Enhanced Target Collision Resistant Hash Functions Revisited

Mohammad-Reza Reyhanitabar, Willy Susilo, and Yi Mu

Centre for Computer and Information Security Research

University of Wollongong

Australia

Outline:

- Introduction
  - Keyless and Dedicated-key Hash Function Settings
  - Conventions
  - Domain Extension
  - MD Transforms
  - Randomized Hashing Construction
  - Related Security Notions
- Our Contributions:
  - eTCR versus CR: Separation Result
  - Domain Extension for eTCR Hash Functions
- Conclusion

Introduction

- Two Settings for Hash Functions:

1. Keyless Setting: $\mathrm{H} : \mathcal{M} \rightarrow \mathcal{C}$

   - Example: $\text{SHA-1} : \{0,1\}^{<2^{64}} \rightarrow \{0,1\}^{160}$

2. Dedicated-key Setting (Functions Family): $\mathcal{H} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

   A member of the family is chosen by a key (index or salt) $K \in \mathcal{K}$ and is a function $\mathrm{H} \triangleq \mathcal{H}_K : \mathcal{M} \rightarrow \mathcal{C}$

   - Some examples:

     ★ CRHF family (Damgård, CRYPTO 1987)
     ★ UOWHF family (Naor and Yung, STOC 1989)
     ★ VSH (Contini, Lenstra, and Steinfeld, EUROCRYPT 2006)

Conventions (in Concrete-security Framework):

- The output length (hash size) is some fixed positive integer *n*, i.e. $\mathcal{C} = \{0,1\}^n$
- The hash function (family) should be able to compress, i.e. $|\mathcal{M}| > |\mathcal{C}|$

- Depending on the input length, we can have:
  - Fixed-input-length (FIL) hash function, usually called a 'Compression Function':
    - Keyless Setting: $\mathrm{h} : \{0,1\}^m \rightarrow \{0,1\}^n$
    - Dedicated-key Setting: $h : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$
  - Variable-input-length (VIL) hash function, <u>usually what is meant by a</u> 'Hash Function':
    - Keyless Setting: $\mathrm{H} : \{0,1\}^{<2^\lambda} \rightarrow \{0,1\}^n$
    - Dedicated-key Setting: $\mathcal{H} : \mathcal{K} \times \{0,1\}^{<2^\lambda} \rightarrow \{0,1\}^n$
  - Arbitrary-input-length (AIL) hash function !: $\mathcal{M} : \{0,1\}^*$

Constructing a (VIL or AIL) Hash Function:

- Two-step Paradigm:

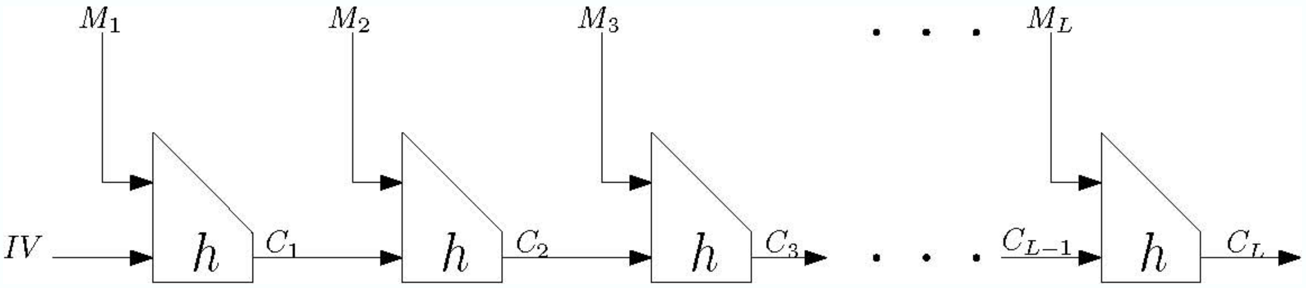
    1. Construct a compression function capable of hashing FIL messages
    2. Apply a domain extension transform to build the full-fledged hash function capable of hashing messages of variable length

