Generic MitM Attack Frameworks on Sponge Constructions

Xiaoyang Dong [1] Boxin Zhao[2(✉)] Lingyue Qin[1] **Qingliang Hou**[3]
Shun Zhang[4] Xiaoyun Wang[1,3]

[1]Tsinghua University, Beijing, China

[2]Zhongguancun Laboratory, Beijing, P.R.China

[3]Shandong University, Qingdao, China

[4]PLA Strategic Support Force Information Engineering University, Zhengzhou, China

CRYPTO 2024 / August 18 - 22, 2024

Outline

Hash Function

Hash Function

A cryptographic hash function H maps a message M of arbitrary length into a short fixed-length h-bit target T.

Security Properties

- Preimage resistance: given T, find x such that $H(x) = T$ by querying at least 2^h H.
- Second preimage resistance: given x, find $x' \neq x$ such that $H(x) = H(x')$ by querying at least 2^h H.
- Collision resistance: find $x \neq x'$, such that $H(x) = H(x')$ by querying at least $2^{h/2}$ H.

Application

Signatures $\left(Sign_{key}(H(m))\right)$, Block Chain, Integrity $(H(m))$, MAC $(H(key, m))$, $\cdots$

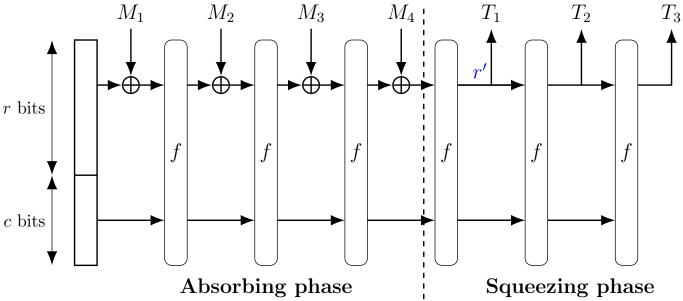
Secure Hash Algorithm-3 (SHA-3)

In 2004-2005, several cryptographic hash algorithms were successfully attacked, like MD5 and SHA-1. Hence, NIST held the SHA-3 competition in 2007.

Timeline

- 2008/10: 64 algorithms were submitted, and 51 algorithms were selected as the first-round candidates.
- 2009/07: 14 algorithms were selected as the second-round candidates.
- 2010/12: 5 third-round candidates: BLAKE, Grøstl, JH, Keccak and Skein.
- 2012/10: Keccak was selected as the winner.
- 2015/08: Keccak was standardized as SHA-3.

The sponge construction of SHA-3

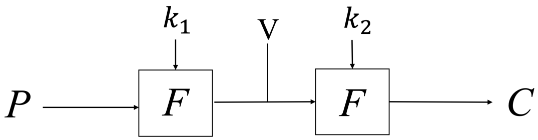


- b-bit Keccak-f permutation, with r-bit outer part (rate) and c-bit inner part (capacity).
- **Absorbing phase**: Given message is padded and divided into several r-bit blocks, i.e., M_i. Each M_i is XOR-ed into the outer part.
- **Squeezing phase**: Output h-bit digest $T_1 \| T_2 \| \ldots$, $h = 224, 256, 384, 512$.

Outline

Meet-in-the-Middle (MitM) Attack

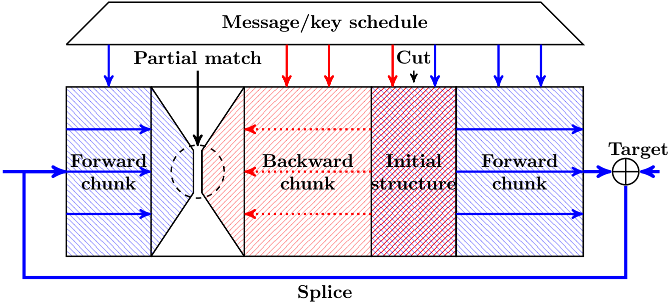
- MitM was first introduced by Diffie and Hellman in 1977 to attack Double-DES.
- Example: $C = E_K(P) = F_{k_2}(F_{k_1}(P))$, $K = k_1 \| k_2$.
  - Neutral sets: k_1 and k_2 are independent of each other.
  - Match: $F_{k_1}(P)$ and $F^{-1}_{k_2}(C)$.
- Time complexity: $2^{|k_1|+|k_2|} \rightarrow 2^{|k_1|+|k_2|-n}$.



