A Flaw in The Internal State Recovery Attack on ALPHA-MAC

Shengbao Wu[1,2], Mingsheng Wang[1], and Zheng Yuan[3]

1. State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PO Box 8718, China
2. Graduate School of Chinese Academy of Sciences, Beijing 100190, China
3. Beijing Electronic Science and Technology Institute, Beijing 100070, China
wushengbao@is.iscas.ac.cn
mingsheng_wang@yahoo.com.cn

Abstract. An distinguisher was constructed by utilizing a 2-round collision differential path of ALPHA-MAC, with about $2^{65.5}$ chosen messages and $2^{65.5}$ queries. Then, this distinguisher was used to recover the internal state([1],[2]). However, a flaw is found in the internal state recovery attack. The complexity of recovering the internal state is up to 2^{81} exhaustive search. And the complexity of the whole attack will be up to 2^{67} chosen messages and 2^{81} exhaustive search. To repair the flaw, a modified 2-round differential path of ALPHA-MAC is present and a new distinguisher based on this path is proposed. Finally, an attack with about $2^{65.5}$ chosen messages and $2^{65.5}$ queries is obtained under the new distinguisher.

1 Introduction

A message authentication code (MAC) algorithm accepts a secret key and a variable-length message as input, and outputs a fixed-length authenticator called MAC, which protects both the message's data integrity and its authenticity. MAC algorithms play a important role in network and security protocols(SNMP, SSH, SSL/TLS, IPsec), and various approaches have been proposed to construct them, for example, MAA([8]), UMAC([9]), OMAC([10]), TMAC([11]), CBC-MAC([12]), HMAC/NMAC([13]), MDx-MAC([14]), etc.

ALRED, proposed by Daemen and Rijmen in FSE 2005, is a MAC construction based on an iterated block cipher([3]). A specific instance of ALRED is ALPHA-MAC, which uses AES as underlying block cipher.

In [3],the authors have proved that the ALRED construction has the same security as the underlying block cipher with respect to the key recovery attacks and any forgery attacks not involving inner collisions. Moreover, a result has shown that, for two messages, a collision could only occur after 5 message blocks in ALPHA-MAC.

A series of work has been done to analyse the ALRED construction,for example,[1],[2],[4],[5]. For ALPHA-MAC,Huang et al. provided a method to find second preimages based on the assumption that a key or an internal state is known. Under the same assumption, the idea could be used to find internal collisions ([4]). Biryukov et al. proposed a side-channel collision attack on ALPHA-MAC and mounted a selective forgery attack after the internal state had been recovered([5]).

In [2] and Part I of [1], firstly, based on the birthday paradox, novel distinguishing attacks on the ALERD construction and ALPHA-MAC with success probability 0.63 are presented, and they can directly lead to forgery attacks. The distinguisher of attacking the ALPHA-MAC is constructed under a 2-round collision differential path of it, with about $2^{65.5}$ MAC queries and $2^{65.5}$ chosen messages. Then, this distinguisher is used to recover the internal state y_0, which is an equivalent subkey. According to the approach of the recovery attack, firstly, when a message pair (M^a, M^b) which follows the 2-round collision differential path is obtained, it can recover 8 bytes of y^a_{t-3} and 8 bytes of y^b_{t-3} respectively (The states when collision occur at t-th iteration are y^a_t and y^b_t, which means that y^a_t is equal to y^b_t. y^a_{t-3} and y^b_{t-3} represent the third state before collision). And then, since only 8 bytes of y^a_{t-3} is unknown, all 2^{64} possible internal states of y_0 can be recovered by searching all the 2^{64} possible values of y^a_{t-3} and taking the corresponding part of inverse input message M^a as decryption subkey. Finally, for each y_0, compute the corresponding y^b_{t-3} with M^b to filter the wrong guesses. The complexity of recovery attack on y_0 is at most 2^{65} exhaustive search since two pair of collision messages can ensure the right y_0. The whole attack's complexity is the same as the distinguishing attack, whose time complexity and data complexity are both $2^{65.5}$.

However,we find a flaw in the first step of recovering the internal state,when the recovery attack attempts to recover 8 bytes of state y^a_{t-3} and 8 bytes of state y^b_{t-3}. The result is that we can only recover 6 bytes of y^a_{t-3} and 6 bytes of y^b_{t-3} and we should guess 10 bytes of y^a_{t-3} to recover y_0. The whole attack's time complexity is now dominated by the exhaustive search, which is 2^{81} at least.