- Domain Extension Transform: Message ‘Padding’ + ‘Iteration’ Construction

MD Construction

Merkle-Damgård Transforms:

- Padding:
  - Plain
  - MD Strengthening (length indicating or suffix-free)
  - Prefix-free (Coron et al., CRYPTO 2005)
  - Split (Yasuda, ASIACRYPT 2008)
- Iteration:



Randomized Hashing Mode

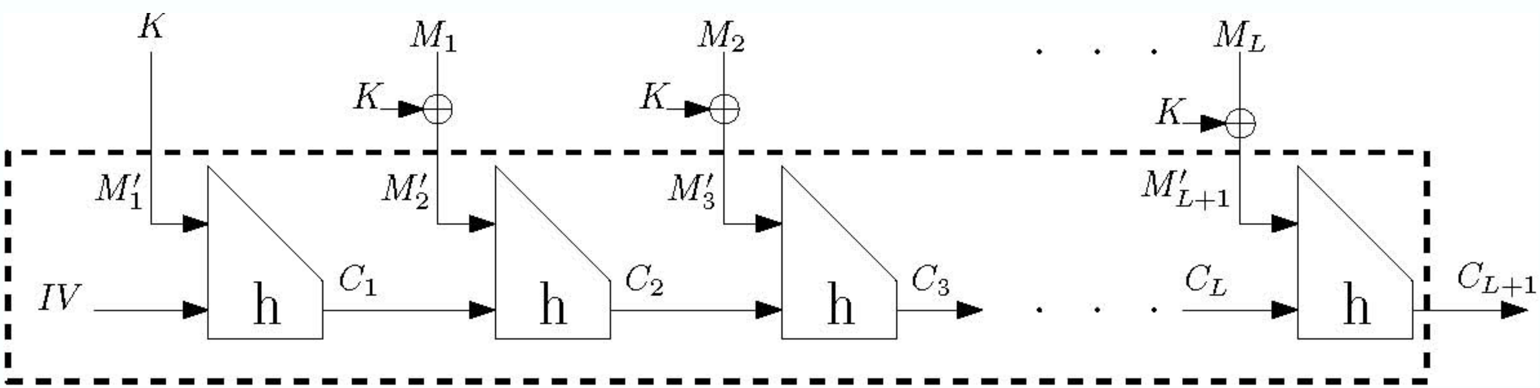
Halevi and Krawczyk at CRYPTO 2006 proposed the following black-box mode of operation for an MD hash function (NIST Draft SP 800-106):

$\mathrm{h} : \{0,1\}^{n+b} \rightarrow \{0,1\}^{n}$ (Keyless)

$\Big\downarrow$ MD

$\mathrm{H} : \{0,1\}^{<2^{\lambda}} \rightarrow \{0,1\}^{n}$ (Keyless) $\xrightarrow{\text{Randomized Hashing (RMX mode)}}$ $\tilde{\mathcal{H}} : \{0,1\}^{b} \times \{0,1\}^{<2^{\lambda}} \rightarrow \{0,1\}^{n}$ (Dedicated-key)

$$\tilde{\mathcal{H}}(K, M) \triangleq \mathrm{H}\big(K \| (M_1 \oplus K) \| \cdots \| (M_L \oplus K)\big)$$



Security Goal for RMX

"The goal is to free practical digital signature schemes from their current reliance on strong collision resistance by basing the security of these schemes on significantly weaker properties of the underlying hash function $\cdots$ (Halevi and Krawczyk, CRYPTO 2006)

Hash-and-Sign:

- ★ $\sigma = Sign(H(M)) \rightarrow$ The hash function H needs to be Collision Resistant
- ★ $\sigma = K, Sign(H_K(M), K) \rightarrow$ The hash function (family) H needs to be UOWHF (=TCR) (Naor and Yung, STOC 1989 - Bellare and Rogaway CRYPTO 1997)
- ★ $\sigma = K, Sign(H_K(M)) \rightarrow$ The hash function (family) H needs to be "enhanced Target Collision Resistant' (Halevi and Krawczyk, CRYPTO 2006)

