Triangulating Rebound Attack on AES-like Hashing

Xiaoyang Dong[1,3] Jian Guo[2] Shun Li[2,4] Phuong Pham[2]

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

[2]School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

[3]National Financial Cryptography Research Center, Beijing, China

[4]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Summary

In this paper, we:

- Improve Rebound Attack with enlarging the Inbound phase to SuperInbound (multiple Inbound phases)

Summary

In this paper, we:

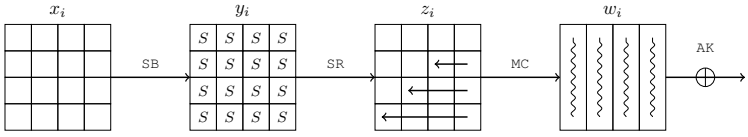
- Improve Rebound Attack with enlarging the Inbound phase to SuperInbound (multiple Inbound phases)
- Solve the system of equations of SuperInbound by an automatic way with Triangulation Algorithm

Summary

In this paper, we:

- Improve Rebound Attack with enlarging the Inbound phase to SuperInbound (multiple Inbound phases)
- Solve the system of equations of SuperInbound by an automatic way with Triangulation Algorithm
- Apply new method to obtain new semi-freestart collision attacks on AES-like Hashings
- Find the first 6-round semi-freestart collision of AES-128-MMO

AES-like Hashing



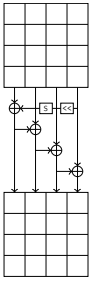
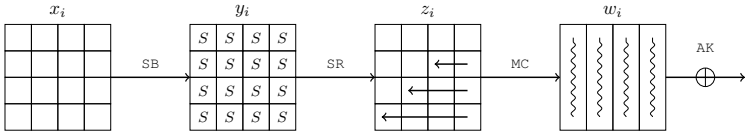


Figure 1: The AES round-function and key-schedule

AES-like Hashing



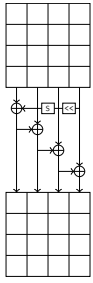
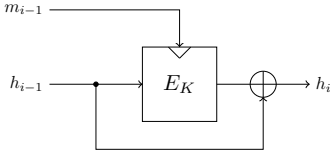
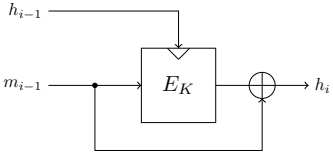


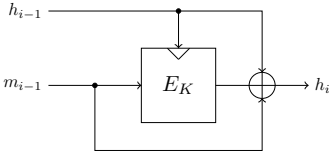
Figure 1: The AES round-function and key-schedule



(DM) Davies-Meyer



(MMO) Matyas-Meyer-Oseas



(MP) Miyaguchi-Preneel

AES System of Equations

$$R_i : \begin{cases} y_i \oplus S(w_{i-1} \oplus k_i) = 0 \\ w_i \oplus \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} y_i[0] & y_i[4] & y_i[8] & y_i[12] \\ y_i[5] & y_i[9] & y_i[13] & y_i[1] \\ y_i[10] & y_i[14] & y_i[2] & y_i[6] \\ y_i[15] & y_i[3] & y_i[7] & y_i[11] \end{pmatrix} = 0 \end{cases} \tag{1}$$

$$KS_i : \begin{cases} k_i[j] \oplus k_i[j-4] \oplus k_{i-1}[j] = 0, & j = 4, \ldots b-1 \\ k_i[0] \oplus k_{i-1}[0] \oplus S(k_{i-1}[b-3]) \oplus \mathbf{RCON}_i = 0 \\ k_i[1] \oplus k_{i-1}[1] \oplus S(k_{i-1}[b-2]) = 0 \\ k_i[2] \oplus k_{i-1}[2] \oplus S(k_{i-1}[b-1]) = 0 \\ k_i[3] \oplus k_{i-1}[3] \oplus S(k_{i-1}[b-4]) = 0 \end{cases} \tag{2}$$

Collision in Classic and Quantum Setting of n-bit hash function

- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value

Collision in Classic and Quantum Setting of n-bit hash function

- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value
- Semi-freestart collision: IV can be chosen by attacker

