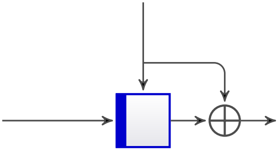
Blockcipher Based Hashing Revisited

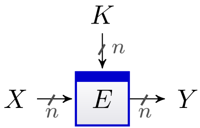
Martijn Stam

EPFL – LACAL

FSE

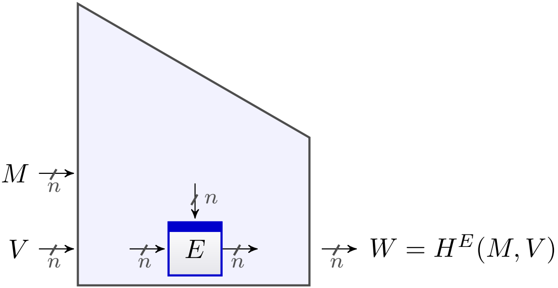
23 February 2009





$$E : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$$

- Block cipher with n-bit key, operating on n bit blocks: $Y = E_K(X)$.
- Compression function H^E from $2n$ bits to n bits (input consists of n bits message and n bits chaining variable).
- Hash function $\mathcal{H}^E$ using Merkle-Damgård transform.

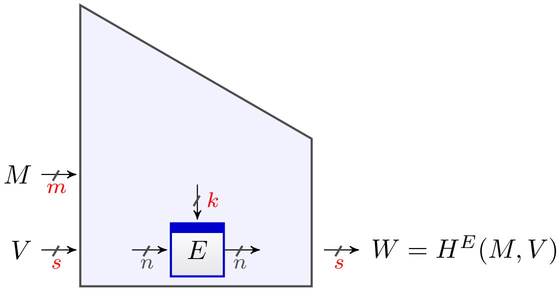


$$E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

- Block cipher with n-bit key, operating on n bit blocks: .
- Compression function H^E from $2n$ bits to n bits (input consists of n bits message and n bits chaining variable).
- Hash function $\mathcal{H}^E$ using Merkle-Damgård transform.

Blockcipher Based Hashing

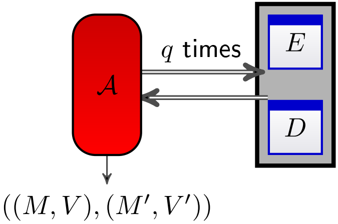
The principle idea



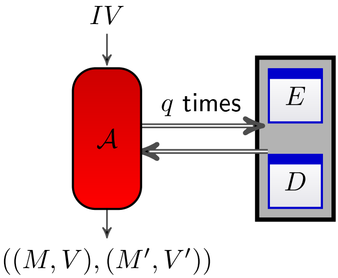
$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$$

- Block cipher with k-bit key, operating on n bit blocks: .
- Compression function H^E from $m+s$ bits to s bits (input consists of m bits message and s bits chaining variable).
- Hash function $\mathcal{H}^E$ using Merkle-Damgård transform.

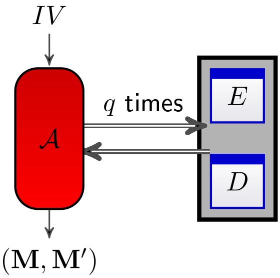
Blockcipher Based Hashing
Collision resistance: A measure of security



$$\mathsf{Adv}^{\mathsf{coll}}_{H}(\mathcal{A}) = \Pr\left[(M,V) \neq (M',V') \text{ and } H^{E}(M,V) = H^{E}(M',V')\right]$$

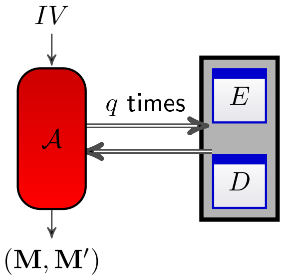


$$\mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A}) = \Pr\left[(M,V) \neq (M',V') \text{ and } H^E(M,V) = \begin{cases} H^E(M',V') \\ IV \end{cases}\right]$$



$$\mathsf{Adv}^{\mathsf{coll}}_{H}(\mathcal{A}) = \Pr\left[(M,V) \neq (M',V') \text{ and } H^{E}(M,V) = \begin{cases} H^{E}(M',V') \\ IV \end{cases}\right]$$

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(\mathcal{A}) = \max_{IV} \Pr\left[\mathbf{M} \neq \mathbf{M}' \text{ and } \mathcal{H}^{E}_{IV}(\mathbf{M}) = \mathcal{H}^{E}_{IV}(\mathbf{M}')\right]$$

