Leaked-State-Forgery Attack Against The Authenticated Encryption Algorithm ALE

Shengbao Wu[1,3], Hongjun Wu[2], Tao Huang[2], Mingsheng Wang[4], Wenling Wu[1]

1. Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, PO Box 8718, China
2. Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
3. Graduate School of Chinese Academy of Sciences, Beijing 100190, China
4.State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

{wushengbao,wwl}@tca.iscas.ac.cn
{wuhj,huangtao}@ntu.edu.sg
wangmingsheng@iie.ac.cn

Abstract. ALE is a new authenticated encryption algorithm published at FSE 2013. The authentication component of ALE is based on the strong Pelican MAC, and the authentication security of ALE is claimed to be 128-bit. In this paper, we propose the leaked-state-forgery attack (LSFA) against ALE by exploiting the state information leaked from the encryption of ALE. The LSFA is a new type of differential cryptanalysis in which part of the state information is known and exploited to improve the differential probability. Our attack shows that the authentication security of ALE is only 97-bit. And the results may be further improved to around 93-bit if the whitening key layer is removed. We implemented our attacks against a small version of ALE (using 64-bit block size instead of 128-bit block size). The experimental results match well with the theoretical results.

Key words: authenticated encryption, forgery attack, ALE

1 Introduction

Confidentiality and message authentication are two fundamental goals in cryptography. In symmetric key cryptography, a block cipher/stream cipher is used to protect the confidentiality of messages; and a message authentication code (MAC) is used to authenticate messages. In the widely used Transport Layer Security (TLS), the MAC-then-Encrypt approach is used: HMAC [27] is applied to authenticate the TCP packets, and AES [9] in CBC mode [26] can be used to encrypt the payload of TCP packets.

In many applications, both confidentiality and message authentication are required. The authenticated encryption algorithm can achieve encryption and authentication simultaneously, and its performance is much better than the combination of separate encryption and authentication. Authenticated encryption has received considerable research interests in recent years. A number of block cipher based authenticated encryption modes have been proposed, *e.g.,* IAPM [21], OCB [28], CCM [29], CWC [23], GCM [24], EAX [4], HBS [19], BTM [18] and McOE [15]. The ISO/IEC 19772:2009 [17] standardized several modes, including EAX, CCM, GCM and OCB 2.0. Besides the authenticated encryption modes, several authenticated encryption algorithms have been proposed, such as Helix [14], Phelix [30], Hummingbird-2 [13], ASC-1 [20], the 3GPP algorithm 128-EIA3 [2] and Grain-128a [3]. The coming competition CAESAR (Competition for Authenticated Encryption: Security, Applicability and Robustness) [7] is expected to attract many new authenticated encryption algorithms.

ALE. ALE (*A*uthenticated *L*ightweight *E*ncryption) is an AES-based authenticated encryption algorithm proposed by Bogdanov *et al.* at FSE 2013 [6]. It is designed for the low-cost embedded systems (such as RFID tags and smart cards) and provides single-pass authenticated encryption with associated data. The keystream generation of ALE uses the idea of the LEX stream cipher [5],

and the tag generation uses the idea of Pelican MAC [10]. It has 256-bit internal state and aims to have a probability of success at most 2^{-128} for a forgery attack.

Pelican MAC is an extremely simple MAC based on AES. In Pelican MAC, any difference being introduced in the forgery attack passes through at least four AES rounds. It ensures that the success rate of a forgery attack is at most 2^{-128}. The state size of Pelican MAC is only 128 bits. The small state size means that the number of messages being authenticated under the same key should be less than 2^{64}. Yuan *et al.* delivered a state recovery attack against the Pelican MAC by exploiting the state collision when more than 2^{64} authentication tags are generated from the same key [33]. The attack given in [33] cannot be applied to ALE. In ALE, the state size is increased to 256 bits, and a new nonce is needed for generating each authentication tag when the same key is used.

The stream cipher LEX is based on AES, and four keystream bytes are extracted from the AES state after each round. LEX suffers from two attacks. The slide attack against LEX recovers the key with negligible complexity when around 2^{60} nonces are used with the same secret key [31]. Another attack recovers the key with around 2^{100} simple operations and 2^{40} keystream bytes [11, 12]. ALE is not vulnerable to these two attacks due to its large state and the changing AES round keys (the round keys in LEX are fixed for the same key).

The design of ALE is similar to the authenticated encryption algorithm ASC-1. In ASC-1, a leaked byte is protected by an additional key byte before it is extracted as keystream byte. However, the additional key byte is not used in ALE for better hardware efficiency. Unfortunately, the lacking of additional key bytes in ALE allows part of the AES state being leaked as keystream, and such leaked state information can be exploited to improve the forgery attack, as demonstrated in this paper.

In this paper, we propose a new attack – leaked-state-forgery attack (LSFA) against ALE. The general idea of this attack is to exploit the leaked state information so as to increase the differential probability. For ALE, there exists four-round AES differential characteristics with probability much larger than 2^{-128} after taking into account the leaked state information. The forgery attack against ALE can reach the success rate of 2^{-97}, which is 2^{31} higher than the claimed probability. We show that the results may be further improved if the whitening key layer is removed. We implemented our attack on a small version of ALE, in which 64-bit block and 4-bit-to-4-bit S-box are used. The experimental results match well with the theoretical results.

Very recently, Khovratovich and Rechberger independently proposed an attack against ALE in SAC 2013 [22] which also exploits the weakness of the ALE scheme. However, we notice that their attack is applied to a variant of ALE which the four bytes are leaked after `SubByte`. And in this work, we optimized the differential characteristics used in our attacks so that lower complexities can be obtained in this paper.

This paper is organized as follows. The specification of ALE is given in Sect. 2. Section 3 describes a basic forgery attack against ALE. Section 4 optimizes the forgery attack. Section 5 discusses the effect of removing the whitening key layer of four-round AES. Section 6 gives the experimental results on ALE with reduced block size. Section 7 concludes this paper.

2 The Specification of ALE

In this section, we give a brief description of the ALE. The full specifications of ALE can be found in the original paper [6].

AES round function. AES-128 is used as an underlying primitive of ALE. A full specification of AES can be found in [9]. There are four operations in an AES round: `SubBytes(SB)`, `ShiftRows(SR)`, `MixColumns(MC)` and `AddRoundKey(ARK)`.

```
AESRound(State, ExpandedKey[i])
{
  SubBytes(State);
  ShiftRows(State);
  MixColumns(State);
```

```
    AddRoundKey(State,ExpandedKey[i]);
}
```

LEX keystream extraction. In the stream cipher LEX, AES round functions are repeatedly applied to a state (the subkeys are fixed). At the end of each AES round, 4 bytes from the state are extracted as the keystream [5]. The positions of leaked bytes are shown in Fig. 1.

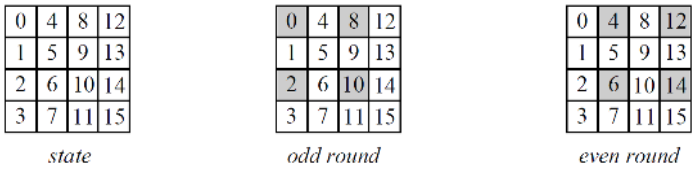

Fig. 1: The positions of the leaked bytes in the even and odd rounds of LEX.

Pelican MAC. In the Pelican MAC, each 128-bit message block is xored to a secret 128-bit state, then the state passes through 4 AES rounds. In Pelican MAC, each difference passes through at least 25 active S-boxes (following directly from the analysis of AES), thus Pelican MAC provides strong security against forgery attack.

Specification of ALE. The encryption/authentication of ALE is shown in Fig. 2. The process of associated data and last partial block are omitted here. The encryption component of ALE is based on LEX, and its authentication component is based on Pelican MAC. A different nonce is used in ALE for the protection of every message. When the verification fails, the plaintext from the decryption should be kept secret so as to prevent state recovery attack. To encrypt/authenticate a

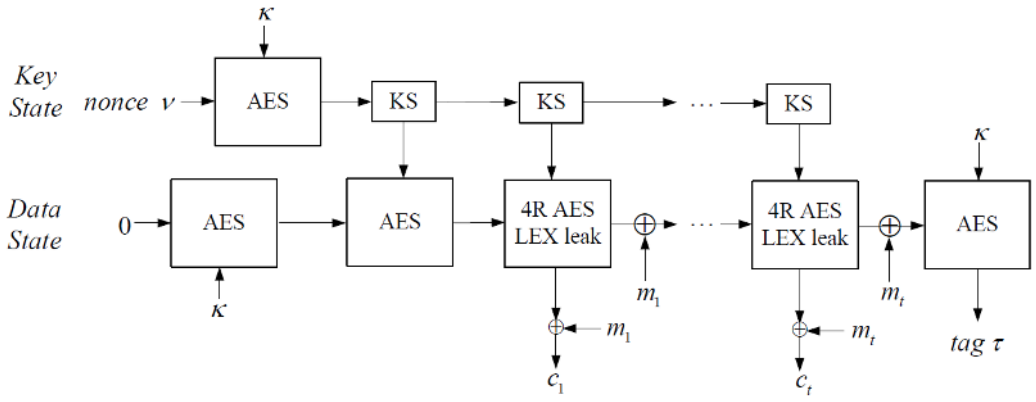

Fig. 2: Encryption and authentication of ALE.

message, ALE takes a 128-bit master key κ, a message μ, associated data α and 128-bit non-zero nonce ν as inputs. And it outputs ciphertext γ of the same length as message and a 128-bit tag τ. The initialization of ALE is given as follows: the nonce ν is encrypted using AES-128 under the master key κ. The 128-bit output is used as the initial key state. A message with value 0 is encrypted using AES-128 under the master key κ to give the data state. The 128-bit output $\text{AES}_\kappa(0)$ is encrypted again using the initial key state as the key. The key state is updated by applying round key schedule of AES-128 to the final round key of last AES encryption with round constant x^{10} in $\mathbb{F}_{2^8}$.

To process a 16-byte message block, the data state is encrypted with 4 rounds of AES using the key state as key. 16 bytes are leaked from the data state in the 4 AES rounds in accordance with the LEX keystream extraction. According to the code provided by the authors of ALE, five round keys

are used during the 4 AES rounds, namely, an initial whitening key is used. And at each AES round, four bytes are leaked after the `AddRoundKey()` function. The leak is xored to the current 16-byte block M for encryption. The final round subkey is updated one more time using the AES round key schedule with byte round constant x^4 in $\mathbb{F}_{2^8}$ to get the key state. The current message block M is xored to the data state so that it would pass through the next 4 AES rounds for authentication purpose (similar to that in Pelican MAC).

The decryption/verification is similar to the encryption/authentication, except that the ciphertext block is xored to the keystream to get the message, as shown in Fig. 3. We provide this figure here since the decryption/verification is important in our attack.