Collision in Classic and Quantum Setting of n-bit hash function

- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value
- Semi-freestart collision: IV can be chosen by attacker
- Free-start collision: $H(IV, m) = H(IV', m')$

Collision in Classic and Quantum Setting of n-bit hash function

- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value
- Semi-freestart collision: IV can be chosen by attacker
- Free-start collision: $H(IV, m) = H(IV', m')$
- Classic complexity: Birthday paradox with $2^{n/2}$

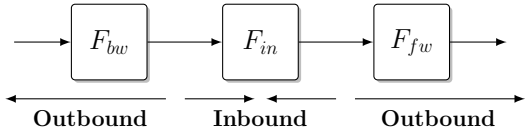
Collision in Classic and Quantum Setting of n-bit hash function

- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value
- Semi-freestart collision: IV can be chosen by attacker
- Free-start collision: $H(IV, m) = H(IV', m')$
- Classic complexity: Birthday paradox with $2^{n/2}$
- Quantum complexity with quantum memory: BHT algorithm with $2^{n/3}$

Collision in Classic and Quantum Setting of n-bit hash function

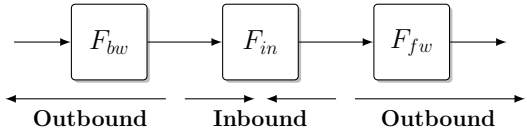
- Collision: find a message pair (m, m') s.t $H(IV, m) = H(IV, m')$, IV is fixed initial value
- Semi-freestart collision: IV can be chosen by attacker
- Free-start collision: $H(IV, m) = H(IV', m')$
- Classic complexity: Birthday paradox with $2^{n/2}$
- Quantum complexity with quantum memory: BHT algorithm with $2^{n/3}$
- Quantum complexity without quantum memory: Rho algorithm with $2^{n/2}$

Rebound Attack



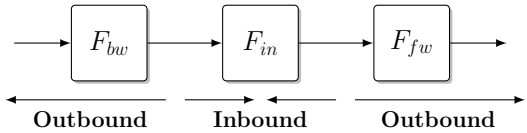
- Introduced by Mendel et al. in [MRST09].
- Splits the internal cipher F into three subparts: $F = F_{fw} \circ F_{in} \circ F_{bw}$.

Rebound Attack



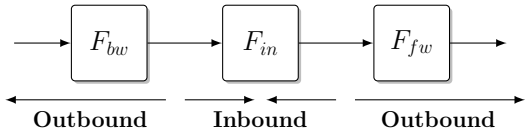
- Introduced by Mendel et al. in [MRST09].
- Splits the internal cipher F into three subparts: $F = F_{fw} \circ F_{in} \circ F_{bw}$.
  - **Inbound phase**: the attackers generate enough pairs fulfill the low-probability part in the middle of the differential trail with a meet-in-the-middle technique.

Rebound Attack



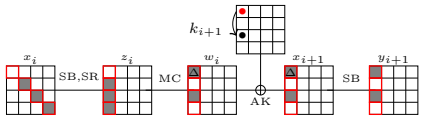
- Introduced by Mendel et al. in [MRST09].
- Splits the internal cipher F into three subparts: $F = F_{fw} \circ F_{in} \circ F_{bw}$.
  - **Inbound phase**: the attackers generate enough pairs fulfill the low-probability part in the middle of the differential trail with a meet-in-the-middle technique.
  - **Outbound phase**: these matched values pairs are computed backward and forward through F_{bw} and F_{fw} to obtain a pair satisfied the outbound differential trail.

