On Chosen Target Forced Prefix preimage-resistance

Michal Rjaško

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics
Department of Computer Science
rjasko@dcs.fmph.uniba.sk

June 23, 2009
Hash function basics

- Hash function, in general, is a function $F : \mathcal{M} \rightarrow \mathcal{Y}$, where $|\mathcal{M}| >> |\mathcal{Y}|$.
  - We consider only functions $F : \{0, 1\}^* \rightarrow \{0, 1\}^y$.

- Cornerstones of current cryptography due to many applications:
  - Digital signatures
  - Message authentication
  - Password storage
  - Data integrity
  - ...

- Many different applications bring many different security properties that cryptographic hash functions should preserve.
Hash function basics

- Hash function, in general, is a function $F : \mathcal{M} \rightarrow \mathcal{Y}$, where $|\mathcal{M}| >> |\mathcal{Y}|$.
  - We consider only functions $F : \{0, 1\}^* \rightarrow \{0, 1\}^y$.

- Cornerstones of current cryptography due to many applications:
  - Digital signatures
  - Message authentication
  - Password storage
  - Data integrity
  - ...

- Many different applications bring many different security properties that cryptographic hash functions should preserve.
Hash function properties

Basic properties of every “good” hash function:

- **Preimage resistance**
  - for given image $Y$ it is hard to find message $M$: $F(M) = Y$.

- **Second-preimage resistance**
  - for given message $M$ it is hard to find message $M' \neq M$: $F(M) = F(M')$.

- **Collision resistance**
  - it is hard to find two different messages $M, M'$: $F(M) = F(M')$. 
Hash function family

\( H : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^y. \)

- Preimage resistance:
  \( \text{Adv}_{H}^{\text{Pre}[\lambda]}(A) = \Pr \left[ K \leftarrow K; M \leftarrow \{0, 1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(K, Y) : H_K(M') = Y \right] \)

- Second-preimage resistance:
  \( \text{Adv}_{H}^{\text{Sec}[\lambda]}(A) = \Pr \left[ K \leftarrow K; M \leftarrow \{0, 1\}^\lambda; M' \leftarrow A(K, M) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \)

- Collision resistance:
  \( \text{Adv}_{H}^{\text{Coll}}(A) = \Pr \left[ K \leftarrow K; (M, M') \leftarrow A(K) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \)

Everywhere and always versions – maximize the advantage over all keys (always) or messages (everywhere) – aPre, ePre, aSec, eSec

- maximizing Coll advantage makes no sense
Hash function family

\[ H : \{0, 1\}^k \times \{0, 1\}^* \to \{0, 1\}^y. \]

- Preimage resistance:
  \[ \text{Adv}_{H}^{\text{Pre}[\lambda]}(A) = \Pr \left[ K \overset{\$}{\leftarrow} \mathcal{K}; M \overset{\$}{\leftarrow} \{0, 1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(K, Y) : H_K(M') = Y \right] \]

- Second-preimage resistance:
  \[ \text{Adv}_{H}^{\text{Sec}[\lambda]}(A) = \Pr \left[ K \overset{\$}{\leftarrow} \mathcal{K}; M \overset{\$}{\leftarrow} \{0, 1\}^\lambda; M' \leftarrow A(K, M) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \]

- Collision resistance:
  \[ \text{Adv}_{H}^{\text{Coll}}(A) = \Pr \left[ K \overset{\$}{\leftarrow} \mathcal{K}; (M, M') \leftarrow A(K) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \]

Everywhere and always versions – maximize the advantage over all keys (always) or messages (everywhere) – aPre, ePre, aSec, eSec

- maximizing Coll advantage makes no sense
Hash function family

\[ H : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^y. \]

- Preimage resistance:
  \[ \text{Adv}^{\text{Pre}[\lambda]}_H(A) = \Pr \left[ K \leftarrow \mathcal{K}; M \leftarrow \{0, 1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(K, Y) : H_K(M') = Y \right] \]

- Second-preimage resistance:
  \[ \text{Adv}^{\text{Sec}[\lambda]}_H(A) = \Pr \left[ K \leftarrow \mathcal{K}; M \leftarrow \{0, 1\}^\lambda; M' \leftarrow A(K, M) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \]

- Collision resistance:
  \[ \text{Adv}^{\text{Coll}}_H(A) = \Pr \left[ K \leftarrow \mathcal{K}; (M, M') \leftarrow A(K) : (M \neq M') \land (H_K(M) = H_K(M')) \right] \]

Everywhere and always versions – maximize the advantage over all keys (always) or messages (everywhere) – \( a\text{Pre}, e\text{Pre}, a\text{Sec}, e\text{Sec} \)

- maximizing Coll advantage makes no sense
Nostradamus attack

Created by Kelsey and Kohno, 2006.