- Security Analysis of Randomized Hashing Construction:
  - New security property for a dedicated-key hash function is introduced: Enhanced Target Collision Resistance (eTCR)
  - New security assumptions for a keyless compression function are introduced: OWH, c-SPR and e-SPR
  - Under the assumption that the compression function is regular, OWH will be implied by other two assumptions (c-SPR and e-SPR).
  - c-SPR and e-SPR are both implied by (i.e. are weaker than) the strong collision resistance assumption on the keyless compression function

$$\begin{array}{lcl} \text{c-SPR and OWH assumptions on h} & \Longrightarrow & \text{eTCR property for } \tilde{\mathcal{H}} \\ \text{e-SPR and OWH assumptions on h} & \Longrightarrow & \text{eTCR property for } \tilde{\mathcal{H}} \end{array}$$

On SPR, c-SPR and e-SPR Assumptions

- These security assumptions for a keyless compression function $\mathrm{h} : \{0,1\}^{n+b} \to \{0,1\}^n$ are defined as follows:

$$\mathrm{Adv}_{\mathrm{h}}^{\mathrm{SPR}}(A) = \Pr\left\{c||m \xleftarrow{\$} \{0,1\}^{n+b};\ (c'||m') \xleftarrow{\$} A(c||m):\ c||m \neq c'||m' \ \wedge\ \mathrm{h}(c||m) = \mathrm{h}(c'||m')\right\}$$

$$\mathrm{Adv}_{\mathrm{h}}^{\mathrm{c\text{-}SPR}}(A) = \Pr\left\{m \xleftarrow{\$} \{0,1\}^{b};\ (c, c'||m') \xleftarrow{\$} A(m):\ c||m \neq c'||m' \ \wedge\ \mathrm{h}(c||m) = \mathrm{h}(c'||m')\right\}$$

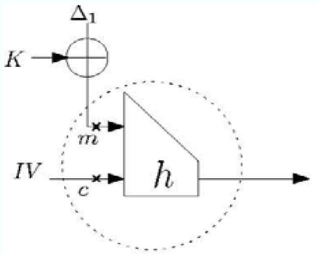
- Generic security level of c-SPR is similar to keyless-CR, i.e. $O(2^{\frac{n}{2}})$

e-SPR Game:

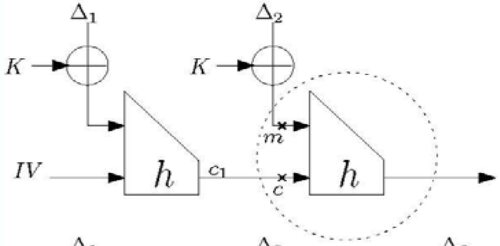
Let H^{c_0} be the MD iteration of h with initial value c_0.
The game is parameterized by the IV$= c_0$. A chooses $l \geq 1$ values $\Delta_i,\ i = 1, \cdots, l$, each of length b bits; then A receives a random $K \in \{0,1\}^b$ and c and m are set to $m = K \oplus \Delta_l$ and $c = H^{c_0}(K \oplus \Delta_1, \cdots, K \oplus \Delta_{l-1})$.
Finally A chooses c', m'.

A wins iff: $(c||m) \neq (c'||m') \ \wedge\ \mathrm{h}(c||m) = \mathrm{h}(c'||m')$

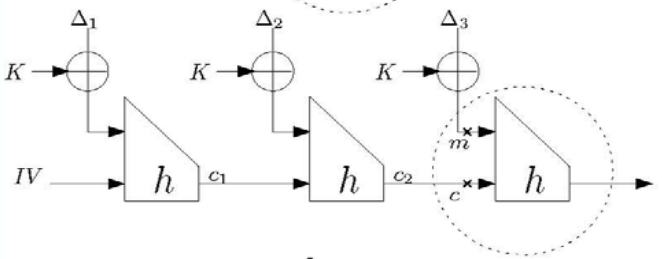
e-SPR(t, L+1, ϵ): A collection of L+1 SPR-like assumptions on h



Case $l = 1$: The variable c in e-SPR game is selected from a distribution D_1 which is the distribution of intial value $IV = c_0$.

