Leaked-State-Forgery Attack against the Authenticated Encryption Algorithm ALE

Shengbao Wu\textsuperscript{1,3}, Hongjun Wu\textsuperscript{2}, Tao Huang\textsuperscript{2}, Mingsheng Wang\textsuperscript{4}, and Wenling Wu\textsuperscript{1}

\textsuperscript{1}Institute of Software, Chinese Academy of Sciences, China
\textsuperscript{2}Nanyang Technological University, Singapore,
\textsuperscript{3}Graduate School of Chinese Academy of Sciences, China
\textsuperscript{4}Institute of Information Engineering, Chinese Academy of Sciences, China
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Authenticated Encryption: Composition of encryption and message authentication

- Encrypt-then-MAC (IPsec)
- MAC-then-Encrypt (TLS)
- Encrypt-and-MAC

Examples of authenticated encryption schemes
- OCB, CCM, GCM, EAX, McOE, ALE,...
Introduction:
Authenticated Encryption Algorithm ALE

- ALE (Authenticated Lightweight Encryption)
  - Designed by Andrey Bogdanov et al. (FSE 2013)
  - Based on AES-128
  - Combine the ideas of LEX and Pelican MAC
  - Lightweight: 2579 GE
    - For low-cost embedded systems
  - Efficient with AES-NI
Introduction:
ALE Encryption and Authentication

Processing of associated data and the last partial block are omitted
Introduction:
LEX Leak for ALE Encryption

- Processing one plaintext block

  A whitening key is XORed with the data state → Four-round AES-128 encryption → Leaked keystream is XORed with plaintext block

5 round keys are used!

- Positions of the leaked bytes

<table>
<thead>
<tr>
<th>state</th>
<th>odd round</th>
<th>even round</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 4 8 12</td>
<td>0 4 8 12</td>
<td>0 4 8 12</td>
</tr>
<tr>
<td>1 5 9 13</td>
<td>1 5 9 13</td>
<td>1 5 9 13</td>
</tr>
<tr>
<td>2 6 10 14</td>
<td>2 6 10 14</td>
<td>2 6 10 14</td>
</tr>
<tr>
<td>3 7 11 15</td>
<td>3 7 11 15</td>
<td>3 7 11 15</td>
</tr>
</tbody>
</table>
Claim 1. State recovery: State recovery with complexity = t data blocks succeeds with prob. at most $t \cdot 2^{-128}$.

Claim 2. Key recovery: Key recovery with complexity = t data blocks succeeds with prob. at most $t \cdot 2^{-128}$, even if state recovered.

Claim 3. Forgery w/o state recovery: Forgery not involving key/state recovery succeeds with prob. at most $2^{-128}$.
Introduction:
Cryptanalysis of ALE

- Khovratovich and Rechberger’s attack (SAC 2013)
  - Forgery attack
    - Bytes are leaked after SubByte – a variant of ALE. The actual leak in ALE is before SubByte
    - Complexity is from $2^{102}$ to $2^{119}$ depending on the amount of data
  - State recovery attack
    - Requires $2^{120}$ forgery attempts of 48 byte messages
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
  - The main idea of the attack
  - Finding a differential characteristic
  - Launching the forgery attack
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Basic Attack: The Main Idea of the 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.

- In ALE, 4 state bytes are leaked at the end of every round
- It is possible to bypass some active S-boxes with probability 1!
Basic Attack:
An example of 1-4-16-4 differential characteristic

Round 1
SB 96 → F3 → F3 → 0E
FD F3 F3

Round 2
SB 0E 42 37 C6 6E
FD F3

Round 3
SB C6 59 84 6E 51 6E 42 6E
97 37 42 B2 C6 37 C6 DC
SB D7 FC 3E FC 4F FC 4F 21 8A E5 4F E5 D7 F9 D7 E5
SR D7 FC 3E FC 4F FC 4F 21 4F E5 E5 8A E5 D7 F9 D7
MC ARK C6 59 84 6E 51 6E 42 6E 97 37 42 B2 C6 37 C6 DC
Round 4
SB 6C 81 81
81
SB 33 6F 6F
6F
SR 33 6F 6F
6F
MC ARK B1 B8 DE 5C 6F 82 6F 55
Basic Attack:
An example of 1-4-16-4 differential characteristic

- **Input difference:**
  \[
  \Delta_{\text{in}} = (0,0,0,0; 0,0,0,0; 0,0,0,0; 0,96,0,0)
  \]

- **Output difference:**
  \[
  \Delta_{\text{out}} = (B1,DE,6F,6F; 0,0,0,0; B8,5C,82,55; 0,0,0,0)
  \]

