Generic Attacks on Double Block Length Hashing

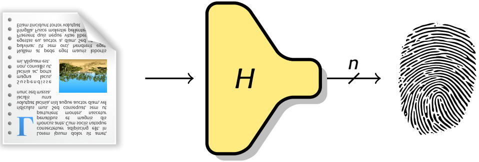
Gaëtan Leurent

Inria, France

GAPS Workshop

Hash functions

- Public function $H : \{0,1\}^* \to \{0,1\}^n$
- Should behave like a random function
  - No structural property
  - Cryptographic properties without any key!
- Concrete security goals



Preimage attack

Given H and $\overline{X}$, find M s.t. $H(M) = \overline{X}$. Ideal security: 2^n.

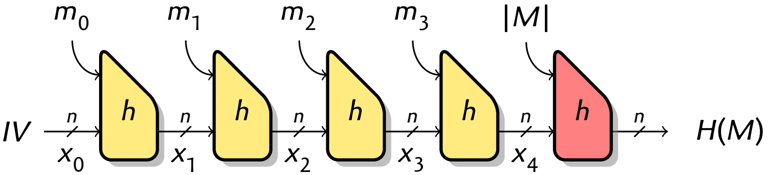
Second-preimage attack

Given H and M_1, find $M_2 \neq M_1$ s.t. $H(M_1) = H(M_2)$. Ideal security: 2^n.

Collision attack

Given H, find $M_1 \neq M_2$ s.t. $H(M_1) = H(M_2)$. Ideal security: $2^{n/2}$.

The Merkle-Damgård construction (SHA-1, SHA-2)



- n-bit state, compression function $h : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^n$
- Padding rule (ignored in this talk for simplicity)
- Finalization using message length (MD strengthening)
- Notation: Iterated compression function h^*
  - $h^*(x, m_0 \parallel m_1 \parallel m_2) = h(h(h(x, m_0), m_1), m_2)$
- Security reductions:
  - Hash collisions imply compression function collision (generic security $2^{n/2}$)
  - Hash preimages imply finalization preimages (generic security 2^n)
- Indifferentiable up to $2^{n/2}$ queries [Coron, Dodis, Malinaud & Puniya, C'05]

Generic attacks on Merkle-Damgård

Many properties "between" collision and preimage broken with birthday complexity, by generic attacks exploiting collisions in smart ways

Second-preimage for long challenges [Kelsey & Schneier, Eurocrypt '05]

Given a long challenge C ($\mathrm{len}(C) = 2^s$), find $M \neq C$ with $H(M) = H(C)$ Complexity $\tilde{\mathcal{O}}(2^{n-s})$

Multicollision [Joux, Crypto '04]

Find a large set of message $\{M_i\}$ s.t. $\forall i,\ H(M_i) = H(M_0)$ Complexity $\tilde{\mathcal{O}}(2^{n/2})$

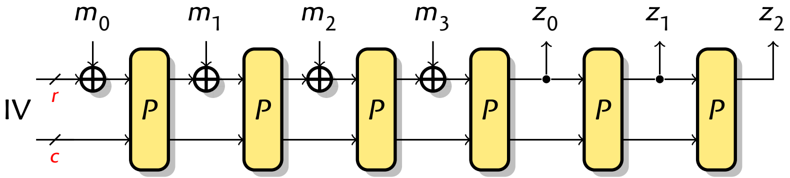
Chosen-prefix collision [Stevens, Lenstra & de Weger, EC'07]

Given challenges C, C', find M, M' s.t. $H(C \parallel M) = H(C' \parallel M')$ Complexity $\mathcal{O}(2^{n/2})$

Diamond structure [Kelsey & Kohno, EC'06]

Given challenges $\{C_i\}$, find $\{M_i\}$ s.t. $\forall i,\ H(C_i \parallel M_i) = H(C_0 \parallel M_0)$ Complexity $\tilde{\mathcal{O}}(\sqrt{|\{C_i\}|}2^{n/2})$

The sponge construction (SHA-3, Ascon)

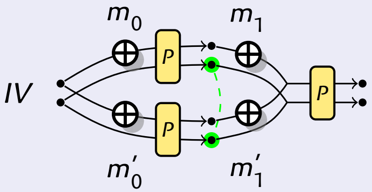


- b-bit state, cryptographic permutation $P : \{0,1\}^b \to \{0,1\}^b$
  - State split into rate r and capacity c: $b = c + r$
- Padding rule (ignored in this talk for simplicity)
- Tight security in the random permutation model:
  - Indifferentiable up to $2^{c/2}$ queries [Bertoni, Daemen, Peters & Van Assche, EC'08]
  - Collision attack in $\min(2^{c/2}, 2^{n/2})$
  - Preimage attack in $\min(\max(2^{c/2}, 2^{n-r}), 2^n)$ [Lefevre & Mennink, Crypto '22]
  - Second-preimage in $\min(2^{c/2}, 2^n)$

Generic attacks on sponge

- Notation:
  - State after absorption and processing: $S(m_1 \parallel m_2 \parallel m_3)$
  - Rate and capacity part of S: $\mathcal{R}(S)$ and $\mathcal{C}(S)$

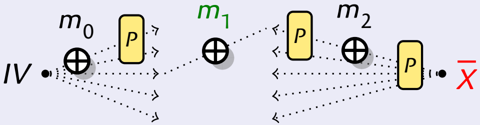
Collision attack



1. Find (m_0, m'_0) colliding on capacity: $\mathcal{C}(S(m_o)) = \mathcal{C}(S(m'_0))$
2. Choose (m_1, m'_1) with $m_1 \oplus m'_1 = \mathcal{R}(S(m_0)) \oplus \mathcal{R}(S(m'_0))$

Total complexity $2^{c/2}$

Preimage attack: meet-in-the-middle



1. Eval $S(m_0) = P(IV + m_0)$ for $2^{c/2}$ m_0
2. Eval $\overleftarrow{S}(m_2) = P^{-1}(P^{-1}(\overline{X} + m_2))$ for $2^{c/2}$ m_2
3. Find (m_0, m_2) colliding on capacity $\mathcal{C}(S(m_0)) = \mathcal{C}(\overleftarrow{S}(m_2))$
4. Choose $m_1 = \mathcal{R}(S(m_0)) \oplus \mathcal{R}(\overleftarrow{S}(m_2))$

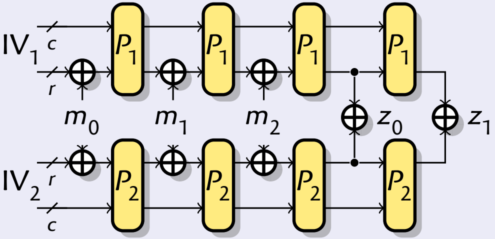
Total complexity $2^{c/2}$

Increasing state size

- Security of hash functions strongly related to state size
  - Indifferentiability bound $2^{n/2}$ for Merkle-Damgård, $2^{c/2}$ for sponge

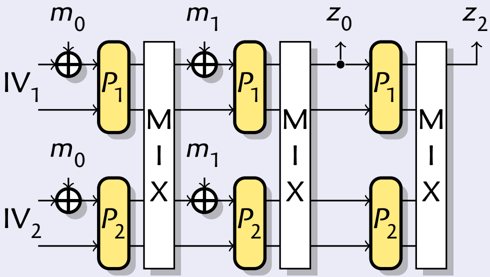
Combiners

- Compute two hash functions H_1, H_2 in parallel and combine output *e.g.* $H : M \mapsto H_1(M) \oplus H_2(M)$
- Motivation: robustness



Double block length

- Use two primitives in parallel and mix states
- *E.g.* double sponge [ToSC'24]



Outline: Generic security of double block length hashing

Goals of the talk

- Identify GAPS between proofs and attacks
- Fill some of them

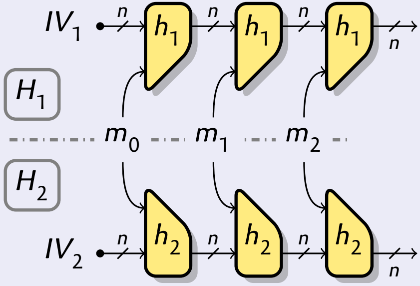
- Combiners with two Merkle-Damgård hash functions
  - Overview of known results: multicollision and interchange structure
- Combiners with two sponge hash functions
  - Folklore generic attacks using multicollisions
  - New distinguisher *(joint work with César Mathéus)*
- Double sponge
  - New distinguisher *(joint work with César Mathéus)*

Outline

Generic attacks against Merkle-Damgård combiners

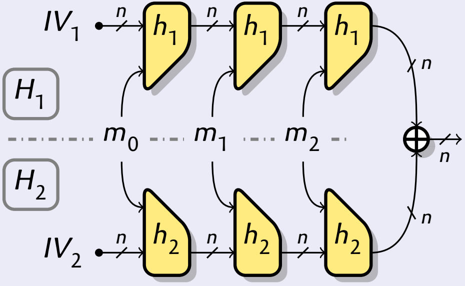
Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- $2n$-bit output



XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- n-bit output



Generic attacks against Merkle-Damgård combiners

Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- $2n$-bit output
- Generic security: attacks / proofs

	attacks	proofs
Collisions:	$2^{n/2}$	$2^{n/2}$
Preimages:	2^n	2^n
Indifferentiability:	$2^{n/2}$	$2^{n/2}$

XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- n-bit output
- Generic security: attacks / proofs

	attacks	proofs
Collisions:	$2^{n/2}$	$2^{n/2}$
Preimages:	$2^{3n/5}$	$2^{n/2}$
Indifferentiability:	$2^{n/2}$	$2^{n/2}$

Multicollision [Joux, C'04]

If H_1 and H_2 are good MD hash functions, $H_1 \parallel H_2$ is not stronger!