Rebound Attack



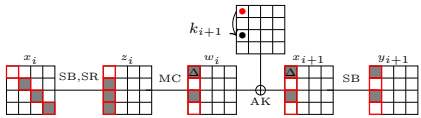
- Introduced by Mendel et al. in [MRST09].
- Splits the internal cipher F into three subparts: $F = F_{fw} \circ F_{in} \circ F_{bw}$.
  - **Inbound phase**: the attackers generate enough pairs fulfill the low-probability part in the middle of the differential trail with a meet-in-the-middle technique.
  - **Outbound phase**: these matched values pairs are computed backward and forward through F_{bw} and F_{fw} to obtain a pair satisfied the outbound differential trail.
  - **Complexity**: p is the probability of outbound trail, then the complexity is roughly $1/p$ in classic setting with $p > 2^{-n/2}$, $1/\sqrt{p}$ in quantum setting $p > 2^{-2n/3}$ with quantum memory, or $p > 2^{-n}$ without quantum memory

Connecting multiple inbound parts

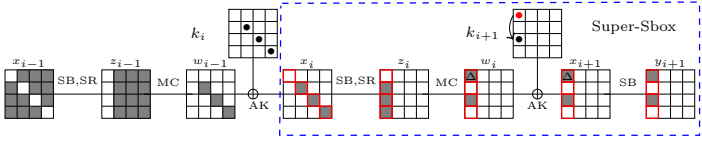


- Active bytes of x_i, x_{i+1} are known, random $k_{i+1}[0] \rightarrow k_{i+1}[2]$ is computed.

Connecting multiple inbound parts

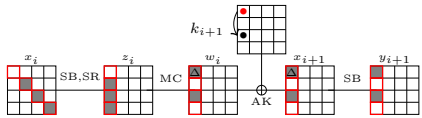


- Active bytes of x_i, x_{i+1} are known, random $k_{i+1}[0] \rightarrow k_{i+1}[2]$ is computed.

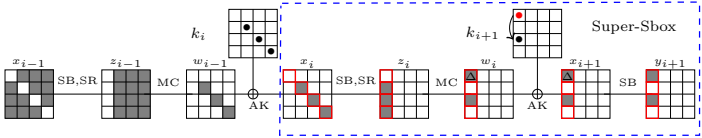


- Active bytes of x_{i-1}, x_i are known, then k_i's bytes are computed accordingly.

Connecting multiple inbound parts



- Active bytes of x_i, x_{i+1} are known, random $k_{i+1}[0] \rightarrow k_{i+1}[2]$ is computed.



- Active bytes of x_{i-1}, x_i are known, then k_i's bytes are computed accordingly.

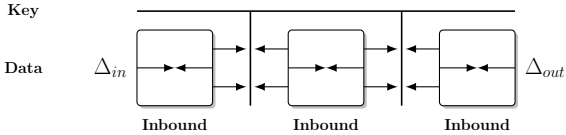
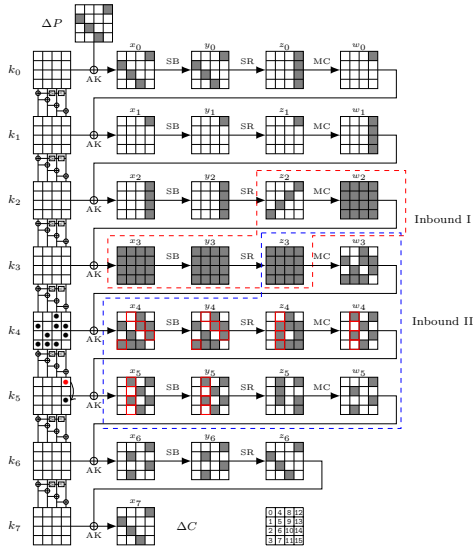


Figure 2: Super-Inbound: bridging multiple local inbound parts

New 7-round AES-128-MMO trail

- Choose the valid differences for z_2, w_3, w_4, w_5
- Assess differential distribution table (DDT) to obtain values of 31 active bytes of x_3, x_4, x_5
- Use k_4, k_5 to connect these known bytes
- Outbound probability: 2^{-56} for MC $w_1 \rightarrow z_1$ and $\Delta_P = \Delta_C$
- Complexity: 2^{56}



Triangulation Algorithm

- An efficient Gaussian-based-algorithm to solve system of bijective equations introduced by Khovratovich, Biryukov, and Nikolić [KBN]

Triangulation Algorithm

- An efficient Gaussian-based-algorithm to solve system of bijective equations introduced by Khovratovich, Biryukov, and Nikolić [KBN]
- Automatically detect a way to solve the nonlinear system.