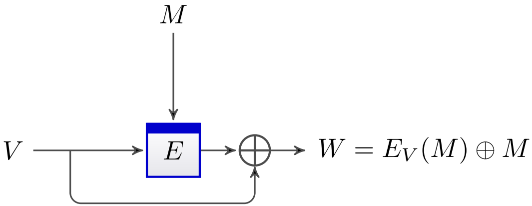


$$\mathsf{Adv}^{\mathsf{coll}}_{H}(\mathcal{A}) = \Pr\left[(M,V) \neq (M',V') \text{ and } H^E(M,V) = \begin{cases} H^E(M',V') \\ IV \end{cases}\right]$$

$$\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(\mathcal{A}) = \max_{IV} \Pr\left[\mathbf{M} \neq \mathbf{M}' \text{ and } \mathcal{H}^E_{IV}(\mathbf{M}) = \mathcal{H}^E_{IV}(\mathbf{M}')\right]$$

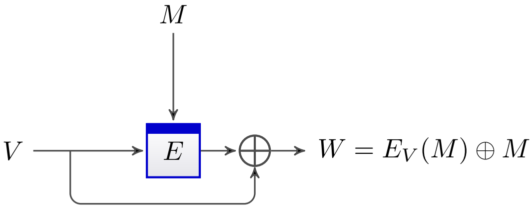
$$\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(q) \leq \mathsf{Adv}^{\mathsf{coll}}_{H}(q)$$

Example: Davies-Meyer Construction



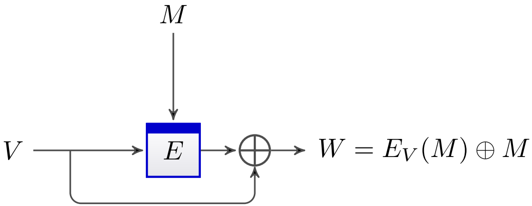
$$
\begin{array}{lll}
K & = & \quad\quad M \\
X & = & \quad\quad\quad\quad\quad V \\
 & & \\
W & = & Y\oplus \quad\quad\quad V
\end{array}
$$

Example: Davies-Meyer Construction



$$
\begin{array}{lcccc}
K & = & & 1 \cdot M & \oplus & 0 \cdot V \\
X & = & & 0 \cdot M & \oplus & 1 \cdot V \\
\\
W & = & Y \oplus & 0 \cdot M & \oplus & 1 \cdot V
\end{array}
$$

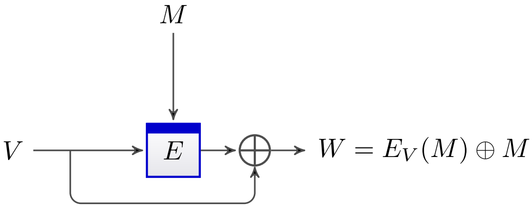
Example: Davies-Meyer Construction



$$\begin{pmatrix} K \\ X \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M \\ V \end{pmatrix}$$

$$W = Y \oplus \begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M \\ V \end{pmatrix}$$

Example: Davies-Meyer Construction

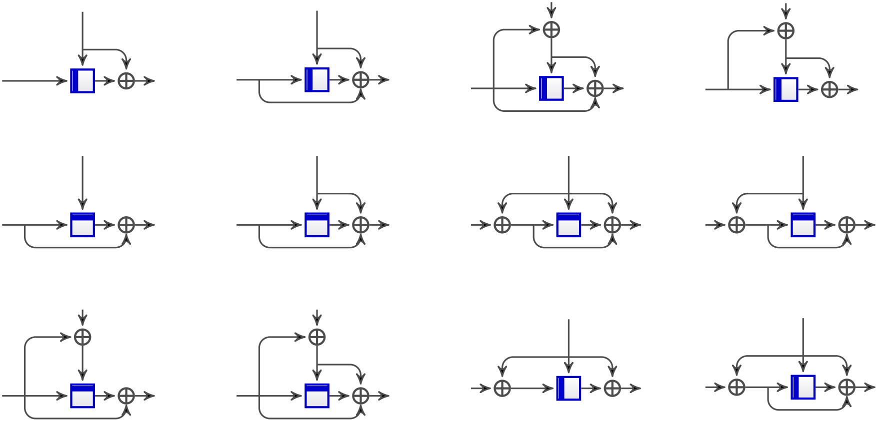


$$\begin{pmatrix} K \\ X \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M \\ V \end{pmatrix} = \begin{pmatrix} \mathbf{K} \\ \mathbf{X} \end{pmatrix} \begin{pmatrix} M \\ V \end{pmatrix}$$

$$W = Y \oplus \begin{pmatrix} 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} M \\ V \end{pmatrix} = Y \oplus \mathbf{U} \begin{pmatrix} M \\ V \end{pmatrix}$$

Where $\mathbf{K}, \mathbf{X}, \mathbf{U} \in \mathbb{Z}_2^2$.
[PGV93]: Examined all $2^6 = 64$ possible schemes, attack-based approach.