Interchange structure [L & Wang, EC'15]

If H_1 and H_2 are good MD hash functions, $H_1 \oplus H_2$ is weaker!

Generic attacks against Merkle-Damgård combiners

Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- $2n$-bit output
- Generic security: attacks / proofs
  - Collisions: $2^{n/2}$ / $2^{n/2}$
  - Preimages: 2^n / 2^n
  - Indifferentiability: $2^{n/2}$ / $2^{n/2}$

XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- n-bit output
- Generic security: attacks / proofs
  - Collisions: $2^{n/2}$ / $2^{n/2}$
  - Preimages: $2^{3n/5}$ / $2^{n/2}$
  - Indifferentiability: $2^{n/2}$ / $2^{n/2}$

Multicollision [Joux, C'04]

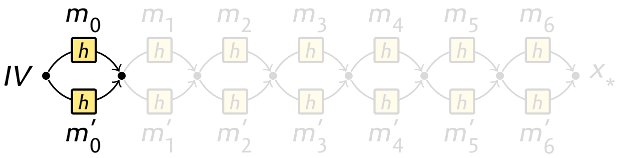
If H_1 and H_2 are good MD hash functions, $H_1 \parallel H_2$ is not stronger!

Interchange structure [L & Wang, EC'15]

If H_1 and H_2 are good MD hash functions, $H_1 \oplus H_2$ is weaker!

Multicollisions

[Joux, Crypto '04]



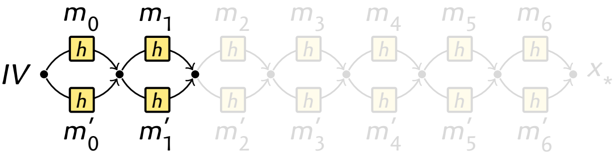
1. Find a collision pair m_0 / m'_0 starting from IV
2. Find a collision pair m_1 / m'_1 starting from $x_1 = h^*(m_0)$
3. Repeat t times
4. This yields 2^t messages with the same hash:

$m_0 m_1 m_2 \dots$	$m'_0 m_1 m_2 \dots$	$m_0 m'_1 m_2 \dots$	$m'_0 m'_1 m_2 \dots$
$m_0 m_1 m'_2 \dots$	$m'_0 m_1 m'_2 \dots$	$m_0 m'_1 m'_2 \dots$	$m'_0 m'_1 m'_2 \dots$

- Complexity $t \cdot 2^{n/2}$ vs. $\approx 2^{\frac{2^t-1}{2^t} n}$ for a random function

Multicollisions

[Joux, Crypto '04]



1. Find a collision pair m_0 / m'_0 starting from IV
2. Find a collision pair m_1 / m'_1 starting from $x_1 = h^*(m_0)$
3. Repeat t times
4. This yields 2^t messages with the same hash:

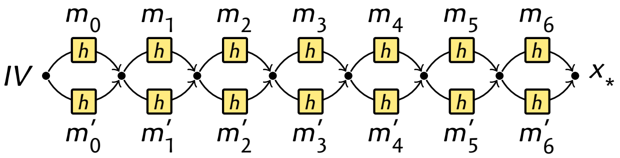
$m_0 m_1 m_2 \dots$ $m'_0 m_1 m_2 \dots$ $m_0 m'_1 m_2 \dots$ $m'_0 m'_1 m_2 \dots$

$m_0 m_1 m'_2 \dots$ $m'_0 m_1 m'_2 \dots$ $m_0 m'_1 m'_2 \dots$ $m'_0 m'_1 m'_2 \dots$

- Complexity $t \cdot 2^{n/2}$ vs. $\approx 2^{\frac{2^t-1}{2^t}n}$ for a random function

Multicollisions

[Joux, Crypto '04]

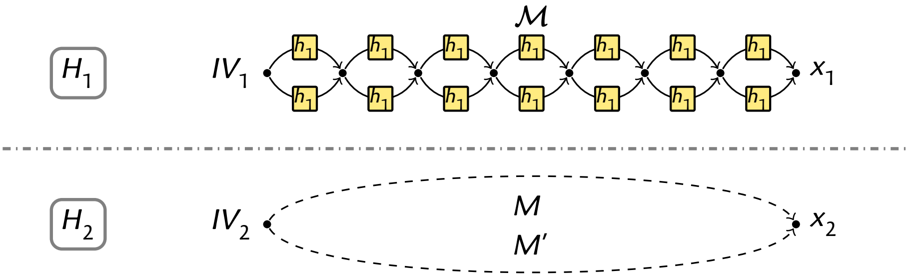


1. Find a collision pair m_0 / m'_0 starting from IV
2. Find a collision pair m_1 / m'_1 starting from $x_1 = h^*(m_0)$
3. Repeat t times
4. This yields 2^t messages with the same hash:

$$\begin{array}{cccc} m_0 m_1 m_2 \dots & m'_0 m_1 m_2 \dots & m_0 m'_1 m_2 \dots & m'_0 m'_1 m_2 \dots \\ m_0 m_1 m'_2 \dots & m'_0 m_1 m'_2 \dots & m_0 m'_1 m'_2 \dots & m'_0 m'_1 m'_2 \dots \end{array}$$

- Complexity $t \cdot 2^{n/2}$ vs. $\approx 2^{\frac{2^t - 1}{2^t} n}$ for a random function

State collision for parallel Merkle-Damgård [Joux, C'04]



1. Build a $2^{n/2}$-multicollision for H_1

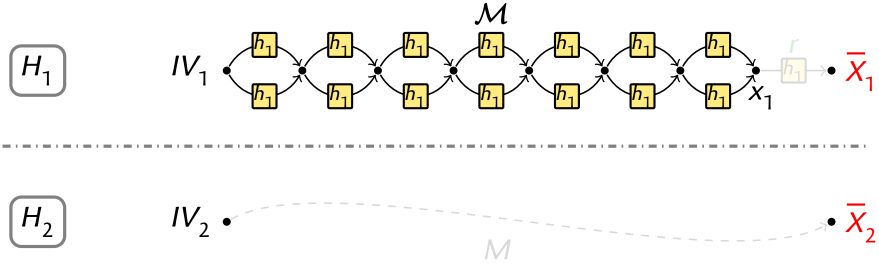
$$\forall M \in \mathcal{M}, H_1(M) = x_1$$

2. Find $M, M' \in \mathcal{M}$ s.t. $H_2(M) = H_2(M')$

- Complexity $\tilde{O}(2^{n/2})$ vs. 2^n for a $2n$-bit hash function.

State preimage for parallel Merkle-Damgård

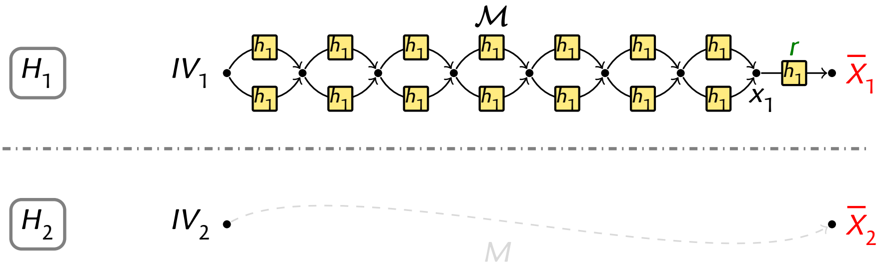
[Joux, C'04]



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 — $\forall M \in \mathcal{M}, h_1^*(M) = x_1$
2. Find a preimage for H_1: $h(h(x_1, r)) = \overline{X}_1$ — $\forall M \in \mathcal{M}, H_1(M \parallel r) = \overline{X}_1$
3. Find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r) = \overline{X}_2$

- Complexity $\tilde{O}(2^n)$ vs. 2^{2n} for a $2n$-bit hash function.