This paper is organized as follows: Section 2, some notations and a brief introduction of ALRED and ALPHA-MAC is given. We also introduce a lemma about computing the expected number of collisions between two sets. In section 3, we point out the flaw in the recovery attack on internal state y_0 in [1],[2] and analyse the complexity of whole attack. In section 4, a modified 2-round differential path is introduced and an attack with complexity of about $2^{65.5}$ chosen messages and $2^{65.5}$ queries is obtained. Finally, we conclude this paper in section 5.

2 Notations And Backgrounds

Some notations are defined and a brief introduction of ALRED and ALPHA-MAC is given in this section firstly. Then, a lemma of computing the expected number of collisions between two sets is introduced.

2.1 Notations

$\mathbf{S}$,$\mathbf{S}^{-1}$	the S-box and inverse S-box of AES.
$\mathbf{T}$,$\mathbf{T}^{-1}$	the matrix of MixColumns transformation and its inverse matrix.
$Rank(\mathbf{A})$	the rank of matrix $\mathbf{A}$
$\triangle X$	the XOR differential X and $X^{'}$
M	a message has the form $M = (x_1, x_2, ...x_t)$
x_i	the i-th message word
y_i	the state after the i-th iteration
z_i	the intermediate state after the Subbytes in i-th iteration

The state in ALPHA-MAC is exhibited as a 4×4 two dimensional array of bytes indexed as:

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

The symbol $y_{i,j}$ presents the j-th byte in the state after the i-th iteration. And the symbol $z_{i,j}$ has similar meaning.

2.2 A Brief Introduction of ALRED And ALPHA-MAC

The MAC construction ALRED is depicted in Fig.1. Its construction is based on an iterated block cipher. The length of key K equals to that of the underlying block cipher. The length of message is a multiple of l_w bits.

For a given message $M = (x_1, x_2, ..., x_t)$, a tag can be computed by executing the following steps in-order.

- Initialization:An all-zero block is adopted as the initial state and the block cipher is applied to it,i.e.,$y_0 = Enc_K(0)$.
- Chaining:For every message word x_i, firstly, maps the bits of the message word to an injection input that has the same dimensions as a sequence of r-round subkeys of the block cipher. Then, a sequence of r-round block cipher function is applied to the state, with the round subkeys replaced by the injection input,i.e.,$y_i = f(y_{i-1,x_i})$,for $i = 1, 2, ..., t$.
- Final transformation:The full block cipher is applied to the state y_t, and the MAC tag is the first l_m bits of the final state,i.e.,$Tag = Trunc(Enc_K(y_t))$.

By using AES as the underlying block cipher and 1-round AES as the iteration function, a specific instance of ALRED named ALPHA-MAC is obtained. Similar to AES, the ALPHA-MAC supports key length of 16, 24 and 32 bytes. The message word length is 4 bytes and the padding method is to append a single 1 followed by the minimum number of 0 bits such that the result is a multiple of 32.

The injection layout places the 4 bytes of each message word $x_i = (x_{i,0}, x_{i,1}, x_{i,2}, x_{i,3})$ into a 4×4 array as follows

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

Since the length of round key in AES is 16 bytes and can be represented in a 4×4 array, the result of injection layout can be adopted as the corresponding 128-bit round key. Like AES, the ALPHA-MAC round function contains four consecutive transformations: AddRoundKey(AK),SubBytes(SB),ShiftRows(SR),and MixColumns(MC).

In this article, $AK^{-1}, SB^{-1}, SR^{-1}$ and MC^{-1} are used to represent the inverse process of AK, SB, SR and MC respectively.