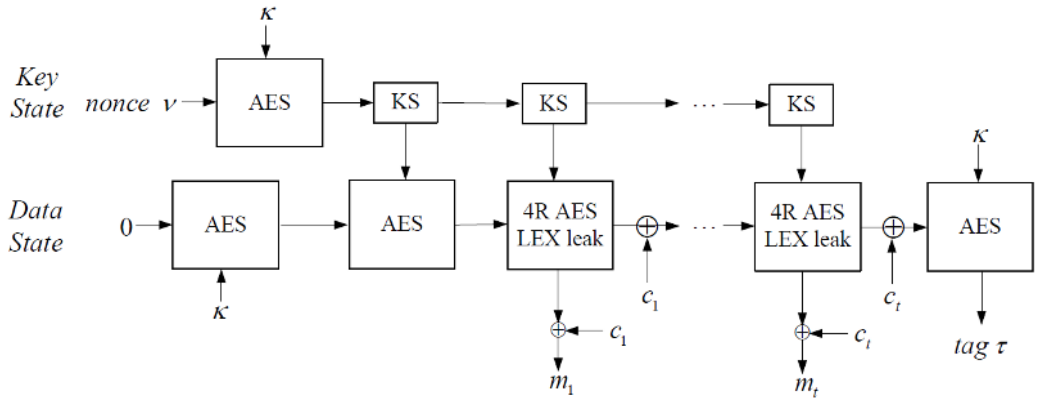

Fig. 3: Decryption and verification of ALE.

The designers of ALE claim that any forgery attack not involving key recovery/internal state recovery has a success probability at most 2^{-128}. It is stated that each secret key is used to protect at most 2^{48} message bits. Such restriction on message bits does not affect the success rate of our forgery attack.

3 A Basic Leaked-State Forgery Attack on ALE

In this section, we present a basic forgery attack against ALE. The chance of successful forgery attack is 2^{-106}, which is 2^{22} larger than the claimed success rate 2^{-128}. This attack requires 2^{41} known plaintext blocks.

3.1 The main idea of the attack

The following property of active S-box will be used in our attack:

Property 1 *For an active S-box, if the values of an input and the input/output difference are known, the output/input difference is known with probability 1.*

Here the active S-box is the S-box with non-zero input difference. In the rest of the paper, we will use a new term *active leaked byte* to denote a leaked byte with difference on it.

In the security analysis of Pelican MAC [10] and ALE [6], the probability of four-round differential characteristic of ALE follows the analysis of AES. It has been shown that for any four-round AES differential characteristic, the number of active S-boxes is at least 25 [8]. For each S-box, the differential probability is at most 2^{-6}. Hence, there is a trivial upper bound for the four-round AES differential probability which is 2^{-150}. However, different from the Pelican MAC, 4 state bytes are leaked at the end of every round in ALE. Using Property 1, it is possible to bypass some active S-boxes with probability 1 when the input bytes to those active S-boxes are leaked. It means that the overall differential probability could be significantly increased.

3.2 Finding a differential characteristic

The first step of the attack is to find a valid four-round AES differential characteristic which passes through 25 (or close to 25) active S-boxes and the differences pass through several leaked bytes in the first three rounds.

There are many differential characteristics for four AES rounds. To categorize those differential characteristics, we use the number of active bytes before the S-box layer in each round to represent a certain type of differential characteristics. For example, the differential characteristic shown in Fig. 4 falls in the type "1–4–16–4". Note that the positions of active bytes are not unique for each type.

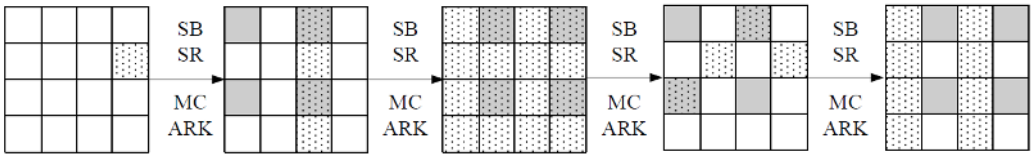

Fig. 4: An example of 1-4-16-4 differential characteristic. Gray squares denote leaked bytes. Squares marked with broken line denote active bytes.

In our basic attack, we use the type of differential characteristic shown in Fig. 4. There are 25 active S-boxes in the differential characteristic, and 8 active leaked bytes are located in the first three rounds.

Next we need to find a differential characteristic with high probability. Note that it is not always guaranteed that the differential probability of each active S-box can reach the maximum value 2^{-6}. The AES S-box has a property that for any input difference δ_1 and output difference δ_2, the probability that equation $S(x) \oplus S(x \oplus \delta_1) = \delta_2$ has a solution is 127/256. Among the 127 solutions, there are 126 solutions have probability 2^{-7} and only one solution has probability 2^{-6}. Hence, for an active S-box, there is a unique output difference reaches the probability 2^{-6} for difference propagation. It shows the conditions to set active S-boxes with difference propagation probability 2^{-6} will limit the number of choices for the possible differential characteristics.

It is thus not surprising that we found no differential characteristic such that every active S-box (except those involving the leaked ones) has the maximum differential probability 2^{-6} after testing all the possible positions of the type "1–4–16–4". In order to find a differential characteristic, we need to allow some active S-box with differential probability 2^{-7}. We managed to find a number of differential characteristics. One of them is given in Fig. 5, and we will use this differential characteristic to demonstrate our basic attack. The differential probability of this differential characteristic is given as $2^{-6\times16+(-7)\times9} = 2^{-159}$ (differential probability 2^{-6} for 16 active S-boxes, 2^{-7} for 9 active S-boxes).

Three differences in Fig. 5 will be used in our attack: the input difference Δ_{in}, the output difference Δ_{out} and the keystream difference Δ_s:

$$\begin{aligned}\Delta_{\text{in}} &= (0,0,0,0;\ 0,0,0,0;\ 0,0,0,0;\ 0,96,0,0);\\ \Delta_{\text{out}} &= (\text{B1,DE,6F,6F};\ 0,0,0,0;\ \text{B8,5C,82,55};\ 0,0,0,0);\\ \Delta_{\text{s}} &= (0,0,\text{E,F3};\ 59,37,\text{6E,F2};\ 0,81,\text{6C},0;\ 0,0,0,0);\end{aligned}$$

Note that the values in Δ_{s} are obtained by simply concatenating the bytes extracted from the states. The order of those bytes has no effect on the attack, as long as this order is fixed.

3.3 Launching the forgery attack

After finding a four-round AES differential characteristic, we need to determine the values of the leaked bytes on the differential characteristic so as to improve the differential probability. The values of the leaked bytes are important for locating the ciphertext bytes that will be modified in the forgery attack.

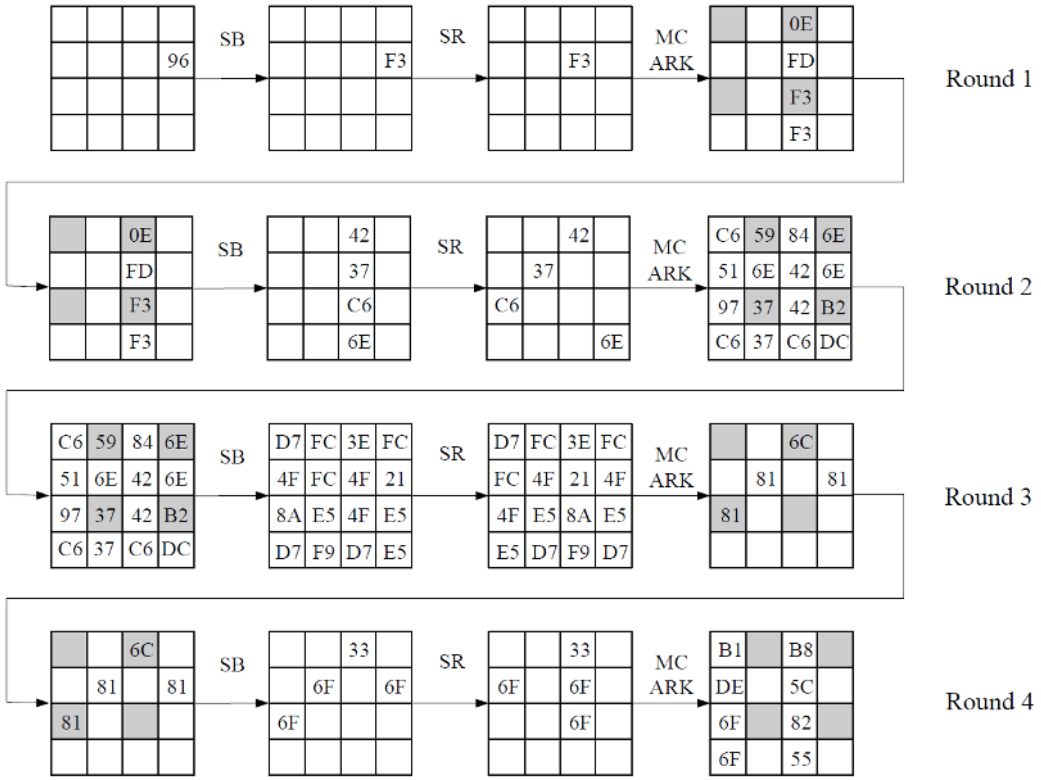

Fig. 5: A differential characteristic of type "1–4–16–4". The hexadecimal numbers indicate the difference values. The empty squares indicate no difference. The squares of leaked bytes are marked with gray color.

In the differential characteristic shown in Fig. 5, the differences at the positions of leaked bytes are known before and after the S-box. Hence, we solve for the values of the active leaked bytes. There are either two or four possible solutions depending on the output difference. We store the possible values of leaked bytes in a table T (Table 5 in Appendix A). Notice that we ignore the conditions on the leaked bytes in the fourth round because that for any leaked values at the end of Round 3, we can always derive the corresponding difference in Round 4.

If the value of a keystream block s_i falls into one of the possible values of table T, we modify the previous ciphertext block c_{i-1} and the current ciphertext block c_i using the differences given in Fig. 5. More specifically, $c'_{i-1} = c_{i-1} \oplus \Delta_{\text{in}}$; $c'_i = c_i \oplus \Delta_{\text{out}} \oplus \Delta_s$. The modified ciphertext is sent for decryption/verification.

We illustrate here how the above attack works. From the decryption, the difference $\Delta m_{i-1} = (c_{i-1} \oplus s_{i-1}) \oplus (c'_{i-1} \oplus s'_{i-1}) = \Delta_{\text{in}}$ because $\Delta s_{i-1} = 0$; the difference $\Delta m_i = (c_i \oplus s_i) \oplus (c'_i \oplus s'_i) = \Delta_{\text{out}}$ because $c'_i \oplus c_i = \Delta_{\text{out}} \oplus \Delta_s$. Then Δm_{i-1} is introduced to the data state, and after four rounds, Δm_i is introduced to cancel the difference in the state. The difference propagation follows that in Fig. 5.

Complexity of the attack. In the attack above, the differential probability of the differential characteristic is 2^{-159} before considering the leaked bytes. There are eight leaked bytes being involved in the differential characteristic, with 5 of them being introduced to the active S-boxes with probability 2^{-7}, and another 3 of them being introduced to the active S-boxes with probability 2^{-6}. According to Property 1, the differential probabilities of those eight active boxes involving the leaked bytes become 1. The overall differential probability becomes $2^{-159} \times 2^{7\times5} \times 2^{6\times3} = 2^{-106}$. The success rate of the above attack is thus 2^{-106}.

In this attack, eight leaked keystream bytes are considered, and the values of 6 leaked bytes (from the first two rounds) should be one of the 128 entries in Table T (as explained above). A

random keystream block satisfies the requirement with probability $128/2^{6\times 8} = 2^{-41}$. We thus need 2^{41} known plaintext blocks in this attack.

4 Optimizing the Leaked-State-Forgery Attack against ALE

In this section, we optimize the LSFA against ALE. In Sect. 4.1, we improve the success rate of the forgery attack. The optimal success rate of a forgery attack can reach 2^{-97}, while 2^{56} known plaintext blocks are needed. In Sect. 4.2, the number of known plaintext blocks can be reduced to $2^{8.4}$ for achieving a success rate 2^{-102}. Note that the known plaintext blocks can be related to different keys or different nonces.