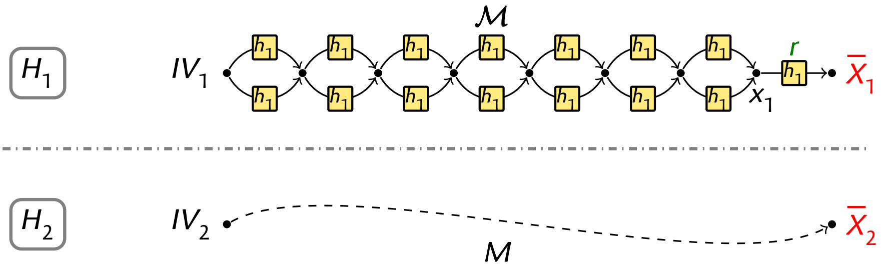
State preimage for parallel Merkle-Damgård [Joux, C'04]



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 — $\forall M \in \mathcal{M}, h_1^*(M) = x_1$
2. Find a preimage for H_1: $h(h(x_1, r)) = \overline{X}_1$ — $\forall M \in \mathcal{M}, H_1(M \parallel r) = \overline{X}_1$
3. Find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r) = \overline{X}_2$

- Complexity $\tilde{O}(2^n)$ vs. 2^{2n} for a $2n$-bit hash function.

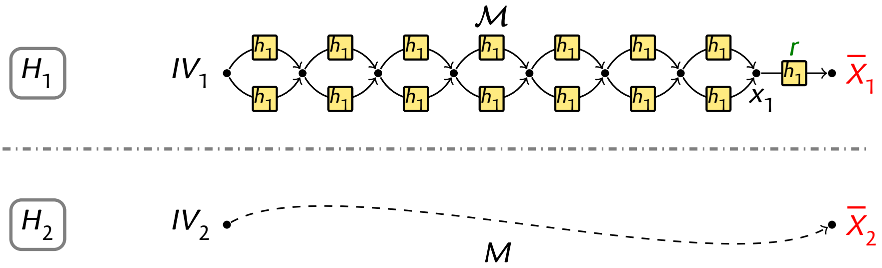
State preimage for parallel Merkle-Damgård [Joux, C'04]



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 — $\forall M \in \mathcal{M}, h_1^*(M) = x_1$
2. Find a preimage for H_1: $h(h(x_1, r)) = \overline{X}_1$ — $\forall M \in \mathcal{M}, H_1(M \parallel r) = \overline{X}_1$
3. Find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r) = \overline{X}_2$

- Complexity $\tilde{O}(2^n)$ vs. 2^{2n} for a $2n$-bit hash function.

State preimage for parallel Merkle-Damgård [Joux, C'04]



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 — $\forall M \in \mathcal{M}, h_1^*(M) = x_1$
2. Find a preimage for H_1: $h(h(x_1, r)) = \overline{X}_1$ — $\forall M \in \mathcal{M}, H_1(M \parallel r) = \overline{X}_1$
3. Find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r) = \overline{X}_2$

- Complexity $\tilde{\mathcal{O}}(2^n)$ vs. 2^{2n} for a $2n$-bit hash function.

Generic attacks against Merkle-Damgård combiners

Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- $2n$-bit output
- Generic security: attacks / proofs
  - Collisions: $2^{n/2}$ / $2^{n/2}$
  - Preimages: 2^n / 2^n
  - Indifferentiability: $2^{n/2}$ / $2^{n/2}$

XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- n-bit output
- Generic security: attacks / proofs
  - Collisions: $2^{n/2}$ / $2^{n/2}$
  - Preimages: $2^{3n/5}$ / $2^{n/2}$
  - Indifferentiability: $2^{n/2}$ / $2^{n/2}$

Multicollision [Joux, C'04]

If H_1 and H_2 are good MD hash functions, $H_1 \parallel H_2$ is not stronger!

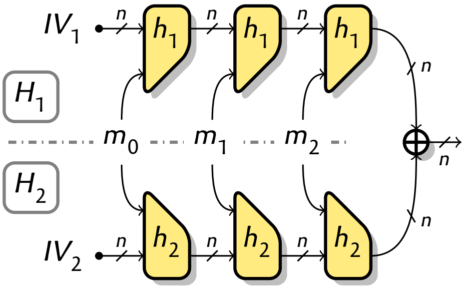
Interchange structure [L & Wang, EC'15]

If H_1 and H_2 are good MD hash functions, $H_1 \oplus H_2$ is weaker!

Preimage on the XOR of two Merkle-Damgård

[L & Wang, EC'15]

$H(M) = H_1(M) \oplus H_2(M)$

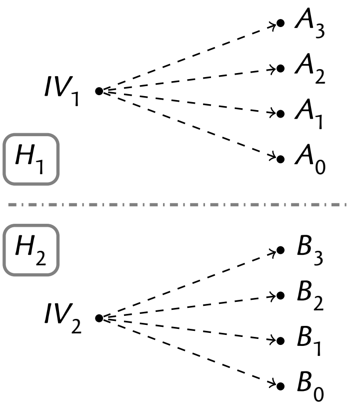


Strategy:

1. Structure to control H_1 and H_2 independently:
   - Sets of states $\mathcal{A} = \{A_j\}$, $\mathcal{B} = \{B_k\}$
   - Set of messages $\{\mathbf{M}_{jk}\}$ with
     $$h_1^*(\mathbf{M}_{jk}) = A_j$$
     $$h_2^*(\mathbf{M}_{jk}) = B_k$$
2. Preimage search for $\overline{X}$:
   - For random blocks r, match $\{g_1(h_1(A_j, r))\}$ and $\{g_2(h_2(B_k, r)) \oplus \overline{X}\}$
   - If there is a match (j, k): Get $\mathbf{M}_{jk}$, preimage is $M = \mathbf{M}_{jk} \parallel r$
   - Complexity $\mathcal{O}(2^n / \min\{|\mathcal{A}|, |\mathcal{B}|\})$

Preimage on the XOR of two Merkle-Damgård [L & Wang, EC'15]

$H(M) = H_1(M) \oplus H_2(M)$



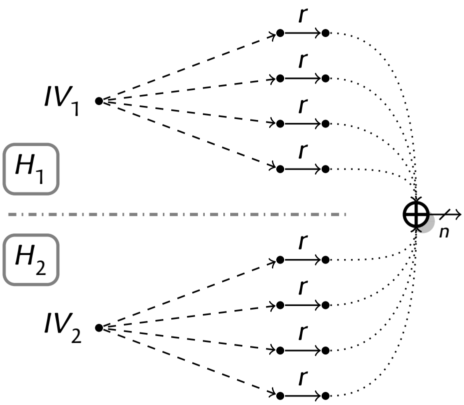
Strategy:

1. Structure to control H_1 and H_2 independently:
   - Sets of states $\mathcal{A} = \{A_j\}$, $\mathcal{B} = \{B_k\}$
   - Set of messages $\{\mathbf{M}_{jk}\}$ with
     $$h_1^*(\mathbf{M}_{jk}) = A_j$$
     $$h_2^*(\mathbf{M}_{jk}) = B_k$$
2. Preimage search for $\overline{X}$:
   - For random blocks r, match $\{g_1(h_1(A_j, r))\}$ and $\{g_2(h_2(B_k, r)) \oplus \overline{X}\}$
   - If there is a match (j, k): Get $\mathbf{M}_{jk}$, preimage is $M = \mathbf{M}_{jk} \parallel r$
   - Complexity $\mathcal{O}(2^n / \min\{|\mathcal{A}|, |\mathcal{B}|\})$

Preimage on the XOR of two Merkle-Damgård

[L & Wang, EC'15]

$H(M) = H_1(M) \oplus H_2(M)$



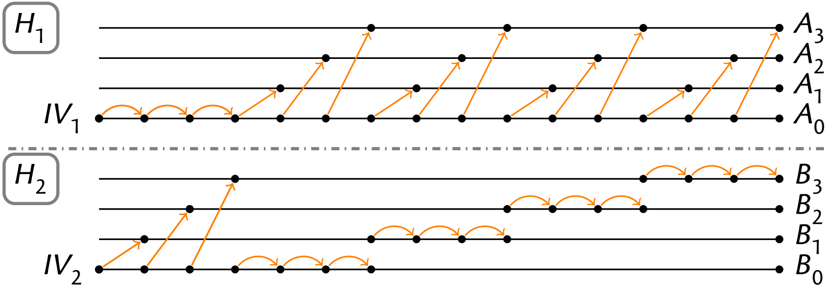
Strategy:

1. Structure to control H_1 and H_2 independently:
   - Sets of states $\mathcal{A} = \{A_j\}$, $\mathcal{B} = \{B_k\}$
   - Set of messages $\{\mathbf{M}_{jk}\}$ with
     $$h_1^*(\mathbf{M}_{jk}) = A_j$$
     $$h_2^*(\mathbf{M}_{jk}) = B_k$$
2. Preimage search for $\overline{X}$:
   - For random blocks r, match $\{g_1(h_1(A_j, r))\}$ and $\{g_2(h_2(B_k, r)) \oplus \overline{X}\}$
   - If there is a match (j, k): Get $\mathbf{M}_{jk}$, preimage is $M = \mathbf{M}_{jk} \parallel r$
   - Complexity $\mathcal{O}(2^n / \min\{|\mathcal{A}|, |\mathcal{B}|\})$

Interchange structure

[L & Wang, EC'15]

