Second Preimage Attacks on Dithered Hash Functions

Elena Andreeva\textsuperscript{1}  
Charles Bouillaguet\textsuperscript{2}  
Pierre-Alain Fouque\textsuperscript{2}  
Jonathan J. Hoch\textsuperscript{3}  
John Kelsey\textsuperscript{4}  
Adi Shamir\textsuperscript{2,3}  
Sebastien Zimmer\textsuperscript{2}

\textsuperscript{1}K.U. Leuven, ESAT/COSIC, Leuven-Heverlee, Belgium  
\textsuperscript{2}École Normale Supérieure, Paris, France  
\textsuperscript{3}Weizmann Institute of Science, Rehovot, Israel  
\textsuperscript{4}NIST, Gaithersburg, MD, USA

EUROCRYPT 2008
Iterated Hash Functions

Hash Functions Cryptanalysis

\[ H : \{0, 1\}^* \mapsto \{0, 1\}^n \]

Should behave “like a random oracle”.
Hash Functions Cryptanalysis

\[ H : \{0, 1\}^* \mapsto \{0, 1\}^n \]

Should behave “like a random oracle”.

**Collision attack** Find \( M_1 \neq M_2 \) s.t. \( H(M_1) = H(M_2) \).
Ideal security: \( 2^{n/2} \).

**Second-preimage attack** Given \( M_1 \), find \( M_2 \neq M_1 \) s.t. \( H(M_1) = H(M_2) \).
Ideal security: \( 2^n \).

**Preimage attack** Given \( y \), find \( M \) s.t. \( H(M) = y \).
Ideal security: \( 2^n \).
Hash Functions Cryptanalysis

\[ H : \{0, 1\}^* \mapsto \{0, 1\}^n \]

Should behave “like a random oracle”.

**Collision attack** Find \( M_1 \neq M_2 \) s.t. \( H(M_1) = H(M_2) \).
Ideal security: \( 2^{n/2} \).

**Second-preimage attack** Given \( M_1 \), find \( M_2 \neq M_1 \) s.t.
\[ H(M_1) = H(M_2) \]
Ideal security: \( 2^n \).

**Preimage attack** Given \( y \), find \( M \) s.t. \( H(M) = y \).
Ideal security: \( 2^n \).
Iterated Hash Functions

The Merkle-Damgård Mode of Operation

Most hash functions are **iterated** hash functions:

- Split $M$ into $m$-bit blocks: $M = m_0, m_1, \ldots, m_r$
- Pad the last block (include binary encoding of $|M|$)
- Iterate a compression function $f : \{0, 1\}^{n+m} \to \{0, 1\}^n$

\[
\begin{align*}
IV & \xrightarrow{f} h_1 \\
        & \xrightarrow{f} h_2 \\
        & \xrightarrow{f} h_3 \\
        & \xrightarrow{f} H(M)
\end{align*}
\]
A full hash function is made of

- A compression function
- A mode of operation (i.e., a way of using it)

In this talk

Attacks against the mode of operation

- Works for all $f$: generic attacks
- Model $f$ as a Random Oracle
- Collisions on $f$ cost $2^{n/2}$
Joux’s Multicollision [CRYPTO’04]  
Towards the First Generic Second Preimage Attack

For the cost of $k$ collisions, we can build a $2^k$-multicollision

- At each step, find a colliding block pair starting from the last chaining value
- $2^k$ paths between IV and $h_k$

Works because of the \textit{iterated} structure of $H$!
Kelsey & Schneier Second Preimage Attack [EUROCRYPT’05]

At step $i$, find a collision between a 1-block message and a $(2^i + 1)$-block message

- $|m_1| = 2$
- $|m_2| = 3$
- $|m_3| = 5$

Messages of sizes $[k + 1; 2^{k+1} - 2]$ that hash to $h_k$ ⇒ expandable message

How to use this?
1. Generate an **Expandable Message** $\mathcal{M}$ that hashes to $h_\mathcal{M}$.
Kelsey & Schneier Second Preimage Attack (Cont’d)

