Meet-in-the-Middle Preimage Attacks on Sponge-based Hashing

Lingyue Qin[1] Jialiang Hua[1] Xiaoyang Dong[1(✉)] Hailun Yan[2]
Xiaoyun Wang[1]

[1]Tsinghua University

[2]University of Chinese Academy of Sciences

April 25, 2023

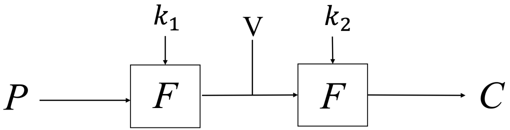
Outline

Meet-in-the-Middle (MITM) Attack

- Proposed by Diffie and Hellman in 1977 [DH77]
- A generic technique for cryptanalysis of symmetric-key primitives
- An efficient exhaustive search way based on the birthday attack

Example: Double Encryption

- $C = E_K(P) = F_{K_2}(F_{K_1}(P)),\ K = K_1 \| K_2$
- The time complexity of a naive exhaust search: $2^{|K_1|+|K_2|}$
- The time complexity of MITM attach: $2^{|K_1|+|K_2|-n}$
    - Meet in the middle: $F_{K_1}(P)\ ?=\ F^{-1}_{K_2}(C)$



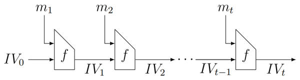
Meet-in-the-Middle (MITM) Attack

- It has been widely applied on block ciphers and hash functions.
- Various techniques improve the framework of MitM attack
  - internal state guessing, splice-and-cut, initial structure, bicliques, 3-subset MitM, indirect-partial matching, sieve-in-the-middle, match-box, dissection, differential-aided MitM, nonlinear constrained neutral words...
- There are also some MILP-based automatic tools.
  - Sasaki at IWSEC 2018, Bao et al. at EUROCRYPT 2021 and CRYPTO 2022, Dong et al. at CRYPTO 2021...
- The MITM attack and its variants have broken:
  - MD4, GOST, MD5, HAVAL, GEA-1/2 ...

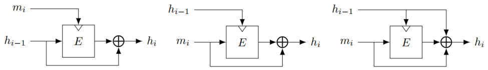
MITM Preimage Attacks on Hash Functions

Most MITM attacks targeted on Merkle-Damgård hash functions

- The feed-forward mechanism → A closed computation path



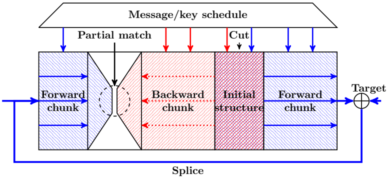
Merkle-Damgård construction

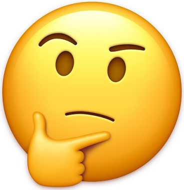


The compression functions f: block cipher + PGV modes

The Splice-and-Cut MITM attacks

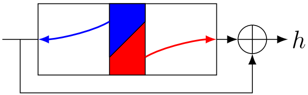
- The chunk separation.
- The neutral sets (■/■): the degree of freedom (DoF) for each chunk.
- The partial matching: the filtering ability (degree of matching, DoM).



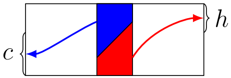


Does it work for Sponge-based Hashing?

MITM Attack on Sponge-based Hashing vs. MD Hashing



(a) MITM on DM

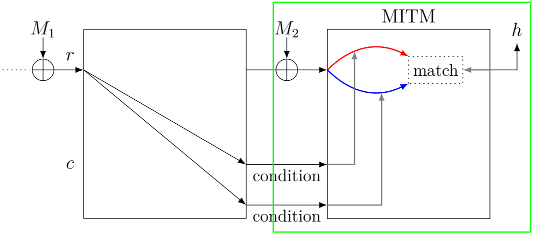


(b) MITM on Sponge

- The complexity of exhaustive search: 2^h
- For MITM attack on the MD hashing
  - The search space $< 2^h$
- For MITM attack on the sponge-based hashing
  - The search space is $2^{(h+c)}$ to meet not only h but also c. Not a good idea!