4.1 Improving the differential probability

From the attack presented in Sect. 3, we notice that the success rate of forgery attack is determined by the probability of the differential characteristic after taking into account of the leaked bytes. To evaluate the success rate of the forgery attack against ALE, we use the term *effective active S-boxes* to represent the active S-boxes which cannot be bypassed by exploiting the leaked bytes. In the following, we will analyze different cases to find the smallest number of effective active S-boxes.

We start with recalling some properties of the AES round function. The function `MixColumns` has a property that if it is active, the total number of active bytes in the input and output will be at least five (the property of the maximum distance separable code). By referring to the Lemma 9.4.1 from [9], we have the following lemma.

Lemma 1. *The number of active S-boxes of any two-round AES differential characteristic is lower bounded by $5N$, where N is the number of active columns in the first round.*

In the four AES rounds in ALE, there are 16 leaked bytes. But the leaked bytes from the fourth round cannot be exploited in the attack as they do not pass through S-boxes directly. Therefore only the leaked bytes in the first three rounds can be exploited, and there are at most 12 active leaked bytes. We use $[l_1,l_2,l_3]$ to indicate the number of active leaked bytes in the first three rounds respectively. For instance, the number of active leaked bytes in the differential characteristic in Fig. 4 is $[2, 4, 2]$. And we use n_i^A $(i = 1, 2, 3, 4)$ to denote the number of active S-boxes at each S-box layer, which will be used in later analysis.

In the following, we will analyze differential characteristics with the smallest number of effective active S-boxes, using the techniques of solving Mixed-Integer Linear Programming (MILP) problems [25, 32]. MILP is a useful technique for proving security bounds against differential cryptanalysis, by evaluating the minimum number of active S-boxes in several rounds of encryption. Designers and cryptanalysts only require to write out simple (in)equations that are input into an MILP solver, then an optimal solution will be returned.

We denote by X_i the input state of round i, then we have

$$X_{i+1} = ARK \circ MC \circ SR \circ SB(X_i)$$

, where $i \in \{1, 2, 3, 4\}$. Let $X_{i,j}$ be the j-th byte of X_i, where $0 \leq j \leq 15$. For a further step, suppose $Y_i = SB(X_i)$, $Z_i = SR(Y_i)$ and $W_i = MC(Z_i)$. We introduce a function χ to catch whether a byte is nonzero, that is , $\chi(x) = 1$ if $x \neq 0$ and $\chi(x) = 0$ if $x = 0$. Here, the value of $\chi(x)$ is a real number. Then, according to the techniques given in [25, 32], the problem of evaluating the minimum number of effective active S-boxes is translated to an MILP problem as follows.

The Objective Function. The objective function is to minimize the value of

$$\sum_{i=1}^{4}\sum_{j=0}^{15}\chi(\Delta X_{i,j}) - \sum_{k=0,2,8,10}(\chi(\Delta X_{2,k}) + \chi(\Delta X_{4,k})) - \sum_{l=4,6,12,14}\chi(\Delta X_{3,l}), \tag{1}$$

since we would like to evaluate the minimum number of effective active S-boxes. In (1), the number of effective active S-boxes is obtained by first counting the number of active S-boxes in four consecutive rounds of AES and then minus the number of active leaked bytes.

Constraints. According to the property of `MixColumns` , we have $\sum_{j=4k}^{4k+3}(\chi(\Delta Z_{i,j})+\chi(\Delta W_{i,j})) = 0$ or ≥ 5, where $1 \leq i \leq 4$ and $0 \leq k \leq 3$. On the other hand, we have $\chi(\Delta Y_{i,j}) = \chi(\Delta X_{i,j})$, $\chi(\Delta Z_{i,j}) = \chi(\Delta Y_{i,5j \bmod 16})$ and $\chi(\Delta X_{i+1,j}) = \chi(\Delta W_{i,j})$ $(0 \leq j \leq 15)$. Thus, two consecutive rounds of AES provide us four constraints:

$$5d_{i,1} \leq \sum_{j=0}^{3}(\chi(\Delta X_{i,5j \bmod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,1}, \tag{2}$$

$$5d_{i,2} \leq \sum_{j=4}^{7}(\chi(\Delta X_{i,5j \bmod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,2}, \tag{3}$$

$$5d_{i,3} \leq \sum_{j=8}^{11}(\chi(\Delta X_{i,5j \bmod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,3}, \tag{4}$$

$$5d_{i,4} \leq \sum_{j=12}^{15}(\chi(\Delta X_{i,5j \bmod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,4}, \tag{5}$$

where $i \in \{1,2,3\}$ and $d_{i,j} \in \{0,1\}$ $(1 \leq j \leq 4)$. Notice that $d_{i,j} = 0$ if and only if all eight differences before and after `MixColumns` are zero and $d_{i,j} = 1$ otherwise. Here, we do not consider the case of $i = 4$ since linear transformations in Round 4 does not influence the probability of a differential characteristic.

Additional Constraints. To avoid trivial solution where the minimum number of active S-boxes is zero, the following constraint

$$\sum_{j=0}^{15}\chi(\Delta X_{1,j}) \geq 1 \tag{6}$$

is added to ensure that at least one S-box is active. For a further step, the constraint

$$\sum_{k=0,2,8,10}(\chi(\Delta X_{2,k}) + \chi(\Delta X_{4,k})) + \sum_{l=4,6,12,14}\chi(\Delta X_{3,l}) = n \text{ (or } \leq n) \tag{7}$$

is added to the system. That is, all differential characteristics are classified by the number of active leaked bytes. Constraint (7) help us quickly locate the pattern of differential characteristics with minimum effective active S-boxes.

Since a four-round differential characteristic has at least 25 active S-boxes, the number of effective active S-boxes is at least $25 - n$ if n active leaked bytes are involved. Experimental results confirm this but bring us more knowledge. We solve 11 MILP problems by setting n to be different values, that is, $n \leq 2, 3, \ldots, 8$ and $n = 9, 10, 11, 12$. Here, we choose Maple software [1] to solve them. The minimum number of effective active S-boxes, denoted by m, classified by the number of active leaked bytes is given in Table 1. Each MILP problem cost few seconds to return the optimal solution by running the code in Appendix B.

Table 1: Minimum number m of effective active S-boxes, if $(\leq)n$ active leaked bytes are included in a differential characteristic

n	≤ 2	≤ 3	≤ 4	≤ 5	≤ 6	≤ 7	≤ 8	9	10	11	12
m	23	22	21	20	19	18	17	16	16	19	18

From Table 1, we conclude that the best probability of a differential characteristic is at most 2^{-96}, since a differential characteristic has at least 16 effective active S-boxes. What is more, exactly 9 or 10 active leaked bytes are involved if a differential characteristic has 16 effective active S-boxes. An interesting observation is that the minimum number of active S-boxes (i.e., $n + m$) may be

greater than 25 if too many active leaked bytes are included in a differential characteristic, because it has to cover too many specific positions in these cases.

Now, we demonstrate that only 4 kinds of differential characteristics may have exactly 16 effective active S-boxes by analyzing the distribution of 9 or 10 active leaked bytes in a four-round differential characteristic. This is done by adding more concrete constraints to the MILP step by step. We choose the case $l_1 + l_2 + l_3 = 10$ to show the way of determining the distribution of the 10 active leaked bytes in each round. Similar process is applied to $l_1 + l_2 + l_3 = 9$. The procedure is summarized in Table 2.

Since $l_1 + l_2 + l_3 = 10$, we have $l_2 = 2, 3$ or 4. The minimum number of effective active S-boxes is 17, 20 and 16 if $l_2 = 2, 3$ and 4, respectively. Thus, to find differential characteristics with exactly 16 effective active S-boxes, we only need to consider $l_2 = 4$, which implies $l_1 + l_3 = 6$. For a further step, we have $l_1 = 2, 3$ or 4. The minimum number of effective active S-boxes is 17, 20 and 16 if $[l_1, l_2] = [2, 4]$, $[l_1, l_2] = [3, 4]$ and $[l_1, l_2] = [4, 4]$, respectively. Therefore, differential characteristics with exactly 10 active leaked bytes and 16 effective active S-boxes exist only if $[l_1, l_2, l_3] = [4, 4, 2]$. Combined with Lemma 1, $l_1 = 4$ implies $n_1 \geq 2$ and $n_1^A + n_2^A \geq 10$ since at least two columns are active in the first `MixColumns` layer; $[l_1, l_2] = [4, 4]$ implies $n_2^A + n_3^A \geq 20$; $[l_2, l_3] = [4, 2]$ implies $n_3^A + n_4^A \geq 15$ and $n_4^A \geq 4$, where $n_4^A \geq 4$ since two active leaked bytes appear in round 4 and at least two active bytes will appear in two non-leaking columns. Thus, for case $[l_1, l_2, l_3] = [4, 4, 2]$, only one possible type of differential characteristic 2-8-12-4 can be appeared.

Table 2: Minimum number m of effective active S-boxes with more constraints, the distribution of 9 or 10 active leaked bytes in these rounds, and the type of possible differential characteristic

n	additional constraints	m	$[l_1, l_2, l_3]$	Type of differential characteristic
10	$l_2 = 2$	17	discard	
	$l_2 = 3$	20	discard	
	$l_2 = 4$, $l_1 = 2$	17	discard	
	$l_2 = 4$, $l_1 = 3$	20	discard	
	$l_2 = 4$, $l_1 = 4$	16	[4,4,2]	2-8-12-4
9	$l_2 = 1$	16	[4,1,4]	4-6-9-6
	$l_2 = 2$	17	discard	
	$l_2 = 3$	21	discard	
	$l_2 = 4$, $l_1 = 1$	16	[1,4,4]	2-3-12-8
	$l_2 = 4$, $l_1 = 2$	17	discard	
	$l_2 = 4$, $l_1 = 3$	21	discard	
	$l_2 = 4$, $l_1 = 4$	16	[4,4,1]	2-8-12-3

Summary of the analysis. From the above discussion, we conclude that the number of effective active S-boxes is at least 16 in a differential characteristic. And there are four types of differential characteristics, "2–3–12–8", "2–8–12–4", "2–8–12–3" and "4–6–9–6", which can reach this lower bound.

After testing these four types of differential characteristics, we conclude that there is no differential characteristic in which each of the effective active S-box reaches the maximum differential probability 2^{-6}. The differential characteristic with best probability is of the type "2–8–12–4", and the details are given in Fig. 6. In this differential characteristic, the probability of one effective active S-box is 2^{-7}. So the overall probability of the differential characteristic is $2^{-6\times 15+(-7)} = 2^{-97}$. This is the best success rate of the forgery attack against ALE. For this differential characteristic, the values of 8 leaked bytes (from the first two rounds) should be one of the 2^8 values given in Table 6 in Appendix A. And the probability of random keystream block satisfying the requirement is $2^8/2^{8\times 8} = 2^{-56}$. If each key is restricted to protect 2^{48} message bits (2^{41} message blocks), we need to observe 2^{15} keys to find a weak keystream block to launch the attack. The experimental results of this attack on a small version of ALE are given in Sect. 6.1.