Triangulation Algorithm

- An efficient Gaussian-based-algorithm to solve system of bijective equations introduced by Khovratovich, Biryukov, and Nikolić [KBN]
- Automatically detect a way to solve the nonlinear system.
- Models an AES-like block cipher as a system of key schedule and round function equations, and the state bytes and key bytes as variables.

Triangulation Algorithm

$$\begin{cases} F(x_1 \oplus x_2) \oplus 2 \times a & = 0 \\ x_1 \oplus 3 \times x_3 \oplus G(x_4) & = 0 \\ x_2 \oplus x_4 \oplus a & = 0 \end{cases}$$

1. Construct a system of equations whose variables are the bytes. The predefined values are fixed to constants.

Triangulation Algorithm

$$\begin{cases} F(x_1 \oplus x_2) \oplus 2 \times a & = 0 \\ x_1 \oplus 3 \times x_3 \oplus G(x_4) & = 0 \\ x_2 \oplus x_4 \oplus a & = 0 \end{cases}$$

1. Construct a system of equations whose variables are the bytes. The predefined values are fixed to constants.
2. All variables and equations are marked as non-processed.

Triangulation Algorithm

$$\begin{cases} F(x_1 \oplus x_2) \oplus 2 \times a & = 0 \\ x_1 \oplus 3 \times x_3 \oplus G(x_4) & = 0 \\ x_2 \oplus x_4 \oplus a & = 0 \end{cases}$$

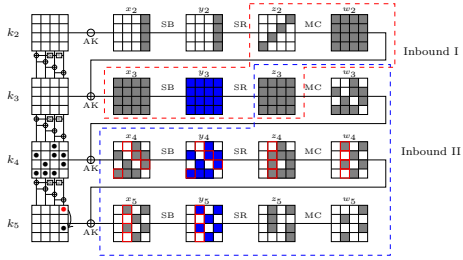
1. Construct a system of equations whose variables are the bytes. The predefined values are fixed to constants.
2. All variables and equations are marked as non-processed.
3. Mark the variable which is involved in only one non-processed equation as processed. Also mark this equation as processed. If no such variable exist, exit.

Triangulation Algorithm

$$\begin{cases} F(x_1 \oplus x_2) \oplus 2 \times a & = 0 \\ x_1 \oplus 3 \times x_3 \oplus G(x_4) & = 0 \\ x_2 \oplus x_4 \oplus a & = 0 \end{cases}$$

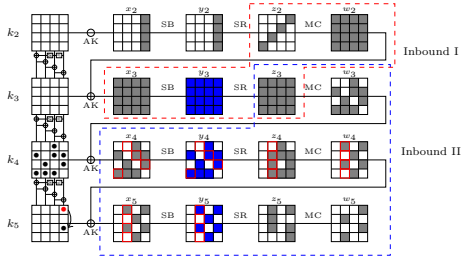
1. Construct a system of equations whose variables are the bytes. The predefined values are fixed to constants.
2. All variables and equations are marked as non-processed.
3. Mark the variable which is involved in only one non-processed equation as processed. Also mark this equation as processed. If no such variable exist, exit.
4. Return to Step 3 if there still exist non-processed equations.
5. Return all non-processed variables as "free variables".

Triangulation Rebound Attack



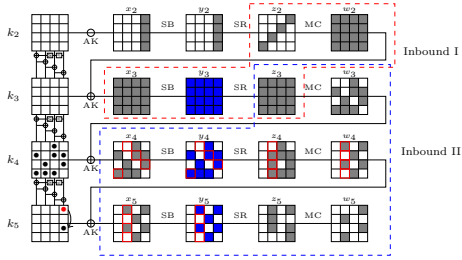
1. Find a truncated differential trail following the *Constrained Programming* based search model from [BGLP21].

Triangulation Rebound Attack



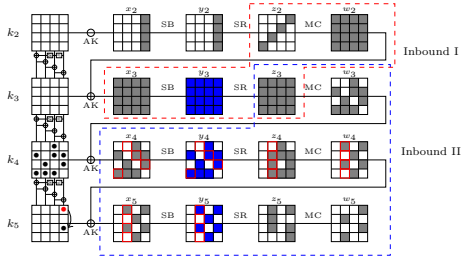
1. Find a truncated differential trail following the *Constrained Programming* based search model from [BGLP21].
2. Choose certain 2 to 4 consecutive rounds as the *Super-Inbound* and the number of active bytes is less than 32.