- Interchange structure for a large set of output states

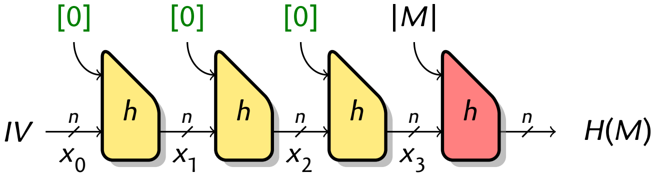


- Complexity $\tilde{\mathcal{O}}(2^{n/2+2t})$ to build a structure with $|\mathcal{A}| = |\mathcal{B}| = 2^t$

- Complexity $\tilde{\mathcal{O}}(2^{5n/6})$ for preimages (tradeoff)

Alternative structure using cycles

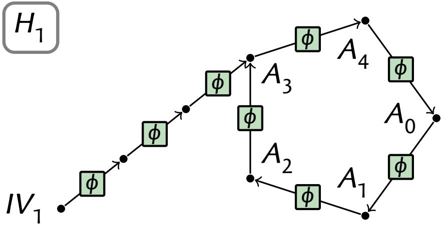
- Alternative presentation of "multicycles" [Bao, Wang, Guo, Gu, C'17]

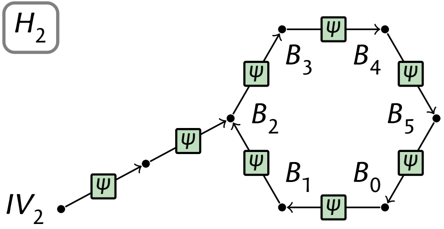


- Using a long message repeating a fixed block $M = [0]^\lambda$, we iterate fixed functions:

$$\phi : x \mapsto h_1(x, [0])$$
$$\psi : x \mapsto h_2(x, [0])$$

Alternative structure using cycles





- Use cyclic nodes as end-point:
  - $\mathcal{A} = H_1$ cycle, length ℓ_1
  - $\mathcal{B} = H_2$ cycle, length ℓ_2
- With suitable naming, for λ large enough:

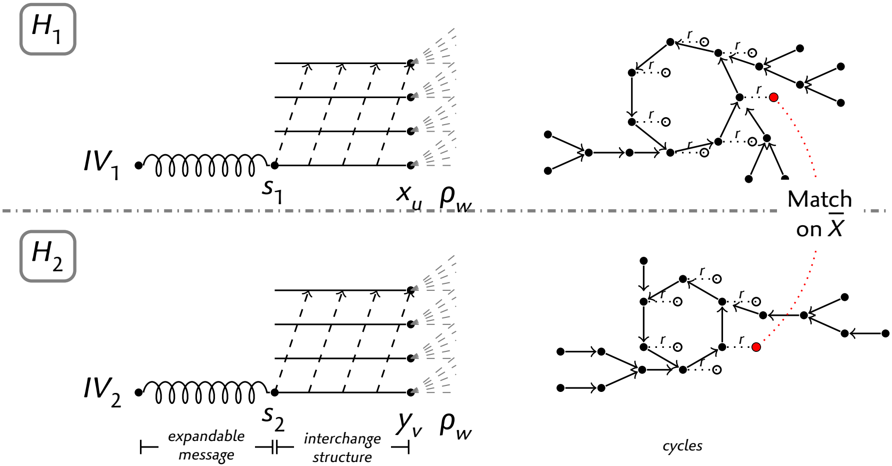
$$h_1^*([0]^\lambda) = A_{\lambda \bmod \ell_1} \qquad h_2^*([0]^\lambda) = B_{\lambda \bmod \ell_2}$$

- To reach (A_j, B_k), use Chinese Remainder

$$\begin{cases} h_1^*([0]^\lambda) = A_j \\ h_2^*([0]^\lambda) = B_k \end{cases} \iff \begin{cases} \lambda \bmod \ell_1 = i \\ \lambda \bmod \ell_2 = j \end{cases}$$

  - λ uniformly distributed in range of size $\ell_1 \ell_2$
  - $\Pr[\lambda < 2^t] \approx 2^{n-t}$
- Complexity $\tilde{\mathcal{O}}(2^{3n/4})$ for preimages (tradeoff)

Advanced preimage attack [BHBLS24]



- Using interchange, small cycles, expandable message
- Complexity $\tilde{\mathcal{O}}(2^{3n/5})$

GAPS: Preimage on the XOR of two Merkle-Damgård

Interchange structure

- Complexity $\tilde{\mathcal{O}}(2^{5n/6})$ [LW15]
- Works for Merkle-Damgård and HAIFA
  - Finalization function, block counter at each round
- Short messages: length $\tilde{\mathcal{O}}(2^{n/3})$

Using cycles

- Complexity $\tilde{\mathcal{O}}(2^{3n/4})$ (simple)
- Complexity $\tilde{\mathcal{O}}(2^{5n/8})$ [BWGG17]
- Complexity $\tilde{\mathcal{O}}(2^{11n/18})$ [BDGLW20]
- Complexity $\tilde{\mathcal{O}}(2^{3n/5})$ [BHBLS24]
- Works only for Merkle-Damgård mode
  - Finalization function, same function at each step
- Long messages: length $\tilde{\mathcal{O}}(2^{3n/5})$

- Security proof (indifferentiability) up to $2^{n/2}$ queries

Outline

Merkle-Damgård Combiners

- Multicollisions
- Preimage attack on the XOR combiner

Sponge Combiners

- Multicollisions
- New 4-sum distinguisher

The Double Sponge

- New 4-sum distinguisher

Generic attacks against sponge combiners

- Consider large n, 2nd-preimage rather than preimage $\implies$ ignore squeezing

Concatenation combiner

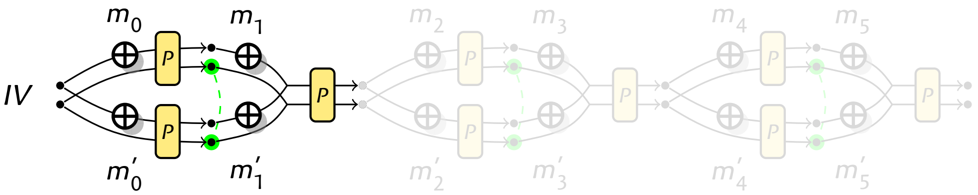
- $H(M) = H_1(M) \| H_2(M)$
- Generic security: attacks / proofs
  - Collisions: ? / $2^{c/2}$
  - 2nd-preimages: ? / $2^{c/2}$
  - Indifferentiability: $2^{c/2}$ / $2^{c/2}$

XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- Generic security: attacks / proofs
  - Collisions: ? / $2^{c/2}$
  - 2nd preimages: ? / $2^{c/2}$
  - Indifferentiability: ? / $2^{c/2}$

- Not much analysis of sponge combiners
- Probably because we can increase sponge security by increasing r
- Combiner could be useful for small b, if the provide security beyond $2^{c/2}$

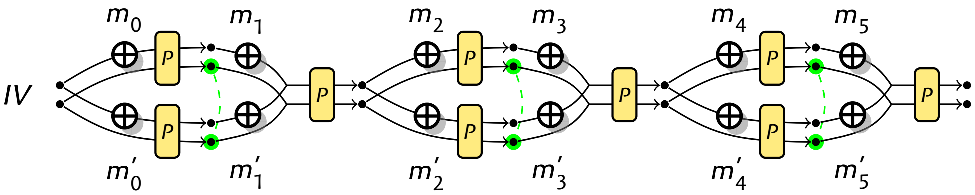
Multicollision for a sponge



1. Find (m_0, m'_0) colliding on capacity: $\mathcal{C}(S(m_o)) = \mathcal{C}(S(m'_0))$
2. Choose (m_1, m'_1) with $m_1 \oplus m'_1 = \mathcal{R}(S(m_0)) \oplus \mathcal{R}(S(m'_0))$
3. Repeat

- Complexity $t \cdot 2^{c/2}$

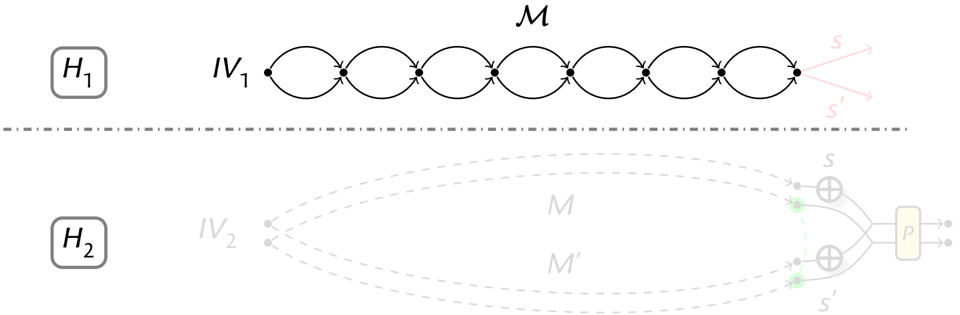
Multicollision for a sponge



1. Find (m_0, m'_0) colliding on capacity: $\mathcal{C}(S(m_o)) = \mathcal{C}(S(m'_0))$
2. Choose (m_1, m'_1) with $m_1 \oplus m'_1 = \mathcal{R}(S(m_0)) \oplus \mathcal{R}(S(m'_0))$
3. Repeat