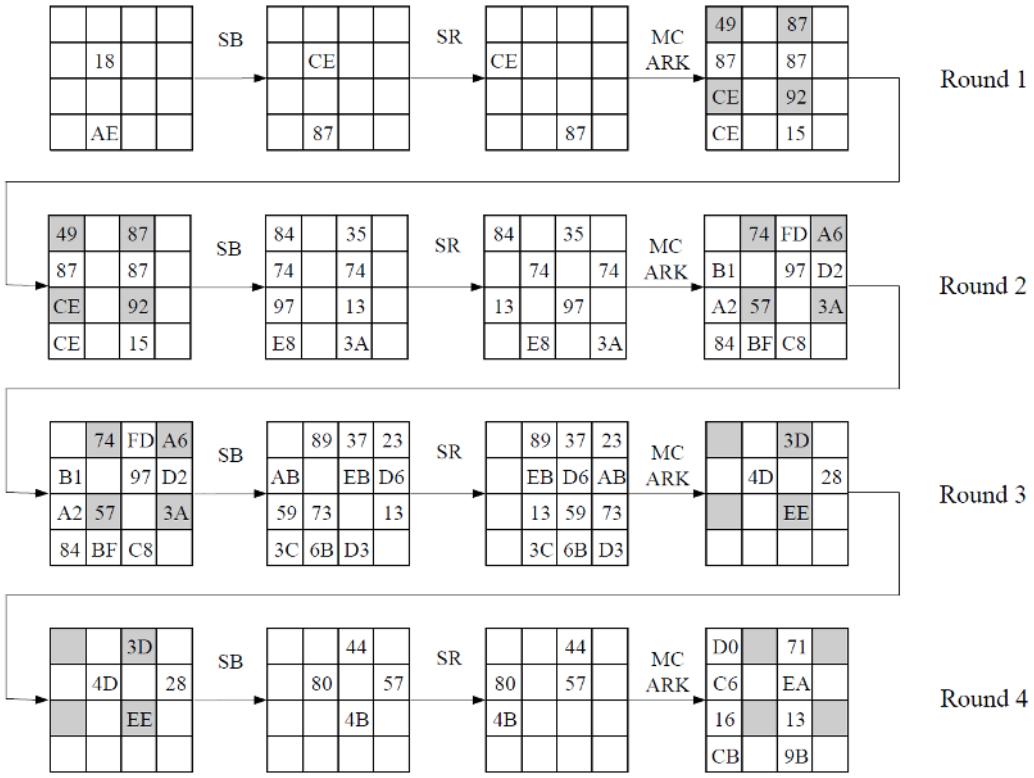

Fig. 6: Differential Path of type "2–8–12–4". The hexadecimal numbers indicate the difference values. The empty squares indicate no difference. The squares of leaked bytes are marked with gray color.

4.2 Reducing the number of known plaintext blocks

There are two approaches to reduce the number of known plaintext blocks required in the attack. One approach is to allow differential probability of 2^{-7} for some effective active S-boxes; another approach is to reduce the number of active leaked bytes in a differential characteristic. In these two approaches, with the reduced success rate, we are able to reduce the number of known plaintext blocks drastically.

Relaxing conditions on effective active S-boxes. When we try to find the best probability for the differential characteristics, it is important to restrict as many as effective active S-boxes to have probability 2^{-6} for the input and output differences. However, if we are not satisfied with the large number of plaintext blocks required to launch the attack, we can relax the condition on some active S-boxes to have probability 2^{-7}. For instance, the probability of random keystream satisfying the requirements for leaked bytes in the differential characteristic presented in Sect. 4.1 is 2^{-56}. However, if we relax the probabilities on two effective active S-boxes to 2^{-7}, this probability increases to at least than 2^{-50} because the increased number of differential characteristics is at least 2^6 by our test. It can be increased further if more conditions on effective active S-boxes are relaxed.

Reducing the number of active leaked bytes in the first two rounds. Another way to reduce the number of known plaintext blocks is to reduce the active leaked bytes in the first two rounds. The reason is that only the active leaked bytes in first two rounds are related to the conditions on leaked bytes. No matter what values the active leaked bytes are taken in Round 3, we can determine the corresponding differences after the S-box layer according to the leaked values. The only cost is an additional pre-computed look-up table. One good choice is let the number of active leaked bytes to be $[4, 0, 4]$, and the type of differential characteristic is "6-4-6-9". In this case, we only need to check conditions on the four active leaked bytes in the first round, yet we can still have a

relatively good differential probability. There are 762408 possible differential characteristics in the first two rounds when all the 17 effective active S-boxes are with probability 2^{-6}, resulting in a success rate 2^{-102} for the forgery attack. The average number of solutions for an active S-box is estimated as $2 \times 126/127 + 4 \times 1/127 = 2^{1.01}$. Therefore, the probability for a random keystream satisfying the conditions on leaked bytes is $2^{1.01 \times 4} \times 762408/2^{32} = 2^{-8.4}$. The details of one of the 762408 differential characteristics are provided in Fig. 7.

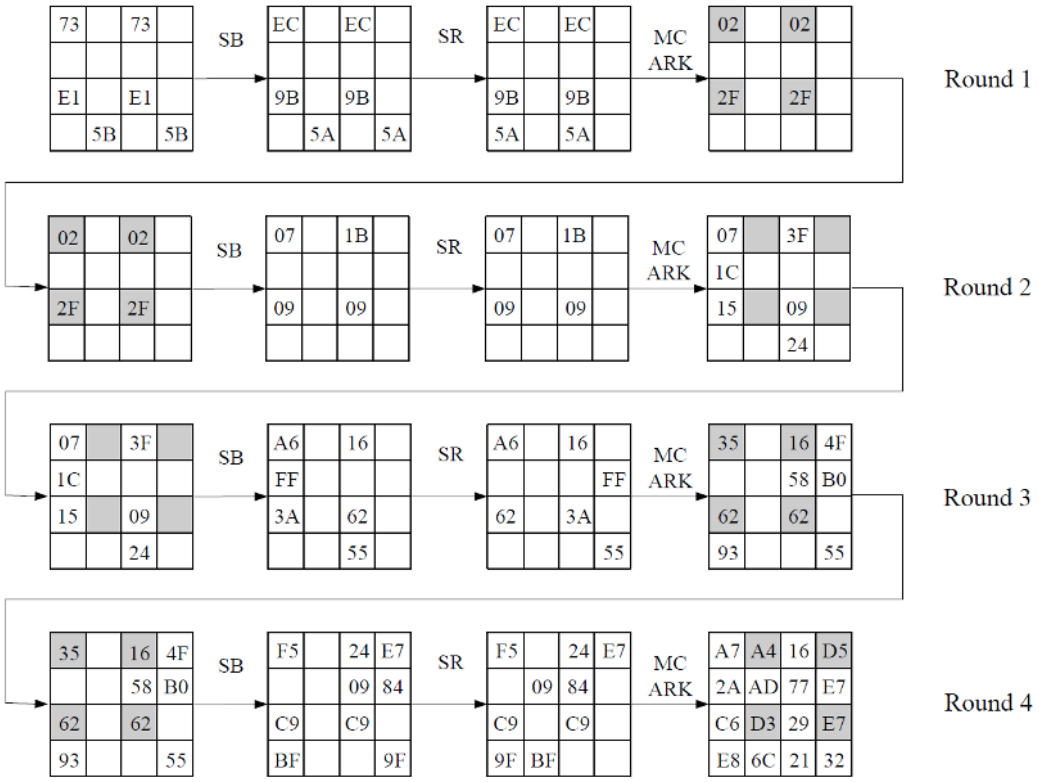

Fig. 7: Differential Path of type "6–4–6–9". The hexadecimal numbers indicate the difference values. The empty squares indicate no difference. The squares of leaked bytes are marked with gray color.

5 Effect of Removing the Whitening Key Layer

In this section, we show that the results may be further improved if the whitening key layer is removed. The success rate of a forgery attack can reach around $2^{-93.1}$, and only one or two plaintext blocks are needed to launch the attack.

Once the whitening key layer is removed, additional four bytes before the first S-box layer are known to an attacker, i.e., byte $X_{1,4}, X_{1,6}, X_{1,12}$ and $X_{1,14}$. They are obtained by xoring the previous message block and the last four leaked bytes of processing the previous message block. Thus, at most 16 leaked bytes can be exploited. In the following discussions, we denote by l_0 the number of active leaked byte before the first S-box layer, while l_1, l_2 and l_3 still indicate the number of active leaked bytes in the first three rounds respectively.

First, we analyze the smallest number of effective active S-boxes in a differential characteristic. The objective function is adjusted to minimize the value of

$$\sum_{i=1}^{4}\sum_{j=0}^{15}\chi(\Delta X_{i,j}) - \sum_{k=4,6,12,14}(\chi(\Delta X_{1,k}) + \chi(\Delta X_{3,k})) - \sum_{l=0,2,8,10}(\chi(\Delta X_{2,l}) + \chi(\Delta X_{4,l})), \quad (8)$$

since now additional four bytes are leaked before the first S-box layer. Similarly, (7) is adjusted to the following constraint

$$\sum_{k=4,6,12,14} (\chi(\Delta X_{1,k}) + \chi(\Delta X_{3,k})) + \sum_{l=0,2,8,10} (\chi(\Delta X_{2,l}) + \chi(\Delta Y_{4,l})) = n. \tag{9}$$

Notice that $n = l_0 + l_1 + l_2 + l_3$.

Table 3: Minimum number m of effective active S-boxes, if n active leaked bytes are included in a differential characteristic

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
m	30	24	23	22	21	20	19	18	17	16	15	19	18	22	21	25	24

Table 4: Minimum number m of effective S-boxes with more constraints, and the distribution of 10 active leaked bytes in these rounds

$l_1 + l_2$	additional constraints	m	$[l_0, l_1, l_2, l_3]$	Case number
	$l_1 = 0$, $\chi(\Delta X_{3,4}) + \chi(\Delta X_{3,14}) = 0$	15	[4,0,2,4]	#1
	$l_1 = 0$, $\chi(\Delta X_{3,4}) + \chi(\Delta X_{3,14}) = 1$	20	discard	
	$l_1 = 0$, $\chi(\Delta X_{3,4}) + \chi(\Delta X_{3,14}) = 2$	15	[4,0,2,4]	#2
2	$l_1 = 1$	20	discard	
	$l_1 = 2$, $\chi(\Delta X_{2,0}) + \chi(\Delta X_{2,2}) = 0$	15	[4,2,0,4]	#3
	$l_1 = 2$, $\chi(\Delta X_{2,0}) + \chi(\Delta X_{2,2}) = 1$	20	discard	
	$l_1 = 2$, $\chi(\Delta X_{2,0}) + \chi(\Delta X_{2,2}) = 2$	15	[4,2,0,4]	#4
3		20	discard	
	$l_1 = 0$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 0$	15	[2,0,4,4]	#5
	$l_1 = 0$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 1$	20	discard	
	$l_1 = 0$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 2$	15	[2,0,4,4]	#6
	$l_1 = 0$, $l_0 = 3$	20	discard	
	$l_1 = 0$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 0$	15	[4,0,4,2]	#7
	$l_1 = 0$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 1$	20	discard	
	$l_1 = 0$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 2$	15	[4,0,4,2]	#8
	$l_1 = 1$	20	discard	
4	$l_1 = 2$	18	discard	
	$l_1 = 3$	20	discard	
	$l_1 = 4$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 0$	15	[2,4,0,4]	#9
	$l_1 = 4$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 1$	20	discard	
	$l_1 = 4$, $l_0 = 2$, $\chi(\Delta X_{1,4}) + \chi(\Delta X_{1,14}) = 2$	15	[2,4,0,4]	#10
	$l_1 = 4$, $l_0 = 3$	20	discard	
	$l_1 = 4$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 0$	15	[4,4,0,2]	#11
	$l_1 = 4$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 1$	20	discard	
	$l_1 = 4$, $l_0 = 4$, $\chi(\Delta X_{4,0}) + \chi(\Delta X_{4,2}) = 2$	15	[4,4,0,2]	#12
5		20	discard	
6		17	discard	
7		20	discard	
8		16	discard	

The minimum number of effective active S-boxes classified by the number of active leaked bytes is given in Table 3. We conclude that a differential characteristic involves at least 15 effective active S-boxes. Thus, the best probability of a differential characteristic is at most 2^{-90}. For a further step, exactly 10 active leaked bytes are included in a differential characteristic with 15 effective active S-boxes, that is, $l_0 + l_1 + l_2 + l_3 = 10$. Similar to the process of Table 2, the distribution of the 10 active leaked bytes in these four rounds is studied by adding more and more constraints to the

MILP problems. This is done by first studying the sum of $l_1 + l_2$, which may be $2, \ldots, 7$ or 8, and then investigating the values of l_i $(0 \leq i \leq 3)$. The results are given in Table 4.

