New Birthday Attacks on Some MACs Based on Block Ciphers

Wei Wang

Joint work with Zheng Yuan, Keting Jia, Guangwu Xu, and Xiaoyun Wang

Santa Barbara, USA

August 18, 2009

Outline

Outline

Main Results

Inner near-collision with some specific differences

- Part I – Distinguishing and forgery attack on ALRED and its AES-based instance ALPHA-MAC
  - Joint work with Zheng Yuan, Keting Jia, and Xiaoyun Wang
  - Distinguishing and forgery attack on ALRED construction
  - Internal state recovery attack on ALPHA-MAC
- Part II – Impossible differential cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES
  - Joint work with Xiaoyun Wang and Guangwu Xu
  - The first impossible differential attack on MACs
  - Recover the internal state of PELICAN, a subkey of MT-MAC-AES, and two 128-bit key of PC-MAC-AES

Outline

1. Main Results
2. Brief Introduction of MAC Algorithms
3. Related Works on Cryptanalysis of MACs
   - A General Forgery Attack on Iterated MACs
   - Distinguishing Attack on HMAC/NMAC-MD5
4. Our Works
   - Distinguishing and Forgery Attacks on ALRED and Its AES-based Instance ALPHA-MAC
   - Impossible Differential Cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES

Message Authentication Code

Secret Key $+$ Message $\Rightarrow$ $\boxed{\text{MAC Algorithm}}$ $\Rightarrow$ Tag (MAC)

Applications:

- Guarantee data integrity and data origin authentication
- Internet security: IPsec, SSL, SSH, SNMP, etc
- Finance: Banking, electronic purses, etc

Constructions:

- Based on hash function with secret key, e.g., HMAC
- Based on block cipher, e.g., CBC-MAC
  Block cipher and reduced block cipher, e.g., PELICAN
- Based on universal hash function, e.g., Wegman-Carter MAC

MAC Security

Three kinds of attacks:

- Distinguishing Attack
  - Distinguishing-R Attack
  - Distinguishing-H Attack
- Forgery Attack
  - Existential Forgery:
    For a new message M, compute a valid MAC
  - Selective Forgery: The adversary can select a message M, and compute $M' \neq M$ with $MAC_K(M') = MAC_K(M)$
  - Universal Forgery:
    For any given message M, compute a valid MAC
- Key Recovery Attack

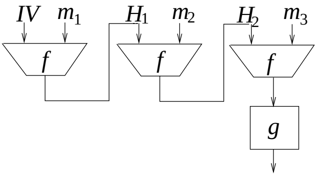
Outline

1. Main Results
2. Brief Introduction of MAC Algorithms
3. Related Works on Cryptanalysis of MACs
   - A General Forgery Attack on Iterated MACs
   - Distinguishing Attack on HMAC/NMAC-MD5
4. Our Works
   - Distinguishing and Forgery Attacks on ALRED and Its AES-based Instance ALPHA-MAC
   - Impossible Differential Cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES

Outline

A General Forgery Attack on Iterated MACs

- Preneel and van Oorschot, Crypto'95
- Applicable to all iterated MACs based on both hash functions and block ciphers



f: compression function

g: output transformation

$\boxed{\text{Detect the } \textit{internal collision}}$

$2^{(n+1)/2}$ randomly chosen M_i

$\xrightarrow{\text{birthday paradox}} \exists(M_j, M_k)$ collide

$\xrightarrow{\text{query with}(M_j||P, M_k||P)}$ if still collide

$\rightarrow$ internal collision

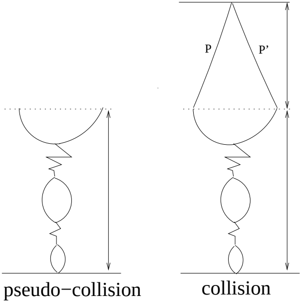
$\rightarrow MAC_K(M_j||Q) = MAC_K(M_k||Q)$

