Meet-in-the-Middle Attacks Revisited: Key-recovery, Collision, and Preimage Attacks

Xiaoyang Dong[1] Jialiang Hua[1(✉)] Siwei Sun[2,3(✉)]
Zheng Li[4] Xiaoyun Wang[1,5] Lei Hu[2,3]

[1]Institute for Advanced Study, BNRist, Tsinghua University, Beijing, China,

[2]State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, China

[3]University of Chinese Academy of Sciences, China

[4]Faculty of Information Technology, Beijing University of Technology, China

[5]School of Cyber Science and Technology, Shandong University, Qingdao, China

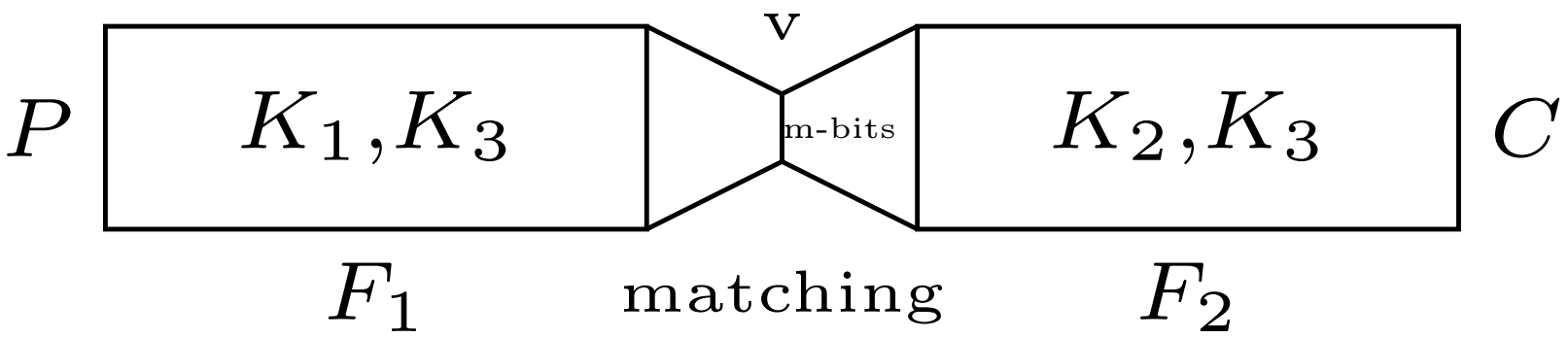
August 5, 2021

Outline

Outline

3-Subset Meet-in-the-Middle Attack

1. MITM Stage: Filter out some wrong key candidates and reduce the key space
2. Key Testing Stage: Test the surviving key candidates in a brute force manner.



3-Subset Meet-in-the-Middla Attack

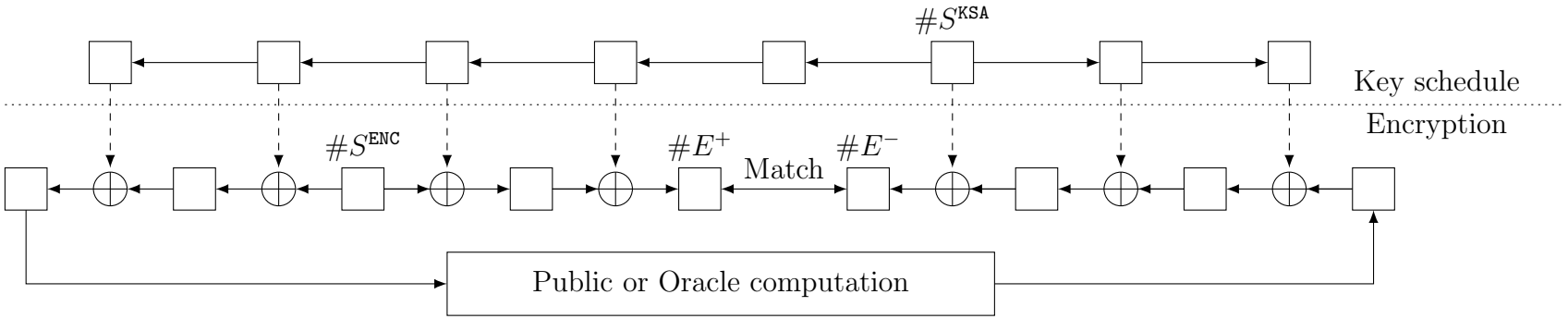
- block size: n-bit key size: l-bit.
  $K = K_1||K_2||K_3$, they are independent.
- m-bit($m < n$) of v can compute by $F_1(P)$ without knowing K_2 and ${F_2}^{-1}(C)$ without knowing K_1.
- $2^{|K_3|}(2^{|K_1|+|K_2|-m}) = 2^{|K|-m} = 2^{l-m}$ remained.
- test the surviving key candidates using some plaintext/ciphertext pairs.

The total computation complexity is estimated as

$$T = \underbrace{2^{|K_3|}(2^{|K_1|} + 2^{|K_2|})}_{MITM\ stage} + \underbrace{(2^{l-m} + 2^{l-m-n} + 2^{l-m-2n} + \cdots)}_{Key\ testing\ stage}.$$

A Formal Description of the MITM Technique

At EUROCRYPT 2021, the MITM preimage attacks on AES-like hashing was throughly modeled as constrained optimization problems by Bao et al.



- Forward chunk: $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}}) \rightarrow \#E^{+}$
- Backward chunk: $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}}) \rightarrow \#E^{-}$
- Partial match: $\#E^{+} \leftrightarrow \#E^{-}$

A Formal Description of the MITM Technique

the cells of $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}})$ are partitioned into different subsets with different meanings:

- neutral words of forward chunk: ■.
- neutral words of backward chunk: ■.
- known constants: ■.
- unknown words in both the forward and backward chunk: □.

A Formal Description of the MITM Technique

- The initial DoF for the forward computation: the number of ■ in $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}})$, defined as λ^+.
- The initial DoF for the backward computation: the number of ■ in $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}})$, defined as λ^-.
- Accumulated DoF consumed in forward chunk: denoted by σ^-
- Accumulated DoF consumed in backward chunk: denoted by σ^+

The remaining degree of freedom(DoF) of forward and backward computation denoted by DoF^+ and DoF^-, the degree of match is denoted by DoM.
So $DoF^+ = \lambda^+ - \sigma^+$, $DoF^- = \lambda^- - \sigma^-$. The objective is to maximize $min\{DoF^+, DoF^-, DoM\}$.