From Table 4, we conclude that a differential characteristic with 15 effective active S-boxes exists only if the concrete distribution of the 10 active leaked bytes satisfies

1) $[l_0, l_1, l_2, l_3] = [4, 0, 2, 4]$, $[4, 2, 0, 4]$, $[2, 0, 4, 4]$, $[4, 0, 4, 2]$, $[2, 4, 0, 4]$ or $[4, 4, 0, 2]$, and
2) $\chi(\Delta X_{i,4}) = \chi(\Delta X_{i,14})$ if $n_i = 2$ and $i \in \{1, 3\}$; $\chi(\Delta X_{i,0}) = \chi(\Delta X_{i,2})$ if $n_i = 2$ and $i \in \{2, 4\}$.

Then, we analyze all the 12 cases of differential characteristics with 15 effective active S-boxes. For each of the 12 cases listed in Table 4, different types of differential characteristics may satisfy it. In this situation, we maximize the number of effective active S-boxes in Round 1 and Round 4, as the differential probability of effective active S-boxes in these two rounds can always reach the maximum value 2^{-6} once the differential characteristic is constructed using the start-from-the-middle technique, which is also employed by the authors in [22]. The best differential characteristics we found are given as follows.

- For each of the 8 cases with $l_1 + l_2 = 4$, that is, case #5 to #12, a differential characteristic with probability of about $2^{-93.1}$ can be construct for almost all of the leaked information. Experimental results show that we can not obtain a differential characteristic for 412, 443, 402 and 373 out of 2^{32} leaked information in case #5 and #6, case #7 and #8, case #9 and #10 and case #11 and #12, respectively. Thus, in average, two plaintext blocks are enough to launch a forgery attack. The differential characteristic of case #10 is given in Appendix C.
- For each of the four cases with $l_1 + l_2 = 2$, that is, case #1 to #4, a class of 1020 differential characteristics with average probability of $2^{-94.1}$ always can be constructed, whatever the leaked information is. Thus, the forgery attack can be launched for any plaintext block. Differential paths of the case #4 are given in Appendix D.

Summary of the analysis. From the above discussion, the whitening key layer is important for ALE. Once it is removed, more internal information will be leaked to an attacker, resulting in forgery attacks with higher success rates and less required plaintext blocks. The success rate of a forgery attack now is about $2^{-93.1}$ to $2^{-94.1}$, and at most 2 plaintext blocks are needed.

6 Experiments on a Reduced Version of ALE

As a proof of concept, we would apply our attacks to ALE (with the whitening key). However, it is impossible to directly attack the original ALE as the complexity is too high. Instead, we choose to attack a reduced ALE construction based on an AES-like light-weight block cipher, LED [16].

The LED block cipher has similar round function as AES except that the operation `AddConstants` is used before the S-box layer in each round, and the round keys are added every four rounds. The S-box in LED has difference propagation probability at most 2^{-2}. Unlike the AES S-box, the output difference may not be unique to attain the best difference propagation probability. And for input difference 14, the probability 2^{-2} can never be obtained. So we need to take care of these differences in the attack.

In our experiments, we modified the LED round function so that it has the same ordered operations: `SubCells`, `ShiftRows`, `MixColumns`, `AddRoundKeys` as AES. Since the differential characteristic is not related to the key schedule, we use random round keys rather than deriving them from the key schedule. In addition, we simplified the input message to the two-block case without considering the initialization, padding and the associated data. The initial state is randomly generated.

6.1 The "2–8–12–4" differential characteristic

In the optimized forgery attacks presented in Sect. 4.1, the differential characteristic of type "2–8–12–4" is one of those have the highest success rate. We will experimentally verify the results on this type of differential characteristics.

Estimations. Using the above reduced ALE, we searched the differential characteristics of type "2–8–12–4". Like the case discussed in original ALE, we need to relax the difference propagation probability of one effective active S-box to find a valid differential characteristic. Fig. 10 in Appendix E illustrates one of the differential characteristics we found.

To estimate the probability that a random keystream block is vulnerable to the attack, we analyze the number of solutions for the values of active leaked bytes in first two rounds. In each of the first two rounds, there are 2^6 possible solutions for the values of the four active leaked bytes. Therefore, the probability of a random keystream block satisfies the conditions on leaked bytes is estimated as $2^6 \times 2^6 \times 2^{(-4)\times 8} = 2^{-20}$. The average number of plaintext blocks needed to get a vulnerable keystream block is thus $1 + 1/2^{-20} = 1 + 2^{20}$. Notice that we need an extra plaintext block to introduce the initial differences.

There are 16 effective active S-boxes in the chosen differential characteristic: 15 of the active effect S-boxes with differential probability 2^{-2}, and one with probability 2^{-3}. So the estimated probability of the differential characteristic is $2^{(-2)\times 15+(-3)\times 1} = 2^{-33}$ which is also the success rate of the forgery attack.

Experimental results. First, we check the probability of the vulnerable keystream blocks. After encrypting $2^{27.1}$ random plaintext blocks, we found 2^7 vulnerable keystram blocks. Hence, the average plaintext blocks needed to find a vulnerable keystream block is $2^{27.1-7} = 2^{20.1}$ which matches the estimated value.

Then, we verify the success rate of the forgery attack. For a vulnerable keystream block, the value of final state is xored with the second message block and stored as t_1. The differences in the final state (thus the leaked bytes) in Round 4 are determined by the values of leaked bytes in Round 3. Then we compute two forged ciphertext blocks similar to the attack procedure in Sect. 3 (but using the difference shown in Fig. 10 in Appendix E). We decrypt the forged ciphertext blocks and xor the second plaintext block from decryption with the final state to get t_2. If the two internal states t_1 and t_2 collide, we get a successful forgery. After examining $2^{36.36}$ vulnerable keystream blocks, we managed to get 10 collisions at the internal states after two blocks. So the average probability for one successful forgery is $2^{-33.04}$. One of the successful forgeries is given in Appendix E.

6.2 The "6–4–6–9" differential characteristic

In Sect. 4.2, the differential characteristics of type "6–4–6–9" (Fig. 7) are observed to require a small number of known plaintext blocks yet have good success rate. We experimentally tested this case on the reduced version of ALE.

Estimations. For this type, we found 1400 differential characteristics for the first two rounds, resulting in 21311 different values for the leaked bytes in the first round. Details of one of the differential characteristics are given in Fig. 11 in Appendix F. It is interesting to notice that certain leaked values may be used in more than one differential characteristic. If we take this into consideration, there are 28657 different leaked values related to the 1400 differential characteristics. Since there are only four active leaked bytes in the first two rounds, the probability that a random keystream is vulnerable is $28657/2^{4\times 4} = 2^{-1.12}$. Thus, the estimated number of plaintext blocks needed to find a vulnerable keystream block is $1 + 1/2^{-1.12} = 2^{1.7}$.

There are 17 effective active S-boxes in the differential characteristic. All of them attain the maximum differential probability 2^{-2}. So the estimated probability of the differential characteristic is $2^{(-2)\times 17} = 2^{-34}$, which is also the success rate of the forgery attack.

Experimental results. In our experiments, $2^{20.7}$ vulnerable keystream blocks are generated from the encryption of $2^{21.6}$ random 2-block plaintexts. So the average number of blocks needed to find one vulnerable keystream block is $2 \times 2^{21.6}/2^{20.7} = 2^{1.9}$, which is close to the estimated value.

After querying $2^{37.7}$ forged ciphertexts, we found 10 collisions in the internal states. So the average probability of successful forgery is around $2^{-34.4}$ which is close to the estimated 2^{-34}. One of the successful forgeries is given in Appendix F.

7 Conclusion

The ALE authenticated encryption algorithm is claimed with a forgery success rate of 2^{-128}. In this paper, we show that the success rate is significantly higher than the claimed rate. By applying the proposed leaked-state-forgery attack, the success rate can reach 2^{-97}. For a success rate 2^{-102}, every one out of $2^{8.4}$ plaintext blocks is vulnerable to the forgery attack. We also show that the whitening key layer is important for ALE, as the complexity of forgery attack can be improved with probabilities from $2^{-93.1}$ to $2^{-94.1}$, and at most two plaintext blocks are needed if the whitening key layer is removed. Our attacks are well-supported by the experimental results on a reduced version of ALE. Our attack confirms again that "it is very easy to accidentally combine secure encryption schemes with secure MACs and still get insecure authenticated encryption schemes" [23]. Hence, in the design of authenticated encryption algorithms, we should be very cautious in analyzing the interaction between encryption and authentication.

Acknowledgements. The authors would like to thank the anonymous reviewers of ASIACRYPT 2013 for their insightful and helpful comments on this paper. Shengbao Wu, Mingsheng Wang and Wenling Wu were supported by the National Basic Research Program of China (Grant No. 2013CB834203 and Grant No. 2013CB338002) and the National Natural Science Foundation of China (Grant No. 61272476, 61232009 and 11171323). Hongjun Wu and Tao Huang were supported by the National Research Foundation Singapore under its Competitive Research Programme (CRP Award No. NRF-CRP2-2007-03) and Nanyang Technological University NAP startup grant (M4080529.110).