1. Generate an **Expandable Message** $\mathcal{M}$ that hashes to $h_{\mathcal{M}}$
2. Find a message block $B$ “connecting” $h_{\mathcal{M}}$ to $M$

\[ f(h_{\mathcal{M}}, B) = h_i \]
1. Generate an Expandable Message $\mathcal{M}$ that hashes to $h_\mathcal{M}$
2. Find a message block $\mathcal{B}$ “connecting” $h_\mathcal{M}$ to $M$
3. Using $\mathcal{M}$, build $P$ of length $i - 1$ that hashes to $h_\mathcal{M}$
Kelsey & Schneier Second Preimage Attack (Cont’d)

1. Generate an Expandable Message $M$ that hashes to $h_M$
2. Find a message block $B$ “connecting” $h_M$ to $M$
3. Using $M$, build $P$ of length $i - 1$ that hashes to $h_M$
4. Assemble all pieces to form a second preimage $M'$

$$f(h_M, B) = h_i$$

$$M' = P \cdot B \cdot M|_{\geq i}$$

$$H(M) = H(M')$$

$$|M'| = |M|$$

$$M' \neq M$$
Cost of the attack:

- **Build Expandable Message $\mathcal{M}$**
  - $k$ collisions
  - $2^k \geq |\mathcal{M}|$
  - Cost: $k \cdot 2^{n/2}$

- “Connect” $h_{\mathcal{M}}$ to target message (i.e., find $\mathcal{B}$).
  - Cost: $2^n/|\mathcal{M}|$.

⇒ If $|\mathcal{M}| = 2^k$, total cost: $k \cdot 2^{n/2} + 2^{n-k}$

  - SHA-1 ($k = 55$, $n = 160$), total cost: $2^{106}$
Cost of the attack:

▶ Build Expandable Message $\mathcal{M}$
  ▶ $k$ collisions
  ▶ $2^k \geq |\mathcal{M}|$
  ▶ Cost: $k \cdot 2^{n/2}$

▶ “Connect” $h_\mathcal{M}$ to target message (i.e., find $\mathbb{B}$).
  ▶ Cost: $2^n/|\mathcal{M}|$.

⇒ If $|\mathcal{M}| = 2^k$, total cost: $k \cdot 2^{n/2} + 2^{n-k}$
  ▶ SHA-1 ($k = 55$, $n = 160$), total cost: $2^{106}$

Conclusion

There is a problem with the Merkle-Damgård mode of operation
Several new modes of operation recently suggested to replace MD.

- Some prevent the 2nd Preimage attack with dithering.
  - Perturb the hash process
  - new input from a fixed dithering sequence $z$.
- HAIFA : dithering with a 64-bit counter
- Rivest : dithering with 2-bit symbols
  (Proposed at the 1st NIST Hash Workshop)
Rivest’s Dithering Proposal

Description

Dithering with a \textit{repetition-free} sequence on 4 letters:

\[ z = abcacdcbcdcadcdbcdbdabacabadbabcdbcba \ldots \]

- no square in sequence
  - square: bana.na

- Perturbs construction of the Expandable Message
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcadcdbcdcadcdbcdbacbabcdbdbcba \ldots \]

- Need to choose/fix dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcadcdbcdcadcdbdabacabdbabcdcbca \ldots \]

- Need to **choose/fix** dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = ab\textit{c}acdc\textit{b}cdcad\textit{c}d\textit{b}dab\textit{a}c\textit{b}\textit{a}b\textit{d}b\textit{d}bc\textit{b}a \ldots \]

- Need to **choose/fix** dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcacdcbcdcadcdbcda \ldots \]

- Need to choose/fix dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcacdcbcdcadcdbhdbabcbdcba \ldots \]