- Complexity $t \cdot 2^{c/2}$

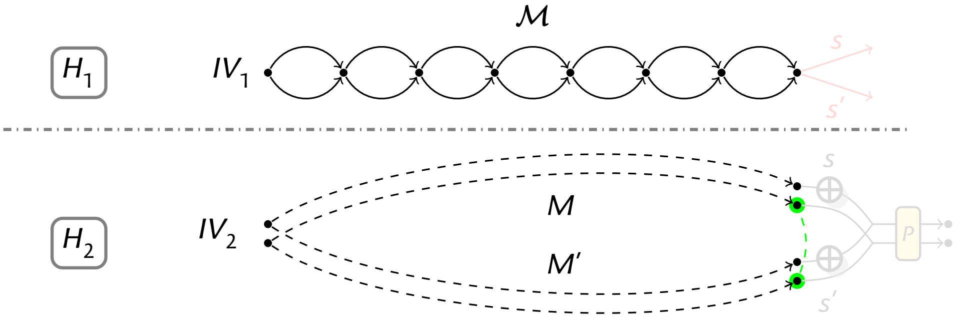
State collision for parallel sponges



1. Build a $2^{c/2}$-multicollision $\mathcal{M}$ for H_1
2. Find a pair $M, M' \in \mathcal{M}$ colliding on the capacity: $\mathcal{C}(S_2(M)) = \mathcal{C}(S_2(M'))$
3. Choose s, s' with $s \oplus s' = \mathcal{R}(S_2(M)) \oplus \mathcal{R}(S_2(M'))$

- Problem: $S_1(M \parallel s) \neq S_1(M' \parallel s')$

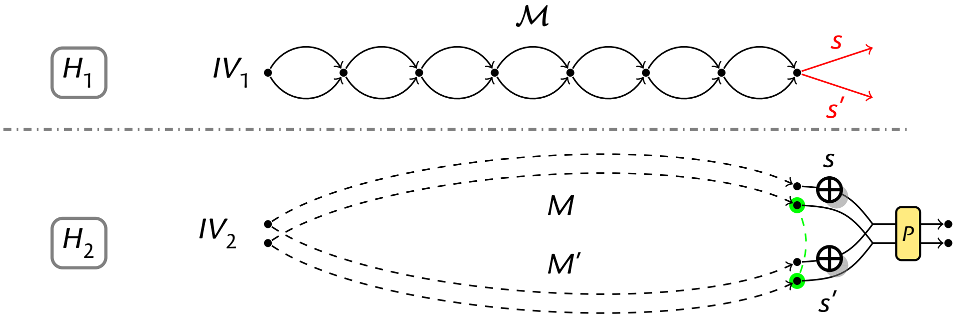
State collision for parallel sponges



1. Build a $2^{c/2}$-multicollision $\mathcal{M}$ for H_1
2. Find a pair $M, M' \in \mathcal{M}$ colliding on the capacity: $\mathcal{C}(S_2(M)) = \mathcal{C}(S_2(M'))$
3. Choose s, s' with $s \oplus s' = \mathcal{R}(S_2(M)) \oplus \mathcal{R}(S_2(M'))$

- Problem: $S_1(M \parallel s) \neq S_1(M' \parallel s')$

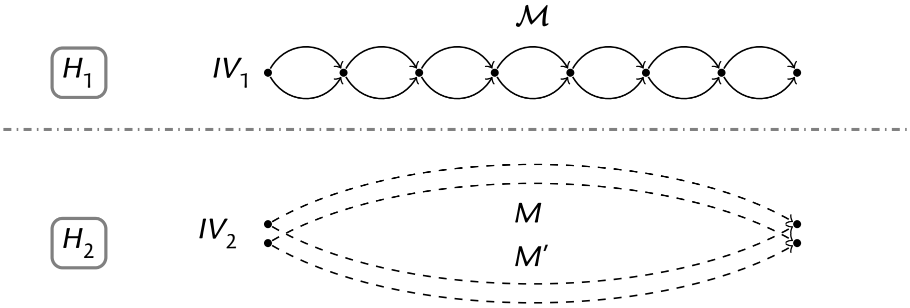
State collision for parallel sponges



1. Build a $2^{c/2}$-multicollision $\mathcal{M}$ for H_1
2. Find a pair $M, M' \in \mathcal{M}$ colliding on the capacity: $\mathcal{C}(S_2(M)) = \mathcal{C}(S_2(M'))$
3. Choose s, s' with $s \oplus s' = \mathcal{R}(S_2(M)) \oplus \mathcal{R}(S_2(M'))$

- Problem: $S_1(M \parallel s) \neq S_1(M' \parallel s')$

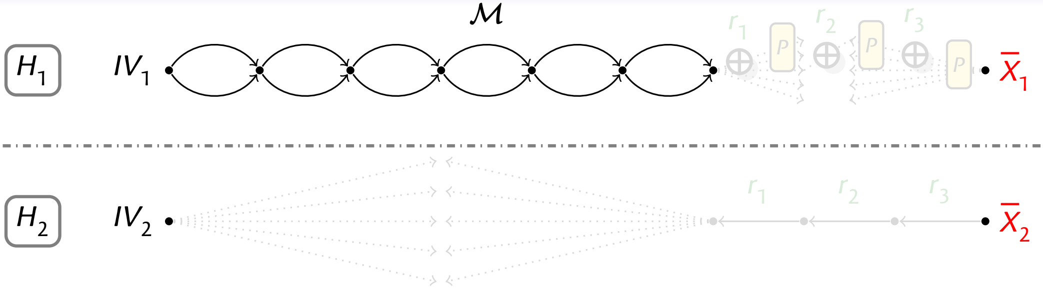
State collision for parallel sponges



1. Build a $2^{b/2}$-multicollision $\mathcal{M}$ for H_1
2. Find a pair $M, M' \in \mathcal{M}$ colliding on the full state: $S_2(M) = S_2(M')$

- Complexity $\tilde{O}(2^{b/2})$

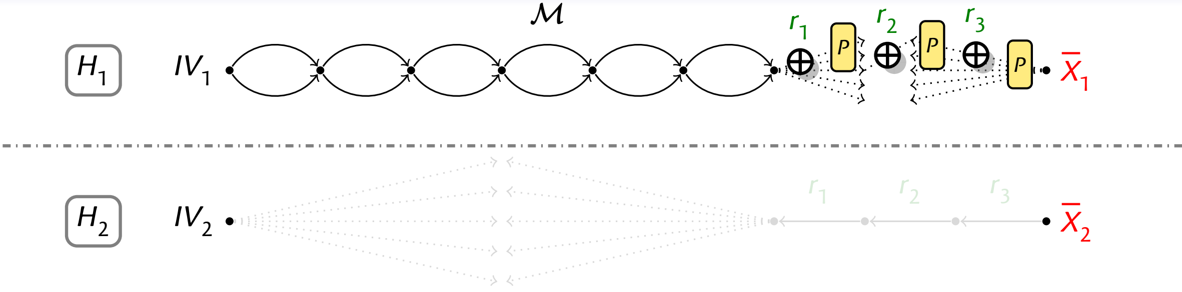
State preimage for parallel sponges



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 $\quad \forall M \in \mathcal{M}, h_1^*(M) =$
2. Using meet-in-the-middle, find H_1 preimage: $\quad \forall M \in \mathcal{M}, H_1(M \parallel r_1 \parallel r_2 \parallel r_3) = \bar{X}_1$
3. Using meet-in-the-middle, find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r_1 \parallel r_2 \parallel r_3) = \bar{X}_2$

- Complexity $\tilde{O}(2^{b/2})$

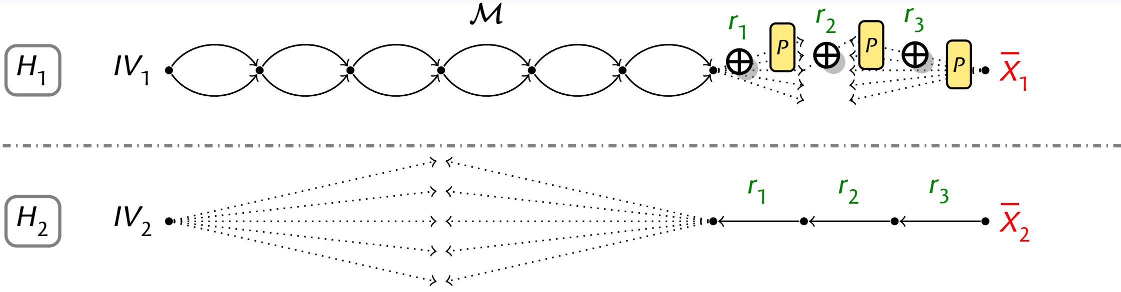
State preimage for parallel sponges



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 — $\forall M \in \mathcal{M}, h_1^*(M) =$
2. Using meet-in-the-middle, find H_1 preimage: $\forall M \in \mathcal{M}, H_1(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_1$
3. Using meet-in-the-middle, find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_2$