- Enhanced techniques: splice-and-cut, initial structure, automated tools, $\cdots$.
- Application to MD constructions: MD4, MD5, SHA-1, Whirpool, AES-MMO, Simpira-DM, $\cdots$.

Splice-and-Cut MitM Attack Framework on MD Hash Functions

- At SAC 2008, Aoki and Sasaki proposed the splice-and-cut technique [AS08].



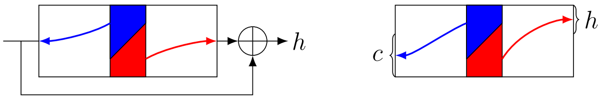
- Splice-and-Cut
- Initial Structure
- Partial Matching

1. For $2^{d_{\mathcal{R}}}$ values of ■, compute backward to the matching points and store them in L_1.
2. For $2^{d_{\mathcal{B}}}$ values of ■, compute forward to the matching points and store them in L_2.
3. Find m-bit partial match between L_1 and L_2.

- Time complexity: $Time = 2^{h-(d_{\mathcal{R}}+d_{\mathcal{B}})} \cdot (2^{\max(d_{\mathcal{R}},d_{\mathcal{B}})} + 2^{d_{\mathcal{R}}+d_{\mathcal{B}}-m}) \simeq 2^{h-\min(d_{\mathcal{R}},d_{\mathcal{R}},m)}$

The Limitation of MitM Attack on Sponge Construction

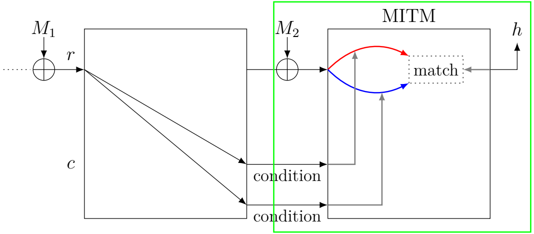
Open problem: How to mount an MitM attack on sponge constructions, like SHA-3, Ascon?



(a) MITM on DM (b) MITM on Sponge

- For DM hashing mode,
  - MitM attack starts at an n-bit internal state in the middle.
  - Two independent chunks meet at the matching point to filter the wrong internal states through the given h-bit target.
  - If $n > h$, at most 2^h internal states are searched to find the preimage.
- For sponge construction, if MitM attack starts at an internal state in the middle,
  - h-bit target in forward computation and c-bit inner part in backward computation should both be satisfied.
  - The search space is 2^{h+c} (preimage security bound usually $\leq 2^h$).

Conditional MitM Attack (EUROCRYPT 2023)



- Two independent neutral sets are divided from the starting state M_2.
- Some conditions determined by M_1 are set to reduce the diffusion of ■ and ■ bits.
  - For the non-linear operation $\chi : b_i = a_i \oplus (a_{i+1} \oplus 1) \cdot a_{i+2}$.
  - If (a_i, a_{i+1}) is (■, ■), then b_i depends on both ■ and ■.
  - If $a_{i+2} = 0$, then b_i only depends on a_i ■.
- Compute backward with the known h-bit target to derive an m-bit matching.

[QHD+23] Lingyue Qin, Jialiang Hua, Xiaoyang Dong, Hailun Yan, Xiaoyun Wang: Meet-in-the-Middle Preimage Attacks on Sponge-Based Hashing. EUROCRYPT 2023

Time Complexity of Conditional MitM Attack

- After finding one proper M_1 satisfying all bit conditions, an MitM episode is performed as follows:
  1. For each of $2^{d_{\mathcal{R}}}$ ■, compute forward to the matching point.
  2. For each of $2^{d_{\mathcal{B}}}$ ■, compute forward to the matching point.
  3. Given the h-bit target, compute backward to derive an m-bit matching point.
  4. Filter states.
- The complexity of one MitM episode is $2^{\max(d_{\mathcal{R}},d_{\mathcal{B}})} + 2^{d_{\mathcal{R}}+d_{\mathcal{B}}-m}$

