New Distinguishing Attack on MAC Using Secret-Prefix Method

Xiaoyun Wang[1,2] , Wei Wang[2],
Keting Jia[2] and Meiqin Wang[2]

1 Tsinghua University
2 Shandong University

清華大學

Tsinghua University

Outline

- Introduction to MAC Algorithms
- Related Distinguishing Attacks on MACs
- Distinguishing Attack on 61-Round LPMAC-SHA1
- Conclusions

Introduction to MAC Algorithms

Definition and Applications

- Definition: MAC=hash function + secret key
- Security properties:
  - Data integrity
  - Data origin authentication
- Practical applications
  - Internet security: IPSec, SSL, SSH, etc.
  - Finance: banking, electronic purses, etc.

Security

- Distinguishing Attack
  - Distinguishing-R Attack: MAC or a random function
  - Distinguishing-H Attack: which cryptographic hash function is embedded in the MAC construction
- Forgery Attack
  - Existential Forgery Attack: compute a valid MAC for a random message
  - Universal Forgery Attack: compute a valid MAC for any given message
- Key Recovery Attack

Remark: Distinguishing-R： $2^{n/2}$complexity(from Preneel and van Oorschot Attack) Ideal complexity: 2^n computations, n is the length of the tag

Three Previous MACs Based on Hash Functions

- Secret prefix： $H(K_1||L||M)$

  L： length of the message M

K_1	$L \| M$

- Secret suffix： $H(M||K_2)$

M	K_2

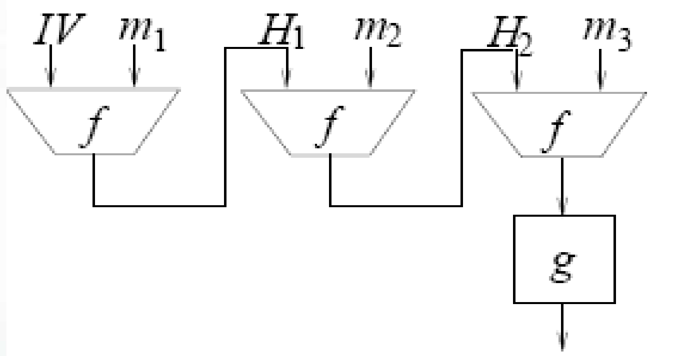
- Envelope： $H(K_1||M||K_2)$

K_1	M	K_2

Related Distinguishing Attacks on MACs

A General Attack on Iterated MACs

- Based on the birthday attack , B. Preneel, P. van Oorschot, Crypto'95
- The attack works with all the iterative MACs: block cipher and hash functions



1. Randomly select $2^{(n+1)/2}$ M_i, Query the corresponding MACs C_i
2. Find (M_j, M_k) such that $C_j=C_k$
3. Query $(M_j||P, M_k||P)$

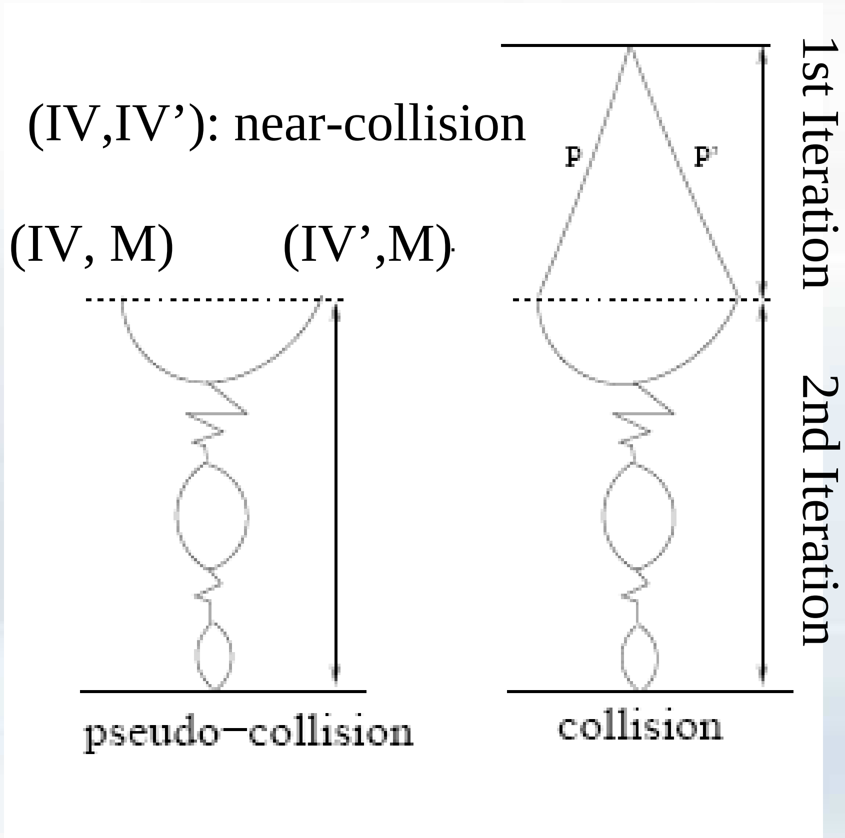
A General Attack on Iterated MACs (2)