- Complexity $\tilde{O}(2^{b/2})$

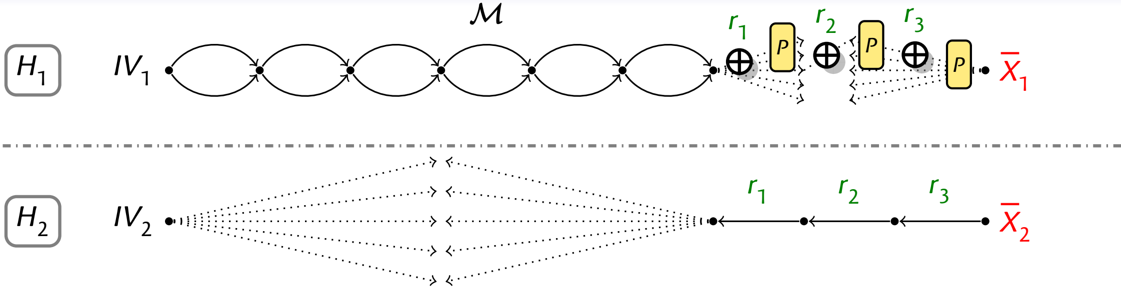
State preimage for parallel sponges



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 $\qquad \forall M \in \mathcal{M}, h_1^*(M) =$
2. Using meet-in-the-middle, find H_1 preimage: $\qquad \forall M \in \mathcal{M}, H_1(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_1$
3. Using meet-in-the-middle, find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_2$

- Complexity $\tilde{O}(2^{b/2})$

State preimage for parallel sponges



1. Build a 2^n-multicollision $\mathcal{M}$ for H_1 $\qquad \forall M \in \mathcal{M}, h_1^*(M) =$
2. Using meet-in-the-middle, find H_1 preimage: $\qquad \forall M \in \mathcal{M}, H_1(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_1$
3. Using meet-in-the-middle, find $M \in \mathcal{M}$ s.t. $H_2(M \parallel r_1 \parallel r_2 \parallel r_3) = \overline{X}_2$

- Complexity $\tilde{\mathcal{O}}(2^{b/2})$

Generic attacks against sponge combiners

- Consider large n, 2^{nd}-preimage rather than preimage $\implies$ ignore squeezing

Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- Generic security: attacks / proofs

	attacks	proofs
Collisions:	$2^{b/2}$	$2^{c/2}$
2^{nd}-preimages:	$2^{b/2}$	$2^{c/2}$
Indifferentiability:	$2^{c/2}$	$2^{c/2}$

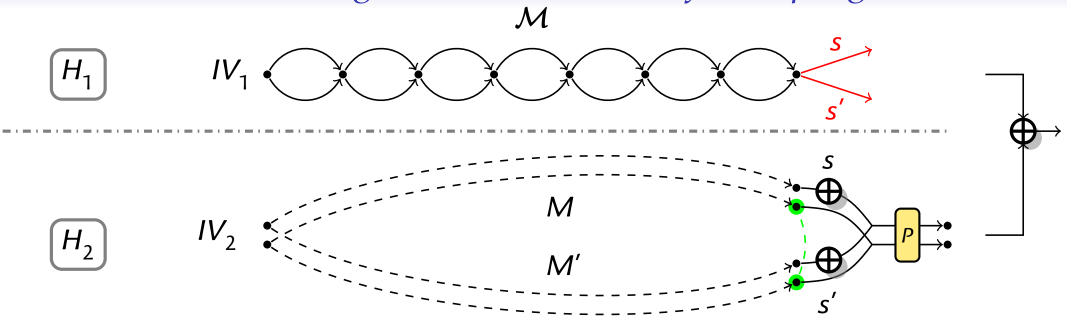
XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- Generic security: attacks / proofs

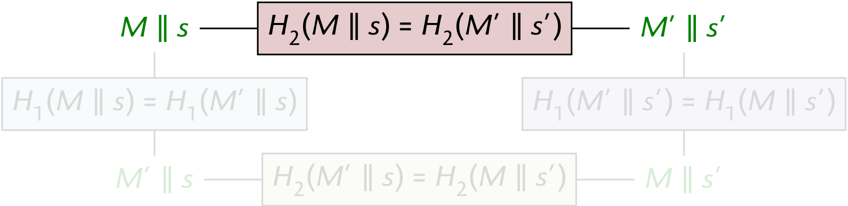
	attacks	proofs
Collisions:	$2^{b/2}$	$2^{c/2}$
2^{nd} preimages:	$2^{b/2}$	$2^{c/2}$
Indifferentiability:	$2^{b/2}$	$2^{c/2}$

- Attacks based on multicollisions have complexity order $2^{b/2} = 2^{c/2+r/2}$
- Rate seems to contribute to the security!
- Focus on indistinguishability gap for XOR combiner

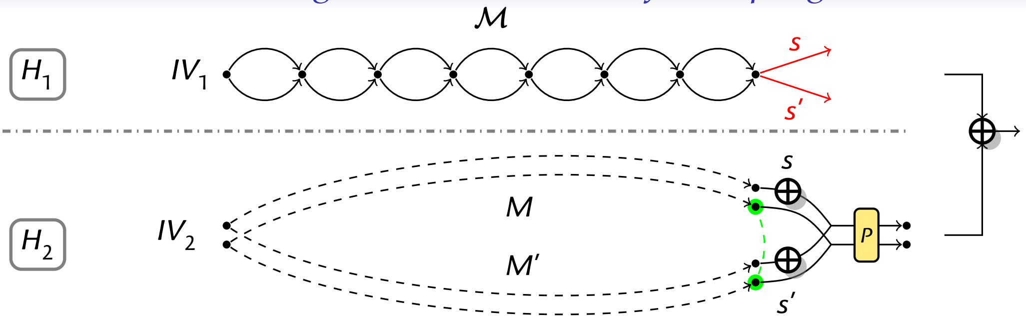
Distinguisher on the XOR of two sponges



- Start from failed collision attempt, use 4 messages



Distinguisher on the XOR of two sponges



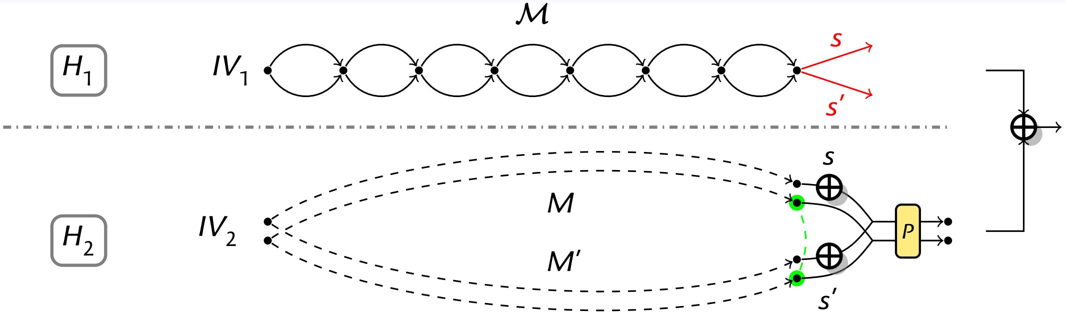
- Start from failed collision attempt, use 4 messages

$M \| s$ — $H_2(M \| s) = H_2(M' \| s')$ — $M' \| s'$

$H_1(M \| s) = H_1(M' \| s)$ $\qquad$ $H_1(M' \| s') = H_1(M \| s')$

$M' \| s$ — $H_2(M' \| s) = H_2(M \| s')$ — $M \| s'$

Distinguisher on the XOR of two sponges



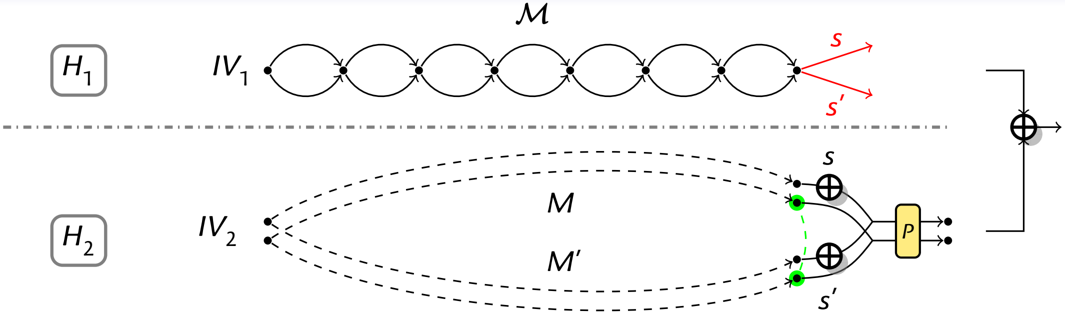
- Start from failed collision attempt, use 4 messages

$M \parallel s$ — $H_2(M \parallel s) = H_2(M' \parallel s')$ — $M' \parallel s'$

$H_1(M \parallel s) = H_1(M' \parallel s)$ $\quad$ $H_1(M' \parallel s') = H_1(M \parallel s')$

$M' \parallel s$ — $H_2(M' \parallel s) = H_2(M \parallel s')$ — $M \parallel s'$