- Applies to Merkle-Damgård hash functions.
- Attack scenario (let $F$ be some hash function):
  1. Nostradamus provides a hash $Y$ of some predictions, e.g. closing stock prices of S&P500.
  2. The prices become public.
  3. Nostradamus has to publish a message $M$ containing the exact closing prices and possibly some other (uncertain) predictions, where $F(M) = Y$. 
Nostradamus attack

Created by Kelsey and Kohno, 2006.

- Applies to Merkle-Damgård hash functions.
- Attack scenario (let $F$ be some hash function):
  1. Nostradamus provides a hash $Y$ of some predictions, e.g. closing stock prices of S&P500.
  2. The prices become public.
  3. Nostradamus has to publish a message $M$ containing the exact closing prices and possibly some other (uncertain) predictions, where $F(M) = Y$. 
Nostradamus attack

Nostradamus

image Y

suffix M

prefix P

Challenger

prefix P

suffix M

F

? image Y
Chosen Target Forced Prefix preimage resistance

- Property that guarantees h.f. security against Nostradamus attack
- Chosen Target – the image $Y$; Forced Prefix – the prefix $P$
- Formal definition:

$$\text{Adv}_{H}^{\text{CTFP}[\lambda]}(A) = \Pr \left[ K \leftarrow \mathcal{K}; (Y, S) \leftarrow A(K); P \leftarrow \{0, 1\}^\lambda; \\ M \leftarrow A(P, S) : H_K(P \parallel M) = Y \right]$$

- always CTFP preimage resistance:

$$\text{Adv}_{H}^{\text{aCTFP}[\lambda]}(A) = \max_{K \in \mathcal{K}} \left( \Pr \left[ (Y, S) \leftarrow A; P \leftarrow \{0, 1\}^\lambda; \\ M \leftarrow A(P, S) : H_K(P \parallel M) = Y \right] \right)$$
Chosen Target Forced Prefix preimage resistance

- Property that guarantees h.f. security against Nostradamus attack
- Chosen Target – the image $Y$; Forced Prefix – the prefix $P$
- Formal definition:

$$\text{Adv}_{H}^{\text{CTFP}[\lambda]}(A) = \Pr \left[ K \xleftarrow{\$} \mathcal{K}; (Y, S) \leftarrow A(K); P \xleftarrow{\$} \{0, 1\}^{\lambda};
M \leftarrow A(P, S) : H_{K}(P || M) = Y \right]$$

- always CTFP preimage resistance:

$$\text{Adv}_{H}^{a\text{CTFP}[\lambda]}(A) = \max_{K \in \mathcal{K}} \left( \Pr \left[ (Y, S) \leftarrow A; P \xleftarrow{\$} \{0, 1\}^{\lambda};
M \leftarrow A(P, S) : H_{K}(P || M) = Y \right] \right)$$
Other properties analyzed in our work:

- Message authentication codes (unforgeability) (MAC):
  \[ \text{Adv}^{\text{MAC}}_H (A) = \Pr [K \leftarrow \mathcal{K}; (M, Y) \leftarrow A^{H_K} : H_K(M) = Y \land M \text{ not queried}] \]

- Pseudo random function (Prf):
  \[ \text{Adv}^{\text{Prf}}_H (A) = \left| \Pr [K \leftarrow \mathcal{K}; 1 \leftarrow A^{H_K(\cdot)}] - \Pr [f \leftarrow \text{Func}(\mathcal{M}, \mathcal{Y}); 1 \leftarrow A^f] \right| \]

- Pseudo random oracle (Pro):
  \[ \text{Adv}^{\text{Pro}}_{H, f, S} (A) = \left| \Pr [K \leftarrow \mathcal{K}; 1 \leftarrow A_{H_K(\cdot), f(\cdot)}(K)] - \Pr [K \leftarrow \mathcal{K}; \mathcal{F} \leftarrow \text{Func}(\mathcal{M}, \mathcal{Y}); 1 \leftarrow A^{F(\cdot), S_F(K, \cdot)}(K)] \right| \]
Relationships

- **Intuition**: $\text{xxx} \rightarrow \text{yyy} \iff (\forall H):$ if $H$ is xxx-secure, then $H$ is yyy-secure.


Two types of implication and separation

- Conventional
- Provisional – the strength depends on a particular hash function
  - e.g. Sec $\rightarrow$ Pre to $2^{y-m}$
    - $H : \{0, 1\}^k \times \{0, 1\}^m \rightarrow \{0, 1\}^y$
Relationships

- Intuition: \( xxx \rightarrow yyy \iff (\forall H): \) if \( H \) is \( xxx \)-secure, then \( H \) is \( yyy \)-secure.