- Distinguishing attack
  - If the MAC value of $M_i||P$ and $M_k||P$ collides, the MAC algorithm is an iterated MAC
  - Otherwise, is a random function.
- Convert to forgery attack directly:
  - Query the corresponding MAC of $M_i||P||P'$, denoted C, where P' is some non-empty string.
  - Obtain a valid MAC of new message $M_k||P||P'$

Distinguishing Attack on HMAC/NMAC-MD5

- To appear in Eurocrypt 09, Wang, Yu, Wang, Zhan: without related key
- Main idea: Collect messages and the corresponding MACs which guarantee inner DBB conditions hold

  DBB conditions: conditions of IV in a pseudo-collision given by den Boer and Bosselaers
- Allure a DBB-collision to occur by appending the same message (high probability 2^{-47} instead of 2^{-128})
- Detect the inner near-collisions



Distinguishing Attack on HMAC/NMAC-MD5

- The distinguishing attack can be utilized to recover a subkey for MD5-MAC
- MD5-MAC is MDx-MAC based on MD5, MDx-MAC was proposed by Preneel and van Oorschot

Distinguishing Attack on 61-Round LPMAC-SHA1

SHA-1 Algorithm

- Input: message $M = (m_0, \cdots, m_{15}), IV = a_0, b_0, c_0, d_0, e_0$
- For *j*=1,2,…,80

$$w_j = \begin{cases} m_j, & j = 0, \cdots, 15; \\ (w_{j-3} \oplus w_{j-8} \oplus w_{j-14} \oplus w_{j-16}) \lll 1, & j = 16, \cdots, 79. \end{cases}$$

$$\begin{aligned} a_j &= (a_{j-1} \lll 5) + f_j(b_{j-1}, c_{j-1}, d_{j-1}) + e_{j-1} + w_{j-1} + k_j, \\ b_j &= a_{j-1},\ c_j = b_{j-1} \lll 30,\ d_j = c_{j-1},\ e_j = d_{j-1}. \end{aligned}$$

- Output: $(a_0 + a_{80}, b_0 + b_{80}, c_0 + c_{80}, d_0 + d_{80}, e_0 + e_{80})$

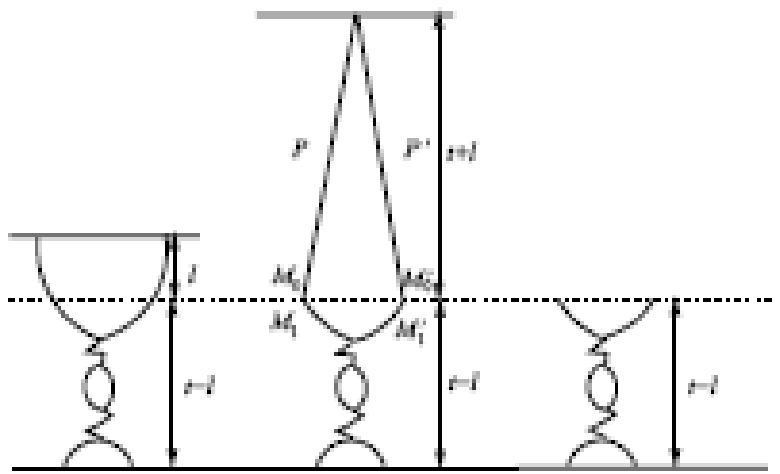
- Boolean functions and constants

j		f_j	k_j
1-20	IF	: $(x \wedge y) \vee (\neg x \wedge z)$	0x5a827999
21-40	XOR	: $x \oplus y \oplus z$	0x6ed6eba1
41-60	MAJ	: $(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$	0x8fabbcdc
61-80	XOR	: $x \oplus y \oplus z$	0xca62c1d6