Triangulation Rebound Attack



1. Find a truncated differential trail following the *Constrained Programming* based search model from [BGLP21].
2. Choose certain 2 to 4 consecutive rounds as the *Super-Inbound* and the number of active bytes is less than 32.
3. Build equation system that connects the inbound parts and through key schedule. Check if the system is solvable by triangulation algorithm.

Triangulation Rebound Attack



1. Find a truncated differential trail following the *Constrained Programming* based search model from [BGLP21].
2. Choose certain 2 to 4 consecutive rounds as the *Super-Inbound* and the number of active bytes is less than 32.
3. Build equation system that connects the inbound parts and through key schedule. Check if the system is solvable by triangulation algorithm.
4. For AES, a basis of $n + k$ variables, if m bytes are fixed, the algorithm should output $n + k - m$ bytes

Comparison with previous work

Previous work:

- Inbound pairs are found manually [GP10, LMR$^+$09, Nay11]
- Only quantum collision attack on 7-round AES-hash [HS]
- Inbound part has 2 rounds [GP10, LMR$^+$09]

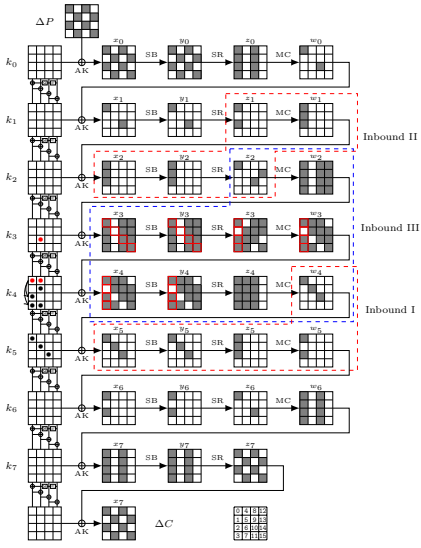
Comparison with previous work

Previous work:
- Inbound pairs are found manually [GP10, LMR$^+$09, Nay11]
- Only quantum collision attack on 7-round AES-hash [HS]
- Inbound part has 2 rounds [GP10, LMR$^+$09]

Our work:
- Inbound part is opened to **3 to 4** rounds for AES
- The first semi-free-start collision attack on 7-round AES-MMO in classic setting
- Automatic way to find the inbound pairs with Triangulation algorithm

New 8-round AES-128-MMO trail

- 30 active bytes in x_2, x_3, x_4, x_5 are known due to DDT.
- Use k_3, k_4, k_5 to connect these known bytes
- Triangulation algorithm output 3 free bytes: $k_3[6], k_4[0,4]$ and 1 filter
- Outbound probability: 2^{-64} for $\Delta_P = \Delta_C$ and 1 filter 2^{-8}
- Complexity: 2^{34} in quantum setting



First 6-round semi-freestart collision on AES-128-MMO

- By cutting the first and last round of 8-round trail

Plaintext			
P	13622301 f4ad7096 c7cea69e e26646e5	K_{-1}	6aa0c09f 92854210 9411daed 8db5f736
P'	eb622301 f4ad7096 c7cedd9e e26646e5		
Ciphertext			
C	ede24da9 8f76183b b23536aa 34dbd668		
C'	15e24da9 8f76183b b2354daa 34dbd668		
Hash output after feedback			
H	fe806ea8 7bdb68ad 75fb9034 d6bd90cd		
H'	fe806ea8 7bdb68ad 75fb9034 d6bd90cd		

Table 1: Pair found by semi-free-start collision attack on 6-round AES-128

Attack on SKINNY-hash-128-384

- MC operation is non-MDS:

$$MC\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} a \oplus c \oplus d \\ a \\ b \oplus c \\ a \oplus c \end{pmatrix}$$

- The values of k_i are the XOR of subkeys and constants of AK operator.
- Linear key-schedule