- Need to choose/fix dithering symbols when building \( \mathcal{M} \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7. \) \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8. \) \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcacdcbcd cadcdbcdbacabdbabcbdbcba \ldots \]

- Need to **choose/fix** dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abc\text{ac}dc\text{bc}d\text{c}cadcdbdabacabadb\text{abc}dbd\text{bc}ba \ldots \]

- Need to choose/fix dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcacdcbcdcadcdbgabacabadbabcdbcbca \ldots \]

- Need to choose/fix dithering symbols when building \( M \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7. P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8. P = m'_1.m'_2.m_3 \)
Rivest’s Dithering Proposal
Effectiveness

\[ z = abcacdcbcdcadcdbcdabacabdbabcdbcba \ldots \]

- Need to choose/fix dithering symbols when building \( \mathcal{M} \)
- How? Need to match the actual sequence...
  - e.g. \( \ell = 7 \). \( P = m_1.m'_2.m_3 \)
  - e.g. \( \ell = 8 \). \( P = m'_1.m'_2.m_3 \)

Conclusion

Kelsey and Schneier’s attack does not work with dithering
The new attack relies on the diamond structure from the herding attack of Kelsey and Kohno [EUROCRYPT’06].

- Complete binary tree of height $\ell$
- Node $\simeq$ chaining values
- Edges $\simeq$ message blocks
The “Diamond” Structure

The new attack relies on the diamond structure from the herding attack of Kelsey and Kohno [EUROCRYPT’06].

- Complete binary tree of height \( \ell \)
- Node \( \simeq \) chaining values
- Edges \( \simeq \) message blocks
- Collision tree
- Maps \( 2^\ell \) chaining values to \( h_\diamond \) (paths of \( \ell \) blocks in the tree)

\[
f(x_5, m) = f(x_6, m') = x_2
\]
The "Diamond" Structure

The new attack relies on the diamond structure from the herding attack of Kelsey and Kohno [EUROCRYPT’06].

- Complete binary tree of height $\ell$
- Node $\simeq$ chaining values
- Edges $\simeq$ message blocks
- Collision tree
- Maps $2^\ell$ chaining values to $h_\Diamond$ (paths of $\ell$ blocks in the tree)
- Built in time $2^{n/2+\ell/2+2}$

\[ f(x_5, m) = f(x_6, m') = x_2 \]
Putting the “Diamond” at Work

Replaying Kelsey and Schneier’s attack, but with a diamond

\[ \begin{align*}
2\ell & \\
\{ x_3, x_4, x_5, x_6 \} & \rightarrow h_0 \rightarrow H(M)
\end{align*} \]
Putting the “Diamond” at Work

Replaying Kelsey and Schneier’s attack, but with a diamond

\[ f(h_\diamond, B_1) = h_i \]
Putting the “Diamond” at Work

Replaying Kelsey and Schneier’s attack, but with a diamond

\[
f(h_\Diamond, B_1) = h_i\]
Putting the “Diamond” at Work

Replaying Kelsey and Schneier’s attack, but with a diamond

- Build diamond
- Connect $h_\Diamond$ to $M$
- Choose prefix $P$
- Connect $P$ to a leaf $x_j$
A New Generic Second Preimage Attack against plain-MD

Putting the “Diamond” at Work

Replaying Kelsey and Schneier’s attack, but with a diamond

- Build diamond
- Connect $h_\Diamond$ to $M$
- Choose prefix $P$
- Connect $P$ to a leaf $x_j$
- Assemble parts
How much does this cost? Assume $|M| = 2^k$.