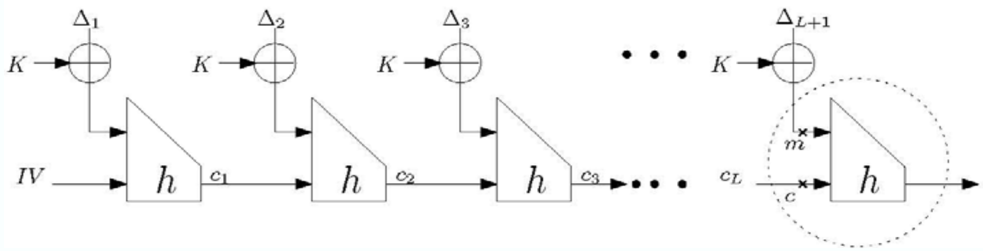


Case $l = 2$: The variable c in e-SPR game is selected from a distribution D_2 induced by $h(IV, K \oplus \Delta_1)$, i.e. the induced distribution on c_1.



Case $l = 3$: The variable c in e-SPR game is selected from a distribution D_3 induced by $h(c_1, K \oplus \Delta_2)$, that is the induced distribution on c_2 .

⋮



Case $l = L + 1$: The variable c in e-SPR game is selected from a distribution D_{L+1} induced by $h(c_{L-1}, K \oplus \Delta_{L+1})$, that is the induced distribution on c_L.

Definitions: CR, TCR, and eTCR

Formal definitions in dedicated-key setting (Rogaway and Shrimpton, FSE 2004):

$$\mathrm{Adv}_{\mathcal{H}}^{CR}(A) = \Pr\left\{K \xleftarrow{\$} \mathcal{K};\ (M, M') \xleftarrow{\$} A(K):\ M \neq M' \ \wedge\ \mathcal{H}_K(M) = \mathcal{H}_K(M')\right\}$$

$$\mathrm{Adv}_{\mathcal{H}}^{TCR}(A) = \Pr\left\{(M, State) \xleftarrow{\$} A_1();\ K \xleftarrow{\$} \mathcal{K};\ M' \xleftarrow{\$} A_2(K, State):\ M \neq M' \ \wedge\ \mathcal{H}_K(M) = \mathcal{H}_K(M')\right\}$$

$$\text{CR} \xrightarrow{\text{implies}} \text{TCR}$$

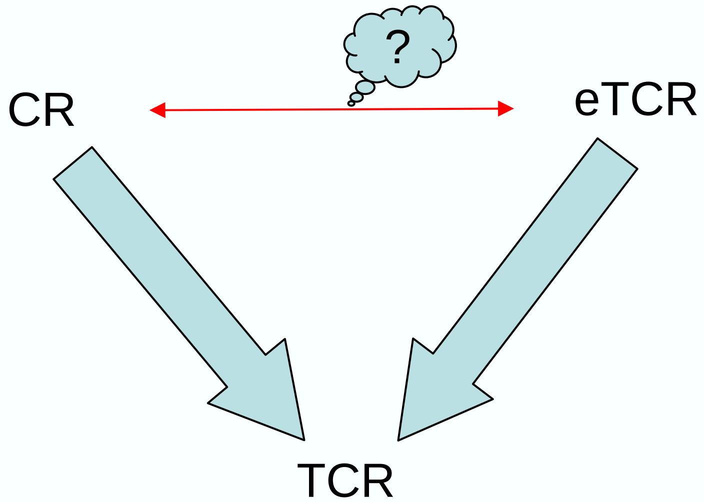
For any dedicated-key hash function $\mathcal{H} : \mathcal{K} \times \mathcal{M} \rightarrow \{0,1\}^n$,
if $\mathcal{H}$ is CR secure then it is TCR secure too.