References

1. Maple. Maple Software. http://www.maplesoft.com/products/maple/.
2. 3GPP. Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3, Document 1, 128-EEA3 and 128-EIA3 specification. The 3rd Generation Partnership Project (3GPP), 2010.
3. M. Agren, M. Hell, T. Johansson, and W. Meier. Grain-128a: A New Version of Grain-128 with Optional Authentication. In *International Journal of Wireless and Mobile Computing, Vol. 5, No. 1*, pages 48–59. 2011.
4. M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In *Fast Software Encryption*, pages 389–407. Springer, 2004.
5. A. Biryukov. A new 128-bit key stream cipher LEX. *eSTREAM, ECRYPT Stream Cipher Project, Report*, 13:2005, 2005.
6. A. Bogdanov, F. Mendel, F. Regazzoni, V. Rijmen, and E. Tischhauser. ALE: AES-Based Lightweight Authenticated Encryption. In *Fast Software Encryption*, 2013.
7. CAESAR. Competition for Authenticated Encryption: Security, Applicability, and Robustness. http://competitions.cr.yp.to/caesar.html.
8. J. Daemen and V. Rijmen. The Wide Trail Design Strategy. In B. Honary, editor, *Cryptography and Coding*, volume 2260 of *LNCS*, pages 222–238. Springer Berlin Heidelberg, 2001.
9. J. Daemen and V. Rijmen. *The Design of Rijndael: AES–the Advanced Encryption Standard*. Springer, 2002.
10. J. Daemen and V. Rijmen. The Pelican MAC Function. IACR ePrint Archive, Report 2005/212, 2005.
11. O. Dunkelman and N. Keller. A New Attack on the LEX Stream Cipher. In *ASIACRYPT 2008*, pages 539–556. Springer, 2008.
12. O. Dunkelman and N. Keller. Cryptanalysis of the Stream Cipher LEX. In *Des. Codes Cryptogr.*, volume 67, pages 357–373. Springer, 2013.
13. D. W. Engels, M. O. Saarinen, P. Schweitzer, and E. M. Smith. The Hummingbird-2 Lightweight Authenticated Encryption Algorithm. In *RFIDSec 2011*, pages 19–31. Springer, 2011.
14. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, and T. Kohno. Helix, Fast Encryption and Authentication in a Single Cryptographic Primitive. In *Fast Software Encryption–FSE 2003*, pages 330–346. Springer, 2003.
15. E. Fleischmann, C. Forler, and S. Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In *Fast Software Encryption–FSE 2012*, pages 196–215. Springer, 2012.
16. J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw. The LED Block Cipher. In B. Preneel and T. Takagi, editors, *Cryptographic Hardware and Embedded Systems–CHES 2011*, volume 6917 of *LNCS*, pages 326–341. Springer Berlin Heidelberg, 2011.

17. ISO/IEC 19772:2009. *Information technology – Security techniques – Authenticated encryption.* ISO, Geneva, Switzerland, 2009.
18. T. Iwata and K. Yasuda. BTM: A Single-Key, Inverse-Cipher-Free Mode for Deterministic Authenticated Encryption. In *Selected Areas in Cryptography – SAC 2009*, pages 313–330. Springer, 2009.
19. T. Iwata and K. Yasuda. HBS: A Single-Key Mode of Operation for Deterministic Authenticated Encryption. In *Fast Software Encryption–FSE 2009*, pages 394–415. Springer, 2009.
20. G. Jakimoski and S. Khajuria. ASC-1: An Authenticated Encryption Stream Cipher. In *Selected Areas in Cryptography – SAC 2011*, pages 356–372. Springer, 2011.
21. C. S. Jutla. Encryption Modes with Almost Free Message Integrity. In *Advances in Cryptology – EUROCRYPT 2001*, pages 529–544. Springer, 2001.
22. D. Khovratovich and C. Rechberger. The LOCAL attack: Cryptanalysis of the authenticated encryption scheme ALE. In *Selected Areas in Cryptography – SAC 2013*. Springer Berlin Heidelberg, 2013.
23. T. Kohno, J. Viega, and D. Whiting. CWC: A High-Performance Conventional Authenticated Encrption Mode. In *Fast Software Encryption – FSE 2004*, pages 408–426. Springer, 2004.
24. D. McGrew and J. Viega. The Galois/Counter Mode of Operation (GCM). http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm/gcm-spec.pdf.
25. N. Mouha, Q. Wang, D. Gu, and B. Preneel. Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming. In C.-K. Wu, M. Yung, and D. Lin, editors, *Information Security and Cryptology*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer Berlin Heidelberg, 2012.
26. NIST. Recommendation for Block Cipher Modes of Operation. NIST special publication 800–38A, 2001 Edition.
27. NIST. The Keyed-Hash Message Authentication Code (HMAC). FIPS PUB 198.
28. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 196–205. ACM, 2001.
29. D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). Available from csrc. nist. gov/encryption/modes/proposedmodes/ccm/ccm. pdf, 2003.
30. D. Whiting, B. Schneier, S. Lucks, and F. Muller. Phelix: Fast Encryption and Authentication in a Single Cryptographic Primitive. eSTREAM, ECRYPT Stream Cipher Project Report 2005/027.
31. H. Wu and B. Preneel. Resynchronization Attacks on WG and LEX. In *Fast Software Encryption – FSE 2006*, pages 422–432. Springer, 2006.
32. S. Wu and M. Wang. Security Evaluation against Differential Cryptanalysis for Block Cipher Structures. Cryptology ePrint Archive: Report 2011/551, 2011. `http://eprint.iacr.org/`.
33. Z. Yuan, W. Wang, K. Jia, G. Xu, and X. Wang. New Birthday Attacks on Some MACs Based on Block Ciphers. In *Advances in Cryptology – CRYPTO 2009*, pages 209–230. Springer, 2009.

A Values of Leaked-Bytes

The values of leaked bytes for the differential characteristic used in the basic LSFA in Sect. 3 are given in Table 5. The index is the byte position in the keystream block. δ_{in} and δ_{out} are the input and output differences for the S-box. α and β can be arbitrary values extracted from the leaked bytes in Round 3. From the table, the total number of possible values at the active leaked bytes in first two rounds is $2 \times 2 \times 2 \times 2 \times 4 \times 2 = 128$.

Table 5: Possible values of leaked bytes in hexadecimal for the basic LSFA. "-" indicates no difference. "$\star$" indicates arbitrary values. α and β are values from the leaked bytes.

Index	δ_{in}	δ_{out}	Value(s)
0 – 1	-	-	$\star$
2	E	42	11 or 1F
3	F3	C6	F, FC
4	59	FC	23, 7A
5	37	E5	19, 2E
6	6E	FC	0, 6E, 8C, E2
7	B2	E5	46, F4
8	-	-	$\star$
9	81	$S(\alpha) \oplus S(81 \oplus \alpha)$	α
10	6C	$S(\beta) \oplus S(6C \oplus \beta)$	β
11 – 15	-	-	$\star$

The values of leaked byes for the differential characteristic used in the optimized LSFA in Sect. 4.2 are given in Table 6. The total number of possible values at the active leaked bytes in first two rounds is 2^8.

Table 6: Possible values of leaked bytes in hexadecimal for the optimized LSFA in Sect. 4.2. "-" indicates no difference. "$\star$" indicates arbitrary values. α and β are values from the leaked bytes.

Index	δ_{in}	δ_{out}	Value(s)
0	49	84	1D or 54
1	CE	97	33, FD
2	87	35	44, C3
3	92	13	5E, CC
4	74	89	10, 64
5	57	73	B0, E7
6	A6	23	6D, CB
7	3A	13	08, 32
8 – 9	-	-	$\star$
10	3D	$S(\alpha) \oplus S(3D \oplus \alpha)$	α
11	EE	$S(\beta) \oplus S(EE \oplus \beta)$	β
12 – 15	-	-	$\star$

B Maple Program for Solving MILP Problems

We employ the function "LPSolve" included in the "Optimization" package of Maple software to solve MILP Problems. To simplify the variables in the MILP problems given in Sect. 4.1, we compress $\chi(\Delta X_{i,j})$ and $d_{i,j}$ to xij and dij here. Then, results in Table 1 are obtained by running the following program.

```
with(Optimization);
%if n<=8, the last constraint x20+x22+...+x48+x410>=n will be removed.
n:=9;
LPSolve(x10+x11+x12+x13+x14+x15+x16+x17+x18+x19+x110+x111+x112+x113
+x114+x115+x21+x23+x24+x25+x26+x27+x29+x211+x212+x213+x214
+x215+x30+x31+x32+x33+x35+x37+x38+x39+x310+x311+x313+x315
+x41+x43+x44+x45+x46+x47+x49+x411+x412+x413+x414+x415,
{x10+x15+x110+x115+x20+x21+x22+x23>=5*d11,
 x10+x15+x110+x115+x20+x21+x22+x23<=8*d11,
 x14+x19+x114+x13+x24+x25+x26+x27>=5*d12,
 x14+x19+x114+x13+x24+x25+x26+x27<=8*d12,
 x18+x113+x12+x17+x28+x29+x210+x211>=5*d13,
 x18+x113+x12+x17+x28+x29+x210+x211<=8*d13,
 x112+x11+x16+x111+x212+x213+x214+x215>=5*d14,
 x112+x11+x16+x111+x212+x213+x214+x215<=8*d14,
 x20+x25+x210+x215+x30+x31+x32+x33>=5*d21,
 x20+x25+x210+x215+x30+x31+x32+x33<=8*d21,
  x24+x29+x214+x23+x34+x35+x36+x37>=5*d22,
 x24+x29+x214+x23+x34+x35+x36+x37<=8*d22,
 x28+x213+x22+x27+x38+x39+x310+x311>=5*d23,
 x28+x213+x22+x27+x38+x39+x310+x311<=8*d23,
 x212+x21+x26+x211+x312+x313+x314+x315>=5*d24,
 x212+x21+x26+x211+x312+x313+x314+x315<=8*d24,
 x30+x35+x310+x315+x40+x41+x42+x43>=5*d31,
 x30+x35+x310+x315+x40+x41+x42+x43<=8*d31,
 x34+x39+x314+x33+x44+x45+x46+x47>=5*d32,
 x34+x39+x314+x33+x44+x45+x46+x47<=8*d32,
 x38+x313+x32+x37+x48+x49+x410+x411>=5*d33,
 x38+x313+x32+x37+x48+x49+x410+x411<=8*d33,
 x312+x31+x36+x311+x412+x413+x414+x415>=5*d34,
 x312+x31+x36+x311+x412+x413+x414+x415<=8*d34,
 x14+x16+x112+x114+x10+x11+x12+x13+x111+x110+x15+x17+x18+x19+x113+x115>=1,
 x20+x22+x28+x210+x34+x36+x312+x314+x40+x42+x48+x410<=n,
 x20+x22+x28+x210+x34+x36+x312+x314+x40+x42+x48+x410>=n
},assume=binary);
```

C Case #10: $[l_0, l_1, l_2, l_3] = [2, 4, 0, 4]$ with $\chi(\Delta X_{1,4}) = \chi(\Delta X_{1,14}) = 1$

The type of a differential characteristic is proposed in Fig. 8. The distribution of active S-boxes in these rounds is $9 \to 6 \to 4 \to 6$, totally 25 active S-boxes. In Fig. 8, from ΔX_1 to ΔZ_4, squares marked with broken line are active, squares marked with backslash should be chosen to satisfy some conditions, and empty squares have no difference.

We denote by MC the matrix used in the `MixColumns` layer. Based on the MDS property of matrix MC, once any four out of the eight differences before and after the matrix MC are given, then another four differences are uniquely determined and can be calculated efficiently.

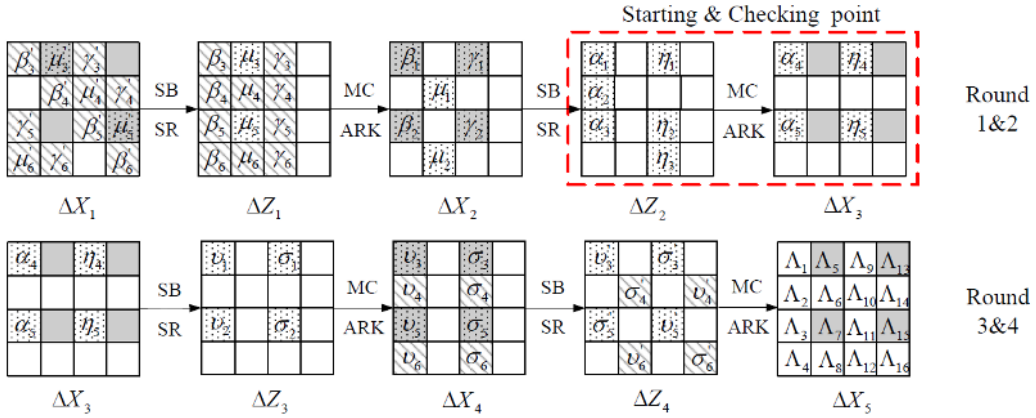

Fig. 8: A differential characteristic with $[l_0, l_1, l_2, l_3] = [2, 4, 0, 4]$ and $\chi(\Delta X_{1,4}) = \chi(\Delta X_{1,14}) = 1$. Gray squares denote leaked bytes. Squares marked with broken line are active, squares marked with backslash should be chosen to satisfy some conditions, and empty squares have no difference.