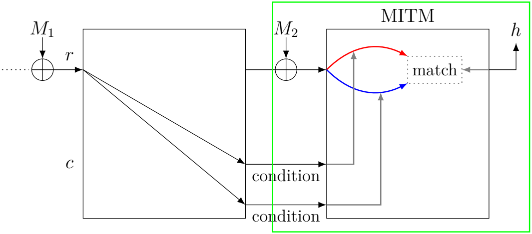
We need to search for a more compact space.

Framework of the MITM Attack on Sponge-based Hashing



- Start from the r-bit rate part and search for a h-bit subspace (if $r > h$)
- Only forward computations are involved
- Specify the configurations: the two neutral sets of the outer part, the two independent forward computation chunks, the matching points
- Partially solve the inverse of the permutation from the h-bit target
- Set conditions to control the characteristic propagation

The MITM Episode on Sponge-based Hashing

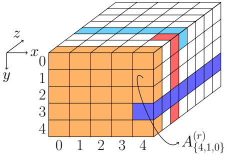


- For 2^{d_1} values of ■ neutral set, compute forward to the matching points;
- For 2^{d_2} values of ■ neutral set, compute forward to the matching points;
- Compute backward with the known h-bit target to the matching points to derive an m-bit matching and filter states.

Automatic MITM Attack Model on Sponge-based Hashing

Example: Keccak

- Keccak-f permutation: $A^{(r)} \xrightarrow{\theta} \theta^{(r)} \xrightarrow{\rho} \rho^{(r)} \xrightarrow{\pi} \pi^{(r)} \xrightarrow{\chi} \chi^{(r)} \xrightarrow{\iota} A^{(r+1)}$



MILP model: The Objective Function + Constraints

- Modelling the Starting State
- Modelling the Attribute Propagation: XOR (θ) and S-box (χ)
- Modelling the Matching Phase
- Auxiliary Techniques: Conditions + Linear Structure + CP-Kernel

Encoding Scheme

3-bit encoding scheme: $(\omega_0, \omega_1, \omega_2)$

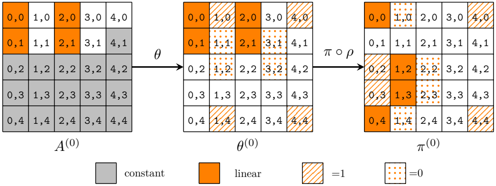
- $\textcolor{gray}{\blacksquare}$: $(1, 1, 1)$, global constant bits
- $\textcolor{red}{\blacksquare}$: $(0, 1, 1)$, depend on $\textcolor{gray}{\blacksquare}$ and $\textcolor{red}{\blacksquare}$
- $\textcolor{blue}{\blacksquare}$: $(1, 1, 0)$, depend on $\textcolor{gray}{\blacksquare}$ and $\textcolor{blue}{\blacksquare}$
- $\textcolor{green}{\blacksquare}$: $(0, 1, 0)$, depend on $\textcolor{gray}{\blacksquare}/\textcolor{red}{\blacksquare}/\textcolor{blue}{\blacksquare}$, but the expression does not contain the product of $\textcolor{red}{\blacksquare}$ and $\textcolor{blue}{\blacksquare}$
- $\square$: $(0, 0, 0)$, depend on the product of $\textcolor{blue}{\blacksquare}$ and $\textcolor{red}{\blacksquare}$

Remark

- In previous MILP-aid MITM models: ($\square$, $\textcolor{gray}{\blacksquare}$, $\textcolor{blue}{\blacksquare}$, $\textcolor{red}{\blacksquare}$)
- In our model: ($\square$, $\textcolor{gray}{\blacksquare}$, $\textcolor{blue}{\blacksquare}$, $\textcolor{red}{\blacksquare}$, $\textcolor{green}{\blacksquare}$)
- The addition of $\textcolor{blue}{\blacksquare}$ and $\textcolor{red}{\blacksquare}$ (not multiplied) can also be used

Modelling the Starting State with the Linear Structure

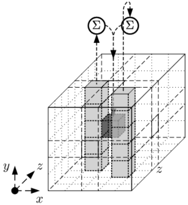
- The linear structure (Guo et al. ASIACRYPT 2016)