Outline

Automatic MITM Key-recovery Attacks

Divide λ^+ to $\lambda^+_{\mathtt{ENC}}$ and $\lambda^+_{\mathtt{KSA}}$ which from $\#S^{\mathtt{ENC}}$ and $\#S^{\mathtt{KSA}}$.
Divide σ^+ to $\sigma^+_{\mathtt{ENC}}$ and $\sigma^+_{\mathtt{KSA}}$ which is the accumulated DoF be consumed from $\#S^{\mathtt{ENC}}$ and $\#S^{\mathtt{KSA}}$.

Two requirements

In MITM key-recovery attack, the full key space must be tested and we prefer not to exhaust the full codebook.

1. $\lambda^+_{\mathtt{ENC}}$ and $\lambda^-_{\mathtt{ENC}}$ should be used up, $\lambda^+_{\mathtt{KSA}}$ and $\lambda^-_{\mathtt{KSA}}$ cannot be depleted.
2. there is at least one ■ cell in the plaintext state.

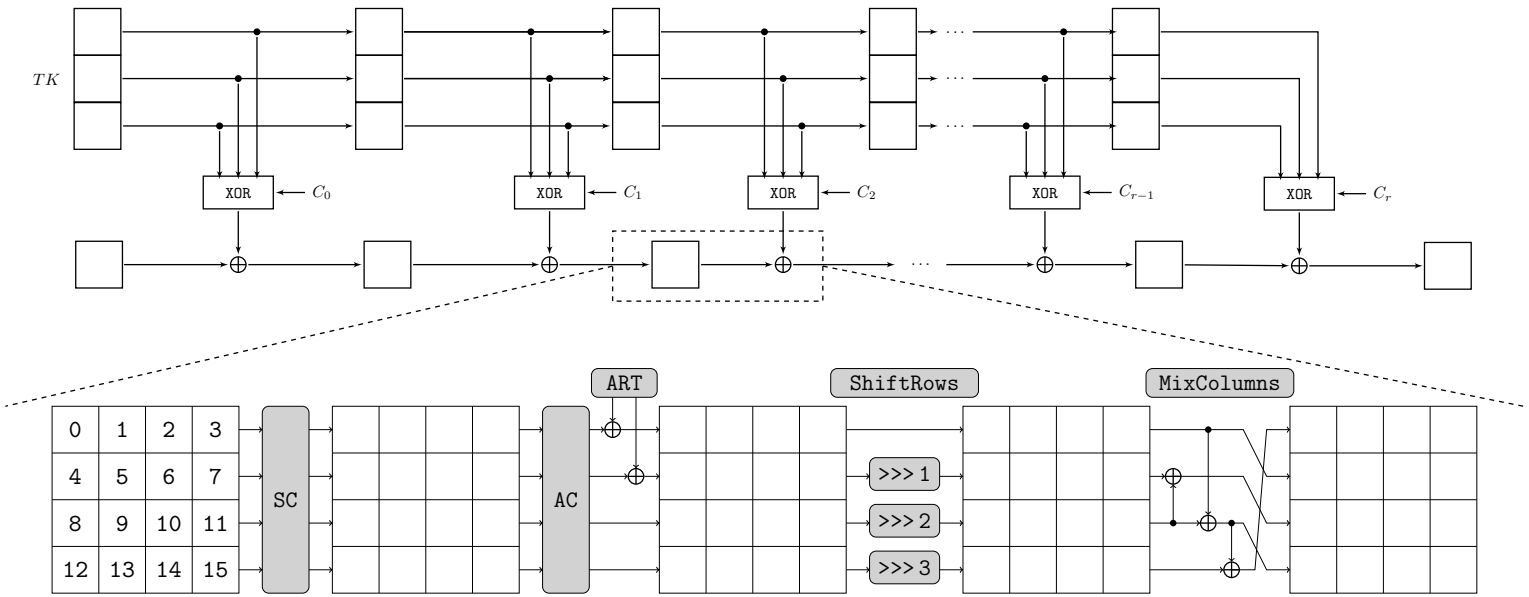
Then we have $\lambda^+_{\mathtt{ENC}} - \sigma^+_{\mathtt{ENC}} = 0$, $\lambda^-_{\mathtt{ENC}} - \sigma^-_{\mathtt{ENC}} = 0$

$$\lambda^+_{\mathtt{KSA}} - \sigma^+_{\mathtt{KSA}} >= 1,\ \lambda^-_{\mathtt{KSA}} - \sigma^-_{\mathtt{KSA}} >= 1$$

so $\mathrm{DoF}^+ = \lambda^+_{\mathtt{KSA}} - \sigma^+_{\mathtt{KSA}}$, $\mathrm{DoF}^- = \lambda^-_{\mathtt{KSA}} - \sigma^-_{\mathtt{KSA}}$

