Cryptanalysis of Lightweight Authenticated Cipher ACORN [*]

Meicheng Liu and Dongdai Lin

State Key Laboratory of Information Security, Institute of Information Engineering
Chinese Academy of Sciences, Beijing 100093, P. R. China
meicheng.liu@gmail.com

Abstract. In this paper, we study the lightweight authenticated cipher ACORN designed by H. Wu for CAESAR competition. First we describe slid properties of keys and IVs for the cipher. For each (Key, IV) pair, the probability that there is another pair which generates an identical state up to a clock difference is found to be 1. Meanwhile, for each key, the probability that there exist at least two IVs which generate an identical state up to a clock difference using this key is also shown to be 1. Then we propose state recovering attacks on ACORN-128 using guess-and-determine and differential-algebraic techniques. The time complexities of these attacks are respectively about 2^{180} and 2^{130} CPU cycles in the single (Key, IV) pair setting and the two related (Key, IV) pairs setting. The success probability of the latter is 0.56. Since the state-update function of ACORN-128 is invertible, we can mount a key recovering attack from a state recovering attack. The results show that ACORN-128 has weak boundary of 128-bit security under related-key state recovering attacks and chosen-IV state recovering attacks.

Keywords: Authenticated Cipher, Related Key, Guess and Determine, Algebraic Attack

1 Introduction

The lightweight authenticated stream cipher ACORN was recently proposed by H. Wu [1] and submitted to CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness). This cipher is very simple and has only 128-bit version at present. ACORN-128 contains a 293-bit internal state, and is designed to protect up to 2^{64} bits of associated data (AD) and up to 2^{64} bits of plain-text by using a 128-bit secret key, a 128-bit initial value (IV) and an ℓ_{tag}-bit authentication tag where $64 \leq \ell_{tag} \leq 128$. The use of a 128-bit tag is recommended by the designer. The following requirements are claimed by the designer to be satisfied when using ACORN.

1. Each key should be generated in a secure way.

2. Each key and IV pair should not be used to protect more than one message; and each key and IV pair should not be used with two different tag sizes.

3. If verification fails, the decrypted plain-text and the wrong authentication tag should not be given as output.

In this paper, we analyze the cipher ACORN-128 mostly from the point of view of confidentiality. At first, we describe slid properties for the cipher. Two distinct (Key, IV) pairs which generate an identical state up to a clock difference are said to be slidable. There are a series of papers on the topic related to slid pairs, e.g. [2]. The number of slid pairs for ACORN-128 whose relations are determined is found to be more than 2^{227}. For each pair of (Key, IV), there always exist another pair which generate a same state as that of (Key, IV) up to a clock difference. There also exist slid pairs with identical keys. Such keys are called to be slidable. That is, for a slid key, there exist two distinct IVs which generate an identical state up to a clock difference.

[*] This work was supported by the National 973 Program of China under Grant 2011CB302400, the National Natural Science Foundation of China under Grant 61303258, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06010701, and SKLOIS Research Project under Grant 2014-ZD-04.

The number of slid keys determined by explicit formulations is more than 2^{92}. Theoretically, the probability that a random key is slidable is shown to be one.

Further we propose state recovering attacks on ACORN-128 in both contexts of using a single (Key, IV) and two related (Key, IV) pairs. In the state recovering attacks, given a keystream z of length n an attacker attempts to recover the internal state of the cipher. In the related-key state recovering attacks, given keystreams of some lengths generated by related keys an attacker attempts to recover the internal state of the cipher, where the values of these keys are initially unknown but some mathematical relationship connecting the keys is known to or can be controlled by the attacker. Related-key attacks were introduced by Biham [3]. In the related-key attacks on ACORN-128, we assume that two slid pairs are used with chosen associated data. For the case of slid keys, related-key attacks become chosen-IV single-key attacks.

The attack for the single case is based on guess-and-determine technique. In this attack, 130 values are guessed and an approximately linear system of 293 equations with probability $2^{-33.6}$ is set up. The time complexity is $c \cdot 2^{163.6}$ CPU cycles and the data complexity is about $2^{33.6}$ keystream bits, where $c \approx 2^{16}$ is the time complexity of solving a sparse linear system of 293 equations over $GF(2)$. The average-case time complexity of Gaussian elimination for solving a linear system of n equations over $GF(2)$ is about $n^3/6$ operations, and equals around $n^3/384$ CPU cycles when carefully implemented on a modern 64-bit processor. For a sparse linear system, the average running time becomes about $2n$ when implemented on a parallel hardware architecture proposed by Bogdanov *et al.* [4]. In this paper, we evaluate the time complexity for solving a sparse linear system to be $n^3/384$ CPU cycles.

For the case of two related (Key, IV) pairs, we present a chosen plain-text state recovering attack on ACORN-128. The attack are based on differential-algebraic and guess-and-determine techniques. Note that the nonlinear confusion of the cipher ACORN-128 heavily depends on a Boolean function $xy+xz+yz$. We describe a differential-algebraic property for this function and use this property to set up linear or approximately linear equations. In the attack, we guess 108 values, and get a probabilistic linear system of 293 equations with probability $2^{-6.64}$. This attack has time complexity of $c \cdot 2^{114.64}$ with success probability 0.56, where $c \approx 2^{16}$, by using a pair of 49-bit chosen and 115-bit known plain-texts. This result shows that ACORN-128 does not have 128-bit security if an attacker considers related-key state recovering attacks and chosen-IV state recovering attacks. Also, it shows that slid keys are weak keys.