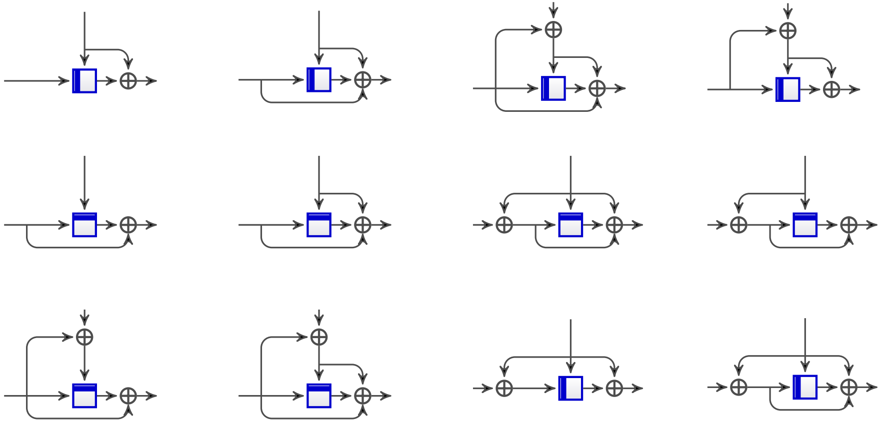
12 Collision Resistant Compression Functions



[PGV93] Schemes deemed secure

[BRS02] Provable collision resistance: $\mathrm{Adv}_H^{\mathsf{coll}}(q) \leq \frac{1}{2}q(q+1)/(2^n - q)$.

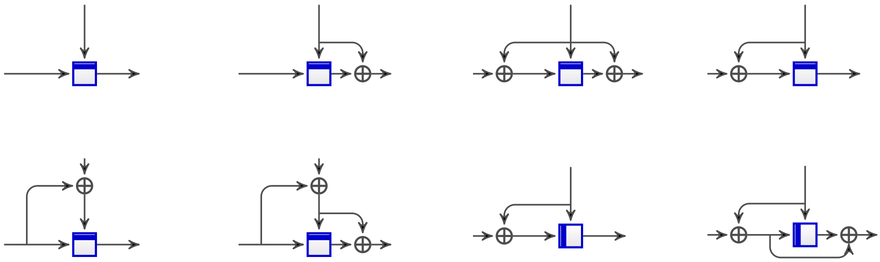
12 Collision Resistant Compression Functions



[PGV93] Schemes deemed secure

[BRS02] Provable collision resistance: $\mathrm{Adv}_H^{\mathrm{coll}}(q) \le \frac{1}{2}q(q+1)/(2^n - q)$.

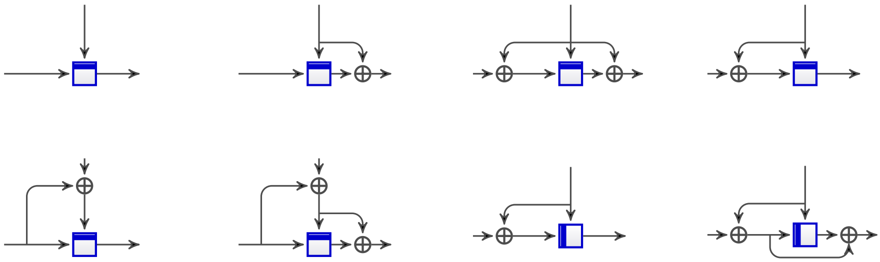
Further 8 Collision Resistant *Hash* Functions



[BRS02] Provable secure in the iteration: $\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(q) \leq 3q(q+1)/2^n$

[DL06] Improved bounds: $\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(q) \leq \frac{1}{2}q(q+1)/(2^n - q)$

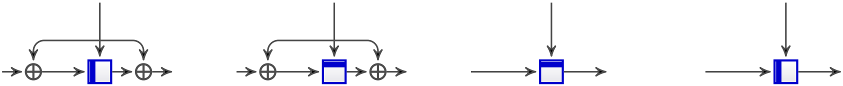
Further 8 Collision Resistant *Hash* Functions



[BRS02] Provable secure in the iteration: $\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(q) \leq 3q(q+1)/2^n$

[DL06] Improved bounds: $\mathsf{Adv}^{\mathsf{coll}}_{\mathcal{H}}(q) \leq \frac{1}{2}q(q+1)/(2^n - q)$

Questions

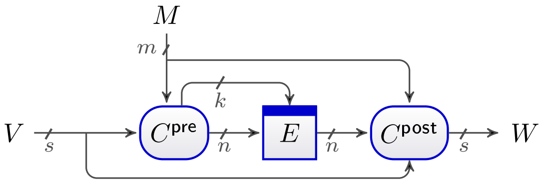


- Why these 12 and 8 schemes?
  What makes them special?
  What do they have in common?
- What happens if for instance
  - we want to chop the output in the end?
  - we want to use addition modulo 2^n instead of XOR?
  - we want to use a blockcipher with keys larger than the blocksize?
  - we want security beyond the blocksize?

