Domain extension for enhanced target collision-resistant hash functions

Ilya Mironov
Microsoft Research, Silicon Valley Campus
| Domain extension for enhanced target collision-resistant hash functions |
Target Collision-Resistance (TCR)

- **Collision-resistance** keyed $H_k: \{0,1\}^* \to \{0,1\}^n$

  Adversary

  $k$  
  random $k$

  $x, y$

  $H_k(x) = H_k(y)$  
  $x \neq y$

- **Target collision-resistance** keyed $H_k: \{0,1\}^* \to \{0,1\}^n$

  Adversary$_1$

  $x$

  random $k$

  $H_k(x) = H_k(y)$  
  $x \neq y$

  Adversary$_2$

  $y$

  random $k$

  $H_k(x) = H_k(y)$  
  $x \neq y$

[Naor–Yung’89]
Hash-and-Sign Paradigm

Signature $\sigma$ defined over fixed-length inputs
Public key $PK$, secret key $SK$

- Collision-resistance:
  - add random $k$ to $PK$
  - $\sigma(H_k(M))$ is as secure as $\sigma$

- Target collision-resistance:
  - generate random $k$ for each message
  - $\sigma(k, H_k(M))$ is secure
Black-box Constructions

One-way functions

TCR functions
PRF, MAC, signatures, symmetric-key encryption

Collision-resistant functions

[Simon’98]
**Enhanced Target Collision–Resistance**

- **(Plain) TCR**
  - Adversary$_1$
    - $x$
    - Random $k$
  - Adversary$_2$
    - $y$
    - $H_k(x) = H_k(y)$
    - $x \neq y$

- **Enhanced TCR**
  - Adversary$_1$
    - $x$
    - Random $k$
  - Adversary$_2$
    - $k'$, $y$
    - $H_k(x) = H_{k'}(y)$
    - $(x, k) \neq (y, k')$

[Halevi–Krawczyk’06]
**Hash-and-Sign Paradigm**

Signature \( \sigma \) defined over fixed-length inputs

Public key \( PK \), secret key \( SK \)

- **TCR:**
  - generate random \( k \) for each message
    - \( \sigma(k, H_k(M)) \) is secure

- **Enhanced TCR:**
  - generate random \( k \) for each message
    - \( k, \sigma(H_k(M)) \) is secure

- **Basis of RMX**
  - message pre-processing
  - retrofits existing code

[Halevi–Krawczyk’06]
Support of HMAC, PRF

If a construct is specified for the use of the candidate algorithm in a randomized hashing scheme, the construct must, with overwhelming probability, provide $n$ bits of security against the following attack:

- The attacker chooses a message, $M_1$.
- The specified construct is then used on $M_1$ with a randomization value $r_1$ that has been randomly chosen without the attacker’s control after the attacker has supplied $M_1$.
- Given $r_1$, the attacker then attempts to find a second message $M_2$ and randomization value $r_2$ that yield the same randomized hash value.
Black-box Constructions

One-way functions

TCR functions
PRF, MAC, signatures, symmetric-key encryption

eTCR?

Collision-resistant functions

eTCR?

eTCR?

eTCR?
From TCR to eTCR

TCR ← eTCR — by definition

TCR → eTCR

$h_k$ — TCR

$H_k(x) = h_k(x) || k$ — eTCR

A₁

$\xrightarrow{x}$

random $k$

A₂

$\xleftarrow{k', y}$

$h_k(y) || k = h_{k'}(y) || k'$

$(x, k) \neq (y, k')$
Black-box Constructions

One-way functions

TCR functions
PRF, MAC, signatures, symmetric-key encryption eTCR

Collision-resistant functions
Given $h_k: \{0,1\}^m \rightarrow \{0,1\}^n$ construct $H_K: \{0,1\}^* \rightarrow \{0,1\}^n$

Merkle–Damgård:

Collision-resistant
Domain Extension for TCR

Given $h_k : \{0,1\}^m \rightarrow \{0,1\}^n$ construct $H_K : \{0,1\}^* \rightarrow \{0,1\}^n$

Merkle–Damgård:

Target collision-resistant

[Bellare–Rogaway’97]
Domain Extension for TCR

**Linear hash:**
Key: $k_1, k_2, k_3, k_4, k_5$

[NY89]

**Shoup scheme:**
Key: $k, k_0, k_1, k_2$

[Shoup’00]
Domain Extension for eTCR

Linear hash:
Key: $k_1, k_2, k_3, k_4, k_5$

$M_1 \rightarrow h_{k_1} \rightarrow h_{k_2} \rightarrow h_{k_3} \rightarrow h_{k_4} \rightarrow h_{k_5}$

[NY89]

Shoup scheme:
Key: $k, k_0, k_1, k_2$

$M_1 \rightarrow h_k \rightarrow h_k \rightarrow h_k \rightarrow h_k \rightarrow h_k$

$IV \rightarrow h_k \rightarrow h_k \rightarrow h_k \rightarrow h_k \rightarrow h_k$

[Shoup’00]
[RSM09] from FSE’09:

- Merkle–Damgård
- RMX
- XOR Linear Hash
- Shoup
- Linear hash (with length encoding)
Efficient Domain Extender for eTCR?

- Linear Hash:
  - linear key expansion
  - extends eTCR

- Shoup scheme:
  - logarithmic key expansion
  - extends TCR

- If $h_k(x)$ is TCR, then $h_k(x) || k$ is eTCR...
Main Construction

Message $M$

TCR hash

Key $K_1$

Key $K_2$

eTCR hash

eTCR
Instantiation

Subtleties:
- Length encoding
- IV
Outline

Domain extension for enhanced target collision-resistant hash functions
Summary

- eTCR is interesting and appealing property
- Domain extension scheme with logarithmic key blowup
- Open questions:
  - More efficient domain extension?
  - Impossibility results
  - Attacks on the eTCR property of SHA-3 candidates