enhanced Target Collision Resistance (Halevi and Krawczyk, CRYPTO 2006):

$$\mathrm{Adv}_{\mathcal{H}}^{eTCR}(A) = \Pr\left\{\begin{array}{l} (M, State) \xleftarrow{\$} A_1(); \\ K \xleftarrow{\$} \mathcal{K}; \\ (K', M') \xleftarrow{\$} A_2(K, State); \end{array} : (K, M) \neq (K', M') \ \wedge\ \mathcal{H}_K(M) = \mathcal{H}_{K'}(M')\right\}$$

$$\text{eTCR} \xrightarrow{\text{implies}} \text{TCR}$$

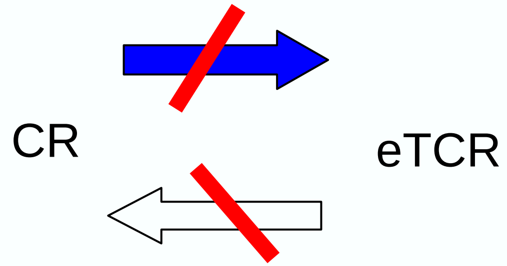
eTCR versus CR



Result (Separation):

1. eTCR property is not implied by the CR property

2. CR property is not implied by the eTCR property



CR $\not\Rightarrow$ eTCR

Assume that we have a hash function $\mathcal{H} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$ which is $(t, \epsilon) - CR$.

Select (and *fix*) an arbitrary message $M^* \in \{0,1\}^m$ and an arbitrary key $K^* \in \{0,1\}^k$.

The hash function $\mathcal{G} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$ shown below is $(t', \epsilon') - CR$, where $t' = t - cT_H$ and $\epsilon' = \epsilon + 2^{-k}$, but it is completely insecure in eTCR sense.

$$\mathcal{G}_K(M) = \begin{cases} M^*_{1\cdots n} & \text{if } M = M^* \bigvee K = K^* & (1) \\ \mathcal{H}_K(M^*) & \text{if } M \neq M^* \bigwedge K \neq K^* \bigwedge \mathcal{H}_K(M) = M^*_{1\cdots n} & (2) \\ \mathcal{H}_K(M) & \text{otherwise} & (3) \end{cases}$$

eTCR $\not\Rightarrow$ CR

Assume that we have a hash function $\mathcal{H} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$, with $m > k \geq n$, which is $(t, \epsilon) - eTCR$.

The hash function $\mathcal{G} : \{0,1\}^k \times \{0,1\}^m \rightarrow \{0,1\}^n$ shown below is $(t', \epsilon') - eTCR$, where $t' = t - c$, $\epsilon' = \epsilon + 2^{-k+1}$, but it is completely insecure in CR sense.

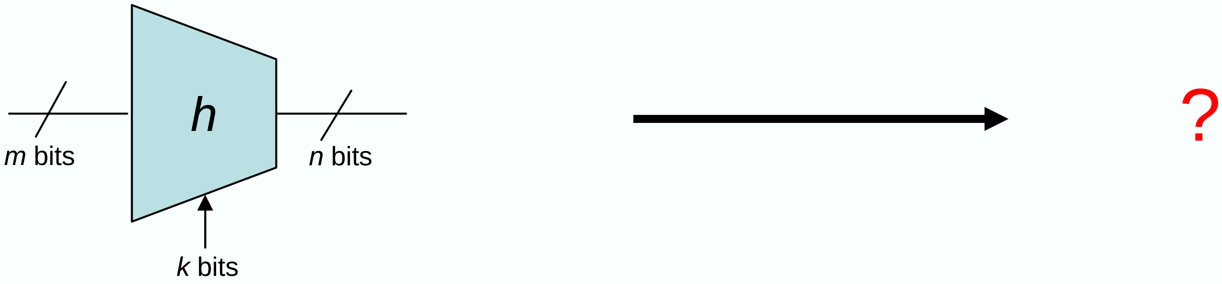
$$\mathcal{G}_K(M) = \begin{cases} \mathcal{H}_K(0^{m-k}||K) & \text{if } M = 1^{m-k}||K \\ \mathcal{H}_K(M) & \text{otherwise} \end{cases}$$

eTCR Preserving Domain Extension

- Given a compression function which is eTCR secure, how can one construct a full-fledged hash function which is eTCR secure?