Questions

- Why these 12 and 8 schemes?
  What makes them special?
  What do they have in common?
- What happens if for instance
  - we want to chop the output in the end?
  - we want to use addition modulo 2^n instead of XOR?
  - we want to use a blockcipher with keys larger than the blocksize?
  - we want security beyond the blocksize?

General Single Call Scenario



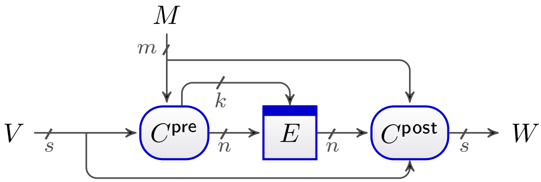
Classical: $s = n$, $m + s = n + k$
Includes PGV/BRS (for $k = n$).

Chopped: $s < n$, $m + s = n + k$
Includes Grindahl (for $k = 0$).

Overloaded: $s = n$, $m + s > n + k$
Includes sponges (for $k = 0$).

Supercharged: $s > n$, $m + s = n + k$
Allows security beyond the birthday bound!

General Single Call Scenario



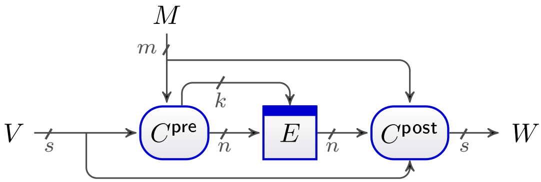
Classical: $s = n$, $m + s = n + k$
Includes PGV/BRS (for $k = n$).

Chopped: $s < n$, $m + s = n + k$
Includes Grindahl (for $k = 0$).

Overloaded: $s = n$, $m + s > n + k$
Includes sponges (for $k = 0$).

Supercharged: $s > n$, $m + s = n + k$
Allows security beyond the birthday bound!

General Single Call Scenario



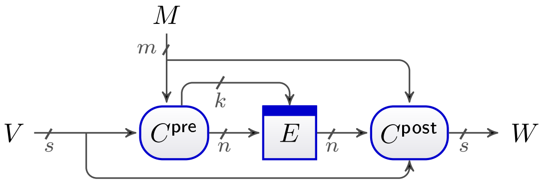
Classical: $s = n$, $m + s = n + k$
Includes PGV/BRS (for $k = n$).

Chopped: $s < n$, $m + s = n + k$
Includes Grindahl (for $k = 0$).

Overloaded: $s = n$, $m + s > n + k$
Includes sponges (for $k = 0$).

Supercharged: $s > n$, $m + s = n + k$
Allows security beyond the birthday bound!

General Single Call Scenario



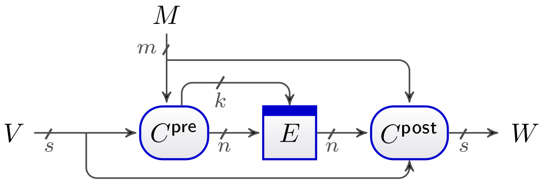
Classical: $s = n$, $m + s = n + k$
Includes PGV/BRS (for $k = n$).

Chopped: $s < n$, $m + s = n + k$
Includes Grindahl (for $k = 0$).

Overloaded: $s = n$, $m + s > n + k$
Includes sponges (for $k = 0$).

Supercharged: $s > n$, $m + s = n + k$
Allows security beyond the birthday bound!

General Single Call Scenario



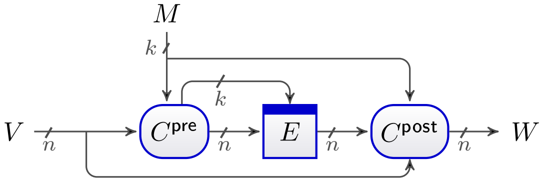
Classical: $s = n$, $m + s = n + k$
Includes PGV/BRS (for $k = n$).

Chopped: $s < n$, $m + s = n + k$
Includes Grindahl (for $k = 0$).