Outline

Distinguishing Attack on HMAC/NMAC-MD5

- Wang et al., EuroCrypt'09
- The first attack on HMAC/NMAC-MD5 without related-key

Detect the *inner near-collision with some specific difference*



- A pseudo-collision differential path of MD5 with prob. 2^{-46} (den Boer and Bosselaers, EuroCrypt'93)
- 2^{66} randomly chosen P $\xrightarrow{\text{birthday paradox}} \exists(P, P')$ s. t. conditions on IV
- $\xrightarrow{\text{Query with } 2^{47} (P||M, P'||M)}$ internal collision/ dBB collision/ others
- Partial key recovery attack on MD5-MAC

Outline

Outline

1. Main Results
2. Brief Introduction of MAC Algorithms
3. Related Works on Cryptanalysis of MACs
   - A General Forgery Attack on Iterated MACs
   - Distinguishing Attack on HMAC/NMAC-MD5
4. Our Works
   - Distinguishing and Forgery Attacks on ALRED and Its AES-based Instance ALPHA-MAC
   - Impossible Differential Cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES

ALRED Construction

- Daemen and Rijmen, FSE 2005
- For a message $M = (x_1, x_2, \cdots, x_t)$,
  1. Apply the block cipher to the state of all-zero block

     $$y_0 = \mathrm{Enc}_k(0)$$

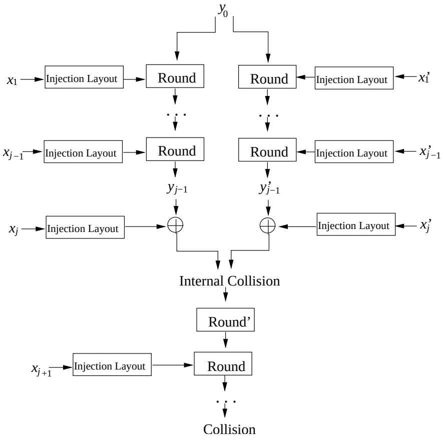
  2. Perform an iteration for each message word x_i

     $$y_i = \mathrm{ReducedEnc}_{x_i}(y_{i-1}), i = 1, 2, \cdots, t$$

  3. Apply the block cipher to the state again, and truncate the first l_m bits of the state as the output

     $$C = Trunc(\mathrm{Enc}_k(y_t))$$

Distinguishing Attack on ALRED Construction



- $2^{(n+1)/2}$ randomly chosen M^i
- $\xrightarrow{\text{birthday paradox}}$

  $\exists(M^a, M^b)$ collide, where

  $$M^a = (x_1^a, \ldots, x_j^a, x_{j+1}, \ldots, x_t)$$
  $$M^b = (x_1^b, \ldots, x_j^b, x_{j+1}, \ldots, x_t)$$
- Query the MAC with

  $$\overline{M^a} = (x_1^a, \ldots, x_{j-1}^a, \overline{x_j^a}),$$
  $$\overline{M^b} = (x_1^b, \ldots, x_{j-1}^b, \overline{x_j^b}),$$

  where $\overline{x_j^a} \oplus \overline{x_j^b} = x_j^a \oplus x_j^b$

  – Collide $\Rightarrow$ ALRED

  – Else $\Rightarrow$ a random function

$\boxed{\text{Collision} \Rightarrow \text{Internal collision} \Rightarrow \textit{Inner near-collision with } \Delta y_{j-1} = \Delta InLayout(x_j)}$

Forgery Attack on ALRED Construction

Obtain a colliding pair (M^a, M^b), where $M^a = (x_1^a, \ldots, x_{j-1}^a, x_j^a)$, $M^b = (x_1^b, \ldots, x_{j-1}^b, x_j^b)$, and $\Delta y_{j-1} = \Delta x_j$

