n/2} \)
  - preimage attack on two-chains Rabin \( \approx 2^{n/2+n/16} \) using multi-collisions

Single block hash

- \( e : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \)
- 12 secure ones (Preneel 93, Black et al 02), here three
  - Davies-Meyer
    \[ h_i = e_{m_i}(h_{i-1}) \oplus h_{i-1} \]
  - Matyas-Meyer-Oseas
    \[ h_i = e_{h_{i-1}}(m_i) \oplus m_i \]
  - Preneel-Miyaguchi
    \[ h_i = e_{h_{i-1}}(m_i) \oplus m_i \oplus h_{i-1} \]
- Hash rates. About \( 1/(1+1) \) (1/2 for DES and AES)
- Collisions (birthday attack) in \( 2^{n/2} \) operations

MD4-family

- MD4, Rivest 1990
- MD5, Rivest 1991
- SHA-0, 1993
- SHA-1, 1994
- all hash functions of Davies-Meyer form
- “block ciphers” with feed-forward
- hash rates for Davies-Meyer can be (arbitrarily) high

Double block hash - based on block ciphers

- Based on \( e : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n \)
- Length of hash, \( 2n \) bits
- Aim: \( 2^n \) security level for collisions
  - Merkle, 1989
  - PBGV, QG, LOKI-DBH, ...., 1990s
  - Hirose, Nandi, 2005
Merkle’s double block schemes with DES (1989)

- “DES can be used to build a one-way hash function which is secure”
- if DES fails “it seems almost certain that some block cipher exist with the desirable properties”
- proof of security in ideal cipher model
- collisions \(\approx 2^{55}\), inconvenient block sizes, low hash rates
- “recent proposal from IBM looks very hopeful”, but no proof..

MDC-2

- designed for DES but can be used with any block cipher
- hash rate \(1/(2+2)\) \(= 1/4\) for DES and AES

MCD-2 used with DES and AES

(Best known attacks)

<table>
<thead>
<tr>
<th>Attack</th>
<th>DES</th>
<th>AES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preimage attack</td>
<td>(2^{83})</td>
<td>(2^{192})</td>
</tr>
<tr>
<td>2nd preimage attack</td>
<td>(2^{83})</td>
<td>(2^{192})</td>
</tr>
<tr>
<td>Collision attack</td>
<td>(2^{55})</td>
<td>(2^{128})</td>
</tr>
<tr>
<td>Hash rate</td>
<td>(1/4)</td>
<td>(1/4)</td>
</tr>
</tbody>
</table>

- For use with AES, “proof” that collision requires \(> 2^{75}\) operations (Steinberger 2007)
Abreast-DM & Tandem-DM - Lai, Massey 1990

e : \{0, 1\}^\kappa \times \{0, 1\}^n \to \{0, 1\}^n, \kappa > n \quad f(x, y) = e_x(y) \oplus y

Abreast-DM scheme:
\[
\begin{align*}
h^1_i &= f(h^2_{i-1} || m_i, h^1_{i-1}) \\
h^2_i &= f(m_i || h^1_{i-1}, \overline{h^2_{i-1}})
\end{align*}
\]

Tandem-DM scheme:
\[
\begin{align*}
h^1_i &= f(h^2_{i-1} || m_i, h^1_{i-1}) \\
h^2_i &= f(m_i || (h^1_i \oplus h^1_{i-1}), h^2_{i-1})
\end{align*}
\]

AES-256, (128-bit block, 256-bit key), hash rate 1/4, conjectured security level for collisions $2^{128}$

Knudsen-Preneel 1996

\[ f_i(x, y) = e_x(y) \oplus y \]

Compress: \((h^1_{i-1}, \ldots, h^5_{i-1}, m_i) \to (h^1_i, \ldots, h^5_i)\)

\[
\begin{align*}
h^1_i &= f_1(h^1_{i-1}, h^2_{i-1}) \\
h^2_i &= f_2(h^3_{i-1}, h^4_{i-1}) \\
h^3_i &= f_3(h^5_{i-1}, m_i) \\
h^4_i &= f_4(h^3_{i-1} \oplus h^3_{i-1} \oplus h^5_{i-1}, h^3_{i-1} \oplus h^5_{i-1} \oplus m_i) \\
h^5_i &= f_5(h^3_{i-1} \oplus h^3_{i-1} \oplus h^4_{i-1} \oplus m_i, h^2_{i-1} \oplus h^3_{i-1} \oplus h^5_{i-1} \oplus m_i)
\end{align*}
\]

Constructed from \([5, 3, 3]\) code over GF(2^2): rate 1/(5+5)
Claimed security against collision attacks is $2^n$

Knudsen-Preneel, more examples

Better rates using codes over larger fields

<table>
<thead>
<tr>
<th>GF(2^2)</th>
<th>GF(2^4)</th>
<th>Collision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Rate</td>
<td>Code</td>
</tr>
<tr>
<td>[5, 3, 3]</td>
<td>1/(5+5)</td>
<td>[6, 4, 3]</td>
</tr>
<tr>
<td>[8, 5, 3]</td>
<td>2/(8+8)</td>
<td>[8, 6, 3]</td>
</tr>
<tr>
<td>[12, 9, 3]</td>
<td>6/(12+12)</td>
<td>[12, 10, 3]</td>
</tr>
</tbody>
</table>

AES-128, rate 1/3, conjectured security level for collisions $2^{128}$

Hirose’s double block mode 2006

Based on work by Nandi, 2005
\[ e : \{0, 1\}^\kappa \times \{0, 1\}^n \to \{0, 1\}^n, \kappa > n, c \text{ nonzero constant} \]

\[
\begin{align*}
h^1_i &= e_{h^2_{i-1} || m_i} (h^1_{i-1} \oplus \overline{h^1_{i-1}}) \\
h^2_i &= e_{h^2_{i-1} || m_i} (h^1_{i-1} \oplus c) \oplus h^1_{i-1} \oplus c
\end{align*}
\]

- Collision requires $2^n$ operations assuming \(e(\cdot, \cdot)\) is ideal cipher
- AES-256, hash rate 1/3, security level $2^{128}$ for collisions
Hirose’s double block mode, figure

\[ \begin{align*}
  &\text{e} &\rightarrow & h_i^1 \\
  &m_i &\rightarrow & h_i^{2-1} \\
  &c &\rightarrow & h_i^2
\end{align*} \]

Ideal cipher model?