Obstacles I

- SHA-1 hasn't any differential path with high probability, but the probability in the last three rounds is high

- How to avoid the differential path in the first round, and completely explore the probability advantage in the last three rounds



Near-Collision Path for 15-61 Steps SHA-1

Step i	D.V.	XOR Difference of the Input to Step i Δw_{i-1}	Δa_i	Δb_i	Δc_i	Δd_i	Δe_i	Conditions
14	2	2, 31, 32			32	2, 30	7	$a_{13,2}=1, a_{13,30}=0, a_{13,32}=1$ $a_{11,4}=w_{15,2}+1, a_{11,32}=w_{14,30}$ $a_{10,9}=w_{14,7}+1$
15	2	2, 7, 30, 31, 32	2			32	2,30	$a_{15,2}=w_{14,2}, a_{14,32}=1$
16	0	2, 7, 30, 31		2			32	$a_{14,4}=w_{14,2}+w_{16,2},$ $a_{13,4}=w_{14,2}+w_{16,2}+1$
17	0	2, 32			32			$a_{16,32}=0$
18	0					32		$a_{17,32}=1$
19	0						32	
20	0	32						
21	2	2	2					$a_{21,2}=w_{20,2}$
22	0	7		2				$a_{20,4}=a_{19,4}+w_{20,2}+w_{23,7}+1$
23	2		2		32			$a_{23,2}=w_{23,7}+1$
24	0	7,32		2		32		$a_{22,4}=a_{21,4}+w_{23,7}+w_{25,7}$
25	2	32	2		32		32	$a_{25,2}=w_{25,7}+1$
26	0	7		2		32		$a_{24,4}=a_{23,4}+w_{25,7}+w_{26,1}$
27	3	1, 32	1		32		32	$a_{27,1}=w_{26,1}+1$
28	0	6, 7		1		32		$a_{26,3}=a_{25,3}+w_{26,1}+w_{28,1}+1$
29	0	1, 2, 32			31		32	$a_{28,31}=a_{26,1}+w_{26,1}+w_{29,31}$
30	2	2,31	2			31		$a_{30,2}=w_{29,2},$ $a_{29,31}=a_{28,1}+w_{26,1}+w_{30,31}+1$
31	0	7,31, 32		2			31	$a_{29,4}=a_{28,4}+w_{29,2}+w_{31,2}+1$
32	0	2, 31, 32			32			

D.V.: Disturbance Vector

Sufficient Conditions on Message Words

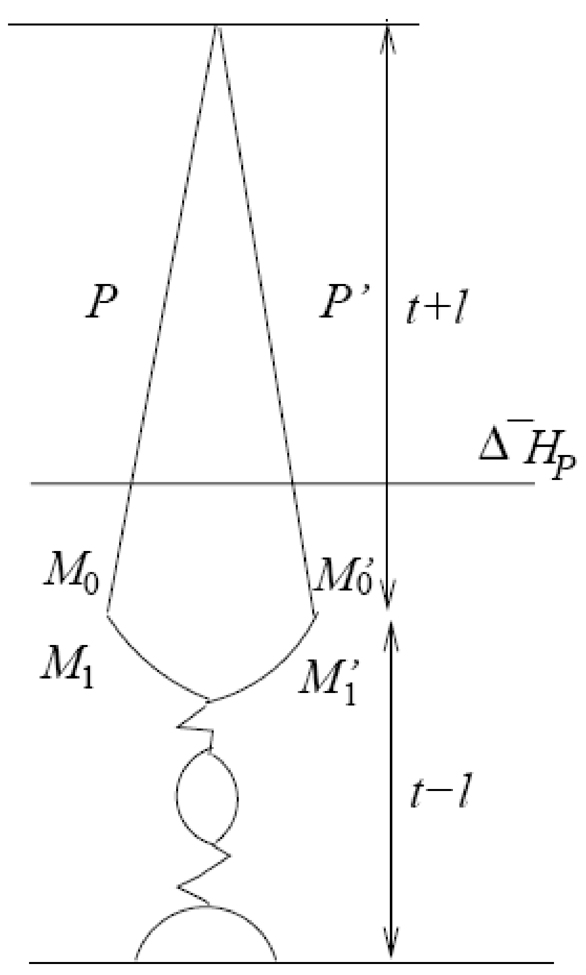
$w_{14,31} = w_{14,30} + 1, w_{15,7} = w_{14,2} + 1, w_{15,30} = w_{14,30}, w_{15,31} = w_{15,30} + 1$
$w_{21,7} = w_{20,2} + 1, w_{27,6} = w_{26,1} + 1, w_{27,7} = w_{26,1}, w_{28,2} = w_{28,1} + 1$
$w_{30,7} = w_{29,2} + 1, w_{31,31} = w_{26,1} + 1, w_{35,7} = w_{34,2} + 1, w_{41,7} = w_{40,2} + 1$
$w_{43,7} = w_{40,2} + 1, w_{44,2} = w_{40,2} + 1, w_{55,8} = w_{54,3} + 1, w_{56,3} = w_{54,3} + 1$
$w_{57,1} = w_{54,3} + 1, w_{58,1} = w_{54,3} + 1, w_{58,9} = w_{57,4} + 1, w_{59,1} = w_{54,3} + 1$
$w_{59,4} = w_{57,4} + 1, w_{59,8} = w_{58,3} + 1$