FIL eTCR function

VIL eTCR function



$$h : \{0,1\}^k \times \{0,1\}^m \to \{0,1\}^n \xrightarrow{\text{transform}} \mathcal{H} : \mathcal{K} \times \{0,1\}^{<2^\lambda} \to \{0,1\}^{n'}$$

where $n' \leq n$ and $|\mathcal{K}| \geq 2^k$

Orthogonality of Property Preservation

Strengthened MD Transform:

- ★ preserves CR (Merkle and Damgård, CRYPTO 1989)
- ★ does not preserve (Pseudo-) Random Oracle (Coron et al., CRYPTO 2005)
- ★ does not preserve TCR (Bellare and Rogaway, CRYPTO 1997)

ideal hash (random oracle)

$\Downarrow$

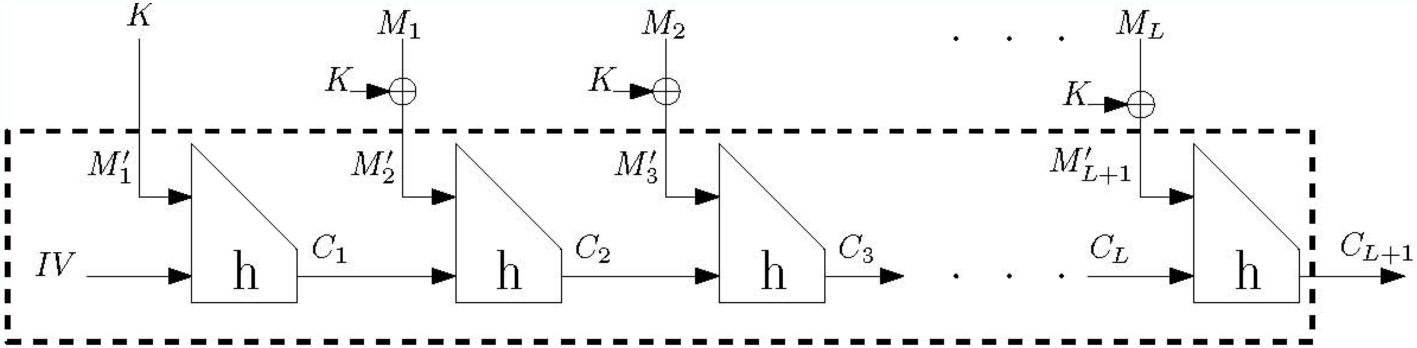
CR

$\Downarrow$

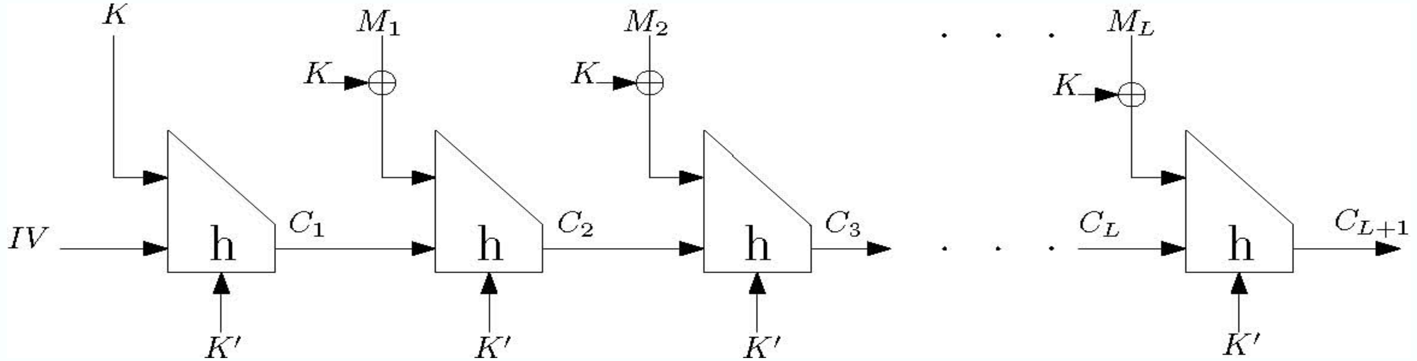
TCR

$\longrightarrow$ In general, from the fact that a domain extension transform is able or unable to preserve a security notion, one cannot conclude about the transform's property preservation capability with regard to other either weaker or stronger security notions.

Can Randomized Hashing Preserve eTCR?



Original Randomized Hashing



Randomized Hashing in the Dedicated-key Setting

Negative Result: Randomized Hashing does not preserve eTCR

