Double-Length Hash Functions with Birthday PRO Security in the Ideal Cipher Model

Yusuke Naito
Mitsubishi Electric Corporation

This talk was presented at SAC 2011
Hash Function

- Hash function: \(\{0,1\}^* \rightarrow \{0,1\}^n\).

- Hash functions are used as:
  - Random Oracle instantiation
  - HMAC
  - Pseudorandom Function
  - ...

- For example, fixed lengths used are 256-bit, 512-bit.
Hash Security

- **Preimage Resistance**
  given \( z \), hard to find \( M \) s.t. \( z = H(M) \)

- **Second Preimage Resistance**
  given \( M \), hard to find \( M' \)
  s.t. \( H(M) = H(M') \) and \( M \neq M' \)

- **Collision Resistance**
  hard to find \( M, M' \)
  s.t. \( H(M) = H(M') \) and \( M \neq M' \)

- **Pseudorandom Oracle (indiff. from RO): Our Goal**
  Stronger property than CR, SPR and PR
Hash Function Design

Blockcipher-based hash and Permutation-based hash

Davies-Meyer Merkle-Damgard

Sponge
Blockcipher-based Double-Length Hash Function (DLHF)

- DLHF is constructed from an existing blockcipher (e.g., AES)

- The output length of blockciphers is too short.
  - e.g., AES (output length: 128 bit)

  ![Diagram](image)

  A collision of 128 bit hash is found with $2^{64}$ complexity

- DLHFs are designed so that the output length is twice of that of the blockcipher.
  - e.g., AES-based hash: the output length is 256 bit
Blockcipher-based Double-Length Hash Function (DLHF)

- Hirose’s scheme, Tandem-DM, Abreast-DM, MJH, MDC-2, ....

- DLHFs are useful on size restricted devices (e.g., RFID, IC card) when implementing both a hash function and a blockcipher. One has only to implement the blockcipher.

- DLHFs are designed from a single blockcipher.

- The security is proven in the ideal cipher model.
Example: Hirose’s Hash

- Constructed from a single blockcipher.
- The Davies-Meyer mode is used twice in one block.

![Diagram of Hirose’s Hash]

**Input**

- IV1
- IV2
- constant values

**Output**

- output1 || output2

**Details**

- 2n-bit
- n-bit
- e.g., AES-256
Ideal Cipher Model

An adversary (or distinguisher) can access to

(ideal) encryption oracle $E$
- query: plain text $x$, key $k$
- response: cipher text $y$

(ideal) decryption oracle $E^{-1}$
- query: cipher text $y$, key $k$
- response: plain text $x$
Ideal Cipher Model

forward procedure

on query \((x, k)\)
\[y \leftarrow_R \{0, 1\}^n\]
Ret \(y\)
Ideal Cipher Model

forward procedure

on query \((x,k)\)
\(y' \leftarrow \mathcal{R}\{0,1\}^n \setminus \{y\}\)
Ret \(y'\)
Ideal Cipher Model

Inverse procedure

on query \((y'',k)\)

\[ x'' \leftarrow_R \{0,1\}^n \setminus \{x,x'\} \]

Ret \(x''\)
Pseudorandom Oracle (PRO) or Indifferentiable from RO

$H^E$ is PRO if $\exists S$ s.t. $\forall D$: $|\Pr[D \Rightarrow 1 \text{ (left)}] - \Pr[D \Rightarrow 1 \text{ (right)}]| \leq \varepsilon$ ($\varepsilon$ is a negl. function for the security parameter)

- (Left) $D$ can make queries to $H$, $E$ and $E^{-1}$.
- (Right) $D$ can make queries to RO and $S$.
- $S$ simulates $E, E^{-1}$ by using RO.

PRO is the important security property
- the security of many cryptosystems is preserved when RO is replaced with $H^E$
  (e.g., IND-CCA security, EUF-CMA security and many others)
Birthday Pseudorandom Oracle Security

The PRO advantage $|\Pr[D \Rightarrow 1 \text{ (left)}] - \Pr[D \Rightarrow 1 \text{ (right)}]|$ is bounded by the birthday bound.

\[ e.g., \]
When $H^E: \{0,1\}^* \rightarrow \{0,1\}^{2n}$ and D can make $q$ queries, the birthday bound is $O(q^2/2^{2n})$.

The query complexity to be differentiable from RO with probability of $1/2$ is $O(2^n)$.
# Previous Security Results (Ideal Cipher Model)

<table>
<thead>
<tr>
<th></th>
<th>Collision Resistance</th>
<th>Pseudorandom Oracle (PRO)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dedicated Hash</td>
<td><img src="#" alt="Circle" /></td>
<td><img src="#" alt="Circle" /></td>
</tr>
<tr>
<td></td>
<td>birthday security</td>
<td>beyond birthday security</td>
</tr>
<tr>
<td>Double-Length Hash</td>
<td><img src="#" alt="Circle" /></td>
<td><img src="#" alt="Triangle" /></td>
</tr>
<tr>
<td>(from a single practical size blockcipher)</td>
<td>not achieve birthday security</td>
<td></td>
</tr>
</tbody>
</table>
Previous Results of Blockcipher-based DLHF

There is no double-length hash function constructed from a single practical size blockcipher and achieving birthday PRO-security.