Distinguisher on the XOR of two sponges



- Output on the 4 messages sums to zero:

$$\begin{aligned} & H(M \parallel s) \oplus H(M \parallel s') \\ \oplus\ & H(M' \parallel s) \oplus H(M' \parallel s') \end{aligned} = \begin{aligned} & H_1(M \parallel s) \oplus H_2(M \parallel s) \oplus H_1(M \parallel s') \oplus H_2(M \parallel s') \\ \oplus\ & H_1(M' \parallel s) \oplus H_2(M' \parallel s) \oplus H_1(M' \parallel s') \oplus H_2(M' \parallel s') \end{aligned} = 0$$

- Also true with arbitrary suffix: strong distinguisher:

$$\forall \sigma,\ H(M \parallel s \parallel \sigma) \oplus H(M' \parallel s \parallel \sigma) \oplus H(M \parallel s' \parallel \sigma) \oplus H(M' \parallel s' \parallel \sigma) = 0$$

The multiple 4-sum problem

Definition (4-sum problem (with random functions) [Wagner, Crypto'02])

Given $f : \{0,1\}^* \to \{0,1\}^n$,
Find distinct (x_1, x_2, x_3, x_4) s.t. $f(x_1) \oplus f(x_2) \oplus f(x_3) \oplus f(x_4) = 0$

- Generic complexity: $\approx 2^{n/4}$

Definition (multiple 4-sum problem)

Given $f : \{0,1\}^* \to \{0,1\}^n$, $\phi_i : \{0,1\}^* \to \{0,1\}^*$, $i \leq m$ (some technical restriction),
Find distinct (x_1, x_2, x_3, x_4) s.t. $\forall i < m,\ f(\phi_i(x_1)) \oplus f(\phi_i(x_2)) \oplus f(\phi_i(x_3)) \oplus f(\phi_i(x_4)) = 0$

- Generic complexity: $\gtrsim 2^{nm/52}$

- ϕ_i are message expansion function: expand quartet (x_1, x_2, x_3, x_4) into m related quartets
- Finding m related 4-sums on n bits is hard if n or m is large

Generic attacks against sponge combiners

- Consider large n, 2^{nd}-preimage rather than preimage $\implies$ ignore squeezing

Concatenation combiner

- $H(M) = H_1(M) \parallel H_2(M)$
- Generic security: attacks / proofs
  - Collisions: $2^{b/2}$ / $2^{c/2}$
  - 2^{nd}-preimages: $2^{b/2}$ / $2^{c/2}$
  - Indifferentiability: $2^{c/2}$ / $2^{c/2}$

XOR combiner

- $H(M) = H_1(M) \oplus H_2(M)$
- Generic security: attacks / proofs
  - Collisions: $2^{b/2}$ / $2^{c/2}$
  - 2^{nd} preimages: $2^{b/2}$ / $2^{c/2}$
  - Indifferentiability: $2^{c/2}$ / $2^{c/2}$

- Distinguisher on the XOR of two sponges with complexity $\tilde{O}(2^{c/2})$
- Tight indistinguishability of the XOR of two sponge: $2^{c/2}$
- GAPS for collision and preimage security

Outline

Merkle-Damgård Combiners

- Multicollisions
- Preimage attack on the XOR combiner

Sponge Combiners

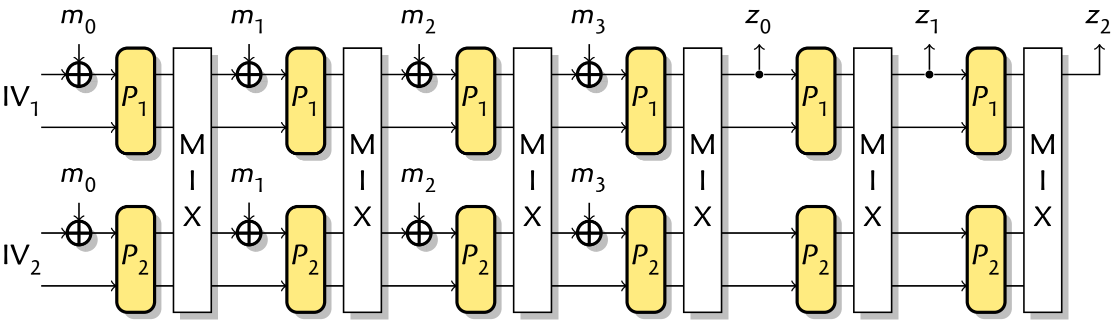
- Multicollisions
- New 4-sum distinguisher

The Double Sponge

- New 4-sum distinguisher

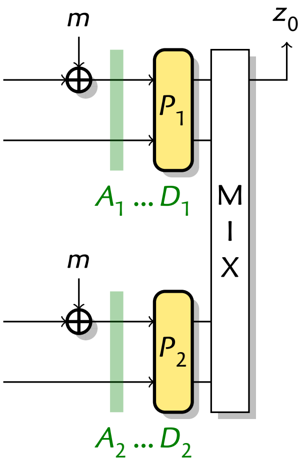
The double sponge construction

[Lefevre & Mennink, ToSC'24]



- $2b$-bit state, 2 permutations $P_1, P_2 : \{0,1\}^b \to \{0,1\}^b$
  - Linear operation MIX to mix both states
  - Notation: State after absorption: $(S_1(m_0 \parallel m_1), S_2(m_0 \parallel m_1))$
- Security beyond the birthday bound
  - Indifferentiability proof up to $2^{2b/3}$ queries
  - Generic attack with complexity $2^{c+r/2}$ (state collision)
  - Simulator-specific attack with complexity $2^{2c/3+r/3}$

4-sum for the double sponge (I)

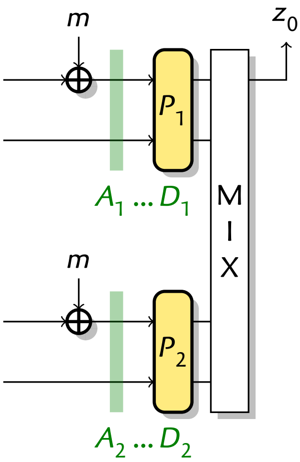


- Consider 4 states A, B, C, D after final message absorption
- Assume pairwise collisions of half-states:

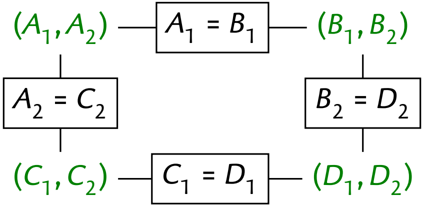
$$
\begin{array}{ccccc}
(A_1, A_2) & - & \boxed{A_1 = B_1} & - & (B_1, B_2) \\
| & & & & | \\
\boxed{A_2 = C_2} & & & & \boxed{B_2 = D_2} \\
| & & & & | \\
(C_1, C_2) & - & \boxed{C_1 = D_1} & - & (D_1, D_2)
\end{array}
$$

- Pairwise collisions preserved by P_i
- In particular, states after P_i sum to zero
- Sum is preserved by linear operation MIX
- Outputs z_0 sum to zero

4-sum for the double sponge (I)

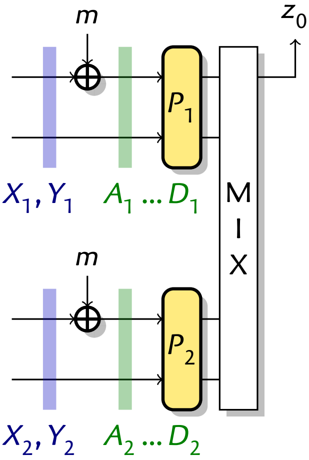


- Consider 4 states A, B, C, D after final message absorption
- Assume pairwise collisions of half-states:



- Pairwise collisions preserved by P_i
- In particular, states after P_i sum to zero
- Sum is preserved by linear operation MIX
- Outputs z_0 sum to zero

4-sum for the double sponge (II)



- 2 prefixes M, M'; states $X = S(M)$, $Y = S(M')$
- 4 messages $M \parallel m_A, M' \parallel m_B, M' \parallel m_C, M \parallel m_D$; corresponding states after last message XOR:

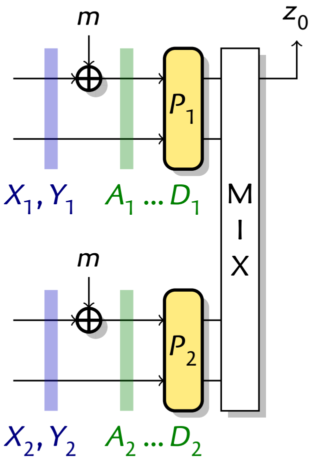
$$A_i = X_i \oplus (m_A \parallel 0^c) \qquad D_i = X_i \oplus (m_D \parallel 0^c)$$
$$B_i = Y_i \oplus (m_B \parallel 0^c) \qquad C_i = Y_i \oplus (m_C \parallel 0^c)$$