Obstacles II

- $\Delta^{-}H_P = H_P - H'_P$ is unknown

  output difference: $\Delta^{-}H_P + \Delta^{-}ch_{61}$

  Birthday attack can't be applied directly
- How to choose messages, and fulfill the birthday attack to detect the inner near-collision



Mathematical Properties of the Differential Path

- If the inner near-collision occurs, replace (M_1 , M_1') with another $(\overline{M}_1,\ \overline{M'_1})$

$$\Pr((P\|M_0\|\overline{M}_1,\ P'\|M'_0\|\overline{M'_1})\ \text{follows the DP}) = 2^{-34}.$$

DP: Differential path

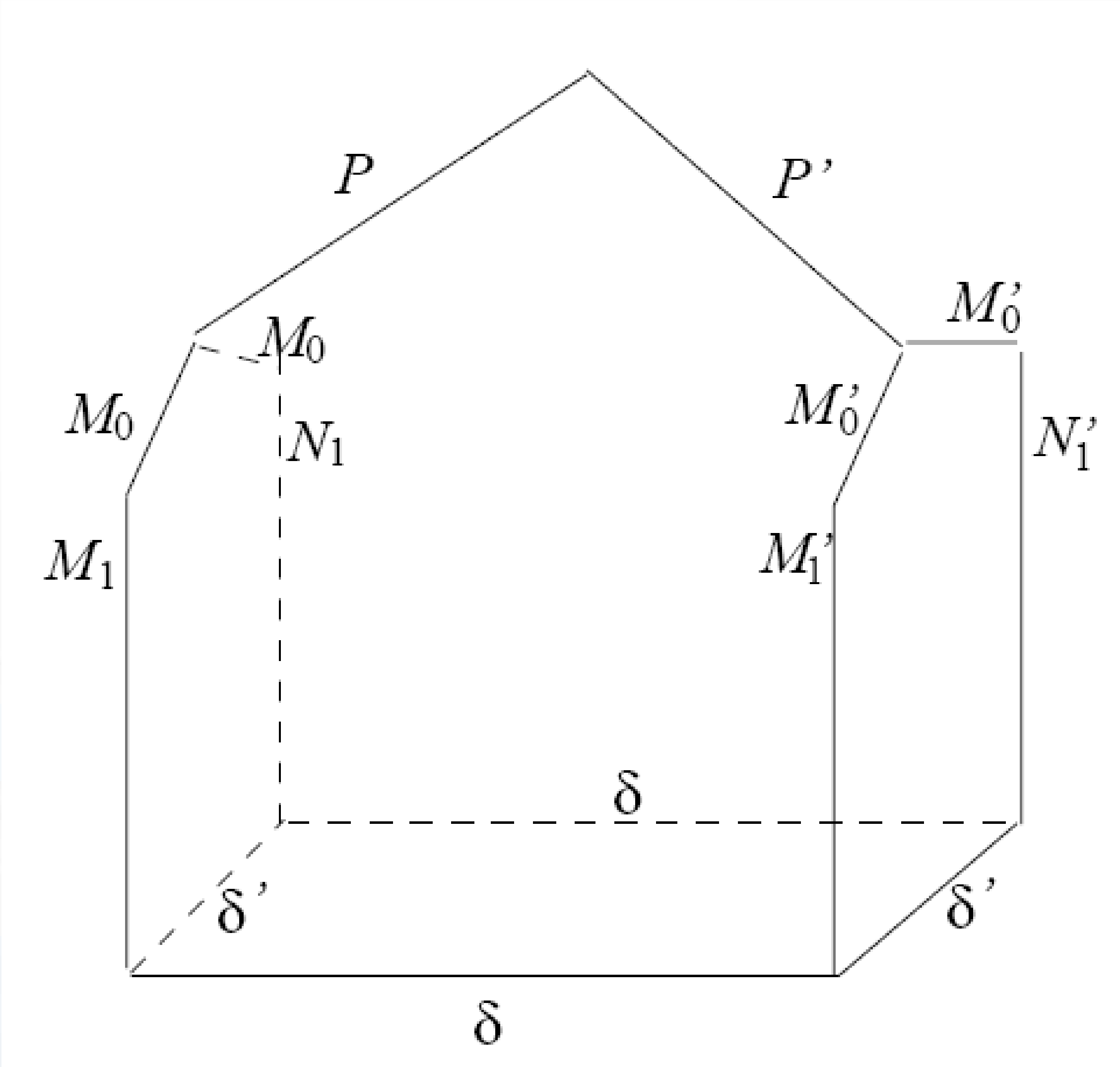
- If $(P\|M_0\|M_1, P'\|M'_0\|M'_1)$ and $(P\|M_0\|N_1, P'\|M'_0\|N'_1)$ s.t.,