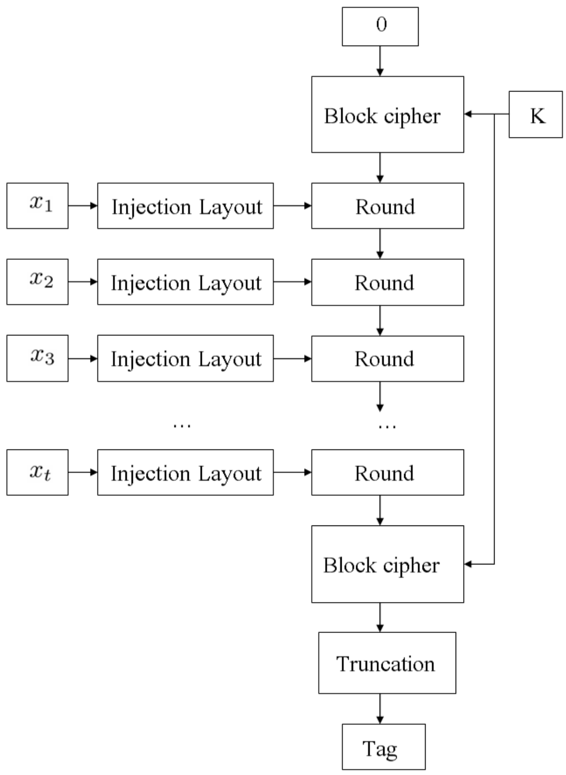

Fig.1. ALRED construction

2.3 Collision Between Two Sets

Given two subsets $\mathbf{N_1}$ and $\mathbf{N_2}$, each obtained by selecting elements at random from a large set $\mathbf{N}$. And both of them have no inner collisions, i.e., there is not exist two elements a and b in $\mathbf{N_1}$(or $\mathbf{N_2}$) satisfy $a = b$. We have

Lemma 1. *([7]) The expected number of collision between* $\mathbf{N_1}$ *and* $\mathbf{N_2}$ *is:* $\frac{n_1 \times n_2}{n}$*, where* n_1*,* n_2 *and* n *represented the number of elements of* $\mathbf{N_1}$*,* $\mathbf{N_2}$ *and* $\mathbf{N}$ *respectively.*

3 The Flaw in the Original Article

Utilizing the 2-round differential path and Fact 1 to Fact 3 in article [1], $(y^a_{t-2,0}, y^b_{t-2,0}, y^a_{t-2,10}, y^b_{t-2,10})$ can be recovered. Use these results and Fact 4 in [1], $(y^a_{t-3,0}, y^b_{t-3,0}, y^a_{t-3,2}, y^b_{t-3,2}, y^a_{t-3,8}, y^b_{t-3,8}, y^a_{t-3,10}, y^b_{t-3,10})$ can be recovered. Then, the correct $(y^a_{t-3,5}, y^a_{t-3,15}, y^b_{t-3,5}, y^b_{t-3,15})$ is claimed to be found by the following four linear equations

$$\triangle z_{t-2,5} = \mathbf{S}(y^a_{t-3,5}) \oplus \mathbf{S}(y^b_{t-3,5}) \quad (1)$$

$$\triangle z_{t-2,15} = \mathbf{S}(y^a_{t-3,15}) \oplus \mathbf{S}(y^b_{t-3,15}) \quad (2)$$

$$y^a_{t-2,0} = 2\mathbf{S}(y^a_{t-3,0} \oplus x^a_{t-2,0}) \oplus 3\mathbf{S}(y^a_{t-3,5}) \oplus \mathbf{S}(y^a_{t-3,10} \oplus x^a_{t-2,3}) \oplus \mathbf{S}(y^a_{t-3,15}) \quad (3)$$

$$y^b_{t-2,0} = 2\mathbf{S}(y^b_{t-3,0} \oplus x^b_{t-2,0}) \oplus 3\mathbf{S}(y^b_{t-3,5}) \oplus \mathbf{S}(y^b_{t-3,10} \oplus x^b_{t-2,3}) \oplus \mathbf{S}(y^b_{t-3,15}) \quad (4)$$

Remark 1. In [1] and [2],the position of coefficients 2 and 3 in equations (3) and (4) has been changed. According to the matrix of $\mathbf{T}$ ([6]), it's not right.

However, we find that the solution of these four linear equations is not unique.

Proposition 1. *From equation (1) to equation (4),* 2^8 *solutions can be found.*

Proof. Let $\mathbf{X} = (S(y^a_{t-3,5}), S(y^a_{t-3,15}), S(y^b_{t-3,5}), S(y^b_{t-3,15}))^T$ and $\mathbf{Y} = (\triangle z_{t-2,5}, \triangle z_{t-2,15}, y^a, y^b)^T$,where $y^i = y^i_{t-2,0} \oplus 2S(y^i_{t-3,0} \oplus x^i_{t-2,0}) \oplus S(y^i_{t-3,10} \oplus x^i_{t-2,3})$,$i = a$ or b. Then we can rewrite equation (1) to (4) as

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{Y} \quad (5)$$

where

$$\mathbf{A} = \begin{pmatrix} 1\,0\,1\,0 \\ 0\,1\,0\,1 \\ 3\,1\,0\,0 \\ 0\,0\,3\,1 \end{pmatrix} \quad (6)$$