<table>
<thead>
<tr>
<th>Security</th>
<th>PRO</th>
<th>Collision Resistance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hirose Tandem-DM Abreast-DM ...</td>
<td>☒</td>
<td>☀</td>
</tr>
<tr>
<td>prefix-free Merkle-Damgård using PBGV</td>
<td>☐($O(2^{n/2})$)</td>
<td>☀</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>blockcipher</th>
<th>key size</th>
<th>output size</th>
<th>hash size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hirose Tandem-DM Abreast-DM ...</td>
<td>2n</td>
<td>n</td>
<td>2n</td>
</tr>
<tr>
<td>prefix-free Merkle-Damgård using PBGV</td>
<td>2n</td>
<td>n</td>
<td>2n</td>
</tr>
</tbody>
</table>

The size is supported by AES-256.
Our double-length hash functions can be constructed from a single practical size blockcipher and achieves the birthday PRO security!

<table>
<thead>
<tr>
<th>Our Shemes</th>
<th>Security</th>
<th>blockcipher</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PRO</td>
<td>Collision Resistance</td>
<td>key size</td>
<td>output size</td>
<td>hash size</td>
</tr>
<tr>
<td>Our Shemes</td>
<td>$O(2^n)$</td>
<td>$\bigcirc$</td>
<td>$2n$</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
<tr>
<td>Hirose Tandem-DM Abreast-DM</td>
<td>$\times$</td>
<td>$\bigcirc$</td>
<td>$2n$</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
<tr>
<td>prefix-free Merkle-Damgård</td>
<td>$\bigcirc$</td>
<td>$\bigcirc$</td>
<td>$2n$</td>
<td>$n$</td>
<td>$2n$</td>
</tr>
</tbody>
</table>

The size is supported by AES-256
Our Double-Length Hash Functions

- Constructed from a single blockcipher such as AES-256

- Strengthened Merkle-Damgård + post-processing function (SMD) (PPF)

Input via suffix free padding


IV CF CF CF

c1 c2 rv1 rv2

rv1||rv2 is the output

Hirose Tandem-DM Abreast-DM
DLHF using Hirose’s Scheme

Strengthened Merkle-Damgård post-processing function

$E_{c1} \oplus E_{c2}$

$c1, c2$ are the outputs

$E_{rv1} \oplus E_{rv2}$

$rv1 \parallel rv2$ is the output

$E$ is the encryption function

$M[1], M[2], M[i]$ are the input messages

$IV1, IV2$ are the initialization vectors

2n bits

n bits
The query complexity to be differentiable from RO with probability 1/2 is $O(2^n)$.

Our DLHFs achieve the birthday PRO-security!
Step 1:
Compression functions of Hirose’s scheme, Tandem-DM, and Abreast-DM are Preimage Aware (PrA)
⇒ The following NMAC hash function is PRO

$$\text{compression function (CF)}: \text{Hirose, Tandem-DM, Abreast-DM}$$
Step 1 (outline)

- The PrA security of Hirose, Tandem-DM, Abreast-DM = Collision Resistant (CR) + Preimage Resistant (PR)
  - birthday security ($O(2^n)$)
  - beyond birthday security ($O(2^{2n})$)
  (Since the PrA notion is complex, the detail is skipped)

- The following NMAC hash functions satisfy birthday PRO security ($O(2^n)$)
**Step 2**

Step 2:
The post-processing function (PPF) is $\text{PRO} \implies \text{RO}$ can be replaced by the PPF.

\[
\begin{align*}
& \text{M}[1] \quad \text{M}[2] \\
& \text{M}[i] \\
& \text{RO} \quad \text{RO} \\
& \text{E}' \quad \text{E}' \\
& \text{c}_1 \quad \text{c}_2 \\
& \text{rv}_1 \quad \text{rv}_2
\end{align*}
\]
Step 2 (intuition)

rv1 is randomly chosen from \(\{0,1\}^n\)
rv2 is randomly chosen from \(\{0,1\}^n \setminus \{rv1\}\)

Birthday security

Since PPF: \(rv1 \neq rv2\),
if RO: \(rv1 \neq rv2\), then PPF is RO
\[\Rightarrow \text{birthday PRO security (O}(2^n))\]
Result from Step 2

\[ \text{CF} \quad \text{M}[1] \quad \text{M}[2] \quad \text{IV} \quad \text{CF} \quad \text{CF} \quad \text{RO} \quad \text{Birthday security} \quad \text{M}[1] \| \text{M}[2] \| \ldots \| \text{M}[i] \quad \text{RO} \]

\[ \text{Birthday security} \quad \text{Birthday security} \quad \text{Birthday security} \]

\[ \text{CF} \quad \text{M}[1] \quad \text{M}[2] \quad \text{IV} \quad \text{CF} \quad \text{CF} \quad \text{CF} \quad \text{C} \quad \text{C} \quad \text{E}' \quad \text{E}' \quad \text{rv1} \quad \text{rv2} \quad \text{Birthday security} \]
Step 3:
The single-blockcipher based DLHF (our DLHF) is indifferentiable from two-blockcipher based DLHF.
Step 3 (intuition)

Since the output of $E$ is almost (n-bit) random, the complexity that a random value is equal to $c_1$ or $c_2$ is $O(2^n)$.
Result from Step 3
Conclusion

First time DLHFs

- achieve birthday PRO security
- constructed from a single practical size blockcipher such as AES-256
Thank you for your attention!