$$\begin{aligned} & H_K(P\|M_0\|M_1) - H_K(P'\|M'_0\|M'_1) \\ = \ & H_K(P\|M_0\|N_1) - H_K(P'\|M'_0\|N'_1) \\ = \ & \Delta^- H_P + \Delta^- ch_{61} = \delta. \end{aligned}$$

$$\Rightarrow$$

$$\begin{aligned} & H_k(P\|M_0\|M_1) - H_K(P\|M_0\|N_1) \\ = \ & H_K(P'\|M'_0\|M'_1) - H_K(P'\|M'_0\|N'_1) = \delta'. \end{aligned}$$

Distinguisher



Distinguishing Attack Details

- (1) Randomly choose a structure S, which consists of $2^{84.5}$ different one-block messages
- (2) For all P in S, compute the following two structures of differences

$$S_1 = \{LPMAC(P||M_0||M_1) - LPMAC(P||M_0||N_1) \mid P \in S\},$$

$$S_2 = \{LPMAC(P||M_0'||M_1') - LPMAC(P||M_0'||N_1') \mid P \in S\}.$$

Search all the collisions between two structures by the birthday attack

Distinguishing Attack Details

- (3) For each collision, compute

  $$\delta = LPMAC(P\|M_0\|M_1) - LPMAC(P'\|M_0'\|M_1').$$

  Substitute M_1 and M1' with 2^{34} different $\overline{M_1}, \overline{M_1'}$ respectively, compare

  $$LPMAC(P\|M_0\|\overline{M_1}) - LPMAC(P'\|M_0'\|\overline{M_1'}) \text{ with } \delta.$$

  - If one match found, LPMAC is based on 61-step SHA-1.
  - Else, go to step 4.
- (4) Choose another structure *S*, and repeat steps (2)-(3)

  If the number of structures exceeds 2^{68}, then a random function

The complexity is about 2

Comparison with the Previous Distinguishing Attacks on MACs Based on SHA-1

	MAC	Steps	Data
Kim *et al.* [9]	HMAC	43	$2^{154.9}$
Rechberger *et al.* [13]	HMAC	50	$2^{153.5}$
This paper	LPMAC	43	$2^{124.5}$
		50	$2^{136.5}$
		61	$2^{154.5}$