Overloaded: $s = n$, $m + s > n + k$
Includes sponges (for $k = 0$).

Supercharged: $s > n$, $m + s = n + k$
Allows security beyond the birthday bound!

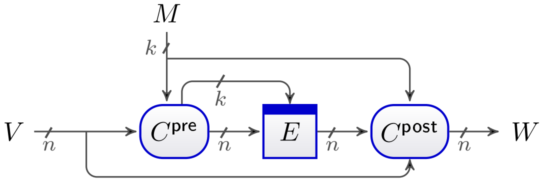
Type I: Secure Compression (Classical)



Create a list of tuples $V \overset{M}{\to} W$ such that $W = H^E(M, V)$. Then

Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

Type I: Secure Compression (Classical)

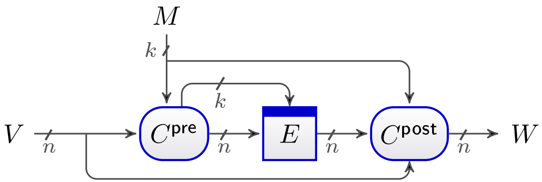


Create a list of tuples $V \overset{M}{\to} W$ such that $W = H^E(M, V)$. Then

Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

- Minimize the size of this list (given q)
- The W's distributed roughly independent uniform.

Type I: Secure Compression (Classical)

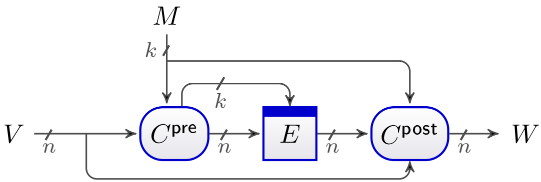


Create a list of tuples $V \overset{M}{\to} W$ such that $W = H^E(M, V)$. Then

Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

- Minimize the size of this list (given q)
- The W's distributed roughly independent uniform.

Type I: Secure Compression (Classical)



Create a list of tuples $V \stackrel{M}{\to} W$ such that $W = H^E(M, V)$. Then

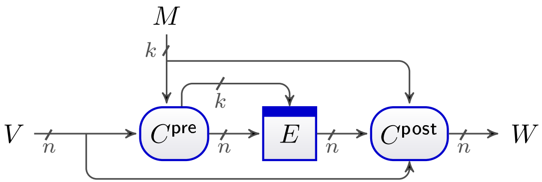
Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

- Minimize the size of this list (given q)
- The W's distributed roughly independent uniform.

Then you might expect birthday bound behaviour

$$\mathsf{Adv}_H^{\mathsf{coll}}(\mathcal{A}) \approx \frac{(\text{Size of list})^2}{2^n}$$

Type I: Secure Compression (Classical)



Create a list of tuples $V \xrightarrow{M} W$ such that $W = H^E(M, V)$. Then

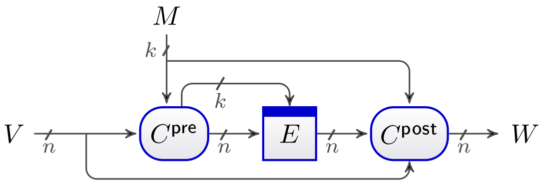
Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

- Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
- The W's distributed roughly independent uniform.

Then you might expect birthday bound behaviour

$$\mathsf{Adv}_H^{\text{coll}}(\mathcal{A}) \approx \frac{(\text{Size of list})^2}{2^n} = \frac{q^2}{2^n}$$

Type I: Secure Compression (Classical)

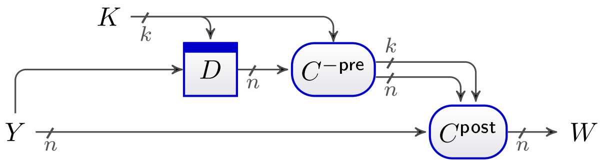


Create a list of tuples $V \xrightarrow{M} W$ such that $W = H^E(M, V)$. Then

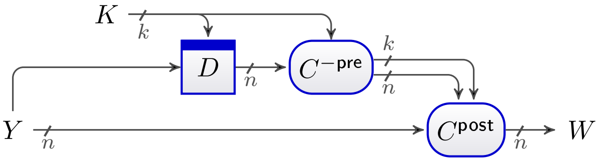
Collision in $H \Leftrightarrow$ "Collision" in list (W-component)

- Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
- The W's distributed roughly independent uniform.