1. Build diamond: $2^{n/2+\ell/2+2}$
2. Connect $h_\Diamond$ to $M$: $2^{n-k}$
3. Generate $P$: free
4. Connect $h_P$ to Diamond: $2^{n-\ell}$
5. Assemble parts: free

Total: $2^{n/2+\ell/2+2} + 2^{n-k} + 2^{n-\ell}$

Take $\ell \approx n/3$. Complexity becomes $\approx 5 \cdot 2^{2n/3} + 2^{n-k}$

SHA-1 ($n = 160$, $k = 55$, $\ell = 53$): complexity $= 2^{109.5}$
How To Cope With Rivest’s Dithering?

\[ z = abcacdcbcdcadcdbhacababacbdcdbcba \ldots \]

**Question**

How does this affect the attack?

\[ \implies \text{We have to fix dithering symbols:} \]

1. Inside the diamond
2. When connecting \( h_\diamond \) to \( M \)

**Key Ideas**

- Fix a dithering symbol for each level of the diamond
  \[ \omega_i \text{ at level } i \quad (1 \leq i \leq \ell) \]
- **guess** the right symbol \( (\omega_{\ell+1}) \) for the connection
How To Cope With Rivest’s Dithering (cont’d) ?

```
H(M) = \mu \mu_1 \ldots \mu_{\ell - 1} \mu_\ell
```

IV → M → H(M)

abcacdcbcdcadcdabcababcdbabcbacbcdbacbcdbcacba ...
How To Cope With Rivest’s Dithering (cont’d)?

\[ f \left( h_\diamond, \omega_{\ell+1}, \mathbb{B}_1 \right) = h_i \]
How To Cope With Rivest’s Dithering (cont’d)?

\[ f(h_\diamond, \omega_{\ell+1}, B_1) = h_i \]

must be the same

\[ \omega_{\ell+1} \]

What if \( \omega \) does not match \( z \)?

⇒ Diamond does not converge!

⇒ Connection fails!

\[ f(h_\diamond, z_i, B_1) \neq h_i \]
How To Cope With Rivest’s Dithering (cont’d)?

With Dithering

\[
\begin{align*}
\omega_1 & \quad \ldots \quad \omega_{\ell-1} \quad \omega_\ell \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
x_3 & \quad x_4 & \quad x_5 & \quad x_6 & \quad x_1 \\
\omega_\ell & \quad \vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
\end{align*}
\]

must be the same

\[
f \left( h_\diamond, \omega_{\ell+1}, B_1 \right) = h_i
\]

must be the same

IV \quad M \quad h_i \quad H(M)

\[abcacdcbcdcadcldbabacabadbabcdbbcbacbcdcacba \ldots\]
How To Cope With Rivest’s Dithering (cont’d) ?

What if $\omega$ does not match $z$ ?

$\implies$ Diamond does not converge !

$\implies$ Connection fails !

$f (h_\diamond, z_i, B_1) \neq h_i$

must be the same

IV $\xrightarrow{}$ M $\xrightarrow{}$ $h_i$ $\xrightarrow{}$ $H(M)$

$abcacdcbcdcadcdbcdbdabacabadbabcdbdbcbaabcacdcacba$
How To Cope With Rivest’s Dithering? (end)

With dithering, the diamond (and connection) only works at certain positions, where $\omega_1(\ell+1)$ matches $z$.

**Question**

How to choose $\omega$? Probability that $\omega$ matches $z$ where $B_1$ connects?

**(Partial) Answer**

Depends on $z$.
- Should choose a frequently-occurring factor of $z$
- Probability depends on how often it appears in $z$

**Attack?**

Could there be frequently-occurring factors in $z$?
Analysis of Rivest’s dithering sequence
Or: How a Cryptanalyst Becomes a Sequence-Theorist for a While

Answer: YES

Theorem (Cobham, 1972, “Uniform Tag Sequences”)
The number of different factors of size s in z is linear in s

- There is a very low number of different factors in z
  \[ \implies \text{so at least one of them occur frequently.} \]
- Would have been exponential for a pseudo-random sequence...