Conclusions

Main Contribution

- This paper: distinguish an inner near-collision occur inside one iteration
  - To distinguish 61-round LPMAC-SHA1
- Previous distinguishing techniques
  - Distinguish an inner collision between iterations such that $(M_1 \| M_2)$, $(M_1' \| M_2)$, $H(K, M_1)$ and $H(K, M_1')$ is a collision

    Available to iterative MACs
  - Distinguish an inner near-collision between iterations such that $(M_1 \| M_2)$, $(M_1' \| M_2)$, $H(K, M_1)$ and $H(K, M_1')$ is a near-collision

    Available for some important specific iterative MACs

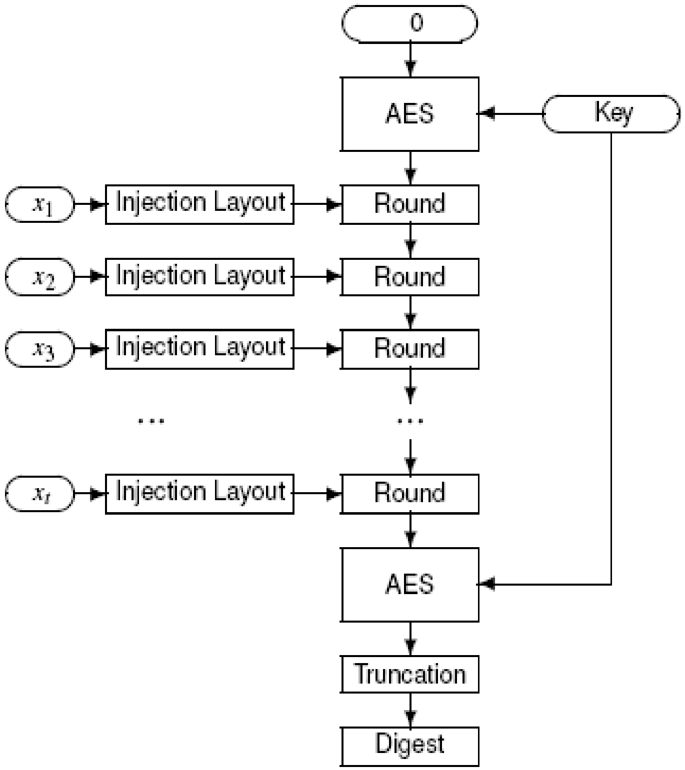
Further Research Results

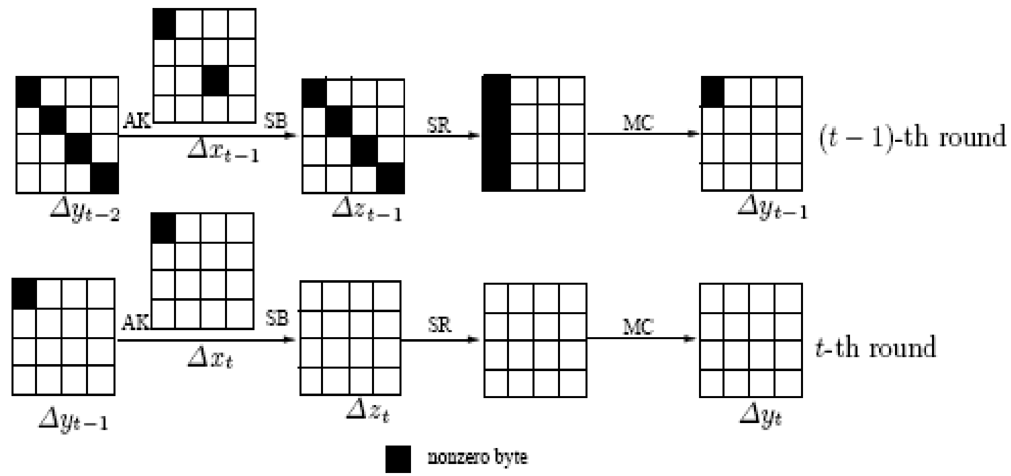
- Distinguish inner near-collisions or inner collisions with specific truncated differential path
  - To distinguish the instantiated MAC from a random function
  - To recover the subkey or equivalent subkey