For forward queries,
$C^{\text{post}}(M, V, \cdot) : \{0,1\}^n \rightarrow \{0,1\}^n$ bijective for all M, V.



$$C^{\mathsf{aux}}(K, X, Y) = C^{\mathsf{post}}(C^{-\mathsf{pre}}(K, X), Y)$$



$$C^{\mathsf{aux}}(K,X,Y) = C^{\mathsf{post}}(C^{-\mathsf{pre}}(K,X),Y)$$

For inverse queries,
$C^{\mathsf{aux}}(K,\cdot,Y) : \{0,1\}^n \to \{0,1\}^n$ bijective for all K,Y
gives $V \stackrel{M}{\to} W$ with W's distributed roughly independent uniform.

Type I: Secure Compression (Classical)

1. The preprocessing C^{pre} is bijective.
2. For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For all K, Y the modified postprocessing $C^{\mathsf{aux}}(K, \cdot, Y)$ is bijective.

Type I: Secure Compression (Classical)

1. The preprocessing C^{pre} is bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{X}}$ is invertible (6 possible matrices).
2. For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
   [PGV/BRS] Automatically satisfied.
3. For all K, Y the modified postprocessing $C^{\mathsf{aux}}(K, \cdot, Y)$ is bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{U}}$ is invertible (2 possibilities per matrix).

Type I: Secure Compression (Classical)

1. The preprocessing C^{pre} is bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{X}}$ is invertible (6 possible matrices).
2. For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
   [PGV/BRS] Automatically satisfied.
3. For all K, Y the modified postprocessing $C^{\mathsf{aux}}(K, \cdot, Y)$ is bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{U}}$ is invertible (2 possibilities per matrix).

$\Rightarrow$ Gives exactly the 12 Type-I PGV schemes.

Type II: Security in the Iteration (Classical)

The Duo-Li proof technique uses that list of $V \xrightarrow{M} W$ satisfy:

1. Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
2. For a forward query W is distributed roughly independent uniform $\Rightarrow$ For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For an inverse query V is distributed roughly independent uniform

Type II: Security in the Iteration (Classical)

The Duo-Li proof technique uses that list of $V \xrightarrow{M} W$ satisfy:

1. Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
2. For a forward query W is distributed roughly independent uniform $\Rightarrow$ For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For an inverse query V is distributed roughly independent uniform

Type II: Security in the Iteration (Classical)

The Duo-Li proof technique uses that list of $V \overset{M}{\to} W$ satisfy:

1. Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.

2. For a forward query W is distributed roughly independent uniform $\Rightarrow$ For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For an inverse query V is distributed roughly independent uniform For all K, $C^{-\mathsf{pre}}(K, \cdot)$ restricted to V is bijective.

Type II: Security in the Iteration (Classical)

The Duo-Li proof technique uses that list of $V \xrightarrow{M} W$ satisfy:

1. Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{X}}$ is invertible (6 matrices possible).
2. For a forward query W is distributed roughly independent uniform
   $\Rightarrow$ For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For an inverse query V is distributed roughly independent uniform
   For all K, $C^{-\mathsf{pre}}(K, \cdot)$ restricted to V is bijective.
   [PGV/BRS] The key is message dependent, $K = M$ or $K = M \oplus V$.
   $\Rightarrow$ Only 4 matrices possible, $\mathbf{U}$ unrestricted.

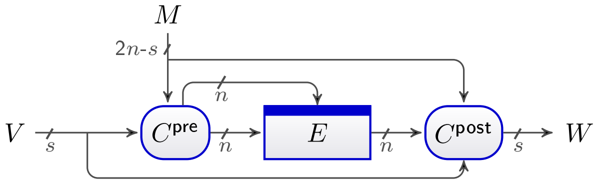
Type II: Security in the Iteration (Classical)

The Duo-Li proof technique uses that list of $V \stackrel{M}{\rightarrow} W$ satisfy:

1. Minimize the size of this list (given q) $\Rightarrow$ C^{pre} bijective.
   [PGV/BRS] $\binom{\mathbf{K}}{\mathbf{X}}$ is invertible (6 matrices possible).
2. For a forward query W is distributed roughly independent uniform
   $\Rightarrow$ For all M, V the postprocessing $C^{\mathsf{post}}(M, V, \cdot)$ is bijective.
3. For an inverse query V is distributed roughly independent uniform
   For all K, $C^{-\mathsf{pre}}(K, \cdot)$ restricted to V is bijective.
   [PGV/BRS] The key is message dependent, $K = M$ or $K = M \oplus V$.
   $\Rightarrow$ Only 4 matrices possible, $\mathbf{U}$ unrestricted.

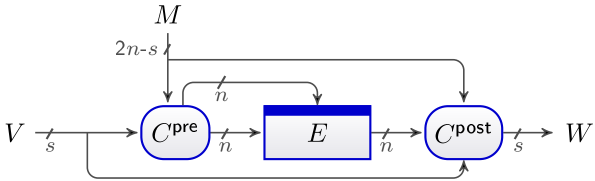
$\Rightarrow$ 16 Type-II schemes: 8 as identified by [BRS02] + 8 that are Type-I.