- Control the diffusion of θ operation
  - $A^{(0)}_{\{0,0,z\}}$ and $A^{(0)}_{\{0,1,z\}}$ should be the same color
  - $A^{(0)}_{\{0,0,z\}} \oplus A^{(0)}_{\{0,1,z\}}$ should be constant
- Add conditions to reduce the diffusion over χ
  - For the row $\pi^{(0)}_{\{*,0,z\}}$, set $\pi^{(0)}_{\{1,0,z\}} = 0$ and $\pi^{(0)}_{\{4,0,z\}} = 1$
- Model $A^{(1)}$ only considering the linear operation $\pi \circ \rho$ from $A^{(0)}$

Modelling the θ operation

$$\theta: \quad \theta^{(r)}_{\{x,y,z\}} = A^{(r)}_{\{x,y,z\}} \oplus \sum_{y'=0}^{4} (A^{(r)}_{\{x-1,y',z\}} \oplus A^{(r)}_{\{x+1,y',z-1\}})$$



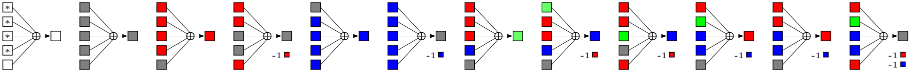
We depose the θ operation to three steps in our model:

$$C^{(r)}_{\{x,z\}} = A^{(r)}_{\{x,0,z\}} \oplus A^{(r)}_{\{x,1,z\}} \oplus A^{(r)}_{\{x,2,z\}} \oplus A^{(r)}_{\{x,3,z\}} \oplus A^{(r)}_{\{x,4,z\}},$$
$$D^{(r)}_{\{x,z\}} = C^{(r)}_{\{x-1,z\}} \oplus C^{(r)}_{\{x+1,z-1\}},$$
$$\theta^{(r)}_{\{x,y,z\}} = A^{(r)}_{\{x,y,z\}} \oplus D^{(r)}_{\{x,z\}}.$$

Modelling the θ operation (Cont.)

The rule of XOR with an arbitrary number of inputs

- XOR-RULE-1: If the inputs have (0,0,0) □ bit, the output is □.
- XOR-RULE-2: If the inputs are all (1,1,1) ■ bits, the output is ■.
- XOR-RULE-3: If the inputs have (1,1,0) ■ (≥ 1) and ■ (≥ 0) bits:
  - the output is ■ without consuming DoF, or ■ by consuming 1 DoF of ■.
- XOR-RULE-4: If the inputs have (0,1,1) ■ (≥ 1) and ■ (≥ 0) bits,
  - the output is ■ without consuming DoF, or ■ by consuming 1 DoF of ■.
- XOR-RULE-5: If the inputs have at least two kinds of ■, ■ and ■ bits:
  - the output can be ■ without consuming DoF.
  - the output can be ■ (or ■) by consuming one DoF of ■ (or ■).
  - the output can be ■ by consuming one DoF of ■ and one DoF of ■.



Modelling the θ operation (Cont.)

Constraints

- Define three 0-1 variables ν_i ($i \in \{0, 1, 2\}$), where $\nu_0 = 1$ if and only if all the ω_0's of the 5 input bits are 1, similar to the cases $i = 1, 2$.
- The above five rules can be represented by (ν_0, ν_1, ν_2):
  1. $(\nu_0, \nu_1, \nu_2) = (*, 0, *)$, `XOR-RULE-1` is applied.
  2. $(\nu_0, \nu_1, \nu_2) = (1, 1, 1)$, `XOR-RULE-2` is applied.
  3. $(\nu_0, \nu_1, \nu_2) = (1, 1, 0)$, `XOR-RULE-3` is applied.
  4. $(\nu_0, \nu_1, \nu_2) = (0, 1, 1)$, `XOR-RULE-4` is applied.
  5. $(\nu_0, \nu_1, \nu_2) = (0, 1, 0)$, `XOR-RULE-5` is applied.
- Define the output bit as $(\omega_0^O, \omega_1^O, \omega_2^O)$, the consumed DoF of ■ bits and ■ bits are $(\delta_{\mathcal{R}}, \delta_{\mathcal{B}})$:

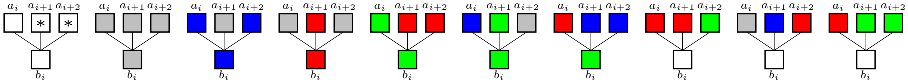
$$\begin{cases} \omega_0^O - \nu_0 \geq 0, \quad -\omega_0^O + \nu_1 \geq 0, \\ \omega_1^O - \nu_1 = 0, \\ \omega_2^O - \nu_2 \geq 0, \quad -\omega_2^O + \nu_1 \geq 0, \end{cases} \qquad \begin{cases} \delta_{\mathcal{R}} - \omega_0^O + \nu_0 = 0, \\ \delta_{\mathcal{B}} - \omega_2^O + \nu_2 = 0. \end{cases}$$

Modelling the χ operation

$\chi : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^n,\ b_i = a_i \oplus (a_{i+1} \oplus 1) \cdot a_{i+2},\ i = 0, 1, ..., 4$

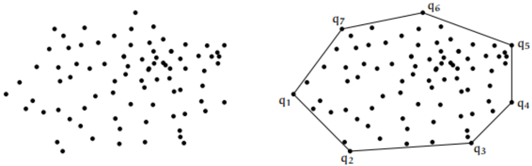
- If there are □ bits in (a_i, a_{i+1}, a_{i+2}), the output is □
- If there are all ■ bits, the output is ■
- If there are only 🟦 (≥ 1) and ■ (≥ 0) bits, the output will be 🟦
- If there are only 🟥 (≥ 1) and ■ (≥ 0) bits, the output will be 🟥
- If there are 🟩, or more than two kinds of 🟦, 🟥 and 🟩 bits in (a_i, a_{i+1}, a_{i+2}):
  - if a_{i+1} and a_{i+2} are all 🟦 (or 🟥), the output is 🟩
  - if a_{i+1} or a_{i+2} is ■, the output is 🟩
  - if a_{i+1} and a_{i+2} are of arbitrarily two kinds of 🟦, 🟥, 🟩, the output is □

Modelling the χ operation (Cont.)



Constraints

Derive linear inequalities by using the convex hull computation (Sun et al. ASIACRYPT 2014)



(a) Input. (b) Output.

Modelling the Matching Phase

Property of χ (Guo et al. ASIACRYPT 2016)

When there are three known consecutive output bits, two linear equations of the input bits can be constructed. Eg. Assuming that (b_0, b_1, b_2) are known, two linear equations on (a_0, a_1, a_2, a_3) are constructed as

$$b_0 = a_0 \oplus (b_1 \oplus 1) \cdot a_2,$$

$$b_1 = a_1 \oplus (b_2 \oplus 1) \cdot a_3.$$

For Keccak-512, we have:

$$A^{(r+1)}_{\{0,1,z\}} = \pi^{(r)}_{\{0,1,z\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot \pi^{(r)}_{\{2,1,z\}}$$
$$A^{(r+1)}_{\{1,1,z\}} = \pi^{(r)}_{\{1,1,z\}} \oplus (A^{(r+1)}_{\{2,1,z\}} \oplus 1) \cdot \pi^{(r)}_{\{3,1,z\}}$$

Modelling the Matching Phase (Cont.)

Apply the inverse of $\rho \circ \pi$, and add the same known $\theta^{(r)}_{\{x,x,z\}}$, we have

$$A^{(r+1)}_{\{0,1,z\}} \oplus \underline{\theta^{(r)}_{\{3,3,z-\gamma[3,0]\}}} \oplus \underline{(A^{(r+1)}_{\{1,1,z\}} \oplus 1)} \cdot \underline{\theta^{(r)}_{\{0,0,z-\gamma[0,2]\}}}$$
$$= \theta^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus \underline{\theta^{(r)}_{\{3,3,z-\gamma[3,0]\}}} \oplus \underline{(A^{(r+1)}_{\{1,1,z\}} \oplus 1)} \cdot (\theta^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus \underline{\theta^{(r)}_{\{0,0,z-\gamma[0,2]\}}})$$

$$A^{(r+1)}_{\{1,1,z\}} \oplus \underline{\theta^{(r)}_{\{4,4,z-\gamma[4,1]\}}} \oplus \underline{(A^{(r+1)}_{\{2,1,z\}} \oplus 1)} \cdot \underline{\theta^{(r)}_{\{1,1,z-\gamma[1,3]\}}}$$
$$= \theta^{(r)}_{\{4,1,z-\gamma[4,1]\}} \oplus \underline{\theta^{(r)}_{\{4,4,z-\gamma[4,1]\}}} \oplus \underline{(A^{(r+1)}_{\{2,1,z\}} \oplus 1)} \cdot (\theta^{(r)}_{\{1,3,z-\gamma[1,3]\}} \oplus \underline{\theta^{(r)}_{\{1,1,z-\gamma[1,3]\}}})$$

Modelling the Matching Phase (Cont.)