Further Research Results

----Attack on ALPHA-MAC

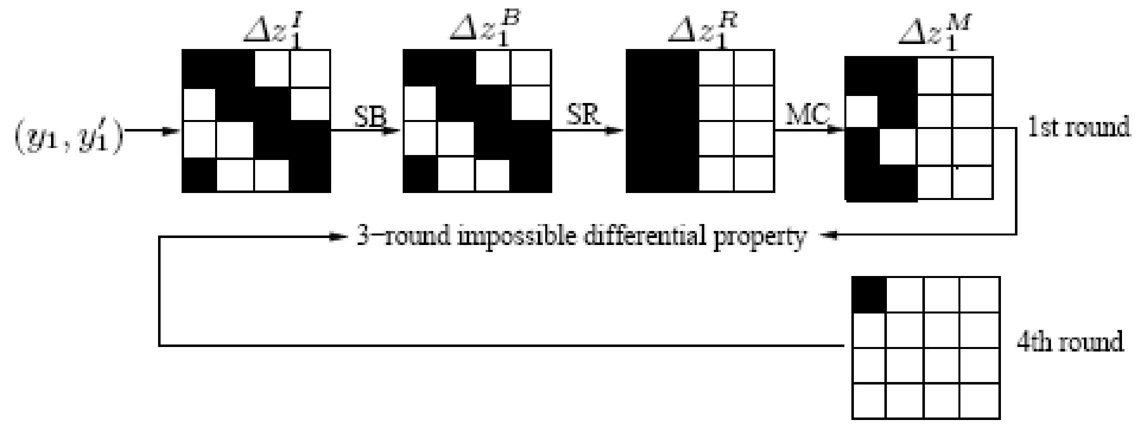
- A successful example: ALPHA (Alred MAC with AES operation), FSE 2005. Designers: Daemen and Rijmen
- Distinguish an inner collision with 2-round differential path
- Recover the inner state which is an equivalent subkey with $2^{65.5}$ computations

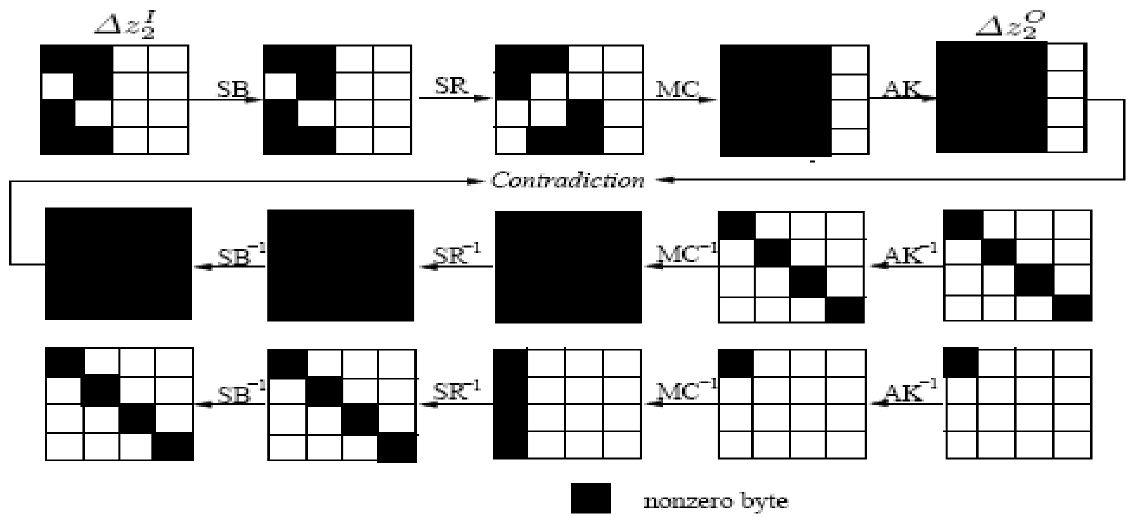




Further Research Results
---Attack on Pelican, MT-MAC and PC-MAC Based on 4-Round AES

- To distinguish an inner near-collision or inner collision with specific differential path
- Choose message pairs to allure an impossible differential path to occur under a wrong subkey





Related References

- Impossible Differential Cryptanalysis of Pelican, MT-MAC-AES and PC-MAC-AES, IACR ePrint
- Distinguishing and Second-Preimage Attack on CBC-like MACs, IACR ePrint
- Distinguishing and Forgery Attacks on Alred and Its AES-based Instance Alpha-MAC, IACR ePrint
- Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC, To appear in Eurocrypt 09

Thank You！