1. Query the MAC oracle with $\widetilde{M^a} = (x_1^a, \ldots, x_{j-1}^a, \widetilde{x_j^a}, s)$, where s is an arbitrary message string
2. Get the forgery of $\widetilde{M^b} = (x_1^b, \ldots, x_{j-1}^b, \widetilde{x_j^a} \oplus \Delta x_j, s)$

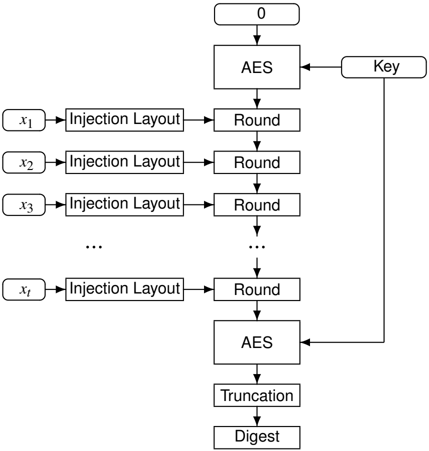
Work for:
MACs based on block ciphers, e.g., CBC-MAC, OMAC, TMAC

$$H_i = f(H_{i-1}, x_i) = E_k(H_{i-1} \oplus x_i)$$

MACs based on CFB mode

$$H_i = f(H_{i-1}, x_i) = E_k(H_{i-1}) \oplus x_i$$

AES Based Instance: ALPHA-MAC



- Round: 1-round AES
- Round: AK, SB, SR, MC
- Message $M = (x_1, x_2, \cdots, x_t)$
- 32-bit word $x_i = (x_{i,0}, x_{i,1}, x_{i,2}, x_{i,3})$
- Injection layout $(x_i) =$

$$\begin{pmatrix} x_{i,0} & 0 & x_{i,1} & 0 \\ 0 & 0 & 0 & 0 \\ x_{i,2} & 0 & x_{i,3} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Other Attacks on ALPHA-MAC

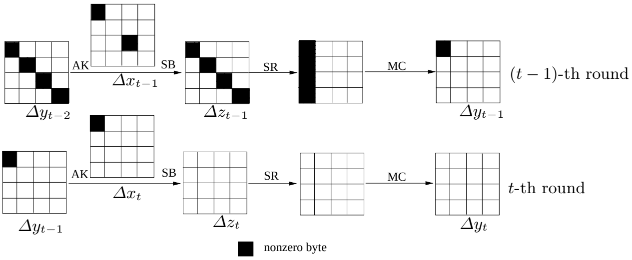
- Huang et al. exploited the algebraic properties of the AES, and applied a selective forgery attack on ALPHA-MAC, on the assumption that a key or an internal state is known
- Biryukov et al. proposed a side-channel collision attack on ALPHA-MAC recovering its internal state, and mounted a selective forgery attack

All forgery attacks are based on the recovery of the internal state

Distinguishing Attack on ALPHA-MAC I

Fact

Given two messages $M = (x_1, \ldots, x_{t-1}, x_t)$ and $M' = (x'_1, \ldots, x'_{t-1}, x'_t)$ following the 2-round collision differential path, there exists an algorithm to find another message pair $\overline{M} = (x_1, \ldots, \overline{x_{t-1}}, x_t)$ and $\overline{M'} = (x'_1, \ldots, \overline{x'_{t-1}}, x'_t)$ satisfying the 2-round collision differential path with 2^9 queries and 2^9 chosen messages.