Outline

Specifications of Skinny



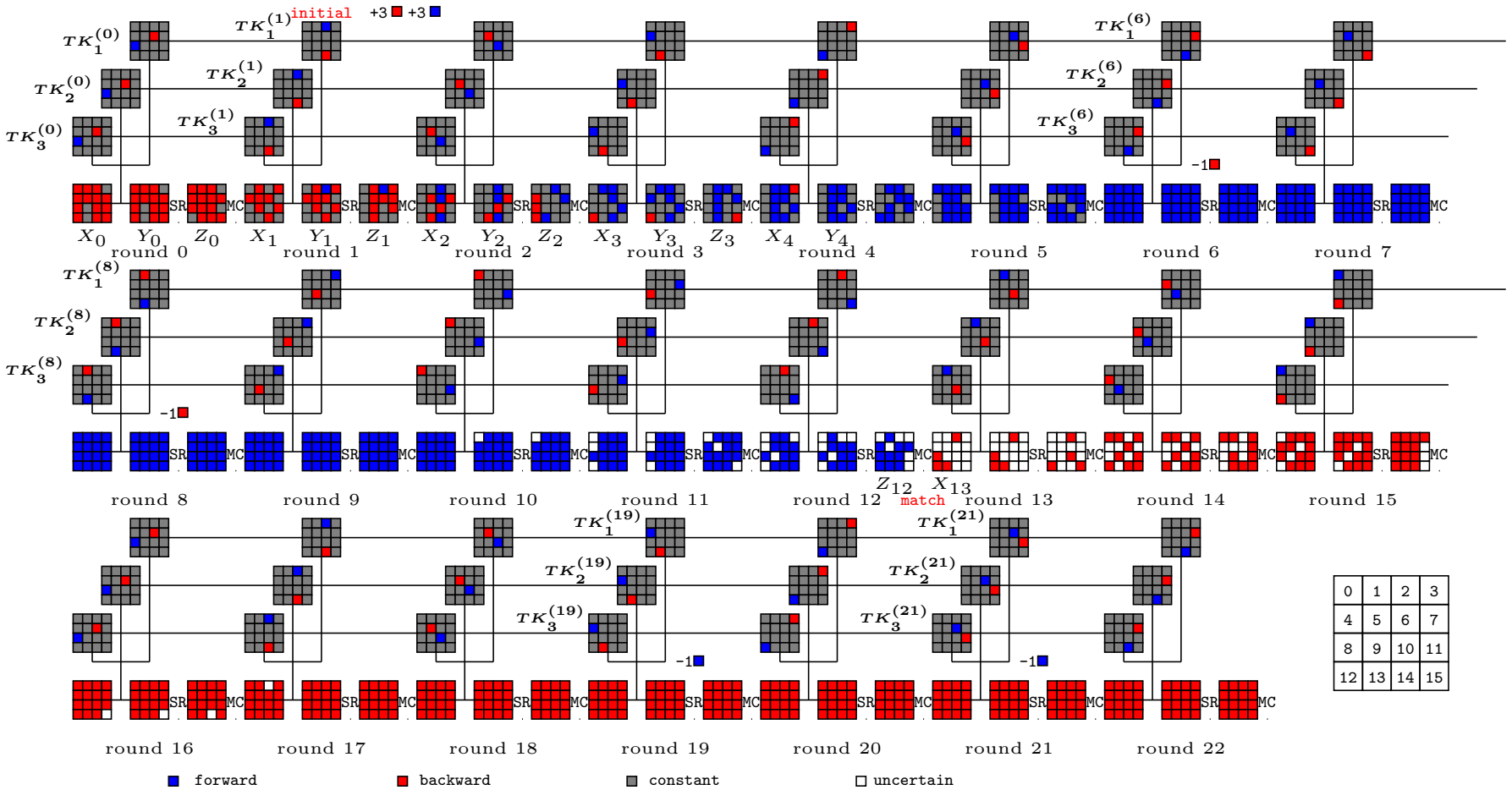
In each round, the state is updated with five operations: `SubCells (SC)`, `AddConstants (AC)`, `AddRoundTweakey (ART)`, `ShiftRows (SR)` and `MixColumns (MC)`.

Programming the MITM Attacks on SKINNY-n-3n with MILP

- Objective function: $max\{\min\{\mathrm{DoF}^+, \mathrm{DoF}^-, \mathrm{DoM}\}\}$
- Constraints for the Starting States: each cell in $(\#S^{\mathtt{ENC}}, \#S^{\mathtt{KSA}})$, $\alpha_i = 1$ if it is Blue and $\beta_i = 1$ it is Red. $\lambda^+_{\mathtt{ENC}} = \sum_i \alpha_i^{\mathtt{ENC}}$, $\lambda^+_{\mathtt{KSA}} = \sum_i \alpha_i^{\mathtt{KSA}}$, $\lambda^-_{\mathtt{ENC}} = \sum_i \beta_i^{\mathtt{ENC}}$, $\lambda^-_{\mathtt{KSA}} = \sum_i \beta_i^{\mathtt{KSA}}$
- Constraints for the Ending States: enumerate all possible patterns and convert it to linear inequalities.
- Constraints Imposed by the Computation Paths: Build rules for XOR, ART and MC.

$$\begin{cases} \lambda^+_{\mathtt{ENC}} - \sigma^+_{\mathtt{ENC}} = 0, \quad \lambda^-_{\mathtt{ENC}} - \sigma^-_{\mathtt{ENC}} = 0 \\ \mathrm{DoF}^+ = \lambda^+_{\mathtt{KSA}} - \sigma^+_{\mathtt{KSA}} \geq 1, \quad \mathrm{DoF}^- = \lambda^-_{\mathtt{KSA}} - \sigma^-_{\mathtt{KSA}} \geq 1 \end{cases}.$$

The MITM Key-recovery Attack on SKINNY-n-3n



The MITM Key-recovery Attack on SKINNY-n-3n

1 $X_0[3, 9, 13] \leftarrow 0$, $X_1[0, 2, 8, 10, 13] \leftarrow 0$, $X_2[1, 3, 9, 11] \leftarrow 0$, $Y_2[5] \leftarrow 0$, $X_3[0, 8] \leftarrow 0$, $Y_4[3] \leftarrow 0$

2 Collecting structure of plaintext-ciphertext pairs and store them in table H, which traverses the non-constant 16-3=13 cells in the plaintext.

3 **for** *All possilbe values of the* ■ *cells in* $(TK_1^{(0)}, TK_2^{(0)}, TK_3^{(0)})$ **do**

4 for $(a_1, a_2, b_1, b_2) \in \mathbb{F}_2^{4w}$ **do**

5 $Y_0[3] \leftarrow TK_1^{(0)}[3] \oplus TK_2^{(0)}[3] \oplus TK_3^{(0)}[3]$, $Y_0[9, 13] \leftarrow X_0[9, 13]$, $Z_0[3, 11, 12] \leftarrow Y_0[3, 9, 13]$, $X_1[12] \leftarrow X_1[0] \oplus Z_0[12]$, $X_1[7] \leftarrow Z_0[3]$, $X_1[15] \leftarrow Z_0[3] \oplus Z_0[11]$, $X_2[15] \leftarrow X_2[3] \oplus Z_1[15]$, $X_3[4] \leftarrow Z_2[0]$