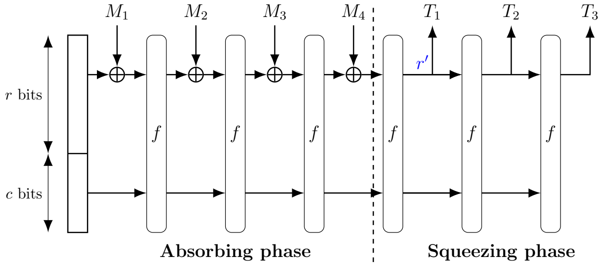
Time complexity

In order to find a h-bit target preimage, the episode should be repeated $2^{h-(d_{\mathcal{R}}+d_{\mathcal{B}})}$ times,

$$Time \simeq C + 2^{h-(d_{\mathcal{R}}+d_{\mathcal{B}})} \times \left(2^{\max(d_{\mathcal{R}},d_{\mathcal{B}})} + 2^{d_{\mathcal{R}}+d_{\mathcal{B}}-m}\right) = C + 2^{h-\min\{d_{\mathcal{R}},d_{\mathcal{B}},m\}}$$

where C is the time complexity to find M_1.

Tight Preimage Security Bound of Sponge Construction [LM22]



- For SHA-3, $h = c/2$, the general bound of preimage attack is 2^h.
- For other sponge constructions, like Ascon-Hash, the general bound was proved to be $\min\{\max\{2^{h-r'}, 2^{c/2}\}, 2^h\}$ [LM22].

How to Attack General Sponge Construction

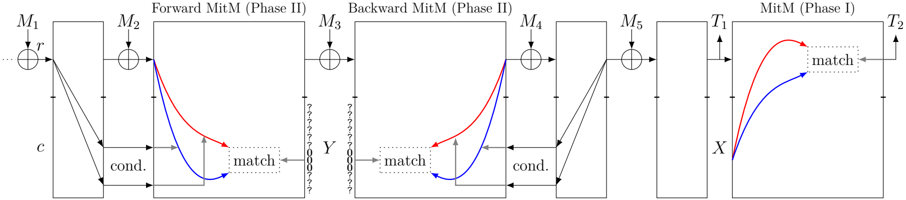
- E.g., SPHINCS$^+$-Haraka, with $b = 512, h = c = r = r' = 256$, then $\min\{\max\{2^{h-r'}, 2^{c/2}\}, 2^h\} = 2^{128}$.
- The time complexity of Qin's model is at least $2^{h-\min\{d_{\mathcal{R}}, d_{\mathcal{B}}, m\}}$.
- At least one MitM episode should be performed, the optimal complexity is achieved when $d_{\mathcal{R}} = d_{\mathcal{B}} = m = h/2$, i.e., $Time \simeq 2^{h/2}$.
- For SPHINCS$^+$-Haraka, $2^{h/2} = 2^{128}$. Qin's MitM model can not achieve preimage attack with complexity better than $2^{h/2}$.

Analysis

With $b = 512, h = c = r = r' = 256$, it leads to $h - r' < c/2 < h$. Hence, $2^{c/2}$ becomes the security bound.

Outline

Attack Framework



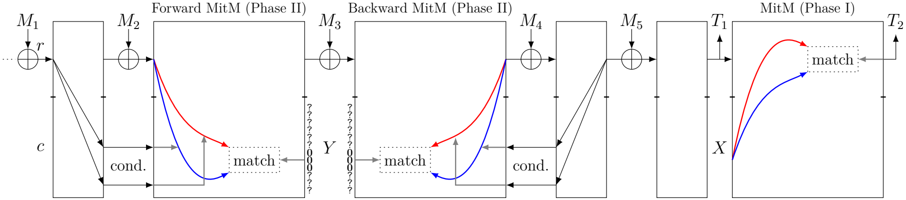
- Phase I: To find a capacity state X, such that $squeeze(T_1 \| X) = T_2 \| T_3 \| \cdots$. With Qin's MitM model [QHD+23], the time to find X can be reduced to

$$2^{|T|-|T_1|-\min(d^l_{\mathcal{R}},d^l_{\mathcal{B}},m^l)} = 2^{h-r'-\min(d^l_{\mathcal{R}},d^l_{\mathcal{B}},m^l)}, \quad (1)$$