Distinguishing Attack on ALPHA-MAC II

1. Construct two structures, each with $2^{65.5}$ texts, where $\Delta x_{t-1}, \Delta x_t$ as shown in the above collision path
   $T_1 = \{M^a = (x_1^a, x_2^a, \ldots, x_{t-1}^a, x_t)\}$
   $T_2 = \{M^b = (x_1^b, x_2^b, \ldots, x_{t-1}^b, x_t \oplus (\eta, 0, 0, 0))\}$
2. Search for (M^a, M^b) s.t. $C^a = C^b$ by the birthday attack. Query the MAC with the new message pair $(\overline{M^a}, \overline{M^b})$, where $\overline{M^a} = (x_1^a, \ldots, x_{t-1}^a, \overline{x_t^a})$, $\overline{M^b} = (x_1^b, \ldots, x_{t-1}^b, \overline{x_t^b})$, $\Delta \overline{x_t} = \Delta x_t$
   - If collide $\Rightarrow$ ALRED-MAC, and goto step 3
   - Otherwise $\Rightarrow$ a random function
3. Randomly choose 2^8 different $(\overline{x_{t-1,0}^a}, \overline{x_{t-1,0}^b})$ to replace $(x_{t-1,0}^a, x_{t-1,0}^b)$. Query the MACs of the new messages.
   - If a collision appears $\Rightarrow$ ALPHA-MAC
   - Otherwise $\Rightarrow$ a random function

Internal State Recovery of ALPHA-MAC I

$$y_0 \longleftarrow \cdots \longleftarrow \Delta y_{t-3} = \begin{pmatrix} * & ? & * & ? \\ ? & * & ? & * \\ * & ? & * & ? \\ ? & * & ? & * \end{pmatrix}$$

$$\xleftarrow{AK^{-1}\ SB^{-1}} \Delta z_{t-2} = \begin{pmatrix} * & ? & * & ? \\ ? & * & ? & * \\ * & ? & * & ? \\ ? & * & ? & * \end{pmatrix} \xleftarrow{SR^{-1}} \begin{pmatrix} * & ? & * & ? \\ * & ? & * & ? \\ * & ? & * & ? \\ * & ? & * & ? \end{pmatrix} \xleftarrow{MC^{-1}} \Delta y_{t-2} = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & ? \end{pmatrix}$$

$$\xleftarrow{AK^{-1}\ SB^{-1}} \Delta z_{t-1} = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{pmatrix} \xleftarrow{SR^{-1}} \begin{pmatrix} * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ * & 0 & 0 & 0 \end{pmatrix} \xleftarrow{MC^{-1}} \Delta y_{t-1} = \begin{pmatrix} \Delta x_{t,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Internal State Recovery of ALPHA-MAC II

1. Recover $(y^a_{t-2,0}, y^b_{t-2,0}, y^a_{t-2,10}, y^b_{t-2,10})$

$$\begin{cases} S(y^a_{t-2,0} \oplus x^a_{t-1,0}) \oplus S(y^b_{t-2,0} \oplus \underline{x^b_{t-1,0}}) = \Delta z_{t-1,0} \\ S(y^a_{t-2,0} \oplus \overline{x^a_{t-1,0}}) \oplus S(y^b_{t-2,0} \oplus x^b_{t-1,0}) = \Delta z_{t-1,0} \end{cases}$$

$$\Delta y_{t-2} = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & ? & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & ? \end{pmatrix} \xleftarrow{AK^{-1}\ SB^{-1}}$$

$$\Delta z_{t-1} = \begin{pmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{pmatrix} \xleftarrow{SR^{-1}\ MC^{-1}} \Delta y_{t-1} = \begin{pmatrix} \Delta x_{t,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Internal State Recovery of ALPHA-MAC III

2. Recover $(y^a_{t-3,0}, y^b_{t-3,0}, y^a_{t-3,2}, y^b_{t-3,2}, y^a_{t-3,8}, y^b_{t-3,8}, y^a_{t-3,10}, y^b_{t-3,10})$
   Making use of the property of the S-box and MixColumn
   $S(y_{t-3,0} \oplus x^a_{t-2,0}) \oplus S(y'_{t-3,0} \oplus x^b_{t-2,0}) = \Delta z_{t-2,0} \Rightarrow 2^8$ candidates
   –Right candidate can lead to a collision with prob. 2^{-8}
   –Wrong candidates produce a collision with prob. $\leq 2^{-16}$