6 Derive the solution space of the ■ cells in the TK by

7 $$TK_1^{(6)}[7] \oplus TK_2^{(6)}[7] \oplus TK_3^{(6)}[7] = a_1$$

8 $$TK_1^{(8)}[1] \oplus TK_2^{(8)}[1] \oplus TK_3^{(8)}[1] = a_2$$

9 Derive the solution space of the ■ cells in the TK by

10 $$TK_1^{(19)}[4] \oplus TK_2^{(19)}[4] \oplus TK_3^{(19)}[4] = b_1$$

11 $$TK_1^{(21)}[6] \oplus TK_2^{(21)}[6] \oplus TK_3^{(21)}[6] = b_2$$

12 Initialize L to be an empty hash table

13 **for** *the value in the solution space of* ■ *cells in* TK **do**

14 Compute $X_{13}[8]$ along the backward computation path: $X_4 \rightarrow X_0 \rightarrow E_K(X_0) \rightarrow X_{13}$ by accessing H

15 Insert relative information into L indexed by $X_{13}[8]$

16 **end**

17 **for** *the value in the solution space of* ■ *cells in* TK **do**

18 Compute $Z_{12}[4]$ and $Z_{12}[8]$ along the forward computation path: $X_1 \rightarrow Z_{12}$

19 **for** *Candidate keys in* $L[Z_{12}[4] \oplus Z_{12}[8]]$ **do**

20 Test the guessed key with several plaintext-ciphertext pairs

21 **end**

22 **end**

23 **end**

24 **end**

A summary of SKINNY and ForkSkinny

SKINNY							
Version	Rounds	Data	Time	Memory	Attack	Setting	Ref.
64-192	22	$2^{47.84}$	$2^{183.97}$	$2^{74.84}$	ID	SK	[16]
	23	2^{52}	2^{188}	2^{4}	MITM	SK	Ours
128-384	22	2^{96}	$2^{382.46}$	$2^{330.99}$	DS-MITM	SK	[15]
	22	$2^{92.22}$	$2^{373.48}$	$2^{147.22}$	ID	SK	[16]
	23	2^{104}	2^{376}	2^{8}	MITM	SK	Ours
ForkSkinny							
64-192	24	2^{52}	2^{188}	2^{4}	MITM	SK	Ours
128-384	24	2^{104}	2^{376}	2^{8}	MITM	SK	Ours
128-256	24	$2^{122.5}$	$2^{124.5}$	$2^{97.5}$	ID	RK	[2]
	26	2^{127}	$2^{250.3}$	2^{160}	ID	RK	[2]

Outline

Computing the solution spaces of nonlinear constrained neutral words

- In MITM attack on Skinny, we have to solve two systems of equations. It is easy to compute if they are linear.
- For nonlinear constrained neutral words, We present a table based technique to solve it.

Algorithm 1: Computing the solution spaces of the neutral words

Input: $(\#S^{\mathrm{ENC}}[\mathcal{G}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{G}^{\mathrm{KSA}}]) \in \mathbb{F}_2^{w\cdot(|\mathcal{G}^{\mathrm{ENC}}|+|\mathcal{G}^{\mathrm{KSA}}|)}$
Output: V, U

$V \leftarrow [\,]$, $U \leftarrow [\,]$
for $(\#S^{\mathrm{ENC}}[\mathcal{B}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{B}^{\mathrm{KSA}}]) \in \mathbb{F}_2^{w\cdot(|\mathcal{B}^{\mathrm{ENC}}|+|\mathcal{B}^{\mathrm{KSA}}|)}$ **do**
  $\boldsymbol{v} \leftarrow \boldsymbol{\pi}^{+}(\#S^{\mathrm{ENC}}[\mathcal{G}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{G}^{\mathrm{KSA}}], \#S^{\mathrm{ENC}}[\mathcal{B}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{B}^{\mathrm{KSA}}])$
  Insert $(\#S^{\mathrm{ENC}}[\mathcal{B}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{B}^{\mathrm{KSA}}])$ into V at index $\boldsymbol{v}$
end
for $(\#S^{\mathrm{ENC}}[\mathcal{R}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{R}^{\mathrm{KSA}}]) \in \mathbb{F}_2^{w\cdot(|\mathcal{R}^{\mathrm{ENC}}|+|\mathcal{R}^{\mathrm{KSA}}|)}$ **do**
  $\boldsymbol{u} \leftarrow \boldsymbol{\pi}^{-}(\#S^{\mathrm{ENC}}[\mathcal{G}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{G}^{\mathrm{KSA}}], \#S^{\mathrm{ENC}}[\mathcal{R}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{R}^{\mathrm{KSA}}])$
  Insert $(\#S^{\mathrm{ENC}}[\mathcal{R}^{\mathrm{ENC}}], \#S^{\mathrm{KSA}}[\mathcal{R}^{\mathrm{KSA}}])$ into U at index $\boldsymbol{u}$
end

MITM preimage attack on AES-like hashing with non-linearly constrained neutral words

Applying the technique to the MITM preimage attack can deal with nonlinearly constrained neutral words.