Firstly, the linear equation system (5) is educed form the message pair which satisfies the 2-round differential path. We know that it has a solution at least. Then, $Rank(\mathbf{A})$ is 3 means that the number of $\mathbf{X}$ which satisfies this linear equation system is 2^8 in the field $\mathbf{F}_{2^8}$. Since the S-box of AES is a bijection, we can find 2^8 $(y^a_{t-3,5}, y^a_{t-3,15}, y^b_{t-3,5}, y^b_{t-3,15})$ to satisfy linear equation (1) to (4). □

In fact, if we obtain a solution of $(y^a_{t-3,5}, y^a_{t-3,15}, y^b_{t-3,5}, y^b_{t-3,15})$,we can construct all solutions. They have the form $(\mathbf{S}^{-1}(\mathbf{S}(y^a_{t-3,5}) \oplus \delta), \mathbf{S}^{-1}(\mathbf{S}(y^a_{t-3,15}) \oplus 3\delta), \mathbf{S}^{-1}(\mathbf{S}(y^b_{t-3,5}) \oplus \delta), \mathbf{S}^{-1}(\mathbf{S}(y^b_{t-3,15}) \oplus 3\delta))$,where $\delta \in \mathbf{F}_{2^8}$.

Similarly, from the equations

$$\begin{aligned}
\triangle z_{t-2,7} &= \mathbf{S}(y^a_{t-3,7}) \oplus \mathbf{S}(y^b_{t-3,7}),\\
\triangle z_{t-2,13} &= \mathbf{S}(y^a_{t-3,13}) \oplus \mathbf{S}(y^b_{t-3,13}),\\
y^a_{t-2,10} &= \mathbf{S}(y^a_{t-3,2} \oplus x^a_{t-2,1}) \oplus \mathbf{S}(y^a_{t-3,7}) \oplus 2\mathbf{S}(y^a_{t-3,8} \oplus x^a_{t-2,2}) \oplus 3\mathbf{S}(y^a_{t-3,13}),\\
y^b_{t-2,10} &= \mathbf{S}(y^b_{t-3,2} \oplus x^b_{t-2,1}) \oplus \mathbf{S}(y^b_{t-3,7}) \oplus 2\mathbf{S}(y^b_{t-3,8} \oplus x^b_{t-2,2}) \oplus 3\mathbf{S}(y^b_{t-3,13}).
\end{aligned}$$

given in [1],we have

Proposition 2. 2^8 *solutions can be found by solving the four linear equations above to recover* $(y^a_{t-3,7}, y^a_{t-3,13}, y^b_{t-3,7}, y^b_{t-3,13})$.

Proof. Since the rank of the coefficient matrix of these four equations is also 3. Similar to proposition 1, we know 2^8 solutions can be obtained. □

Next, we analyse the recovery attack's complexity in [1] and [2]. According to proposition 1, 2 and the form of solutions, we can only recover 6 effective bytes of y^a_{t-3} and 6 effective bytes of y^b_{t-3} respectively. So, if we want to recover the internal state y_0, besides guessing all the 2^{64} possibilities of the rest 8 bytes of y^a_{t-3}(i.e. $y^a_{t-3,1}$, $y^a_{t-3,3}$, $y^a_{t-3,4}$, $y^a_{t-3,6}$, $y^a_{t-3,9}$, $y^a_{t-3,11}$, $y^a_{t-3,12}$, $y^a_{t-3,14}$), we have to guess one byte of $(y^a_{t-3,5}, y^a_{t-3,15})$ and one byte of $(y^a_{t-3,7}, y^a_{t-3,13})$ respectively. For each collision message pair (M^a, M^b), 2^{80} different y_0 can be computed from y^a_{t-3}. However, since we only know 6 bytes of y^b_{t-3}, a y_0 survives randomly with probability 2^{-48} after the filter-out process. So, 2^{32} y_0 can be obtained from a collision message pair and the correct y_0 must be in them. More collision message pairs are needed to find out the right y_0.

How many collision message pairs should we obtain to make sure we can detect the right y_0? Under the assumption that every element which survives the filter-out process is random, 2 collision message pairs is enough to detect the right y_0 by Lemma 1. Because the expected number of collision is $\frac{2^{32} \times 2^{32}}{2^{128}} = 2^{-64}$ except the right y_0.

Another problem is the data complexity. From the birthday paradox, the distinguishing attack's success rate is 0.63, which is also the success rate of a collision occurs. At least 3 structures should be constructed to ensure 2 collisions occurs with probability greater than $\frac{1}{2}$.

Complexity Evaluation. Both the time complexity and the data complexity are now dominated by the recovery attack. The time complexity is $2 \times 2^{80} = 2^{81}$ exhaustive search and the data complexity is $3 \times 2^{65.5} \approx 2^{67}$ chosen messages.