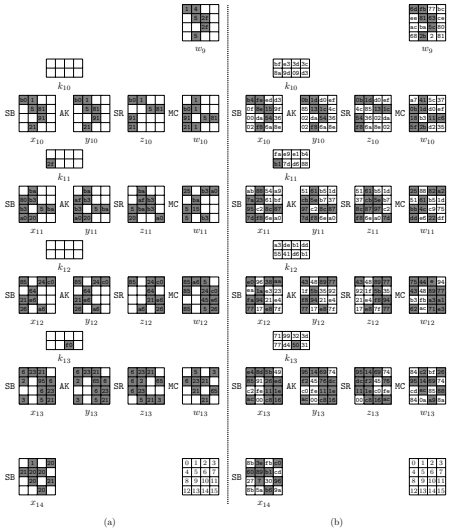


Figure 3: (a) Differences, (b) Values.

Results

Target	Attack	Rounds	Time	C-Mem	qRAM	Setting	Ref.
AES Hash	Collision	6/10	2^{56}	-	0	Quantum	[GP10, LMR+09]
		8/10	$2^{55.53}$	0	0	Quantum	Ours
Compression function AES-128-MMO/MP	Semi-free	6/10	2^{24}	-	-	Classic	Ours
	Semi-free	7/10	2^{56}	2^{16}	-	Classic	Ours
	Semi-free	8/10	2^{34}	-	-	Quantum	Ours
Compression function Saturnin-hash	Semi-free	7/16	2^{86}	-	-	Quantum	Ours
	Free-start	8/16	$2^{122.5}$	-	-	Quantum	[DZS+21]
	Free-start	8/16	$2^{89.65}$	-	-	Quantum	Ours
	Free-start	10/16	$2^{127.2}$	-	-	Quantum	Ours
Compression function Grostl	Semi-free	6/14	2^{180}	2^{64}	-	Classic	[Sch11]
		7/14	2^{214}	-	-	Quantum	
Compression v0	Semi-free	8/14	2^{244}	-	-	Quantum	Ours
Compression function SKINNY-hash	Free-start	16/56	$2^{59.8}$	-		Quantum	[DZS+21]
		19/56	$2^{51.2}$	-		Classic	Ours
		21/56	$2^{46.2}$	-		Quantum	Ours

Table 2: A summary of the results.

References I

[BGLP21] Zhenzhen Bao, Jian Guo, Shun Li, and Phuong Pham.
Quantum Multi-Collision Distinguishers.
Cryptology ePrint Archive, Report 2021/703, 2021.
https://ia.cr/2021/703.

[DZS+21] Xiaoyang Dong, Zhiyu Zhang, Siwei Sun, Congming Wei, Xiaoyun Wang, and Lei Hu.
Automatic classical and quantum rebound attacks on AES-like hashing by exploiting related-key differentials.
In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Singapore, December 6-10, 2021, Proceedings, Part I*, volume 13090 of *Lecture Notes in Computer Science*, pages 241–271. Springer, 2021.

[GP10] Henri Gilbert and Thomas Peyrin.
Super-Sbox cryptanalysis: Improved attacks for AES-like permutations.
In *FSE 2010, Seoul, Korea, February 7-10, 2010*, pages 365–383, 2010.

References II

[HS] 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.

[KBN] Dmitry Khovratovich, Alex Biryukov, and Ivica Nikolic.
Speeding up collision search for byte-oriented hash functions.
In *CT-RSA 2009, Proceedings*, volume 5473, pages 164–181.

[LMR+09] Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schläffer.
Rebound distinguishers: Results on the full Whirlpool compression function.
In *ASIACRYPT 2009, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 126–143, 2009.

References III

[MRST09] Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S. Thomsen.
The rebound attack: Cryptanalysis of reduced Whirlpool and Grøstl.
In *FSE 2009, Leuven, Belgium, February 22-25, 2009*, pages 260–276, 2009.

[Nay11] María Naya-Plasencia.
How to improve rebound attacks.
In *CRYPTO 2011, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 188–205, 2011.

[Sch11] Martin Schläffer.
Updated differential analysis of Grøstl.
Grøstl website (January 2011), 2011.