- **Keystream difference:**
  \[
  \Delta_{s} = (0,0,E,F3; 59,37,6E,F2; 0,81,6C,0; 0,0,0,0)
  \]
Basic Attack: Launching the Forgery Attack

- Determine possible values of leaked bytes. Store the values in a table $T$
  - Example: For $\delta_{in} = 0xf3$, $\delta_{out} = 0xc6$, the values are 0xf or 0xfc
- Find a keystream block $s_i$ which falls into one of the possible values of table $T$
- Modify ciphertext blocks: $c'_{i-1} = c_{i-1} \oplus \Delta_{in}$, $c'_i = c_i \oplus \Delta_{out} \oplus \Delta_s$
- Send the modified ciphertext for decryption/verification
Basic Attack: Launching the Forgery Attack

- In decryption/verification:
  - $\Delta m_{i-1} = (c_{i-1} \oplus s_{i-1}) \oplus (c'_{i-1} \oplus s'_{i-1}) = \Delta_{in}$, because $\Delta s_{i-1} = 0$
  - $\Delta m_i = (c_i \oplus s_i) \oplus (c'_i \oplus s'_i) = \Delta_{out}$, because $c_i \oplus c'_i = \Delta_{out} \oplus \Delta_s$
  - when $\Delta m_{i-1}$ is introduced to the data state, after four rounds, $\Delta m_i$ will cancel the difference in the state

- Complexity of the Attack
  - Before considering the leaked bytes: $2^{-6 \times 16 + (-7) \times 9} = 2^{-159}$
  - 8 active leaked bytes: 5 with prob. $2^{-7}$, 3 with prob. $2^{-6}$
  - Overall probability: $2^{-159} \times 2^{7 \times 5} \times 2^{6 \times 3} = 2^{-106}$
  - Number of known plaintext blocks: $128/2^{6 \times 8} = 2^{-41}$
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
  - Improving the differential probability
  - Reducing the number of known plaintext blocks
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Improving the Differential Probability

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.

- Use the Mixed-Integer Linear Programming (MILP) technique [Mouha, Wang, Gu, Preneel ’11] to study the smallest number of effective active S-boxes.
Improving the Differential Probability

Let $X_i$ be the input state of round $i$, $X_{i,j}$ be the $j$-th byte of $X_i$. We introduce a function $\chi(x)$ such that $\chi(x) = 1$ if $x \neq 0$ and $\chi(x) = 0$ if $x = 0$.

The objective function is to minimize:

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

- Constraints from Property 1:

\[ 5d_{i,1} \leq \sum_{j=0}^{3} (\chi(\Delta X_{i,5j \mod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,1}, \]

\[ 5d_{i,2} \leq \sum_{j=4}^{7} (\chi(\Delta X_{i,5j \mod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,2}, \]

\[ 5d_{i,3} \leq \sum_{j=8}^{11} (\chi(\Delta X_{i,5j \mod 16}) + \chi(\Delta X_{i+1,j})) \leq 8d_{i,3}, \]

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

where \( i \in \{1, 2, 3\} \) and \( d_{i,j} \in \{0, 1\} \) (1 \leq j \leq 4)
Improving the Differential Probability

- Additional Constraints
  - Avoid trivial solution:
    \[ \sum_{j=0}^{15} \chi(\Delta X_{1,j}) \geq 1 \]
  - when number of active leaked byte is \( n \) or \( \leq n \)

\[ \sum_{k=0,2,8,10} (\chi(\Delta X_{2,k}) + \chi(\Delta X_{4,k})) + \sum_{l=4,6,12,14} \chi(\Delta X_{3,l}) = n \text{ (or } \leq n) \]
Improving the Differential Probability

- Use Maple to solve 11 MILP problems when $n \leq 2, 3, \ldots, 8$ and $n = 9, 10, 11, 12$. Minimum number of effective active S-boxes is:

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\leq 2$</th>
<th>$\leq 3$</th>
<th>$\leq 4$</th>
<th>$\leq 5$</th>
<th>$\leq 6$</th>
<th>$\leq 7$</th>
<th>$\leq 8$</th>
<th>$9$</th>
<th>$10$</th>
<th>$11$</th>
<th>$12$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$</td>
<td>23</td>
<td>22</td>
<td>21</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>16</td>
<td>19</td>
<td>18</td>
</tr>
</tbody>
</table>

- At least 16 effective active S-boxes in a differential char.
- Four possible types, “2-3-12-8”, “2-8-12-4”, “2-8-12-3” and “4-6-9-6”, can reach this lower bound.
Improving the Differential Probability

- The differential characteristic with best probability is of the type “2-8-12-4”.
Improving the Differential Probability