Success Rate. The success rate of distinguishing attack is 0.63 when we run it once. Now, we run it 3 times to find out the right y_0. So, the distinguishing attack's success rate is

$$1 - (1 - 0.63)^3 \approx 0.95.$$

And the success rate of the recovery attack is

$$3 \times 0.63^2 \times 0.37 + 0.63^3 \approx 0.69.$$

4 Modified Differential Path and Internal State Recovery

The correspondence of the coefficients in equation (3) and (4) leads to the nonunique solutions in proposition 1 and proposition 2. In this section, we present a modified 2-round differential path and use this differential path to construct a new distinguisher. Finally, the distinguisher is used to recover the internal state with about $2^{65.5}$ chosen messages and $2^{65.5}$ queries.

4.1 Modified Differential Path

The modified differential path is shown in Fig.2.

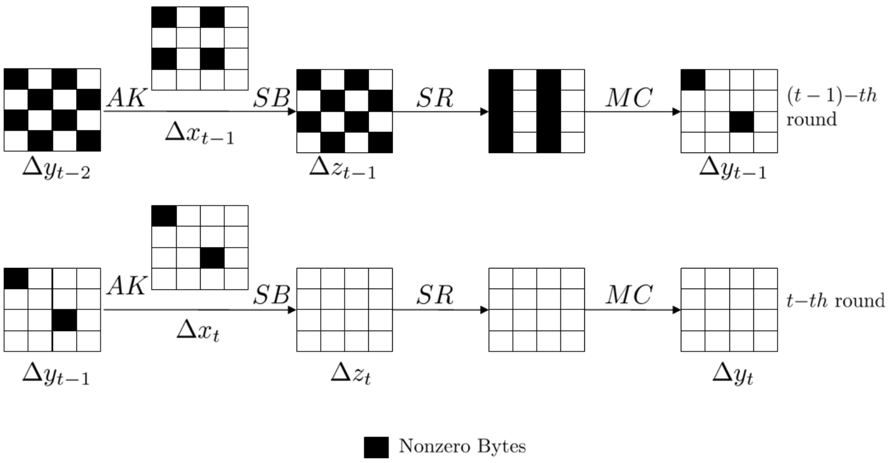

Fig.2. Modified Differential Path

From the differential path, we observe that all bytes in $\triangle y_{t-1}$ are zero except $\triangle y_{t-1,0}$ and $\triangle y_{t-1,10}$,which equal to $\triangle x_{t,0}$ and $\triangle x_{t,2}$ respectively. We have

$$(\triangle z_{t-1,0}, \triangle z_{t-1,5}, \triangle z_{t-1,10}, \triangle z_{t-1,15})^T = \mathbf{T}^{-1}(\triangle x_{t,0}, 0, 0, 0) \quad (7)$$

$$(\triangle z_{t-1,2}, \triangle z_{t-1,7}, \triangle z_{t-1,8}, \triangle z_{t-1,13})^T = \mathbf{T}^{-1}(0, 0, \triangle x_{t,2}, 0) \quad (8)$$

Since all elements in the $\mathbf{T}^{-1}$ are nonzero, there are 8 nonzero bytes in $\triangle z_{t-1}$ as shown in Fig.2.

Given two messages $M^a = (x_1^a, x_2^a, ..., x_{t-1}^a, x_t^a)$ and $M^b = (x_1^b, x_2^b, ..., x_{t-1}^b, x_t^b)$ that follow the 2-round differential path in Fig.2, Fact 1 to Fact 3 in [1] are still correct(the difference is that we choose the messages corresponding to the modified differential path here). And we can use the modified differential path to recover some more information of y_{t-2}^a and y_{t-2}^b.

Proposition 3. *Given M^a and M^b as shown above, $(y^a_{t-2,2}, y^b_{t-2,2})$ and $(y^a_{t-2,8}, y^b_{t-2,8})$ can be recovered with 2^{16} XOR operations and 2^9 chosen messages respectively.*

Proof. The proof is similar to that of Fact 2 in [1]. We only need to replace $(x^a_{t-1,2}, x^b_{t-1,2})$ by different $(\overline{x^a_{t-1,2}}, \overline{x^b_{t-2,2}})$ and replace $(x^a_{t-1,8}, x^b_{t-1,8})$ by different $(\overline{x^a_{t-1,8}}, \overline{x^b_{t-2,8}})$. □