$$\Delta y_{t-3} = \begin{pmatrix} \boxed{*} & ? & * & ? \\ ? & * & ? & * \\ * & ? & * & ? \\ ? & * & ? & * \end{pmatrix}$$

$$\xleftarrow{AK^{-1}\ SB^{-1}} \Delta z_{t-2} = \begin{pmatrix} \boxed{*} & ? & * & ? \\ ? & * & ? & * \\ * & ? & * & ? \\ ? & * & ? & * \end{pmatrix} \xleftarrow{SR^{-1}\ MC^{-1}} \Delta y_{t-2} = \begin{pmatrix} \boxed{*} & 0 & 0 & 0 \\ \boxed{0} & ? & 0 & 0 \\ \boxed{0} & 0 & * & 0 \\ \boxed{0} & 0 & 0 & ? \end{pmatrix}$$

$$\xleftarrow{AK^{-1}\ SB^{-1}} \Delta z_{t-1} = \begin{pmatrix} \boxed{*} & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{pmatrix} \xleftarrow{SR^{-1}\ MC^{-1}} \Delta y_{t-1} = \begin{pmatrix} \Delta x_{t,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Internal State Recovery of ALPHA-MAC IV

3. Recover $(y^a_{t-3,5}, y^b_{t-3,5}, y^a_{t-3,7}, y^b_{t-3,7}, y^a_{t-3,13}, y^b_{t-3,13}, y^a_{t-3,15}, y^b_{t-3,15})$
Making use of some equations derived from the difference and recovered bytes, such as

$$\begin{cases} \Delta z_{t-2,5} & = & S(y^a_{t-3,5}) \oplus S(y^b_{t-3,5}) \\ \Delta z_{t-2,15} & = & S(y^a_{t-3,15}) \oplus S(y^b_{t-3,15}) \\ y^a_{t-2,0} & = & 3S(y^a_{t-3,0} \oplus x^a_{t-2,0}) \oplus 2S(y^a_{t-3,5}) \oplus S(y^a_{t-3,10} \oplus x^a_{t-2,3}) \oplus S(y^a_{t-3,15}) \\ y^b_{t-2,0} & = & 3S(y^b_{t-3,0} \oplus x^b_{t-2,0}) \oplus 2S(y^b_{t-3,5}) \oplus S(y^b_{t-3,10} \oplus x^b_{t-2,3}) \oplus S(y^b_{t-3,15}) \end{cases}$$

$$\Delta y_{t-3} = \begin{pmatrix} * & ? & * & ? \\ ? & \boxed{*} & ? & * \\ * & ? & * & ? \\ ? & * & ? & \boxed{*} \end{pmatrix} \xleftarrow{AK^{-1}\ SB^{-1}} \Delta z_{t-2} = \begin{pmatrix} * & ? & * & ? \\ ? & \boxed{*} & ? & * \\ * & ? & * & ? \\ ? & * & ? & \boxed{*} \end{pmatrix}$$

Internal State Recovery of ALPHA-MAC V

4. Recover the internal state y_0
Guess all the 2^{64} possibilities of the rest 8 bytes of y^a_{t-3}

$$y_{t-3} \xrightarrow[decryption]{(x_{t-3},\cdots,x_1)} y_0 \xrightarrow[encryption]{(x'_1,\cdots,x'_{t-3})} \overline{y_{t-3}} \begin{cases} \overline{y_{t-3}} = y'_{t-3} & \text{right} \\ \text{Else} & \text{wrong} \end{cases}$$

Second Preimages for ALPHA-MAC

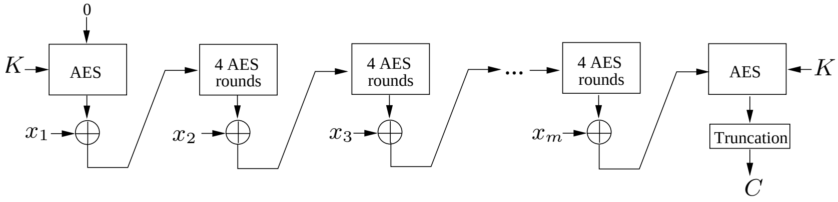
Once the internal state y_0 is recovered, the selective forgery attacks can be performed by Huang et al. or Biryukov et al.'s attacks