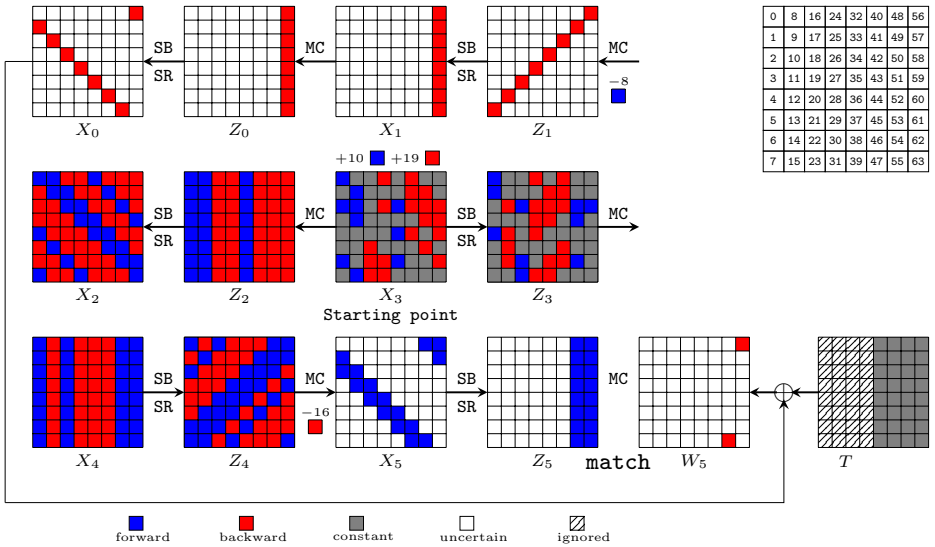
Algorithm 2: The framework of the MITM preimage attack on AES-like hashing with non-linearly constrained neutral words

```
for (#S^ENC[G^ENC], #S^KSA[G^KSA]) ∈ 𝔾 ⊆ 𝔽_2^{w·(|G^ENC|+|G^KSA|)} do
    Call Algorithm 1 to build V, U
    for c^+ = (a_1, ··· , a_{l+}) ∈ 𝔽_2^{w·l^+} do
        for c^- = (b_1, ··· , b_{l-}) ∈ 𝔽_2^{w·l^-} do
            L ← [ ]
            for (#S^ENC[B^ENC], #S^KSA[B^KSA]) ∈ V[c^+] do
                Compute E^+[M^+] along the forward computation path
                Insert (#S^ENC[B^ENC], #S^KSA[B^KSA]) into L indexed by E^+[M^+]
            end

            for (#S^ENC[R^KSA], #S^KSA[R^KSA]) ∈ U[c^-], do
                Compute E^-[M^-] along the backward computation path

                for (#S^ENC[B^ENC], #S^KSA[B^KSA]) ∈ L[E^-[M^-]] do
                    Reconstruct the (candidate) message X
                    if X is a preimage then
                        Output X and Stop.
                    end
                end
            end
        end
    end
end
```

Preimage attack on 6-round output transformation of Grøstl-256



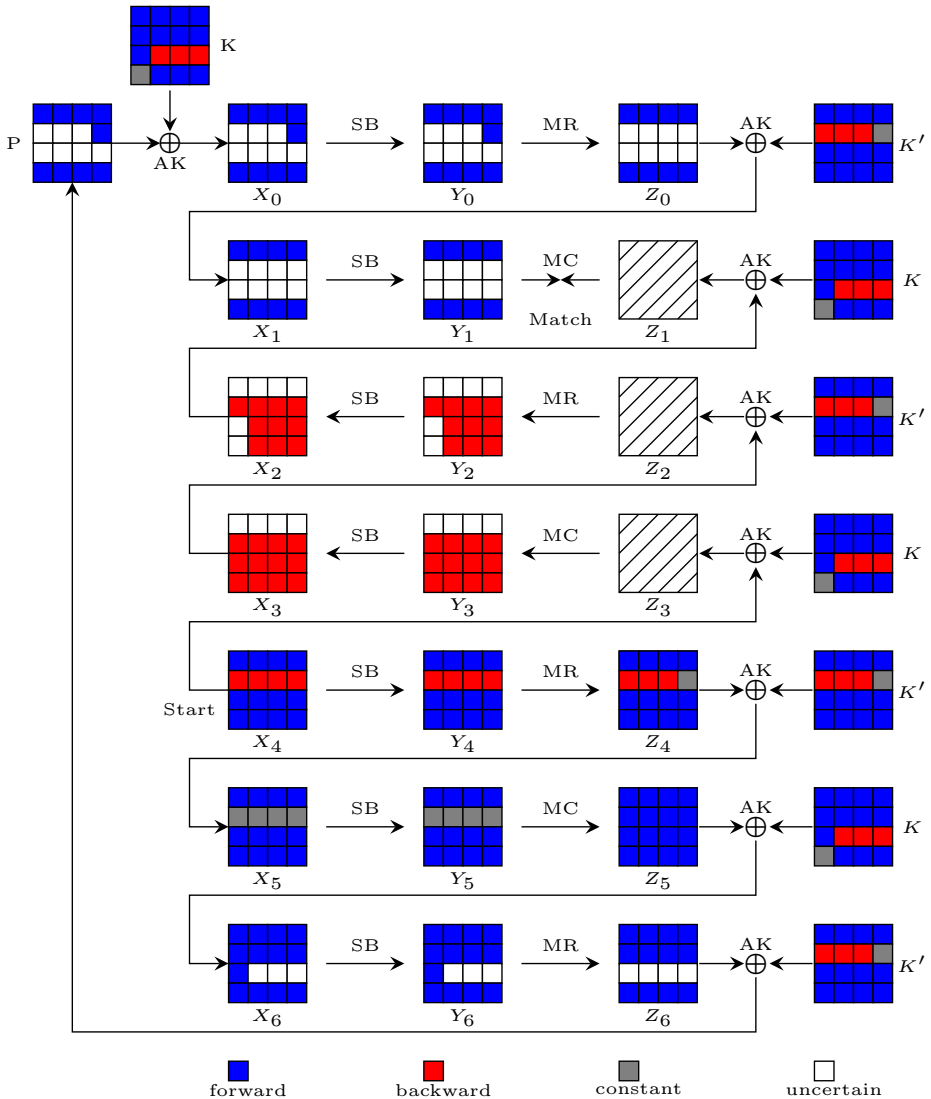
0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

Time complexity:

$$(2^8)^{32-10} + (2^8)^{32-19} + (2^8)^{32-\min(2,\ 3,\ 2)} \approx 2^{240}$$

Memory complexity: $(2^8)^{10} + (2^8)^{19} + (2^8)^{\min(2,\ 3)} \approx 2^{152}$

The 7-round MITM attack on Saturnin-Hash



Time complexity

$$(2^{16})^{16-\min(6,\ 3,\ 3)} \approx 2^{208}.$$

Memory complexity:

$$(2^{16})^3 = 2^{48}$$

.

Outline

MITM-based collision attack

- t-cell partial target preimage attack: can produce a message whose hash value is a random element in a given subspace.
- If there is an algorithm that can produce a different t-cell partial target preimage. We can find a collision by running the algorithm $2^{w(h-t)/2}$ times.
- Given an MITM characteristic, the framework for a collision attack is described in the following algorithm.