Before, for SHA-1, we chose \( \ell = 53 \)

- How many factors of size 54 in z? 772!
- Careful choice of \( \omega \):
  \[ \implies \text{Each connecting block } B_1 \text{ works with probability } \geq 2^{-9} \]
  \[ \implies \text{Just repeat the attack } 2^9 \text{ times!} \]
With Dithering

**Complexity**

Same as before, except that many wrong connecting blocks $B_1$ will be found before $\omega$ matches $z$.

$$2^{n/2+\ell/2+2} + \text{Fact}_z(\ell + 1) \cdot 2^{n-k} + 2^{n-\ell}$$

For comparison with SHA-1, we take $n = 160$ and $k = 55$.

<table>
<thead>
<tr>
<th>Hash function</th>
<th>$\ell$</th>
<th>$\text{Fact}(\ell + 1)$</th>
<th>SHA-1</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain-MD</td>
<td>55</td>
<td></td>
<td>$2^{109.5}$</td>
<td>$5 \cdot 2^{2n/3} + 2^{n-k}$</td>
</tr>
<tr>
<td>Keräänen-Rivest</td>
<td>52</td>
<td>748</td>
<td>$2^{115.5}$</td>
<td>$(k + 40.5) \cdot 2^{n-k+3}$</td>
</tr>
<tr>
<td>Concrete-Rivest</td>
<td>52</td>
<td>33176</td>
<td>$2^{121}$</td>
<td>$2^{n-k+15}$</td>
</tr>
<tr>
<td>Shoup’s UOWHF</td>
<td>53</td>
<td>small</td>
<td>$2^{112}$</td>
<td>$(2k + 3) \cdot 2^{n-k}$</td>
</tr>
</tbody>
</table>

- **Keräänen-Rivest** is what was described before
- **Concrete-Rivest** is Rivest’s “concrete proposal”
  (similar to Keräänen-Rivest, but include a 13-bit counter)
- **Shoup’s UOWHF** was presented at [EUROCRYPT’2000]
Known generic second preimage attacks are long messages attacks.
Possible to find a 2nd preimage of one out of many small messages.

Connection step:
- many small messages \( \simeq \) one big message
  \( \Rightarrow \) Target all of them at the same time.

\[
\begin{align*}
\text{IV} & \quad M_1 \quad H(M_1) \\
\text{IV} & \quad M_2 \quad H(M_2) \\
\text{IV} & \quad M_3 \quad H(M_3)
\end{align*}
\]
Known generic second preimage attacks are long messages attacks. Possible to find a 2nd preimage of one out of many small messages. 

Connection step:
- many small messages $\sim$ one big message
  $\Rightarrow$ Target all of them at the same time
- Now we can find a second preimage of $M_2$!
Faster Second Preimages With (quite a lot) More Precomputation

Hardest step: the connection. Let \( g(\mathbb{B}) = f(h_M, \mathbb{B}) \).

We need to find \( g^{-1} \) for one of the \( h_i \).

- Variation of Hellman's Time-Memory Tradeoff (\( 2^n \) precomputation)
- Also works with shorter messages!

<table>
<thead>
<tr>
<th>range of ( k )</th>
<th>Memory</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k \leq n/4 )</td>
<td>( 2^{2/3(n-k)} )</td>
<td>( 2^{2/3(n-k)} )</td>
</tr>
<tr>
<td>( n/4 \leq k \leq n/2 )</td>
<td>( 2^{n/2} )</td>
<td>( 2^{n/2} )</td>
</tr>
</tbody>
</table>
Conclusion

▶ New generic second preimage attack
  ▶ About the first half of the preimage can be chosen
▶ Attack works in the presence of dithering
  ▶ Rivest’s proposal(s) are broken
  ▶ First Attack on Shoup’s UOWHF, ROX, …
▶ Various extensions of both new and existing attacks
  ▶ Apply attack to collection of small messages
  ▶ Various possibilities for a Time-Memory Tradeoff
▶ Attack is not applicable to HAIFA…