Now, we specify the differential characteristic following the type of Fig. 8. From ΔX_1 to ΔZ_4, bytes without a Greek alphabet have difference zero, and the difference of a byte with a Greek alphabet (i.e., α, β, γ, η, μ, ν and σ) will be determined in the subsequent discussions. Since $\Delta X_5 = MC(\Delta Z_4)$, we obtain the values of Λ_j $(1 \leq j \leq 16)$ once ν_i's and σ_i's $(3 \leq i \leq 6)$ are determined. The procedure of constructing this differential characteristic is given as follows.

1. Construct a differential characteristic from ΔX_2 to ΔZ_3.
   - 1-1. We start at the `MixColumns` layer of round 2, and match the differences $(\alpha_1, \alpha_2, \ldots, \alpha_5)$ first (see the starting point of Fig. 8). That is, we have to choose nonzero $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ and α_5 such that $(\alpha_4, 0, \alpha_5, 0) = (\alpha_1, \alpha_2, \alpha_3, 0) \cdot MC^t$. This is done by choosing an arbitrary difference $\alpha_1 \neq 0$ and computing $(\alpha_2, \alpha_3, \alpha_4, \alpha_5) = (4\alpha_1, 7\alpha_1, 9\alpha_1, B\alpha_1)$.
   - 1-2. Compute $\beta_1 = S^{-1}(\alpha_1 \oplus S(X_{2,0})) \oplus X_{2,0}$ and $\gamma_2 = S^{-1}(\alpha_3 \oplus S(X_{2,10})) \oplus X_{2,10}$.
   - 1-3. Choose β_2 such that one of $\beta_3, \ldots, \beta_6$ is zero, where $(\beta_3, \beta_4, \beta_5, \beta_6)^t = MC^{-1} \cdot (\beta_1, 0, \beta_2, 0)^t$. Thus, $\beta_2 \in \{D^{-1}E\beta_1, B^{-1}9\beta_1, E^{-1}D\beta_1, 9^{-1}B\beta_1\}$. Similarly, choose γ_1 such that one of $\gamma_3, \ldots, \gamma_6$ is zero. Thus, $\gamma_1 \in \{E^{-1}D\gamma_2, 9^{-1}B\gamma_2, D^{-1}E\gamma_2, B^{-1}9\gamma_2\}$.
   - 1-4. Compute $\eta_1 = S(X_{2,8}) \oplus S(X_{2,8} \oplus \gamma_1)$ and $\eta_2 = S(X_{2,2}) \oplus S(X_{2,2} \oplus \beta_2)$. Now, we have to check whether there are nonzero η_3, η_4 and η_5 such that $(\eta_4, 0, \eta_5, 0) = (\eta_1, 0, \eta_2, \eta_3) \cdot MC^t$. It is equivalent to check whether $\eta_1 = 7\eta_2$ (see the checking point of Fig. 8).
   - 1-5. If there is a $(\alpha_1, \beta_2, \gamma_1)$ such that $\eta_1 = 7\eta_2$, compute $(\eta_3, \eta_4, \eta_5) = (4\eta_2, B\eta_2, 9\eta_2)$ and go on. Else, return "construction failure" and abort.
   - 1-6. Choose μ_1, μ_2 such that $Pr(\mu_1 \to \alpha_2) \cdot Pr(\mu_2 \to \eta_3) \neq 0$ and one of μ_4, μ_6 is zero; Choose ν_1, ν_2 such that $Pr(\alpha_4 \to \nu_1) \cdot Pr(\eta_5 \to \nu_2) \neq 0$ and one of ν_4, ν_6 is zero; Choose σ_1, σ_2 such that $Pr(\eta_4 \to \sigma_1) \cdot Pr(\alpha_5 \to \sigma_2) \neq 0$ and one of σ_4, σ_6 is zero.
2. Construct the differences of outer rounds.
   - 2-1. Compute $\mu_3' = S^{-1}(\mu_3 \oplus S(X_{1,4})) \oplus X_{1,4}$ and $\mu_5' = S^{-1}(\mu_5 \oplus S(X_{1,14})) \oplus X_{1,14}$. Choose β_i' $(3 \leq i \leq 6)$ such that $Pr(\beta_i' \to \beta_i) = 2^{-6}$ if $\beta_i \neq 0$ or $\beta_i' = 0$ if $\beta_i = 0$; Choose μ_i' $(i = 4, 6)$ such that $Pr(\mu_i' \to \mu_i) = 2^{-6}$ if $\mu_i \neq 0$ or $\mu_i' = 0$ if $\mu_i = 0$; Choose γ_i' $(3 \leq i \leq 6)$ such that $Pr(\gamma_i' \to \gamma_i) = 2^{-6}$ if $\gamma_i \neq 0$ or $\gamma_i' = 0$ if $\gamma_i = 0$.
   - 2-2. Compute $\nu_3' = S(X_{4,0}) \oplus S(X_{4,0} \oplus \nu_3)$, $\nu_5' = S(X_{4,2}) \oplus S(X_{4,2} \oplus \nu_5)$, $\sigma_3' = S(X_{4,8}) \oplus S(X_{4,8} \oplus \sigma_3)$ and $\sigma_5' = S(X_{4,10}) \oplus S(X_{4,10} \oplus \sigma_5)$. Choose ν_i' $(i = 4, 6)$ such that $Pr(\nu_i' \to \nu_i) = 2^{-6}$ if $\nu_i \neq 0$ or $\nu_i' = 0$ if $\nu_i = 0$; Choose σ_i' $(i = 4, 6)$ such that $Pr(\sigma_i' \to \sigma_i) = 2^{-6}$ if $\sigma_i \neq 0$ or $\sigma_i' = 0$ if $\sigma_i = 0$.
3. Compute $\Delta X_5 = MC(\Delta Z_4)$.

Notice that 9 effective active S-boxes in Round 1 and 4 can always reach the maximum differential probability 2^{-6}. Thus, the probability of this differential characteristic is between $2^{-7 \cdot 6 - 9 \cdot 6} = 2^{-96}$ and $2^{-15 \cdot 6} = 2^{-90}$ if it exists. The existence of this differential characteristic is only related to the existence of a differential characteristic in Round 2 and 3. Two questions **Q1** and **Q2** are experimentally verified to ensure the existence of a differential characteristic from ΔX_2 to ΔZ_3:

Q1: For each $X = (X_{2,0}, X_{2,2}, X_{2,8}, X_{2,10})$, can we find a triple $(\alpha_1, \beta_2, \gamma_1)$ in step 1-1 and step 1-3 such that the condition $\eta_1 = 7\eta_2$ in step 1-4 is satisfied?

For each X, it's very likely to find such a triple, because the choices of $(\alpha_1, \beta_2, \gamma_1)$ are about 2^{12} and the probability of $\eta_1 = 7\eta_2$ is about 2^{-8}. We enumerate all 2^{32} values of X and find that the number of "construction failure" is 402, that is, there is at least one $(\alpha_1, \beta_2, \gamma_1)$ such that $\eta_1 = 7\eta_2$ for $2^{32} - 402$ out of 2^{32} X. For each of these X, we may store a candidate of $(\alpha_1, \beta_2, \gamma_1)$ in a table, which is indexed by the value of X (A redundant triple pair $(0, 0, 0)$ may be included for failure cases). The size of this table is 3×2^{32} bytes. The time complexity of this step is at most 2^{44}.

Q2: For any nonzero (α_2, η_3) (resp. (α_4, η_5) and (η_4, α_5)), can we find a pair of (μ_1, μ_2) (resp. (ν_1, ν_2) and (σ_1, σ_2)) which satisfies the conditions given in step 1-6?

Notice that (α_2, η_3) has 255^2 choices, μ_1 and μ_2 have 127 choices once (α_2, η_3) is given. Thus, **Q2** can be verified in time complexity of about 2^{30}. For a given (α_2, η_3), more than one pair of (μ_1, μ_2) may be found to satisfy the condition given in step 1-6. In this case, we choose the pair (μ_1, μ_2) such that $Pr(\mu_1 \to \alpha_2) \cdot Pr(\mu_2 \to \eta_3)$ is maximum. Experimental results show that the condition given in step 1-6 can be satisfied for each pair of (α_2, η_3), and the maximum probability of $Pr(\mu_1 \to \alpha_2) \cdot Pr(\mu_2 \to \eta_3)$ is 2^{-14}, 2^{-13} and 2^{-12} for 3825, 60690 and 510 pairs of (α_2, η_3), respectively. The average probability of $Pr(\mu_1 \to \alpha_2) \cdot Pr(\mu_2 \to \eta_3)$ is $2^{-13.03}$. Similarly, the condition given in

step 1-6 can be satisfied for each pair of (α_4, η_5) (resp. (η_4, α_5)), and the maximum probability of $Pr(\alpha_4 \to \nu_1) \cdot Pr(\eta_5 \to \nu_2)$ (resp. $Pr(\eta_4 \to \sigma_1) \cdot Pr(\alpha_5 \to \sigma_2)$) is 2^{-14}, 2^{-13} and 2^{-12} for 4312, 60203 and 510 pairs of (α_4, η_5) (resp. (η_4, α_5)), respectively. The average probability of $Pr(\alpha_4 \to \nu_1) \cdot Pr(\eta_5 \to \nu_2)$ (resp. $Pr(\eta_4 \to \sigma_1) \cdot Pr(\alpha_5 \to \sigma_2)$) is $2^{-13.04}$. The best choices of (μ_1, μ_2) and (ν_1, ν_2) (resp. (σ_1, σ_2)) can be stored in two tables.

Thus, the probability of a four-round differential characteristic proposed in this subsection is $2^{-6 \cdot 9 - 13.03 - 2 \cdot 13.04} \approx 2^{-93.1}$ on average. Notice that it always exists and can be easily rebuilt by looking up several tables.

Similar process is done to case #5 to #12 except case #10. Two questions similar to **Q1** and **Q2** are also experimentally verified to check the existence of these differential characteristics. To answer question **Q1**, 2^{32} values of $X = (X_{3,4}, X_{3,6}, X_{3,12}, X_{3,14})$ are enumerated for case #5 to case #8, and 2^{32} values of $X = (X_{2,0}, X_{2,2}, X_{2,8}, X_{2,10})$ are enumerated for case #9, #11 and #12. The number of "construction failure" is 412 for case #5 and #6, 443 for case #7 and #8, 402 for case #9, and 373 for case #11 and #12, respectively. Experimental results show that question **Q2** always can be satisfied. Therefore, we can construct these differential characteristics for almost all cases of the leaked X. The probabilities of these 7 differential characteristics are around $2^{-93.1}$ with a small deviation.

D Case #4: $[l_0, l_1, l_2, l_3] = [4, 2, 0, 4]$ with $\chi(\Delta X_{2,0}) = \chi(\Delta X_{2,2}) = 1$

The type of a differential characteristic is illustrated in Fig. 9. The distribution of active S-boxes in these rounds is $9 \to 6 \to 4 \to 6$, totally 25 active S-boxes. In Fig. 9, from ΔX_1 to ΔZ_4, squares marked with broken line are active, squares marked with backslash should be chosen to satisfy some conditions, and empty squares have no difference.