- Goal: pairwise collisions:

$$\begin{cases} A_1 = B_1 & A_2 = C_2 \\ C_1 = D_1 & B_2 = D_2 \end{cases}$$
$$\iff \begin{cases} X_1 \oplus Y_1 = (m_A \oplus m_B) \parallel 0^c & X_2 \oplus Y_2 = (m_A \oplus m_C) \parallel 0^c \\ X_1 \oplus Y_1 = (m_C \oplus m_D) \parallel 0^c & X_2 \oplus Y_2 = (m_B \oplus m_D) \parallel 0^c \end{cases}$$

- 2^r solutions if $\mathcal{C}(X_1) = \mathcal{C}(Y_1)$ and $\mathcal{C}(X_2) = \mathcal{C}(Y_2)$

4-sum for the double sponge: summary



1. Find messages (M, M') s.t. $X = S(M)$ and $Y = S(M')$ satisfy

$$\mathcal{C}(X_1) = \mathcal{C}(Y_1) \quad \text{and} \quad \mathcal{C}(X_2) = \mathcal{C}(Y_2)$$

2. Solve linear system to find 2^r solutions

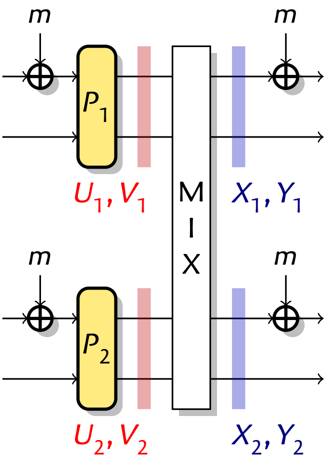
$$m_A = \mathcal{R}(Y_1 \oplus X_2) \oplus i \qquad m_B = \mathcal{R}(X_2 \oplus X_1) \oplus i$$
$$m_C = \mathcal{R}(Y_2 \oplus Y_1) \oplus i \qquad m_D = \mathcal{R}(Y_2 \oplus X_1) \oplus i$$

3. Each solution defines a 4-sum over r bits:

$$H(M \parallel m_A) \oplus H(M' \parallel m_B) \oplus H(M' \parallel m_C) \oplus H(M \parallel m_D) = 0$$

- Multiple 4-sum unlikely with random oracle
- Distinguisher with complexity $\mathcal{O}(2^c)$

Improvement with low-diffusion MIX



Goal

Find messages (M, M') s.t. $X = S(M)$ and $Y = S(M')$ satisfy

$$\mathcal{C}(X_1) = \mathcal{C}(Y_1) \quad \text{and} \quad \mathcal{C}(X_2) = \mathcal{C}(Y_2)$$

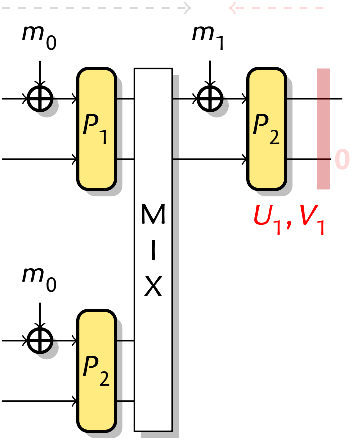
- MIX does not mix rate and capacity parts of state

$$\text{MIX} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

- Sufficient condition on $U = P^{-1}(X)$ and $V = P^{-1}(Y)$

$$\mathcal{C}(U_1) = \mathcal{C}(V_1) \qquad \mathcal{C}(U_2) = \mathcal{C}(V_2)$$
$$U_1[b-1] = V_1[b-1] \qquad U_2[b-1] = V_2[b-1]$$

Meet-in-the-middle with with low-diffusion MIX



Goal

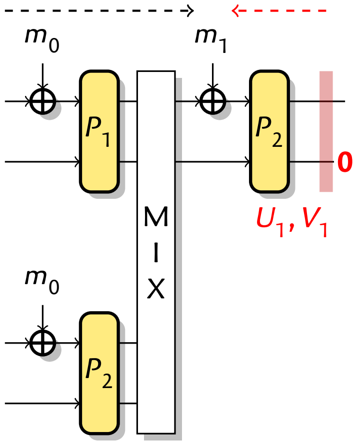
Find (M, M') s.t. $U = \mathrm{MIX}^{-1}(S(M))$ and $V = \mathrm{MIX}^{-1}(S(M'))$ satisfy

$$\mathcal{C}(U_1) = \mathcal{C}(V_1) \qquad \mathcal{C}(U_2) = \mathcal{C}(V_2)$$
$$U_1[b-1] = V_1[b-1] \qquad U_2[b-1] = V_2[b-1]$$

1. Generate $2^{3c/4}$ messages m_0; compute $S_1(m_0)$
2. Generate $2^{3c/4}$ states U_1 with $\mathcal{C}(U_1) = 0$; compute $P_!^{-1}(U_1)$
3. Find $2^{c/2}$ matches on the capacity
   Deduce $2^{c/2}$ messages M_i with $\mathcal{C}(\mathrm{MIX}^{-1}(S(M_i)) = 0)$
4. With high probably, one pair (M_i, M_j) satisfies remaining $c + 2$-bit condition

- Complexity $\mathcal{O}(2^{3c/4})$ if $r \geq 3c/4$

Meet-in-the-middle with with low-diffusion MIX



Goal

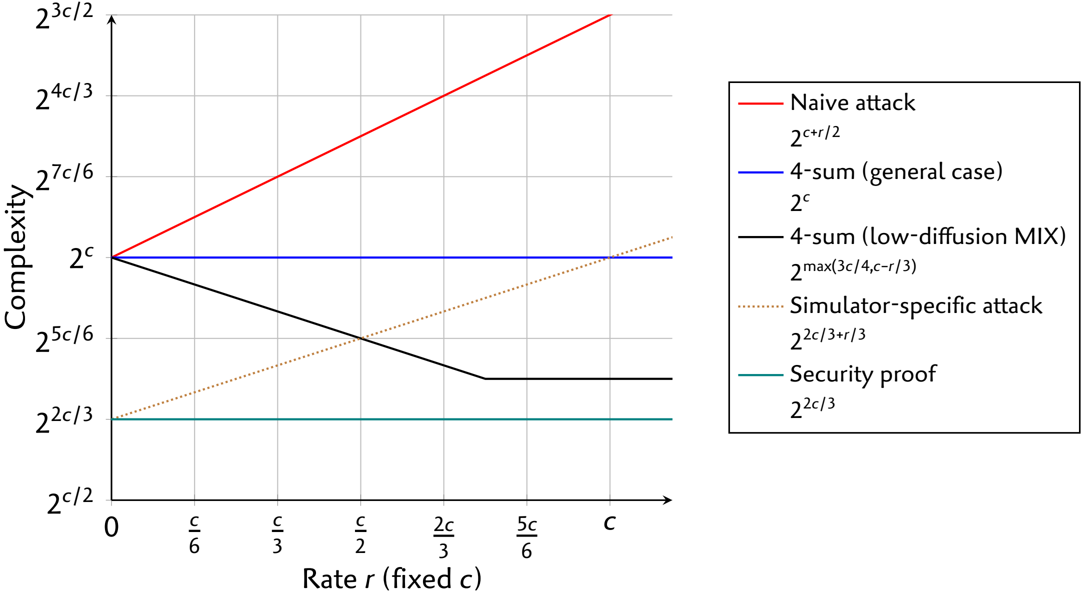
Find (M, M') s.t. $U = \mathrm{MIX}^{-1}(S(M))$ and $V = \mathrm{MIX}^{-1}(S(M'))$ satisfy

$$\mathcal{C}(U_1) = \mathcal{C}(V_1) \qquad \mathcal{C}(U_2) = \mathcal{C}(V_2)$$
$$U_1[b-1] = V_1[b-1] \qquad U_2[b-1] = V_2[b-1]$$

1. Generate $2^{3c/4}$ messages m_0; compute $S_1(m_0)$
2. Generate $2^{3c/4}$ states U_1 with $\mathcal{C}(U_1) = 0$; compute $P_!^{-1}(U_1)$
3. Find $2^{c/2}$ matches on the capacity
   Deduce $2^{c/2}$ messages M_i with $\mathcal{C}(\mathrm{MIX}^{-1}(S(M_i)) = 0)$
4. With high probably, one pair (M_i, M_j) satisfies
   remaining $c + 2$-bit condition

- Complexity $\mathcal{O}(2^{3c/4})$ if $r \geq 3c/4$

Double sponge security



Conclusion

- New distinguishers based on multiple 4-sums
  - Distinguisher on the XOR of 2 sponges with $\tilde{\mathcal{O}}(2^{c/2})$ operations
  - Distinguisher on the double sponge
    - $\mathcal{O}(2^{3c/4})$ operations if $r \geq 3c/4$
    - $\mathcal{O}(2^{c-r/3})$ operations if $r \leq 3c/4$
- Indifferentiability does not increase with rate
- Combiners don't improve indifferentiability bound (sponge and Merkle-Damgård)
  - Merkle-Damgård-XOR has less preimage security than Merkle-Damgård
- Still significant GAPS
  - Double sponge security
  - MD-XOR preimage, sponge-combiner preimage, sponge-combiner collision