- proofs in model give protection against generic attacks
- no real-life cipher is an ideal cipher; “nearly ideal” cipher can be strong for encryption but very weak when used for hashing
- attacker in control of key can invest time in finding key(s) with certain properties

Known-key distinguishers - Knudsen, Rijmen 2007

- Block cipher cryptanalysis with applications to hash functions
- With a given (random) key, produce set of texts with “non-random” statistical behaviour
- Most short-cut attacks on block ciphers exploit statistical properties of plain- and ciphertexts in (reduced) cipher
- If such properties cannot be found given the key, it seems unlikely that they can be found when not given the key

Known-key distinguishers - examples

- Example 1. Generic 7-round Feistel cipher.
  - given a key, one can find (in time $O(1)$) two texts such that
  \[ \Delta(\delta, \alpha) \rightarrow \Delta(\delta, \beta) \]

- Example 2. AES reduced to seven rounds
  - given a key, one can find $2^{56}$ texts balanced in all bytes of plain- and ciphertexts
Known-key distinguishers

- **DES:**
  - key-recovery attack, $2^{43}$ known texts
  - collision attack, $2^{32}$ operations (best known)

- **SHACAL-1:**
  - block cipher built from SHA-1
  - 160-bit blocks, 512-bit keys
  - best known attacks today:
    - key-recovery attack on SHACAL-1 has complexity $\approx 2^{500}$
    - collision attack on SHA-1 has complexity $\approx 2^{60}$

---

**SMASH - Knudsen, 2005**

- Idea: build collision-resistant hash function from one bijective mapping
- Why? we know how to make one, strong bijective mapping (Not a family of bijections !?)
- let $f$ be a strong, bijective mapping of sufficient size
  
  $h(h_{i-1}, m_i) = f(m_1 + h_{i-1}) + m_1 + \theta h_{i-1}$

- Compression function **not** collision-resistant
- 2nd preimages in $2^{n/2}$ operations
- Proposal broken by Rijmen, Rechberger, Pramstaller, 2005

---

**Grindahl - Knudsen, Rechberger, Thomsen 2007**

- Daemen-style hash construction, sponge
- Iterated hash function
- “Rijndael”-state, $4 \times 13$ byte-matrix
- MixColumns, SubBytes same as for AES
- Compression function invertible
- Meet-in-the-middle preimage attack with birthday attack complexity
- Short-cut attack, Peyrin 2007
Hash based on fixed functions

- Preneel, 1992
- Black et al, 2005: Provably secure (collision-resistant) iterated hash functions based on one bijective mapping do not exist (information-theoretic setting)
- Shrimpton-Stam, 2006:
  - let $f_1, f_2, f_3$ be three, distinct functions, then define:
    \[ h(h_{i-1}, m_i) = f_1(m_1) + f_3(f_1(m_1) + f_2(h_{i-1})) \]
  - collisions $\Theta(2^{n/2}/n)$, preimages suboptimal ($2^{2n/3}$)
- Rogaway-Steinberger, 2008
  - at least three bijections needed
  - at least five bijections needed in double-block hash mode

SHA-3 - Call for candidates

- announcement: October 29, 2007
- must provide digests of 224, 256, 384, and 512 bits, not 160.
- available worldwide royalty-free, no IPR
- capable of protecting sensitive information for decades
- should be suitable for
  - digital signatures, FIPS 186-2
  - HMAC, FIPS 198
  - key establishment, SP 800-56A
  - random number generation, SP 800-90
- security strength at least that of the SHA-2s with significantly improved efficiency

SHA-3 - Desirable properties

- efficient integral options, e.g., randomized hashing, that "fundamentally improve security"
- parallelizable
- avoid "generic properties" of Damgård/Merkle constructions
- attack on SHA-2 should not lead to attack on SHA-3
- flexible for a wide variety of implementations
- a single family, except if good arguments for more families
- tunable security parameter, e.g., number of rounds, with recommendations
SHA-3 - Security

Message digest of $n$ bits
- Collisions should take $2^{n/2}$
- Preimages should take $2^n$
- 2nd preimages should take $2^{n-k}$ for messages shorter than $2^k$ bits

Higher levels of security against 2nd preimage will be viewed positively
- NIST open to other designs than Damgård/Merkle

SHA-3 - Timeline

- hard submission deadline: 31/10-2008
- submissions by 31/8-2008 checked by NIST for inconsistencies
- Round 2: $\approx 5$ candidates selected. 12-15 months. Tweaks allowed. Workshop 3.
- AHS(s).
- documentation and testing like AES
- review is public

Outtro

- Hash functions are important for many things in cryptology and we are asking for very strong properties
- No apparent reason why such functions can/should be very fast... ?
- NIST do not really invite for block cipher based proposals
- NIST: “a successful collision attack on an algorithm in the SHA-2 family could have catastrophic effects for digital signatures”
- So better not make a hash of it...