The CP-kernel Property of θ

$$\begin{aligned}
\theta^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus \theta^{(r)}_{\{3,3,z-\gamma[3,0]\}} &= A^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus A^{(r)}_{\{3,3,z-\gamma[3,0]\}},\\
\theta^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus \theta^{(r)}_{\{0,0,z-\gamma[0,2]\}} &= A^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus A^{(r)}_{\{0,0,z-\gamma[0,2]\}},\\
\theta^{(r)}_{\{4,1,z-\gamma[4,1]\}} \oplus \theta^{(r)}_{\{4,4,z-\gamma[4,1]\}} &= A^{(r)}_{\{4,1,z-\gamma[4,1]\}} \oplus A^{(r)}_{\{4,4,z-\gamma[4,1]\}},\\
\theta^{(r)}_{\{1,3,z-\gamma[1,3]\}} \oplus \theta^{(r)}_{\{1,1,z-\gamma[1,3]\}} &= A^{(r)}_{\{1,3,z-\gamma[1,3]\}} \oplus A^{(r)}_{\{1,1,z-\gamma[1,3]\}}.
\end{aligned}$$

Modelling the Matching Phase (Cont.)

With the CP-kernel property of θ, we have

$$A^{(r+1)}_{\{0,1,z\}} \oplus \theta^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot \theta^{(r)}_{\{0,0,z-\gamma[0,2]\}}$$
$$= \theta^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus \theta^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot (\theta^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus \theta^{(r)}_{\{0,0,z-\gamma[0,2]\}})$$

$$\downarrow$$

$$A^{(r+1)}_{\{0,1,z\}} \oplus \theta^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot \theta^{(r)}_{\{0,0,z-\gamma[0,2]\}}$$
$$= A^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus A^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot (A^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus A^{(r)}_{\{0,0,z-\gamma[0,2]\}})$$

Modelling the Matching Phase (Cont.)

Similarly,

$$A^{(r+1)}_{\{1,1,z\}} \oplus \theta^{(r)}_{\{4,4,z-\gamma[4,1]\}} \oplus (A^{(r+1)}_{\{2,1,z\}} \oplus 1) \cdot \theta^{(r)}_{\{1,1,z-\gamma[1,3]\}}$$
$$= \theta^{(r)}_{\{4,1,z-\gamma[4,1]\}} \oplus \theta^{(r)}_{\{4,4,z-\gamma[4,1]\}} \oplus (A^{(r+1)}_{\{2,1,z\}} \oplus 1) \cdot (\theta^{(r)}_{\{1,3,z-\gamma[1,3]\}} \oplus \theta^{(r)}_{\{1,1,z-\gamma[1,3]\}})$$

$$\downarrow$$

$$A^{(r+1)}_{\{1,1,z\}} \oplus \theta^{(r)}_{\{4,4,z-\gamma[4,1]\}} \oplus (A^{(r+1)}_{\{2,1,z\}} \oplus 1) \cdot \theta^{(r)}_{\{1,1,z-\gamma[1,3]\}}$$
$$= A^{(r)}_{\{4,1,z-\gamma[4,1]\}} \oplus A^{(r)}_{\{4,4,z-\gamma[4,1]\}} \oplus (A^{(r+1)}_{\{2,1,z\}} \oplus 1) \cdot (A^{(r)}_{\{1,3,z-\gamma[1,3]\}} \oplus A^{(r)}_{\{1,1,z-\gamma[1,3]\}})$$

Modelling the Matching Phase (Cont.)