From ΔX_1 to ΔZ_4, bytes without a Greek alphabet have difference zero, and the difference of a byte with a Greek alphabet (i.e., α, β, γ, η, μ, ν and σ) will be determined in the subsequent discussions. Since $\Delta X_5 = MC(\Delta Z_4)$, we obtain the value of Λ_j $(1 \le j \le 16)$ once ν_i' and σ_i' $(3 \le i \le 6)$ are determined. The procedure of constructing this differential characteristic is briefly described as follows.

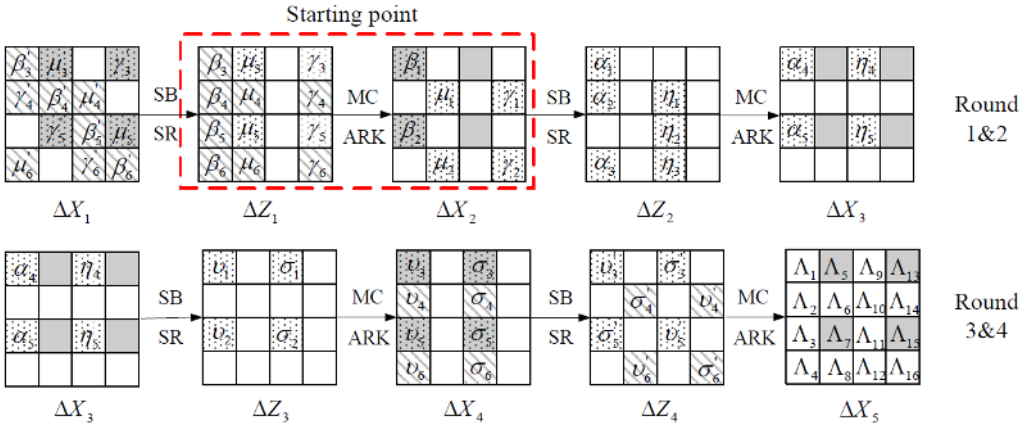

Fig. 9: Differential characteristics with $[l_0, l_1, l_2, l_3] = [4, 2, 0, 4]$ and $\chi(\Delta X_{2,0}) = \chi(\Delta X_{2,2}) = 1$. Gray squares denote leaked bytes. Squares marked with broken line are active, squares marked with backslash should be chosen to satisfy some conditions, and empty squares have no difference.

1. We start at the MC step of Round 1 here, and choose nonzero β_1 and β_2 such that one of $\beta_3, \ldots, \beta_6$ is zero, where $(\beta_3, \beta_4, \beta_5, \beta_6)^t = MC^{-1} \cdot (\beta_1, 0, \beta_2, 0)^t$. Thus, for arbitrary $\beta_1 \neq 0$, we can choose $\beta_2 \in \{D^{-1}E\beta_1, B^{-1}9\beta_1, E^{-1}D\beta_1, 9^{-1}B\beta_1\}$. $\beta_3, \ldots, \beta_6$ are obtained once β_1 and β_2 are determined. Notice that we have 4 choices of β_2 for each $\beta_1 \neq 0$.

2. Compute α_1 and η_2 using the pair $(X_{2,0}, \beta_1)$ and $(X_{2,2}, \beta_2)$, respectively.
3. Compute $\alpha_2, \ldots, \alpha_5$ by solving $(\alpha_4, 0, \alpha_5, 0) = (\alpha_1, \alpha_2, 0, \alpha_3) \cdot MC^t$; Compute η_1, η_3, η_4 and η_5 by solving $(\eta_4, 0, \eta_5, 0) = (0, \eta_1, \eta_2, \eta_3) \cdot MC^t$.
4. Choose (μ_1, μ_2) (resp. (γ_1, γ_2)) such that $Pr(\mu_1 \to \alpha_2) \cdot Pr(\mu_2 \to \eta_3) \neq 0$ (resp. $Pr(\gamma_1 \to \eta_1) \cdot Pr(\gamma_2 \to \alpha_3) \neq 0$) and one of μ_4 and μ_6 (resp. γ_4 and γ_6) is zero. Choose (ν_1, ν_2) (resp. (σ_1, σ_2)) such that $Pr(\alpha_4 \to \nu_1) \cdot Pr(\eta_5 \to \nu_2) \neq 0$ (resp. $Pr(\eta_4 \to \sigma_1) \cdot Pr(\alpha_5 \to \delta_2) \neq 0$) and one of ν_4 and ν_6 (resp. δ_4 and δ_6) is zero.
5. Compute μ'_3, μ'_5, γ'_3 and γ'_5 using the pair $(X_{1,4}, \mu_3)$, $(X_{1,14}, \mu_5)$, $(X_{1,12}, \gamma_3)$ and $(X_{1,6}, \gamma_5)$, respectively. Choose β'_i $(3 \leq i \leq 6)$ such that $Pr(\beta'_i \to \beta_i) = 2^{-6}$ if $\beta_i \neq 0$ or $\beta'_i = 0$ if $\beta_i = 0$; Choose μ'_i $(i = 4, 6)$ such that $Pr(\mu'_i \to \mu_i) = 2^{-6}$ if $\mu_i \neq 0$ or $\mu'_i = 0$ if $\mu_i = 0$; Choose γ'_i $(i = 4, 6)$ such that $Pr(\gamma'_i \to \gamma_i) = 2^{-6}$ if $\gamma_i \neq 0$ or $\gamma'_i = 0$ if $\gamma_i = 0$.
6. Compute ν'_3, ν'_5, σ'_3 and σ'_5 using the pair $(X_{4,0}, \nu_3)$, $(X_{4,2}, \nu_5)$, $(X_{4,8}, \sigma_3)$ and $(X_{4,10}, \sigma_5)$ respectively. Choose ν'_i $(i = 4, 6)$ such that $Pr(\nu'_i \to \nu_i) = 2^{-6}$ if $\nu_i \neq 0$ or $\nu'_i = 0$ if $\nu_i = 0$; Choose σ'_i $(i = 4, 6)$ such that $Pr(\sigma'_i \to \sigma_i) = 2^{-6}$ if $\sigma_i \neq 0$ or $\sigma'_i = 0$ if $\sigma_i = 0$.
7. Compute $\Delta X_{r+4} = MC(\Delta Z_4)$.

The existence of these differential characteristics is only related to the existence of pairs (μ_1, μ_2), (γ_1, γ_2), (ν_1, ν_2) and (σ_1, σ_2) in step 4. Based on the experimental results given in the construction of Fig. 8, they always exist. Thus, we have $255 \times 4 = 1020$ differential characteristics here because β_1 has 255 choices and β_2 has four choices for each β_1. The average probability of them is $2^{-6 \cdot 7 - 13.03 \cdot 2 - 13.04 \cdot 2} = 2^{-94.1}$.

E Details of one Forgery in the "2–8–12–4" Experiment

The initial state is: $0x7745fe4fa948da9$.

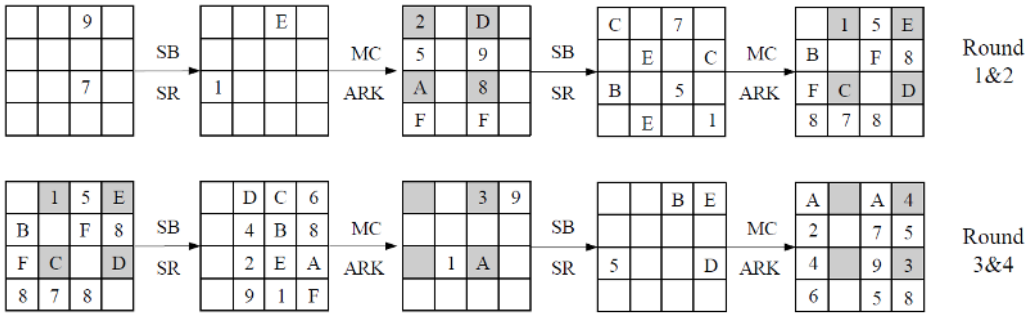

Fig. 10: Differential Path of type "2–8–12–4". The hexadecimal numbers indicate the difference values. The empty squares indicate there is no difference. The squares of leaked bytes are marked with gray color.

Table 7: The values of round keys.

	Round 1	Round 2	Round 3	Round 4
Block 1	$0x27de69bc8bbc6a71$	$0x0eda00f69a70d28f$	$0xcaa2cab4fb3cf8a8$	$0x8034f88c57ed2766$
Block 2	$0xb9cacf23fb387dd8$	$0xe9d293e0d9550016$	$0x7537baeca8ed970e$	$0xe1c9150ac5564aad$

F Details of one Forgery in the "6–4–6–9" Experiment

The initial state is: $0x92304e6d9b7c7373$.

Table 8: The forgery attack on the "2–8–12–4" differential characteristic.

	Plaintext	Ciphertext	Forged Ciphertext	Colliding State
Block 1	$0x37dc069161450099$	$0x6c2b36071e45d85d$	$0x6cbb36071e35d85d$	$0xb23d4f8eeb91a13e$
Block 2	$0xb1469433d739a810$	$0x39d7ac987dd694a8$	$0x53ba102c0d1b4435$	

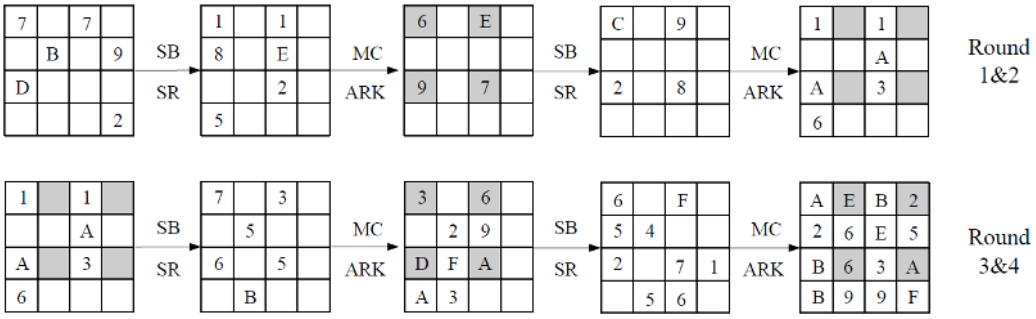

Fig. 11: Differential Path of type "6–4–6–9". The hexadecimal numbers indicate the difference values. The empty squares indicate there is no difference. The squares of leaked bytes are marked with gray color.

Table 9: The values of round keys.

	Round 1	Round 2	Round 3	Round 4
Block 1	$0x60ee23ea2d7054dd$	$0xcf849ed86e6774c0$	$0x569d49934b68af00$	$0x64b01cb5561255c8$
Block 2	$0x36a5467dc8ebe9d2$	$0xbe9da2b83ae39382$	$0x724461aa61be86e2$	$0xa396ceccaa9d57f6$

Table 10: The forgery attack on the "6–4–6–9" differential characteristic.

	Plaintext	Ciphertext	Forged Ciphertext	Colliding State
Block 1	$0x182841a869f5e890$	$0x7bb0dce1e61d0d43$	$0x0bc0d7e8361d0d41$	$0xf134343fa5b20472$
Block 2	$0x35bdb2a519a0818f$	$0xa3398abfcd7fcd1d$	$0x646cac5a462f92a8$	