Outline

1. Main Results
2. Brief Introduction of MAC Algorithms
3. Related Works on Cryptanalysis of MACs
   - A General Forgery Attack on Iterated MACs
   - Distinguishing Attack on HMAC/NMAC-MD5
4. Our Works
   - Distinguishing and Forgery Attacks on ALRED and Its AES-based Instance ALPHA-MAC
   - Impossible Differential Cryptanalysis of PELICAN, MT-MAC-AES and PC-MAC-AES

PELICAN Algorithm I

- Daemen and Rijmen, 2005
- An optimized version of ALPHA-MAC
- x_i: 128-bit message word
- Round function: 4-round AES with round subkeys set to 0



PELICAN Algorithm II

For a message $M = (x_1, x_2, \ldots, x_b)$

1. Initialization: $y_0 = E_K(0)$
   where E is the AES, and K is the secret key

2. Chaining:
   1. $y_1 = y_0 \oplus x_1$
   2. For each message word x_i $(i = 2, \ldots, b)$, perform an iteration operation:
      $y_i = f(y_{i-1}) \oplus x_i$
      where f consists of 4-round AES with the round subkeys set to 0

3. Finalization: $C = Trunc(E_K(y_m))$

Main Idea of the Impossible Differential Cryptanalysis

1. Find an impossible differential path
   - For AES, several 4-round impossible differential paths have been found in literature
   - For PELICAN, a 3-round impossible differential path is OK
2. Collect and sieve the message pairs with the required differences (obstacle)
   - For block ciphers, sieve directly
   - For PELICAN, need new techniques

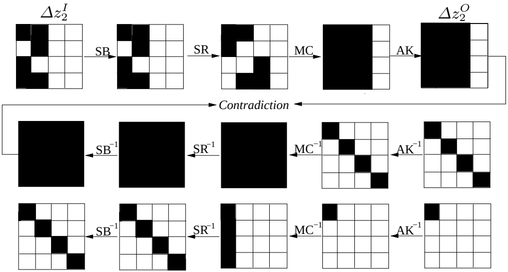
$$\boxed{\text{AES}} \rightarrow \boxed{4\text{-}r\ \text{AES}} \rightarrow \cdots \rightarrow \boxed{4\text{-}r\ \text{AES}} \rightarrow \boxed{\text{AES}}$$

3. For each sieved pair, discard the wrong subkeys (or internal state), and only the correct one is left

Three-Round Impossible Differential of AES

Proposition

For 3-round AES, given an input pair $(z_2^I, z_2^{I\prime})$ *whose bytes equal in all except six indexed by* $(0,1,5,8,12,13)$ *(or* $(0,1,4,5,9,12)$*,* $(0,4,5,8,9,13)$*,* $(1,4,8,9,12,13)$*), the difference of the output pair* $(z_4^O, z_4^{O\prime})$ *can not have exactly one nonzero byte.*

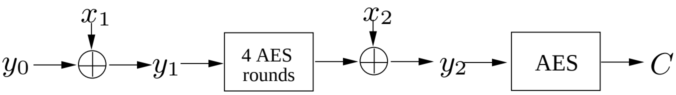


Sieve Useful Message Pairs

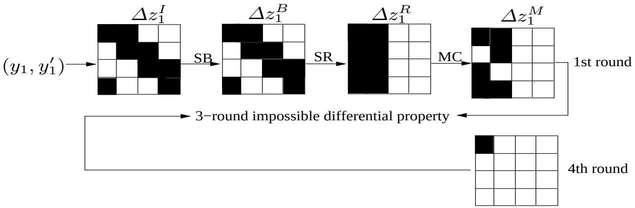
$\boxed{\text{Detect the inner near-collision with specific difference}}$