Chopped Compression Functions ($s < n$)



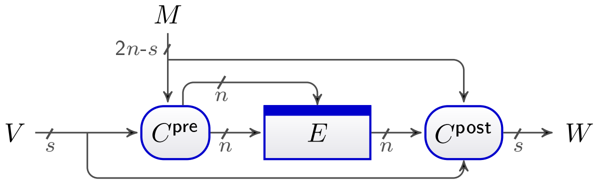
1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\text{post}}(M, V, \cdot)$ is bijective .
3. For all K, Y: $C^{\text{aux}}(K, \cdot, Y)$ is bijective .

Chopped Compression Functions $(s < n)$



1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\text{post}}(M, V, \cdot)$ is ~~bijective~~ balanced .
3. For all K, Y: $C^{\text{aux}}(K, \cdot, Y)$ is ~~bijective~~ balanced .

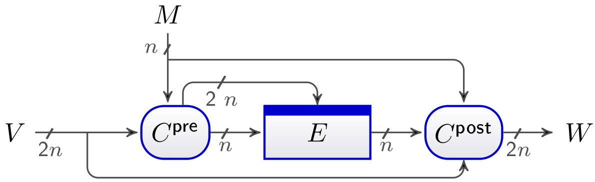
Chopped Compression Functions ($s < n$)



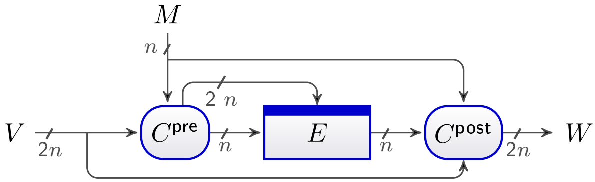
1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\mathsf{post}}(M, V, \cdot)$ is ~~bijective~~ balanced .
3. For all K, Y: $C^{\mathsf{aux}}(K, \cdot, Y)$ is ~~bijective~~ balanced .

$$\mathsf{Adv}_H^{\mathsf{coll}}(q) \le q(q+1)/2^s$$

Immediate consequence: chopping e.g., Davies-Meyer is secure.

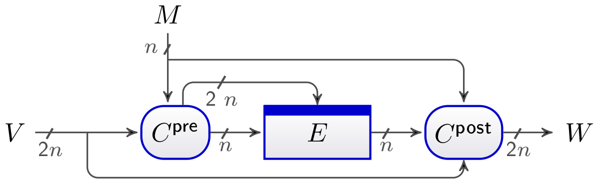


1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\text{post}}(M, V, \cdot)$ is bijective .
3. For all K, Y: $C^{\text{aux}}(K, \cdot, Y)$ is bijective .



1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\mathsf{post}}(M, V, \cdot)$ is ~~bijective~~ injective .
3. For all K, Y: $C^{\mathsf{aux}}(K, \cdot, Y)$ is ~~bijective~~ injective .

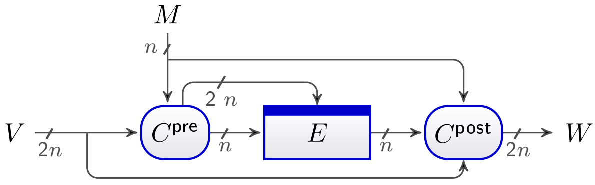
Supercharged Compression Functions
Specified for the double-length scenario



1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\mathsf{post}}(M, V, \cdot)$ is ~~bijective~~ injective .
   Range denoted by $R_{\mathsf{pre},(M,V)}$
3. For all K, Y: $C^{\mathsf{aux}}(K, \cdot, Y)$ is ~~bijective~~ injective .
   Range denoted by $R_{\mathsf{aux},(K,Y)}$

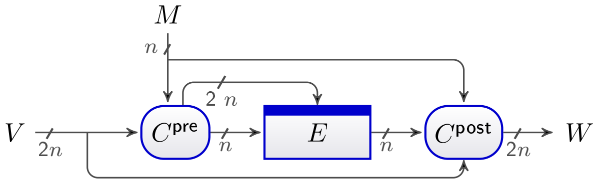
Supercharged Compression Functions

Specified for the double-length scenario



1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\text{post}}(M, V, \cdot)$ is ~~bijective~~ injective .
   Range denoted by $R_{\text{pre},(M,V)}$
3. For all K, Y: $C^{\text{aux}}(K, \cdot, Y)$ is ~~bijective~~ injective .
   Range denoted by $R_{\text{aux},(K,Y)}$