**Two types of implication and separation**

- Conventional

- Provisional – the strength depends on a particular hash function

  - e.g. \( \text{Sec} \rightarrow \text{Pre} \) to \( 2^{y-m} \)

  \( H: \{0, 1\}^k \times \{0, 1\}^m \rightarrow \{0, 1\}^y \)
Relationships II

- We used different, “asymptotic” definitions of implication and separation.
  - $xxx \rightarrow yyy$, if for every h.f.f. $H$ and polynomial adversary $A$, that has non-negligible advantage in $yyy$ sense there exists a polynomial adversary $B$ with non-negligible advantage in $xxx$ sense (against $H$).
- Such definitions are more “general”
- There are cases, where Rogaway and Shrimpton’s definitions do not work
- Asymptotic definitions are less precise
## Result from 2008

<table>
<thead>
<tr>
<th></th>
<th>Pre</th>
<th>aPre</th>
<th>ePre</th>
<th>Sec</th>
<th>aSec</th>
<th>eSec</th>
<th>Coll</th>
<th>MAC</th>
<th>Prf</th>
<th>Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>aPre</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>ePre</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Sec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>aSec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>eSec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Coll</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Mac</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Prf</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
</tr>
<tr>
<td>Pro</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
## Extension to CTFP

<table>
<thead>
<tr>
<th></th>
<th>CTFP</th>
<th>aCTFP</th>
<th>CTFP</th>
<th>aCTFP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>aPre</td>
<td>→</td>
<td>→</td>
<td>x</td>
<td>→</td>
</tr>
<tr>
<td>ePre</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>x</td>
</tr>
<tr>
<td>Sec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>aSec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>eSec</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>MAC</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>CTFP</td>
<td>x</td>
<td>←</td>
<td>x</td>
<td>←</td>
</tr>
<tr>
<td>Prf</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Pro</td>
<td>→</td>
<td>→</td>
<td>→</td>
<td>→</td>
</tr>
<tr>
<td>Coll</td>
<td>→</td>
<td>←</td>
<td>→</td>
<td>←</td>
</tr>
</tbody>
</table>
Example 1: $\text{Coll} \rightarrow \text{CTFP}$

Let $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family

$$
\text{Adv}_{H}^{\text{CTFP}[\lambda]}(A) = \Pr \left[ K \leftarrow \mathcal{K}; (Y, S) \leftarrow A(K); P \leftarrow \{0, 1\}^\lambda; M \leftarrow A(P, S) : H_K(P || M) = Y \right]
$$
Example 1: Coll → CTFP

If $A$ succeeds in the 3rd and 5th line, then $B$ finds a collision.

\[
P_1 \| M_1 \neq P_2 \| M_2
\]

\[
H_K(P_1 \| M_1) = H_K(P_2 \| M_2) = Y
\]
Example 2: CTFP $\not\rightarrow$ Coll

Let $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family.

$$H'_K(M) = H_K(M[1 \ldots |M| - 1] || 0)$$

- $\forall K \in \mathcal{K} : H'_K(01) = H'_K(00)$
- $H'$ is not Coll secure

We need to show: if $H$ is CTFP secure, then so is $H'$.
Example 2: CTFP $\not\rightarrow$ Coll

Let $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family.

$$H'_K(M) = H_K(M[1 \ldots |M| - 1] || 0)$$

- $\forall K \in \mathcal{K} : H'_K(01) = H'_K(00)$
- $H'$ is not Coll secure

We need to show: if $H$ is CTFP secure, then so is $H'$
Example 2: CTFP $\not\rightarrow$ Coll

Let $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family.

$$H'_K(M) = H_K(M[1 \ldots |M| - 1] || 0)$$

- $\forall K \in \mathcal{K} : H'_K(01) = H'_K(00)$
- $H'$ is not Coll secure

We need to show: if $H$ is CTFP secure, then so is $H'$
Example 2: CTFP $\not\rightarrow$ Coll

Adversary $B$

[1$^{st}$ stage with input $K$]

$(Y, S) \leftarrow A(K)$

return $(Y, S||K)$

[2$^{nd}$ stage with input $(P, S||K)$]

$M \leftarrow A(P, S)$

if $H_K(P||M) = Y$ then return $M$

else let $b := M[|M|]$;

return $M[1\ldots|M| - 1]||\overline{b}$

- Consider that $A$ succeeds, i.e. $H'_K(P||M) = Y$
- $H_K(P||M) = Y$ or $H_K(P||M') = Y$ ($M' = M$ but with the last bit inverted)
- Therefore $B$ succeeds
Conclusion

- We formalize the CTFP preimage resistance in hash function family settings,
- Defined always CTFP preimage resistance,
- Worked out all the relationships among the definitions of CTFP, aCTFP and the other security notions (except those that appeared before).

Motivation:
- It is useful to know the relationships among the properties, that we want the hash function to preserve.
Conclusion

- We formalize the CTFP preimage resistance in hash function family settings,
- Defined always CTFP preimage resistance,
- Worked out all the relationships among the definitions of CTFP, aCTFP and the other security notions (except those that appeared before).

Motivation:
- It is useful to know the relationships among the properties, that we want the hash function to preserve.
The End

Thank you for your attention

and

have a nice day.