PELICAN algorithm with two message words:

$$
\begin{aligned}
\text{External Collision} \quad &\Rightarrow \quad \text{Internal Collision} \\
& \qquad AES^{4r}(y_1) \oplus x_2 = AES^{4r}(y'_1) \oplus x'_2 \\
&\Rightarrow \quad \text{Output difference of 4-round AES} \\
& \qquad AES^{4r}(y_1) \oplus AES^{4r}(y'_1) = {\color{red} x_2 \oplus x'_2}
\end{aligned}
$$



Impossible Differential Cryptanalysis of PELICAN I



Message Pairs Collection Phase

1. Construct two structures, each with 2^{64} messages $S_1 = \{(x_1, x_2)\},\ S_2 = \{(x_1', x_2')\}$ where Δx_1 is zero at bytes $(2, 3, 4, 7, 8, 9, 13, 14)$, and Δx_2 has only one nonzero byte

Impossible Differential Cryptanalysis of PELICAN II

2. Query MAC and search collisions between the two structures by the birthday attack
   - If there is no truncation at the final output, this means an inner collision at y_2
   - Else, for all colliding pairs $(x_1\|x_2, x_1'\|x_2')$, query the MAC on $(x_1\|x_2', x_1'\|x_2)$. If still collide, $(x_1\|x_2, x_1'\|x_2')$ must collide at y_2

Internal State Recovery Phase

- Recover 8 bytes of y_0 at position $(0, 1, 5, 6, 10, 11, 12, 15)$ by exhaustive search directly
- Recover the other 8 bytes in a similar manner

Selective Forgery Attack

$\boxed{\text{Recover } y_0 \Rightarrow \text{Control of the internal states}}$

Select $M = (x_1, x_2, \cdots, x_b)$, and obtain the MAC value C
Compute $M' = (x'_1, x'_2, \cdots, x'_b)$ with the same C:

1. Randomly choose x'_1, where $x'_1 \neq x_1$
2. Compute $y_1 = x_1 \oplus y_0, y'_1 = x'_1 \oplus y_0, AES^{4r}(y_1)$ and $AES^{4r}(y'_1)$
3. Set $x'_2 = AES^{4r}(y_1) \oplus AES^{4r}(y'_1) \oplus x_2$, then

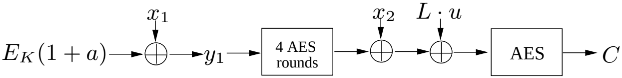
$$y'_2 = AES^{4r}(y'_1) \oplus x'_2 = AES^{4r}(y_1) \oplus x_2 = y_2$$

Set $x'_3 = x_3, \cdots, x'_b = x_b$

Obviously, $MAC_K(M') = C = MAC_K(M)$

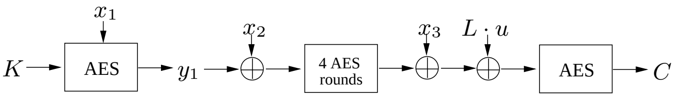
Impossible Differential Cryptanalysis of MT-MAC-AES

- MT-MAC: designed by Minematsu and Tsunoom, FSE'06
- MT-MAC-AES: MT-MAC instantiated with AES and simplified 4-round AES
- Adopt the above attack on PELICAN directly
- Recover the subkey $E_K(1+a)$
- Complexity: $2^{85.5}$ chosen messages and $2^{85.5}$ queries



Impossible Differential Cryptanalysis of PC-MAC-AES

- PC-MAC: designed by Minematsu and Tsunoom, FSE'06
- PC-MAC-AES: PC-MAC instantiated with AES and simplified 4-round AES
- Recover the internal state y_1
- Recover the two 128-bit secret key (K, L) by exhaustive search, respectively
- Complexity: $2^{85.5}$ chosen messages and 2^{128} queries



Thank you very much!