Now, a new distinguisher can be builded similarly as that in [1]. Given a fixed word differential $(\eta, 0, \gamma, 0)$, choose two structures as follows:

$$T_1 = \{M^a = (x^a_1, x^a_2, ..., x^a_{t-1}, x_t)\},$$
$$T_2 = \{M^b = (x^b_1, x^b_2, ..., x^b_{t-1}, x_t \oplus (\eta, 0, \gamma, 0))\},$$

where the message words $(x^a_i, x^b_i)(i = 1, 2, ..., t-1)$ are randomly chosen, i.e., we choose $\triangle x_{t-1}$ and $\triangle x_t$ as shown in Fig.2. The distinguisher works in the following 3 steps:

1. Choose $2^{64.5}$ messages with the form of structure T_1 and T_2 respectively. And query the MAC to obtain the corresponding MACs.
2. Search for collisions between the MACs of T_1 and T_2 by birthday attack, i.e., find message pair (M^a, M^b) such that MAC (M^a)=MAC(M^b),where $M^a \in T_1$ and $M^b \in T_2$. Randomly choose another pair $(\overline{x^a_t}, \overline{x^b_t})$ to replace the last message word (x^a_t, x^b_t) of (M^a, M^b), where $\triangle \overline{x_t} = \triangle x_t$. Obtain the MACs of the new message pair. If a collision occurs, we conclude that the MAC is ALRED-MAC, and go to step 3. Otherwise, the MAC is a random function.
3. Randomly choose 2^8 different $(\overline{x^a_{t-1,0}}, \overline{x^b_{t-1,0}})$ to replace $(x^a_{t-1,0}, x^b_{t-1,0})$. Query the MACs of these new message pairs. If a collision appears among them, the ALRED construction is claimed as the ALPHA-MAC. Otherwise, it's other ALRED MAC instance.

The complexity of this distinguishing attack is also $2^{65.5}$ MAC queries and $2^{65.5}$ chosen messages and its success rate is 0.63 from the birthday paradox.

4.2 Internal State Recovery

In this section, according to the distinguisher which is based on the modified 2-round differential path, we recover the internal state y_0.

Denoted $M^a = (x^a_1, x^a_2, ..., x^a_{t-1}, x^a_t)$ and $M^b = (x^b_1, x^b_2, ..., x^b_{t-1}, x^b_t)$. The process of internal state recovery attack is depicted in Fig.3, where the symbols '*', '?' and '0' have the same meaning as in [1].

Firstly, by proposition 3 and algorithm A_2, A_3 in [1], $(y^a_{t-2,0}, y^b_{t-2,0}, y^a_{t-2,2}, y^b_{t-2,2}, y^a_{t-2,8}, y^b_{t-2,8}, y^a_{t-2,10}, y^b_{t-2,10})$ can be recovered directly.

$$\triangle y_{t-3} = \begin{bmatrix} * & ? & * & ? \\ ? & * & ? & * \\ * & ? & * & ? \\ ? & * & ? & * \end{bmatrix}$$

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

$$\xleftarrow{AK^{-1},SB^{-1}} \triangle z_{t-1} = \begin{bmatrix} * & 0 & * & 0 \\ 0 & * & 0 & * \\ * & 0 & * & 0 \\ 0 & * & 0 & * \end{bmatrix} \xleftarrow{SR^{-1},MC^{-1}} \triangle y_{t-1} = \begin{bmatrix} \triangle x_{t,0} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \triangle x_{t,3} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig.3. Internal State Recovery

Then, based on the following two equations:

$$(\triangle z_{t-2,0}, \triangle z_{t-2,5}, \triangle z_{t-2,10}, \triangle z_{t-2,15})^T = \mathbf{T}^{-1}(\triangle y_{t-2,0}, 0, \triangle y_{t-2,8}, 0)^T \quad (9)$$
$$(\triangle z_{t-2,2}, \triangle z_{t-2,7}, \triangle z_{t-2,8}, \triangle z_{t-2,13})^T = \mathbf{T}^{-1}(\triangle y_{t-2,2}, 0, \triangle y_{t-2,10}, 0)^T \quad (10)$$

we can obtain the values $(\triangle z_{t-2,0}, \triangle z_{t-2,5}, \triangle z_{t-2,10}, \triangle z_{t-2,15}, \triangle z_{t-2,2}, \triangle z_{t-2,7}, \triangle z_{t-2,8}, \triangle z_{t-2,13})$.And since Fact 4 is still correct(The only difference here is the choice of (x_{t-1}, x'_{t-1}).We should let $\overline{\triangle x_{t-1,2}} \neq \triangle x_{t-1,2}$ in addition and select 2^{16} different word pairs (x_{t-1}, x'_{t-1}).), the bytes $(y^a_{t-3,0}, y^b_{t-3,0}, y^a_{t-3,2}, y^b_{t-3,2}, y^a_{t-3,8}, y^b_{t-3,8}, y^a_{t-3,10}, y^b_{t-3,10})$ can be recovered.