- Phase II: To find an inner collision at Y, the time is $2^{c/2}$ trivially. Suppose t-bit of Y are fixed to be 0. Qin's MitM model can find $2^{\frac{c-t}{2}}$ $M_1 \| M_2$ for the forward path where the corresponding t-bit are 0, the time cost is:

$$\mathcal{C}_1 + 2^{\frac{c-t}{2}} \cdot 2^{t-\min(d^{L1}_{\mathcal{R}},d^{L1}_{\mathcal{B}},m^{L1})} = \mathcal{C}_1 + 2^{\frac{c}{2}+\frac{t}{2}-\min(d^{L1}_{\mathcal{R}},d^{L1}_{\mathcal{B}},m^{L1})} \quad (2)$$

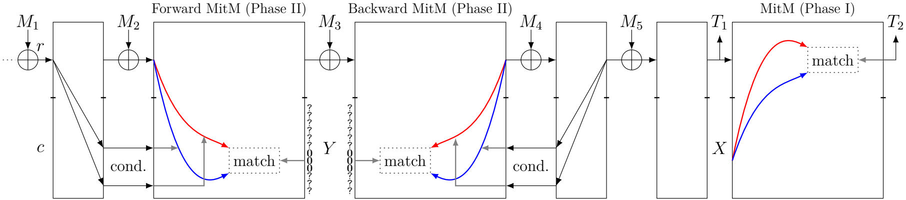
Attack Framework



The tight preimage security bound was proved to be $\min\{\max\{2^{h-r'}, 2^{c/2}\}, 2^h\}$ [LM22]. To beat the general bound, the following three cases are considered:

- Case I: If $h - r' < c/2 < h$, the general bound is $2^{c/2}$. Therefore, **Phase II** is only needed to derive better result than general bound. X in **Phase I** is a fixed constant. E.g., SPHINCS$^+$-`Haraka` with $h = 256$, $r = 256$, $c = 256$, $r' = 256$.

Attack Framework



- Case II: If $h - r' = c/2$, the general bound are $2^{h-r'}$ and $2^{c/2}$. Therefore, **Phase I** and **Phase II** are both needed. E.g., `Gimli-Hash`, `Xoodyak-Hash`, with $h = 256$, $c = 256$, $r = r' = 128$.
- Case III: If $h - r' > c/2$, the general bound is $2^{h-r'}$. Therefore, **Phase I** is only needed. The inner collision in **Phase II** can be performed in time of $2^{c/2}$. E.g., `Ascon-Hash`, `PHOTON`, `SPONGENT` and ACE-$\mathcal{H}$-256.

Description of SPHINCS$^+$-Haraka

- SPHINCS$^+$ is one of the selected Post-Quantum Digital Signature by NIST.
- SPHINCS$^+$-Haraka is instantiated with a sponge-based hashing based on the 512-bit permutation of Haraka v2.
- The 512-bit internal state is the concatenation of 4 AES states.

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

X_0

16	20	24	28
17	21	25	29
18	22	26	30
19	23	27	31

X_1

32	36	40	44
33	37	41	45
34	38	42	46
35	39	43	47

X_2

48	52	56	60
49	53	57	61
50	54	58	62
51	55	59	63

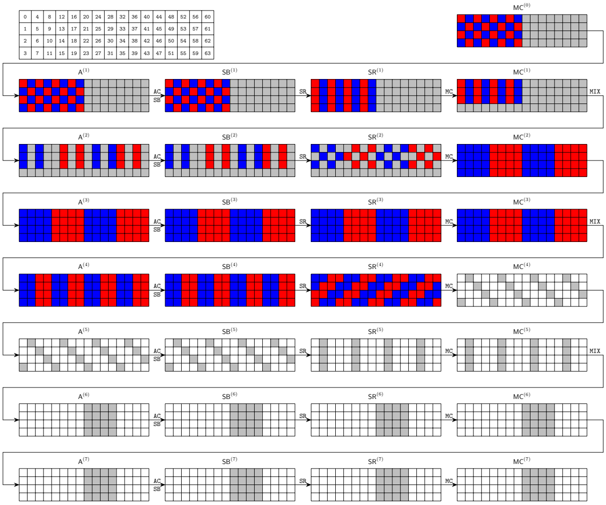
X_3

Two AES rounds are applied individually in each round (total 5 rounds), followed by an MIX operation:

$$0, \cdots, 15 \rightarrow (3, 11, 7, 15), (8, 0, 12, 4), (9, 1, 13, 5), (2, 10, 6, 14)$$

- Our target is the 4-round SPHINCS$^+$-Haraka without the last MIX operation.