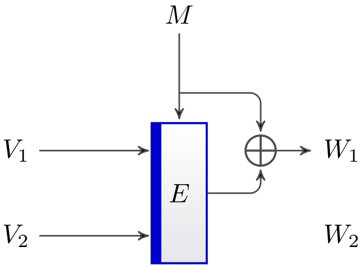
$$\gamma = \max\left\{|R_Z \cap R_{Z'}| : Z, Z' \in \{\text{pre}, \text{aux}\} \times \{0,1\}^{2n+n}, Z \neq Z'\right\}$$



1. The preprocessing C^{pre} is bijective.
2. For all M, V: $C^{\text{post}}(M, V, \cdot)$ is ~~bijective~~ injective .
   Range denoted by $R_{\text{pre},(M,V)}$
3. For all K, Y: $C^{\text{aux}}(K, \cdot, Y)$ is ~~bijective~~ injective .
   Range denoted by $R_{\text{aux},(K,Y)}$

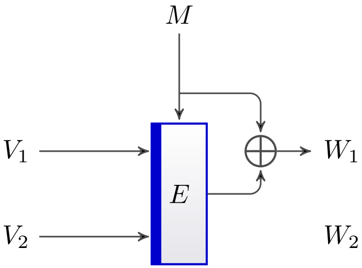
$$\gamma = \max\left\{|R_Z \cap R_{Z'}| : Z, Z' \in \{\text{pre}, \text{aux}\} \times \{0,1\}^{2n+n}, Z \neq Z'\right\}$$

$$\mathsf{Adv}_H^{\text{coll}}(q) \leq \frac{\gamma^{1/2} n q}{2^{n-6}}$$



$$W_2 = MW_1^2 + V_1W_1 + V_2$$

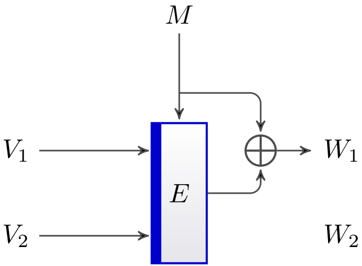
A Rate-1 Double-Length Compression Function
Collision Resistance



$$W_2 = MW_1^2 + V_1W_1 + V_2$$

$$\begin{aligned} R_{\mathsf{pre},(M,V_1,V_2)} &= \left\{(W, MW^2 + V_1W + V_2) | W \in \{0,1\}^n\right\} \\ R_{\mathsf{aux},(K_1,K_2,Y)} &= \left\{(W, W^3 + YW^2 + K_1W + K_2) | W \in \{0,1\}^n\right\} \ . \end{aligned}$$

A Rate-1 Double-Length Compression Function
Collision Resistance



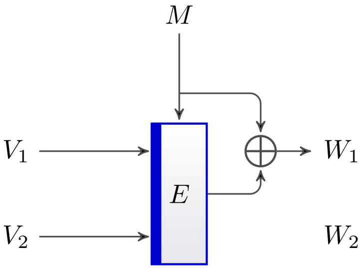
$$W_2 = MW_1^2 + V_1W_1 + V_2$$

$$R_{\mathsf{pre},(M,V_1,V_2)} = \left\{(W, MW^2 + V_1W + V_2) | W \in \{0,1\}^n\right\}$$
$$R_{\mathsf{aux},(K_1,K_2,Y)} = \left\{(W, W^3 + YW^2 + K_1W + K_2) | W \in \{0,1\}^n\right\} .$$

$$\gamma = 3 \quad \Rightarrow \quad \mathsf{Adv}_H^{\mathsf{coll}}(q) \leq 2(4n+2)q/2^n .$$

A Rate-1 Double-Length Compression Function

Collision Resistance



$$W_2 = V_2 W_1^2 + V_1 W_1 + M$$

$$\begin{aligned} R_{\mathsf{pre},(M,V_1,V_2)} &= \left\{(W, V_2W^2 + V_1W + M) | W \in \{0,1\}^n\right\} \\ R_{\mathsf{aux},(K_1,K_2,Y)} &= \left\{(W, K_2W^2 + (K_1+1)W + Y) | W \in \{0,1\}^n\right\} \ . \end{aligned}$$

$$\gamma = 2^n \quad \Rightarrow \quad \mathsf{Adv}_H^{\mathsf{coll}}(q) \le 2(4n+2)q/2^{n/2} \ .$$

Conclusion

- Presented a new framework to capture blockcipher based hashing.
- PGV/BRS results can be derived from it.
- Allows for easy generalization for chopping and overloading.
- Developed theory for supercharging compression functions.
- A new collision resistant rate-1 double length construction.