Now, six equations related to the $(y^a_{t-3,5}, y^a_{t-3,15}, y^b_{t-3,5}, y^b_{t-3,15})$ are listed as follows:

$$\begin{aligned}
\triangle z_{t-2,5} &= \mathbf{S}(y^a_{t-3,5}) \oplus \mathbf{S}(y^b_{t-3,5}), \\
\triangle z_{t-2,15} &= \mathbf{S}(y^a_{t-3,15}) \oplus \mathbf{S}(y^b_{t-3,15}), \\
y^a_{t-2,0} &= 2\mathbf{S}(y^a_{t-3,0} \oplus x^a_{t-2,0}) \oplus 3\mathbf{S}(y^a_{t-3,5}) \oplus \mathbf{S}(y^a_{t-3,10} \oplus x^a_{t-2,3}) \oplus \mathbf{S}(y^a_{t-3,15}), \\
y^b_{t-2,0} &= 2\mathbf{S}(y^b_{t-3,0} \oplus x^b_{t-2,0}) \oplus 3\mathbf{S}(y^b_{t-3,5}) \oplus \mathbf{S}(y^b_{t-3,10} \oplus x^b_{t-2,3}) \oplus \mathbf{S}(y^b_{t-3,15}), \\
y^a_{t-2,8} &= \mathbf{S}(y^a_{t-3,0} \oplus x^a_{t-2,0}) \oplus \mathbf{S}(y^a_{t-3,5}) \oplus 2\mathbf{S}(y^a_{t-3,10} \oplus x^a_{t-2,3}) \oplus 3\mathbf{S}(y^a_{t-3,15}), \\
y^a_{t-2,8} &= \mathbf{S}(y^a_{t-3,0} \oplus x^a_{t-2,0}) \oplus \mathbf{S}(y^a_{t-3,5}) \oplus 2\mathbf{S}(y^a_{t-3,10} \oplus x^a_{t-2,3}) \oplus 3\mathbf{S}(y^a_{t-3,15}).
\end{aligned}$$

The correct $(y^a_{t-3,5}, y^a_{t-3,15}, y^b_{t-3,5}, y^b_{t-3,15})$ can be obtained by solving any four equations of them if their coefficient matrix's rank is 4, for example, the last four equations.

Similarly, we have six equations related to the $(y^a_{t-3,7}, y^a_{t-3,13}, y^b_{t-3,7}, y^b_{t-3,13})$ as follows:

$$\begin{aligned}
\triangle z_{t-2,7} &= \mathbf{S}(y^a_{t-3,7}) \oplus \mathbf{S}(y^b_{t-3,7}),\\
\triangle z_{t-2,13} &= \mathbf{S}(y^a_{t-3,13}) \oplus \mathbf{S}(y^b_{t-3,13}),\\
y^a_{t-2,2} &= 2\mathbf{S}(y^a_{t-3,2} \oplus x^a_{t-2,1}) \oplus 3\mathbf{S}(y^a_{t-3,7}) \oplus \mathbf{S}(y^a_{t-3,8} \oplus x^a_{t-2,2}) \oplus \mathbf{S}(y^a_{t-3,13}),\\
y^b_{t-2,2} &= 2\mathbf{S}(y^b_{t-3,2} \oplus x^b_{t-2,1}) \oplus 3\mathbf{S}(y^b_{t-3,7}) \oplus \mathbf{S}(y^b_{t-3,8} \oplus x^b_{t-2,2}) \oplus \mathbf{S}(y^b_{t-3,13}),\\
y^a_{t-2,10} &= \mathbf{S}(y^a_{t-3,2} \oplus x^a_{t-2,1}) \oplus \mathbf{S}(y^a_{t-3,7}) \oplus 2\mathbf{S}(y^a_{t-3,8} \oplus x^a_{t-2,2}) \oplus 3\mathbf{S}(y^a_{t-3,13}),\\
y^b_{t-2,10} &= \mathbf{S}(y^b_{t-3,2} \oplus x^b_{t-2,1}) \oplus \mathbf{S}(y^b_{t-3,7}) \oplus 2\mathbf{S}(y^b_{t-3,8} \oplus x^b_{t-2,2}) \oplus 3\mathbf{S}(y^b_{t-3,13}).
\end{aligned}$$

And we can solve the right $(y^a_{t-3,7}, y^a_{t-3,13}, y^b_{t-3,7}, y^b_{t-3,13})$ by any four equations of them if their coefficient matrix's rank is 4.