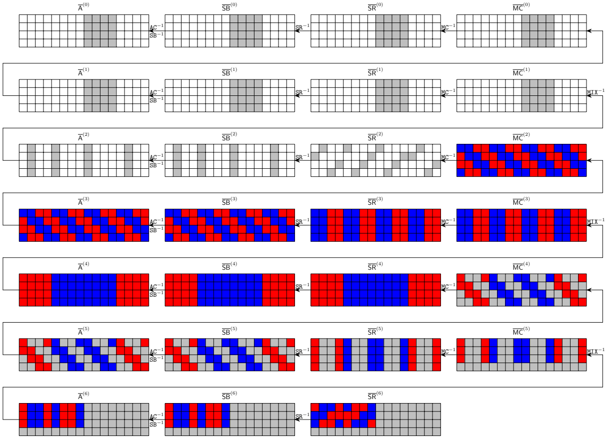
Preimage Attack on 4-round SPHINCS$^+$-Haraka



(a) Forward MitM

1. 16 bytes $\mathrm{MC}^{(7)}[32-47]$ are fixed to be 0 as matching points.
2. Starting state:
   $\mathrm{MC}^{(1)} = \mathtt{AES}(\mathtt{AES}(\mathrm{A}^{(0)}))$.
   $\mathrm{MC}^{(1)}[3,7,11,15,19,23,27,31] \leftarrow 0$,
   $\mathrm{MC}^{(1)}[32\text{-}63]$ is determined by M_1.
3. $d_{\mathcal{R}} = d_{\mathcal{B}} = 12,\ m = 16$.
4. With time of 2^{96}, $2^{96+96-128} = 2^{64}$ M_2 are stored in L_1 indexed by $\mathrm{MC}^{(7)}[48-63]$.

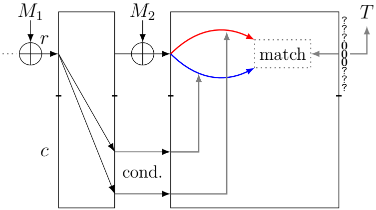
Preimage Attack on 4-round SPHINCS$^+$-Haraka



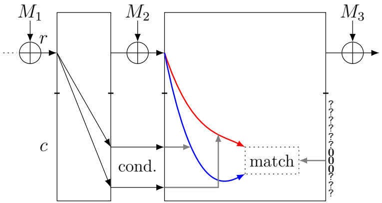
(b) Backward MitM

1. 16 bytes $\overline{A}^{(0)}[32-47]$ are fixed to be 0 as matching points.
2. Starting state: $\overline{SR}^{(6)}$. $\overline{SR}^{(6)}[3,7,11,15,19,23,27,31] \leftarrow 0$, $\overline{SR}^{(6)}$[32-63] is determined by M_5.
3. $d_{\mathcal{R}} = d_{\mathcal{B}} = 12,\ m = 16$.
4. With time of 2^{96}, $2^{96+96-128} = 2^{64}$ M_4 are stored in L_2 indexed by $\overline{A}^{(0)}[48-63]$.
5. Find a collision between L_1 and L_2.
6. $\overline{A}^{(0)}[0-31] = \text{MC}^{(7)}[0-31]$ can be modified by free M_3.

Outline



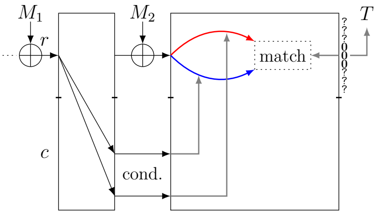
Collision Framework I (h-bit target collision)



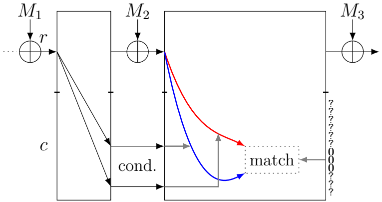
Collision Framework II (c-bit inner collision)

- Collision Framework I: The attack procedure is to find $2^{(h-t)/2}$ messages leading to the same t-bit pre-fixed constants. Time complexity: $\mathcal{C}_I \cdot 2^{(h-t)/2} < 2^{h/2}$.
- Collision Framework II: The attack procedure is to find $2^{(c-t)/2}$ messages leading to the same t-bit pre-fixed constants. The r-bit outer part can be modified by free message. Time complexity: $\mathcal{C}_{II} \cdot 2^{(c-t)/2} < 2^{c/2}$.