MITM-based collision attack

Algorithm 3: The framework of the MITM collision attack on AES-like hashing with non-linearly constrained starting states

```
1  Setting the selected t cells of #T to constants
2  H ← [ ]
3  for (#S^ENC[G^ENC], #S^KSA[G^KSA]) ∈ G ⊆ F_2^{w·(|G^ENC|+|G^KSA|)} do
4      V ← [ ], U ← [ ]
5      Call Algorithm 1 to populate V and U
6      for c^+ = (a_1, ··· , a_{l+}) ∈ F_2^{w·l+} do
7          for c^- = (b_1, ··· , b_{l-}) ∈ F_2^{w·l-} do
8              L ← [ ]
9              for (#S^ENC[B^ENC], #S^KSA[B^KSA]) ∈ V[c^+] do
10                 Compute E^+[M^+] along the forward computation path
11                 Insert (#S^ENC[B^ENC], #S^KSA[B^KSA]) into L indexed by E^+[M^+]
12             end
13             for (#S^ENC[R^KSA], #S^KSA[R^KSA]) ∈ U[c^-], do
14                 Compute E^-[M^-] along the backward computation path
15                 for (#S^ENC[B^ENC], #S^KSA[B^KSA]) ∈ L[E^-[M^-]] do
16                     Reconstruct the (candidate) message X
17                     if X is a t-cell partial target preimage then
18                         Insert X into H indexed by the hash value of X
19                         Stop when there is a collision
20                     end
21                 end
22             end
23         end
```

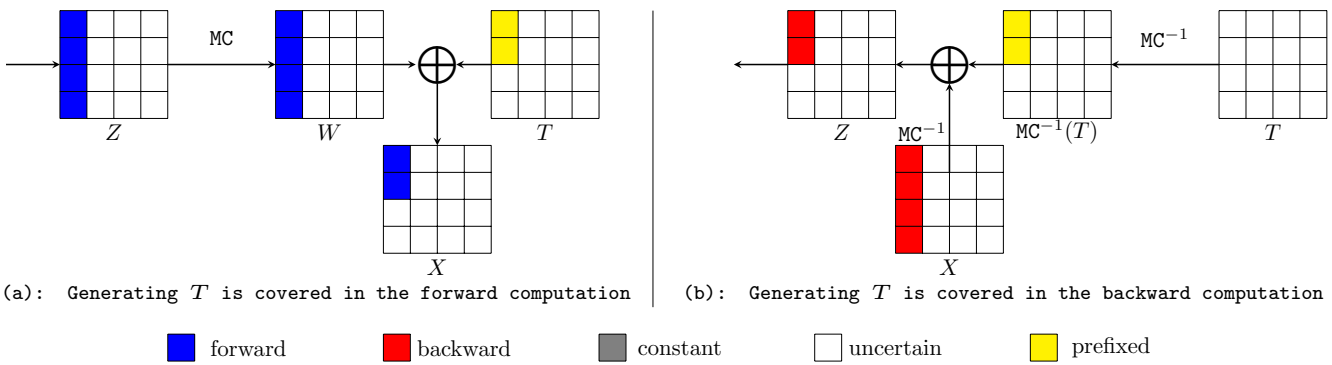
Automatic Search for MITM-based Collision Attacks

The time complexity is about $(2^w)^{\frac{h}{2}-\min\{\mathrm{DoF}^+-\frac{t}{2},\ \mathrm{DoF}^--\frac{t}{2},\ m-\frac{t}{2},\ \frac{t}{2}\}}$

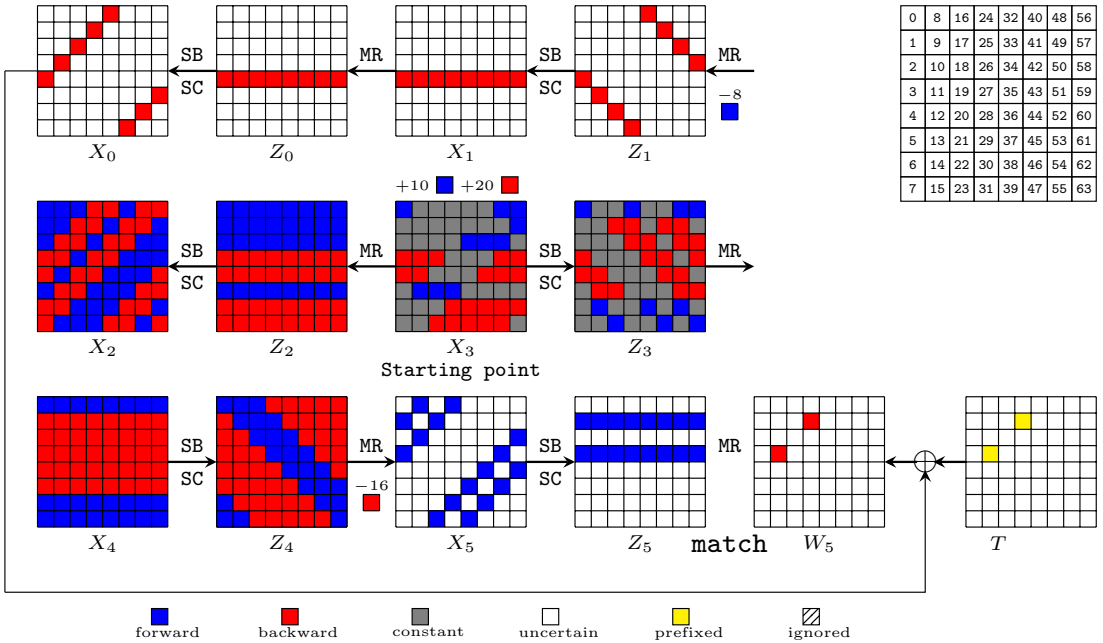
The objective function of the model is to maximize

$$\min(\mathrm{DoF}^+ - \frac{t}{2}, \mathrm{DoF}^- - \frac{t}{2}, m - \frac{t}{2}, \frac{t}{2})$$

- The matching point is placed at the last round: we can ignore the coloring information of T.
- The matching point is not at the last round: we can regard the ■ cells as ■ cells.