Since only one solution can be solved from the 12 linear equations above, we know 8 bytes of y^a_{t-3} and 8 bytes of y^b_{t-3} respectively. We can recover the internal state y_0 by using the same method as in [1]. And the recovery attack on the internal state y_0 is completed.

Finally, we analyse the complexity of the whole attack. Based on the distinguisher constructed by the modified 2-round differential path, we conquer the problem that a unique solution can not be recovered in the step of recovering $(y^a_{t-3,5}, y^b_{t-3,5}, y^a_{t-3,7}, y^b_{t-3,7}, y^a_{t-3,13}, y^b_{t-3,13}, y^a_{t-3,15}, y^b_{t-3,15})$. So, the complexity of the whole attack is dominated by the distinguishing attack, which is about $2^{65.5}$ queries and $2^{65.5}$ chosen messages, with success rate 0.63.

5 Conclusion

In this article, we point out a flaw in the internal state recovery attack. It comes from the limitation of the 2-round differential path constructed in [1]. Then, a modified 2-round differential path which can provide more information to recover the internal state y_0 is presented to conquer the limitation. And we obtain an attack with the same complexity as birthday attack. The second preimage attack can be perform as in [4] and a selective forgery attack can be performed as in [5] if y_0 is known.

References

1. Yuan, Z., Wang, W., Jia, K., Xu, G., Wang, X.: New Birthday Attacks on Some MACs Based on Block Ciphers. Advances in Cryptology - CRYPTO 2009
2. Yuan, Z., Jia, K., Wang, X.: Distinguishing and Forgery Attacks on ALRED and Its AES-based Instance ALPHA-MAC. Cryptology ePrint Archive, Report 2008/516(2008),http://eprint.iacr.org/2008/516
3. Daemen, J., Rijmen, V.: A New MAC Construction ALRED and A Specific Instance ALPHA-MAC. In: Gilber, H., Handschuh, H.(eds.) FSE 2005. LNCS, vol. 3557, pp. 1-17. Springer, Heidelberg(2005)

4. Huang, J., Seberry, J., Susilo, W.: On the Internal Structure of ALPHA-MAC. In: Nguyen, P.Q.(ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 271-285.S pringer, Heidelberg(2006)
5. Biryukov, A., Bogdanov, A., Khovratovich, D., Kasper, T.: Collision Attacks on AES-Based MAC:ALPHA-MAC. In: Paillier, P.,Verbauwhede, I.(eds.) CHES 2007.LNCS, vol. 4727, pp. 166-180. Springer, Heidelberg(2007)
6. J.Daemen and V.Rijmen: The Design of Rijndael:AES-The Advanced Encryption Standard, Springer-Verlag(2002)
7. Joux, A.:Algorithmic Cryptanalysis, pp.190-191. Chapman & Hall(2009)
8. Donald W. Davies: A message authenticator algorithm suitable for a mainframe computer. In: Blakley,G.R.Chaum,D.(eds.) Advances in Cryptology - Proceedings of Crypto '84, LNCS 196, pp. 393-400. Springer-Verlag(1985)
9. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.:UMAC: Fast and Secure Message Authentication. In: Wiener, M,J. (ed.) Advances in Cryptology - Crypto '99. LNCS 1666, pp. 216-233. Springer-Verlag(1999)
10. Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Johansson, T. (ed.) FSE 2003. LNCS, vol. 2887, pp. 129-153. Springer, Heidelberg(2003)
11. Kurosawa, K., Tsunoom, Y.:Provably Secure MACs form Differentially-Uniform Permutations and AES-Based Implementations. In: Robshaw,M.J.B (ed.) FSE 2006. LNCS, vol.4047, pp. 226-241. Springer, Heidelberg(2006)
12. ISO/IEC 9797-1, Information technology - Security Techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using A Block Cipher, ISO(1999)
13. Bellare, M., Canetti, R., Krawczyk, H.:Keying Hash Functions for Message Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 1-15. Springer, Heidelberg(1996)
14. Prenneel, B., Van Oorschot, P.C.: MDx-MAC and building fast MACs from hash functions. In: Coppersmith, D. (ed.) Advances in Cryptology - Proceedings Crypto '95. LNCS 963, pp. 1-14. Springer-Verlag(1995)