- Complexity of the attack
  - 16 effective active S-boxes, 15 with prob. $2^{-6}$, 1 with prob. $2^{-7}$. Hence, prob. of the differential characteristic is $2^{-97}$.
  - The prob. of random keystream block satisfying the requirement is $2^{-56}$. If each key is restricted to protect $2^{48}$ message bits ($2^{41}$ message blocks), we need to observe $2^{15}$ keys to launch the attack.
Reducing the number of known plaintext blocks

- Relaxing conditions on effective active S-boxes
  - Relax the prob. of some effective active S-boxes from $2^{-6}$ to $2^{-7}$ – more choices for differential characteristics.
- Reducing the number of active leaked bytes in the first two rounds
  - Only the active leaked bytes in the first two rounds are considered to satisfy the conditions.
  - The differential characteristic “6-4-9-6” needs $2^{8.4}$ blocks to find one vulnerable keystream block and the success rate is $2^{-102}$
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Effect of Removing the Whitening Key Layer

- When the whitening key layer is removed, additional four bytes before the first S-box layer are known.
- Objective function is changed to:

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

- Constraint on number of active leaked byte is changed to:

\[
\sum_{k=4,6,12,14} (\chi(\Delta X_{1,k}) + \chi(\Delta X_{3,k})) + \sum_{l=0,2,8,10} (\chi(\Delta X_{2,l}) + \chi(\Delta Y_{4,l})) = n
\]
Effect of Removing the Whitening Key Layer

- Minimum number of effective active is reduced to 15.

- 12 cases of differential characteristics.
  - For case #1 to #4, with average prob. of $2^{-94.1}$, a class of 1020 differential characteristics always can be constructed.
  - For case #5 to #12, with average prob. of $2^{-93.1}$, two plaintext blocks are enough to launch a forgery attack.
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Experiments on a Reduced Version of ALE

- Attack a reduced ALE construction based on an AES-like light-weight block cipher LED [Guo, Peyrin’11].

- The settings:
  - Four ordered operations in the round function
    - SubCells, ShiftRows, MixColumns, AddRoundKeys
  - LED S-box is used in SubCells, and random round keys are used instead of deriving them from the key schedule
  - Only consider two-block input message without considering the initialization, padding and the associated data
  - The initial state is randomly generate
Experiments on a Reduced Version of ALE

- Experimental results for the “2-8-12-4” differential char.
  - Average number of blocks to find a vulnerable keystream is $2^{20.1}$ ($2^{20}$ for estimation)
  - Average probability for one successful forgery is $2^{-33.04}$ ($2^{-33}$ for estimation)

- Experimental results for the “6-4-6-9” differential char.
  - Average number of blocks to find a vulnerable keystream is $2^{1.9}$ ($2^{1.7}$ for estimation)
  - Average probability for one successful forgery is $2^{-34.4}$ ($2^{-34}$ for estimation)
Experiments on a Reduced Version of ALE

- The “2-8-12-4” differential characteristic

- An example of the forgery attack

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertext</th>
<th>Forged Ciphertext</th>
<th>Colliding State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1</td>
<td>0x37dc069161450099</td>
<td>0x6c2b36071e45d85d</td>
<td>0x6cbb36071e35d85d</td>
</tr>
<tr>
<td>Block 2</td>
<td>0xb1469433d739a810</td>
<td>0x39d7ac987dd694a8</td>
<td>0x53ba102c0d1b4435</td>
</tr>
</tbody>
</table>
Experiments on a Reduced Version of ALE

- The “6-4-6-9” differential characteristic

- An example of the forgery attack

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Ciphertext</th>
<th>Forged Ciphertext</th>
<th>Colliding State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block 1, $0x\text{182841a869f5e890}$</td>
<td>$0x\text{7bb0dce1e61d0d43}$</td>
<td>$0x\text{0bc0d7e8361d0d41}$</td>
<td>$0x\text{f134343fa5b20472}$</td>
</tr>
<tr>
<td>Block 2, $0x\text{35bdb2a519a0818f}$</td>
<td>$0x\text{a3398abfcd7fcd1d}$</td>
<td>$0x\text{646cac5a462f92a8}$</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Introduction
- A Basic Leaked-State-Forgery Attack on ALE
- Optimized Attack
- Effect of Removing the Whitening Key Layer
- Experiments on a Reduced Version of ALE
- Conclusion
Conclusion

- We proposed the leaked-state-forgery (LSFA) attack against ALE.
  - The authentication security of ALE is only 97-bit rather than 128-bit.
  - If the whitening key layer is removed, the security can be reduced to around 93-bit.
- We experimentally verified our attack against a small 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”. [Kohno, Viega, Whiting’03]
Thank you!