Attack Framework



Collision Framework I (h-bit target collision)



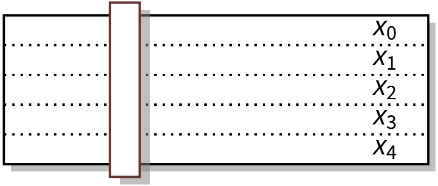
Collision Framework II (c-bit inner collision)

General bound of sponge construction: $\min\{2^{c/2}, 2^{h/2}\}$

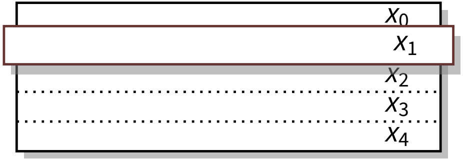
- If $h > c$, the **Collision Framework II** is applied. E.g., `Ascon-XOF`, `Xoodyak-XOF`.
- If $h = c$, the **Collision Framework I** or **II** is applied. E.g., `Ascon-Hash`, `Xoodyak-Hash`.
- If $h < c$, the **Collision Framework I** is applied. E.g., XOF with $h < c$.

Application to Ascon-Hash with Collision Framework II

- Winner of the NIST Lightweight Cryptography Project.
- Description of Ascon permutation: $p_L \circ p_S \circ p_C$



p_S: each column is updated with 5-bit S-box



p_C: each row is diffused with linear functions

- Parameters for Ascon-Hash: $b = 320, h = c = 256, r = r' = 64$.
- x_0 is the r-bit outer part.
- Since $h = c$, **Collision Framework II** can be applied.

Application to Ascon-Hash with Collision Framework II

- p_S applies the 5-bit Ascon S-Box column-wise as $(b_0, b_1, b_2, b_3, b_4) \leftarrow S(a_0, a_1, a_2, a_3, a_4)$. The algebraic normal form (ANF) of the Sbox is as follows:

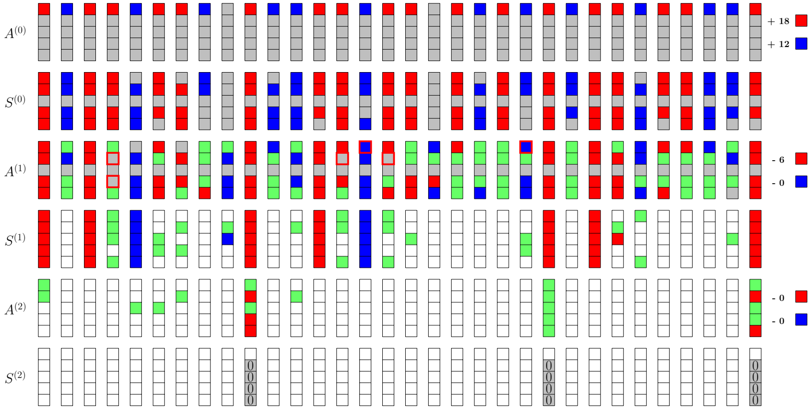
$$\begin{cases} b_0 = a_4a_1 + a_3 + a_2a_1 + a_2 + a_1a_0 + a_1 + a_0 \\ b_1 = a_4 + a_3a_2 + a_3a_1 + a_3 + a_2a_1 + a_2 + a_1 + a_0 \\ b_2 = a_4a_3 + a_4 + a_2 + a_1 + 1 \\ b_3 = a_4a_0 + a_4 + a_3a_0 + a_3 + a_2 + a_1 + a_0 \\ b_4 = a_4a_1 + a_4 + a_3 + a_1a_0 + a_1 \end{cases} \quad (3)$$

Observation 1 (Matching Strategy for Collision Framework II)

If $b_1 = b_2 = b_3 = b_4 = 0$, then $a_0 = 1,\ a_1 \oplus a_2 = 1,\ a_3 = 0,\ a_4 = 0$ can be derived. Therefore, 4 matching equations can be immediately obtained if there are no unknown $\square$ bit in $(a_0, a_1, a_2, a_3, a_4)$.