Leaked linear relations of $A^{(r)}$ from the hash value $A^{(r+1)}$

$$A^{(r)}_{\{3,0,z-\gamma[3,0]\}} \oplus A^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot (A^{(r)}_{\{0,2,z-\gamma[0,2]\}} \oplus A^{(r)}_{\{0,0,z-\gamma[0,2]\}})$$
$$= A^{(r+1)}_{\{0,1,z\}} \oplus \theta^{(r)}_{\{3,3,z-\gamma[3,0]\}} \oplus (A^{(r+1)}_{\{1,1,z\}} \oplus 1) \cdot \theta^{(r)}_{\{0,0,z-\gamma[0,2]\}}$$

Conditions in Matching Points of Keccak

If four bits $(A^{(r)}_{\{3,0,z-\gamma[3,0]\}}, A^{(r)}_{\{3,3,z-\gamma[3,0]\}}, A^{(r)}_{\{0,2,z-\gamma[0,2]\}}, A^{(r)}_{\{0,0,z-\gamma[0,2]\}})$ in $A^{(r)}$ satisfy the following two conditions, there is a 1-bit filter:

(1) No $\square$ in $(A^{(r)}_{\{3,0,z-\gamma[3,0]\}}, A^{(r)}_{\{3,3,z-\gamma[3,0]\}}, A^{(r)}_{\{0,2,z-\gamma[0,2]\}}, A^{(r)}_{\{0,0,z-\gamma[0,2]\}})$.

(2) $(A^{(r)}_{\{3,0,z-\gamma[3,0]\}}, A^{(r)}_{\{3,3,z-\gamma[3,0]\}})$ is of $({\color{red}\blacksquare}, {\color{blue}\blacksquare})$, $({\color{red}\blacksquare}, {\color{green}\blacksquare})$, $({\color{blue}\blacksquare}, {\color{green}\blacksquare})$, $({\color{gray}\blacksquare}, {\color{green}\blacksquare})$, or $({\color{green}\blacksquare}, {\color{green}\blacksquare})$, or opposite order .

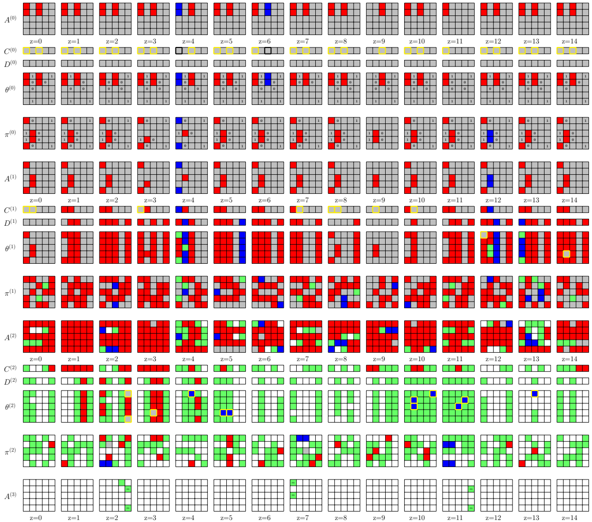
The Objective Function

- Maximize min$\{\mathrm{DoF}_{\mathcal{R}}, \mathrm{DoF}_{\mathcal{B}}, \mathrm{DoM}\}$ to find the optimal attacks
  - $\mathrm{DoF}_{\mathcal{R}} = \lambda_{\mathcal{R}} - l_{\mathcal{R}}$, $l_{\mathcal{R}}$ be the accumulated consumption of DoF of ■
  - $\mathrm{DoF}_{\mathcal{B}} = \lambda_{\mathcal{B}} - l_{\mathcal{B}}$, and $l_{\mathcal{B}}$ be the consumption of DoF of ■
  - $\mathrm{DoM} = \sum \delta_{\mathcal{M}}$

- Maximize v_{obj}

$$\{v_{obj} \leq \mathrm{DoF}_{\mathcal{R}}, v_{obj} \leq \mathrm{DoF}_{\mathcal{B}}, v_{obj} \leq \mathrm{DoM}\}.$$

The MITM preimage attack on 4-round Keccak-512



- ■, ■: DoF of ■ -1
- ■: DoF of ■ -1
- *m*: for matching

- DOF (■)=108
- DOF (■)=8
- DOF =8

The attack model is shown in Figure 10,16,17,18 in our paper.
Perform the attack with (M_1, M_2), the MitM eposide is placed at M_2.