Collision attack on 6-round WHIRLPOOL



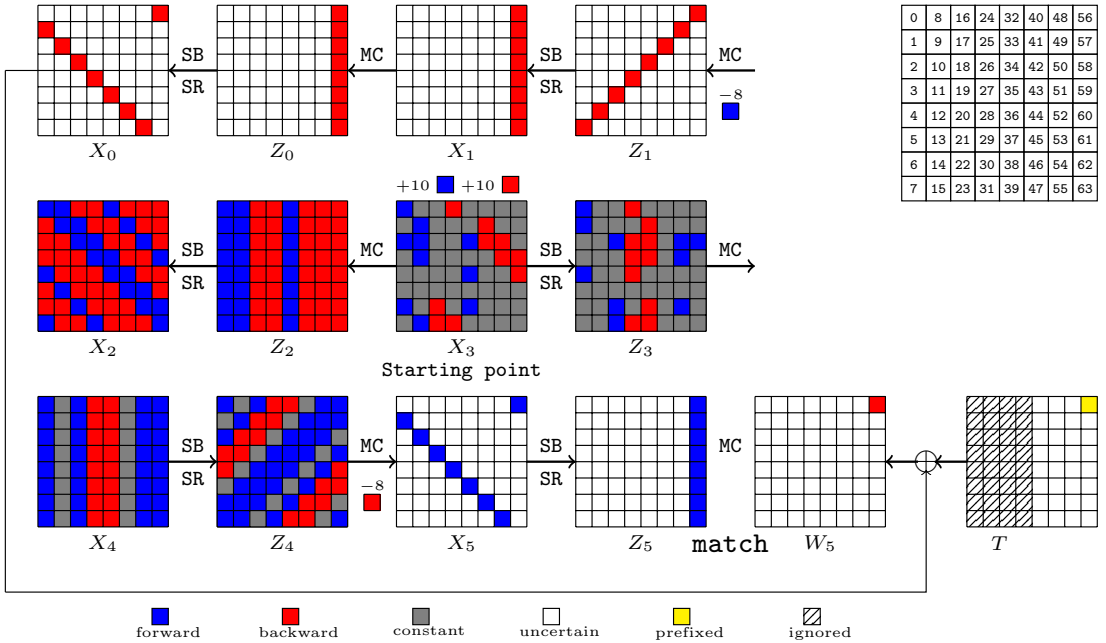
0	8	16	24	32	40	48	56
1	9	17	25	33	41	49	57
2	10	18	26	34	42	50	58
3	11	19	27	35	43	51	59
4	12	20	28	36	44	52	60
5	13	21	29	37	45	53	61
6	14	22	30	38	46	54	62
7	15	23	31	39	47	55	63

The time complexity is

$$(2^8)^{\frac{64}{2}-(10-\frac{2}{2})}+(2^8)^{\frac{64}{2}-(20-\frac{2}{2})}+(2^8)^{\frac{64}{2}-\min\{2-\frac{2}{2},\ 4-\frac{2}{2},\ 2-\frac{2}{2},\ \frac{2}{2}\}} \approx 2^{248}$$

The memory complexity is about 2^{248}.

Collision attack on 6-round Grøstl-256 output transformation



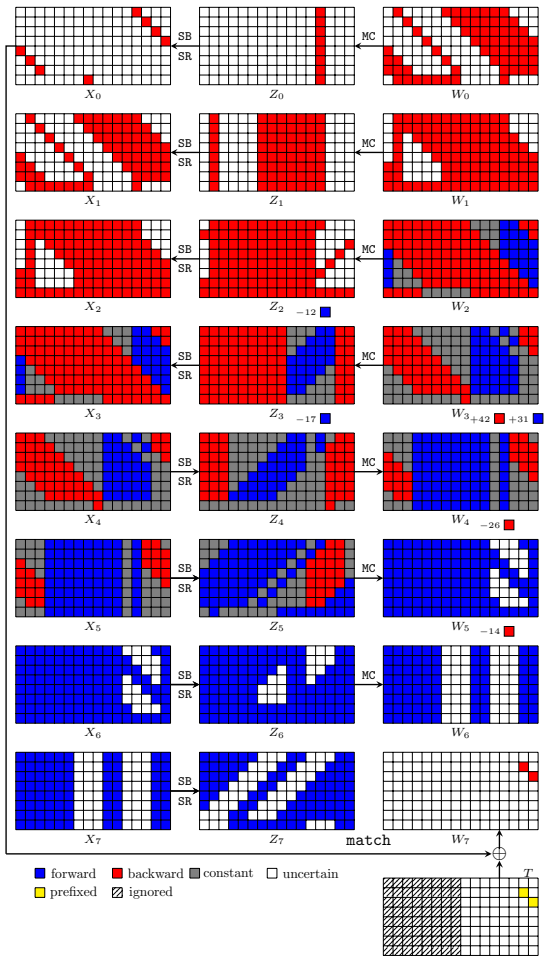
$\mathrm{DoF}^+ = \lambda^+ - l^+ = 2$, $\mathrm{DoF}^- = \lambda^- - l^- = 2$, $m = 1$, and $t = 1$.

The time complexity:

$$(2^8)^{10} + (2^8)^{10} + (2^8)^{\frac{32}{2} - \min\{2-\frac{1}{2},\ 2-\frac{1}{2},\ 1-\frac{1}{2},\ \frac{1}{2}\}} \approx 2^{124}.$$

The memory complexity is about: 2^{124}.

Collision attack on 8-round Grøstl-512 output transformation



$$\mathrm{DoF}^+ = \lambda^+ - l^+ = 2,$$
$$\mathrm{DoF}^- = \lambda^- - l^- = 2,\ m = 2$$
and $t = 2$
The time complexity

$$(2^8)^{\frac{64}{2}-(14-\frac{2}{2})} + (2^8)^{\frac{64}{2}-(16-\frac{2}{2})} + (2^8)^{\frac{64}{2}-\min\{2-\frac{2}{2},\ 2-\frac{2}{2},\ 2-\frac{2}{2},\ \frac{2}{2}\}} \approx 2^{248}$$

The memory cost: 2^{248}.