Application to Ascon-Hash with Collision Framework II

- Collision attack on 3-round Ascon-Hash. The attack parameters are: $d_{\mathcal{R}} = d_{\mathcal{B}} = 24$, $m = t = 24$. The time cost is $2^{\frac{c}{2} - \min\{d_{\mathcal{R}} - \frac{t}{2}, d_{\mathcal{B}} - \frac{t}{2}, m - \frac{t}{2}, \frac{t}{2}\}} = 2^{128-12} = 2^{116}$.



Outline

Summary of applications to preimage and collision attacks

Target	Attacks	Methods	Rounds	Time	Memory	Claim	Generic	Ref.
Ascon-Hash	Collision	Diff. Diff. MitM MitM	2/12 2/12 3/12 4/12	2^{125} 2^{103} $2^{116.74}$ $2^{124.85}$	- - 2^{116} 2^{124}	2^{128}	2^{128}	[ZDW19] [GPT21] ours ours
SPHINCS$^+$-Haraka	Preimage	MitM MitM	3.5/5 4/5	$2^{64.6}$ Q 2^{98}	- 2^{96}	- 2^{128}	$2^{85.3}$ Q 2^{128}	[SS22] ours
PHOTON-80/20/16	Preimage	MitM	4.5/12	2^{60}	2^{24}	2^{64}	2^{64}	ours
ACE-$\mathcal{H}$-256	Preimage	MitM	9/16	2^{160}	2^{128}	2^{192}	2^{192}	ours
Subterranean 2.0	Preimage	MitM	Full	2^{160}	2^{100}	2^{112}	2^{224}	ours
Xoodyak-XOF	Preimage	Neural MitM MitM	1/12 3/12 3/12	- $2^{125.06}$ $2^{121.77}$	- 2^{97} 2^{118}	2^{128}	2^{128}	[LLL$^+$21] [QHD$^+$23] ours
Xoodyak-Hash	Collision	MitM	3/12	$2^{125.23}$	2^{124}	2^{128}	2^{128}	ours

Reference I

[AS08] Kazumaro Aoki and Yu Sasaki.
Preimage attacks on one-block MD4, 63-step MD5 and more.
In SAC 2008, volume 5381, pages 103–119. Springer, 2008.

[GPT21] David Gérault, Thomas Peyrin, and Quan Quan Tan.
Exploring differential-based distinguishers and forgeries for ASCON.
IACR Trans. Symmetric Cryptol., 2021(3):102–136, 2021.

[LLL+21] Guozhen Liu, Jingwen Lu, Huina Li, Peng Tang, and Weidong Qiu.
Preimage attacks against lightweight scheme xoodyak based on deep learning.
In Future of Information and Communication Conference, pages 637–648. Springer, 2021.

[LM22] Charlotte Lefevre and Bart Mennink.
Tight preimage resistance of the sponge construction.
In Yevgeniy Dodis and Thomas Shrimpton, editors, Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV, volume 13510 of Lecture Notes in Computer Science, pages 185–204. Springer, 2022.

Reference II

[QHD+23] Lingyue Qin, Jialiang Hua, Xiaoyang Dong, Hailun Yan, and Xiaoyun Wang.
Meet-in-the-middle preimage attacks on sponge-based hashing.
In Carmit Hazay and Martijn Stam, editors, Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part IV, volume 14007 of Lecture Notes in Computer Science, pages 158–188. Springer, 2023.

[SS22] André Schrottenloher and Marc Stevens.
Simplified MITM modeling for permutations: New (quantum) attacks.
In CRYPTO 2022, Proceedings, Part III, volume 13509, pages 717–747. Springer, 2022.

[ZDW19] Rui Zong, Xiaoyang Dong, and Xiaoyun Wang.
Collision attacks on round-reduced gimli-hash/ascon-xof/ascon-hash.
Cryptology ePrint Archive, Paper 2019/1115, 2019.