Attack Results of Keccak-512

Target	Attacks	Methods	Rounds	Time	Memory	Ref.
Keccak-512	Preimage	Lin.Stru.	2	2^{384}	-	[GLS16]
		Lin.Stru.	2	2^{321}	-	[Raj19]
		Lin.Stru.	2	2^{270}	-	[LIMY21]
		Lin.Stru.	2	2^{252}	-	[HLY22]
		Lin.Stru.	3	2^{482}	-	[GLS16]
		Lin.Stru.	3	2^{475}	-	[Raj19]
		Lin.Stru.	3	2^{452}	-	[LIMY21]
		Lin.Stru.	3	2^{426}	-	[HLY22]
		Rotational	4	2^{506}	-	[MPS13]
		MitM	4	$2^{504.58}$	2^{108}	Ours
	Collision	Diff.	2	Practical	-	[NRM11]
		Diff.	3	Practical	-	[DDS13]

Attack Results of Xoodyak-XOF and Ascon-XOF

Target	Attacks	Methods	Rounds	Time	Memory	Ref.
Xoodyak-XOF	Preimage	Neural	1	-	-	[LLL+21]
		MitM	3	$2^{125.06}$	2^{97}	Ours
Ascon-XOF	Preimage	Cube-like	2	2^{103}	-	[DEMS21]
		MitM	3	$2^{120.58}$	2^{39}	Ours
		MitM	4	$2^{124.67}$	2^{54}	Ours
		Algebraic†	6	$2^{127.3}$	-	[DEMS21]
	Collision	Diff.	2	2^{103}	-	[GPT21]

Conclusion

- Since 1977, the birthday-paradox MitM attack has been widely applied to block ciphers or MD-based hash functions, this is the first attempt to apply it to sponge-based hash functions

- Dedicated bit-level MILP based automatic tools for MitM attacks, leading to improved or first preimage attacks on reduced-round Keccak-512, Ascon-XOF, and Xoodyak-XOF

Future Work

- For other instances of Keccak, it is open problem to apply one or two-round linear structures in the search for MitM attacks

- More tricks can be combined with automatic MILP model to achieve non-negligible improvements
  - Accelerate the search
  - Enlarge the space of solutions

- Extend to MitM collision attacks on sponge-based hash functions

ePrint 2023/518

Lingyue Qin, Boxin Zhao, Jialiang Hua, Xiaoyang Dong, Xiaoyun Wang. Weak-Diffusion Structure: Meet-in-the-Middle Attacks on Sponge-based Hashing Revisited.

Questions or Comments?

Reference I

Itai Dinur, Orr Dunkelman, and Adi Shamir.
Collision attacks on up to 5 rounds of SHA-3 using generalized internal differentials.
In *FSE 2013*, volume 8424, pages 219–240. Springer, 2013.

Christoph Dobraunig, Maria Eichlseder, Florian Mendel, and Martin Schläffer.
Ascon v1.2: Lightweight authenticated encryption and hashing.
J. Cryptol., 34(3):33, 2021.

Whitfield Diffie and Martin E. Hellman.
Special feature exhaustive cryptanalysis of the NBS data encryption standard.
Computer, 10(6):74–84, 1977.

Jian Guo, Meicheng Liu, and Ling Song.
Linear structures: Applications to cryptanalysis of round-reduced Keccak.
In *ASIACRYPT 2016, Proceedings, Part I*, volume 10031, pages 249–274, 2016.

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.

Reference II

Le He, Xiaoen Lin, and Hongbo Yu.
Improved preimage attacks on round-reduced Keccak-384/512 via restricted linear structures.
Cryptol. ePrint Arch., 2022/788, 2022.

Fukang Liu, Takanori Isobe, Willi Meier, and Zhonghao Yang.
Algebraic attacks on round-reduced Keccak.
In *ACISP 2021, Proceedings*, volume 13083, pages 91–110, 2021.

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.

Pawel Morawiecki, Josef Pieprzyk, and Marian Srebrny.
Rotational cryptanalysis of round-reduced Keccak.
In *FSE 2013,*, volume 8424, pages 241–262, 2013.

María Naya-Plasencia, Andrea Röck, and Willi Meier.
Practical analysis of reduced-round Keccak.
In *INDOCRYPT 2011*, volume 7107, pages 236–254, 2011.

Reference III

Mahesh Sreekumar Rajasree.
Cryptanalysis of round-reduced Keccak using non-linear structures.
In *INDOCRYPT 2019*, volume 11898, pages 175–192, 2019.