A Summary of the results

Target	Attack	Rounds	Time	Memory	Setting	Ref.
WHIRLPOOL						
Hash function	Collision	4	2^{120}	2^{16}	Classic	[9]
		5	2^{120}	2^{64}	Classic	[5, 8]
		6	2^{228}	-	Quantum	[6]
		6	2^{248}	2^{248}	Classic	Ours
	Preimage	5	2^{504}	2^{8}	Classic	[11]
		5	$2^{481.5}$	2^{64}		[17]
		6	2^{481}	2^{256}		[13]
Compression function	(Semi-) free-start	5	2^{120}	2^{16}	Classic	[9]
		7	2^{184}	2^{8}		[8]
		8	2^{120}	2^{8}		[13]
Grøstl-256						
Hash function	Collision	3	2^{64}	-	Classic	[14]
		5	2^{120}	2^{64}		[10]
	Pseudo preimage	5	$2^{244.8}$	2^{230}	Classic	[17]
		6	2^{252}	2^{251}		Ours
Compression function	Semi-free-start	6	2^{112}	2^{64}	Classic	[14]
Output Transformation	Preimage	5	2^{206}	2^{48}	Classic	[17]
		6	2^{240}	2^{152}		Ours
	Collision	6	2^{124}	2^{124}	Classic	Ours
Grøstl-512						
Hash function Compression function Output Transformation	Collision	5	2^{240}	2^{64}	Quantum	[4]
		7	2^{152}	2^{56}	Classic	[12]
		8	2^{248}	2^{248}	Classic	Ours
Hash function	Pseudo preimage	8	$2^{507.3}$	2^{507}	Classic	[17]
Saturnin-Hash						
Compression function Hash function	Preimage	7	2^{208}	2^{48}	Classic	Ours
		7	2^{232}	2^{48}		
SKINNY-128-384, Romulus-H, and AES hashing mode						
SKINNY-128-384-DM/MMO	Preimage	23	2^{120}	2^{8}	Classic	Ours
Romulus-H		23	2^{248}	2^{8}		Ours
AES-256		9	2^{120}	2^{8}		[1]
AES-256		10	2^{120}	2^{56}		Ours
Romulus-H compression function	Free-start	23	2^{124}	2^{124}		Ours

Thank you

Reference I


Zhenzhen Bao, Xiaoyang Dong, Jian Guo, Zheng Li, Danping Shi, Siwei Sun, and Xiaoyun Wang.
Automatic search of meet-in-the-middle preimage attacks on AES-like hashing.
Cryptology ePrint Archive, Report 2020/467, 2020.
https://eprint.iacr.org/2020/467.


Augustin Bariant, Nicolas David, and Gaëtan Leurent.
Cryptanalysis of Forkciphers.
IACR Trans. Symmetric Cryptol., 2020(1):233–265, 2020.


Christina Boura, Anne Canteaut, and Christophe De Cannière.
Higher-order differential properties of Keccak and Luffa.
In Antoine Joux, editor, *FSE 2011, Revised Selected Papers*, volume 6733, pages 252–269. Springer, 2011.


Xiaoyang Dong, Siwei Sun, Danping Shi, Fei Gao, Xiaoyun Wang, and Lei Hu.
Quantum collision attacks on AES-like hashing with low quantum random access memories.
In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Proceedings, Part II*, volume 12492, pages 727–757. Springer, 2020.

Reference II


Henri Gilbert and Thomas Peyrin.
Super-Sbox cryptanalysis: Improved attacks for AES-like permutations.
In *FSE 2010, Revised Selected Papers*, pages 365–383, 2010.


Akinori Hosoyamada and Yu Sasaki.
Finding hash collisions with quantum computers by using differential trails with smaller probability than birthday bound.
In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Proceedings, Part II*, volume 12106, pages 249–279. Springer, 2020.


Jérémy Jean, María Naya-Plasencia, and Thomas Peyrin.
Improved rebound attack on the finalist Grøstl.
In Anne Canteaut, editor, *FSE 2012, Revised Selected Papers*, volume 7549, pages 110–126. Springer, 2012.


Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer.
Rebound distinguishers: Results on the full WHIRLPOOL compression function.
In *ASIACRYPT 2009, Proceedings*, pages 126–143, 2009.

Reference III


Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen.
The rebound attack: Cryptanalysis of reduced WHIRLPOOL and Grøstl.
In *FSE 2009, Revised Selected Papers*, pages 260–276, 2009.


Florian Mendel, Vincent Rijmen, and Martin Schläffer.
Collision attack on 5 rounds of Grøstl.
In *FSE 2014, Revised Selected Papers*, pages 509–521, 2014.


Yu Sasaki.
Meet-in-the-middle preimage attacks on AES hashing modes and an application to WHIRLPOOL.
In Antoine Joux, editor, *FSE 2011, Revised Selected Papers*, volume 6733, pages 378–396. Springer, 2011.


Yu Sasaki, Yang Li, Lei Wang, Kazuo Sakiyama, and Kazuo Ohta.
Non-full-active super-sbox analysis: Applications to ECHO and Grøstl.
In *ASIACRYPT 2010, Proceedings*, pages 38–55, 2010.

Reference IV


Yu Sasaki, Lei Wang, Shuang Wu, and Wenling Wu.
Investigating fundamental security requirements on WHIRLPOOL: Improved preimage and collision attacks.
In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012, Proceedings*, volume 7658, pages 562–579. Springer, 2012.


Martin Schläffer.
Updated differential analysis of Grøstl.
In *Grøstl website*, 2011.


Danping Shi, Siwei Sun, Patrick Derbez, Yosuke Todo, Bing Sun, and Lei Hu.
Programming the demirci-selçuk meet-in-the-middle attack with constraints.
In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II*, volume 11273 of *Lecture Notes in Computer Science*, pages 3–34. Springer, 2018.


Mohamed Tolba, Ahmed Abdelkhalek, and Amr M. Youssef.
Impossible differential cryptanalysis of reduced-round SKINNY.
In Marc Joye and Abderrahmane Nitaj, editors, *AFRICACRYPT 2017, Proceedings*, volume 10239, pages 117–134, 2017.

Reference V

Shuang Wu, Dengguo Feng, Wenling Wu, Jian Guo, Le Dong, and Jian Zou. (pseudo) preimage attack on round-reduced Grøstl hash function and others. In *FSE 2012, Revised Selected Papers*, pages 127–145, 2012.