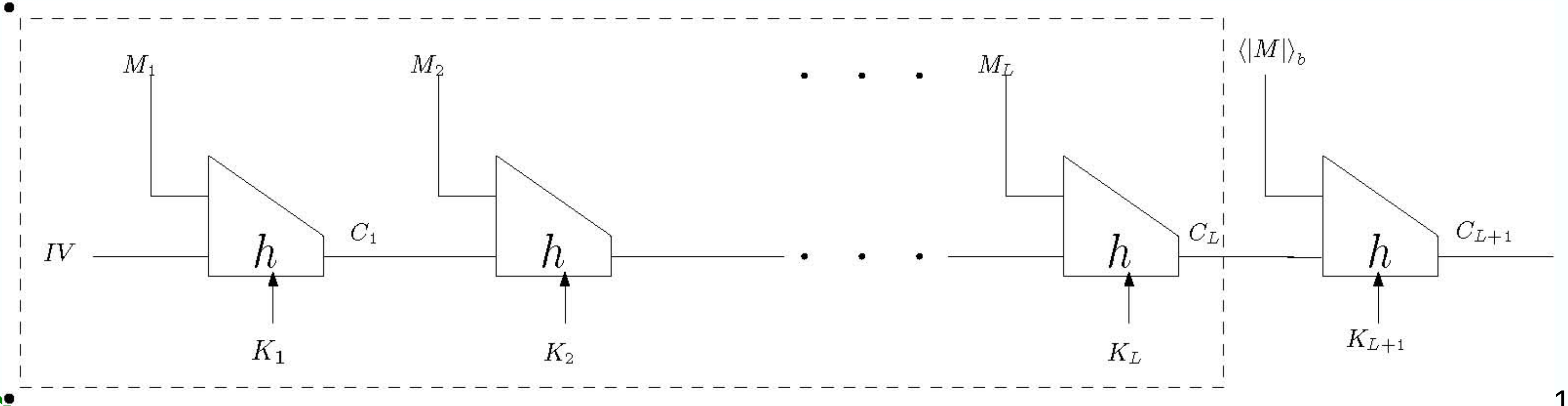
(The proof is done by showing a counterexample)

Other Domain Extenders

Negative Results:

- (Plain, Strengthened, Prefix-free) MD cannot preserve eTCR. (The proof is done by showing a counterexample)
- XOR Masking based transforms for TCR preservation (XLH, Shoup, Enveloped-Shoup, and XTH) are insecure in eTCR sense.

Positive Result: Linear Hash (LH) with a full-final-block strengthening padding ('Nested LH') preserves eTCR.



Conclusion

- There is a separation between CR and eTCR properties (Neither of them implies the other for an arbitrary dedicated-key hash function)
- Current efficient CR and TCR property preserving domain extension transforms (in the standard model) are not capable to preserve eTCR
- The nested LH transform can preserve eTCR but it is inefficient from key length viewpoint.
- Future Research:
  - Design of a new **efficient** eTCR preserving domain extension transform (**without any random oracle**)
  - Showing impossibility results in regard to such efficient eTCR preserving transforms (lower bound on key expansion)

Questions?

Thanks!