The remainder of this paper is organized as follows. In Section 2 the description of ACORN-128 is provided. Section 3 discusses slid properties for the cipher. In section 4, we propose state recovering attacks on ACORN-128. Section 5 concludes the paper.

2 Description of ACORN-128

ACORN-128 uses a 128-bit secret key and a 128-bit initial value, and contains a 293-bit internal state denoted by $s = (s_0, s_1, \cdots, s_{292})$. The state consists of six concatenated LFSRs, as shown in Fig.1.

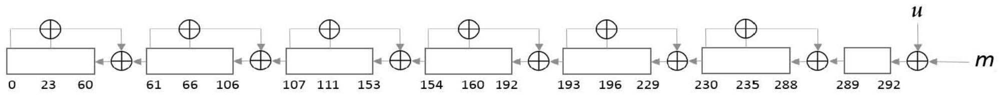

Fig. 1. Internal state of ACORN-128

2.1 Functions of ACORN-128

In this section, the operations "+" and "·" may be considered as addition and multiplication operations modulo 2 respectively, if there is no ambiguity.

Linear Transformation. Taking no account of the shift, the role of LFSRs can be considered as a linear transformation $L : x \mapsto y$ which is described below,

$$\begin{aligned}
&y_{289} = x_{289} + x_{235} + x_{230} \\
&y_{230} = x_{230} + x_{196} + x_{193} \\
&y_{193} = x_{193} + x_{160} + x_{154} \\
&y_{154} = x_{154} + x_{111} + x_{107} \\
&y_{107} = x_{107} + x_{66} + x_{61} \\
&y_{61} = x_{61} + x_{23} + x_0 \\
&y_i = x_i, \text{ for } 0 \le i \le 292 \text{ and } i \notin \{0, 61, 107, 154, 193, 230, 289\}.
\end{aligned} \tag{1}$$

Keystream-Bit Function. The keystream bit z is generated by the Boolean function $g(x)$ whose algebraic expression is listed as follows,

$$g(x) = x_{12} + x_{154} + x_{61}x_{193} + x_{61}x_{235} + x_{193}x_{235}.$$

The output of g only depends the input bits $x_{12}, x_{154}, x_{61}, x_{193}, x_{235}$. Before generating the keystream bit, the internal state s is transformed by L, that is, $z = g(L(s))$.

Feedback-Bit Function. The feedback bit u is generated by the Boolean function $f(x, a, b)$ whose algebraic expression is listed as follows,

$$f(x,a,b) = 1+x_0+x_{66}+x_{107}+x_{23}x_{160}+x_{23}x_{244}+x_{160}x_{244}+x_{66}x_{230}+x_{111}x_{230}+ax_{196}+bg(x),$$

where a and b are constants. For fixed a and b, the output of f only depends the input bits $x_0, x_{23}, x_{66}, x_{107}, x_{111}, x_{160}$,$x_{196}, x_{230}, x_{244}$ and the output of $g(x)$. The constant b is set to be one in the stages of the initialization, processing associated data and finalization and to be zero in the encryption stage. Before generating the feedback bit, the internal state s is transformed by L, that is, $u = f(L(s), a, b)$.

State-Update Function. The state is updated by the function F with input bits $s_0, s_1, \cdots, s_{292}, m, a$ and b. The state-update function $F(s, m, a, b)$ is described in the pseudocode below.

```
$s = L(s)$
$u = f(s, a, b)$
for $i$ from 0 to 291 do
    $s_i = s_{i+1}$
end for
$s_{292} = u + m$
```

The state update function is invertible on s for any given a, b and m.

2.2 The initialization

The initialization is given by the following pseudo-code.

```
$s = (0, \cdots, 0)$
$(m_0, m_1, \cdots, m_{1535}) \leftarrow (K_0, \cdots, K_{127}, IV_0, \cdots, IV_{127}, 1, 0, \cdots, 0)$
for $i$ from 0 to 1535 do
    $(a, b) \leftarrow (1, 1)$
    $s = F(s, m_i, a, b)$
end for
```

2.3 Processing the associated data

After the initialization, the associated data AD is used to update the state.

```
$l \leftarrow$ bit length of associated data $AD$
$(m_0, m_1, \cdots, m_{l+511}) \leftarrow (AD_0, \cdots, AD_{l-1}, 1, 0, \cdots, 0)$
for $i$ from 0 to $l + 255$ do
    $(a, b) \leftarrow (1, 1)$
    $s = F(s, m_i, a, b)$
end for
for $i$ from 0 to 255 do
    $(a, b) \leftarrow (0, 1)$
    $s = F(s, m_{l+256+i}, a, b)$
end for
```

2.4 The encryption

After processing the associated data, at each step of the encryption, one plain-text bit p_i is used to update the state, and p_i is encrypted by XOR z_i.

```
$l \leftarrow$ bit length of plain-text
$(m_0, m_1, \cdots, m_{l+511}) \leftarrow (p_0, \cdots, p_{l-1}, 1, 0, \cdots, 0)$
for $i$ from 0 to $l - 1$ do
    $z_i = g(L(s))$
    $(a, b) \leftarrow (1, 0)$
    $s = F(s, m_i, a, b)$
end for
for $i$ from 0 to 255 do
    $(a, b) \leftarrow (1, 0)$
    $s = F(s, m_{l+i}, a, b)$
end for
for $i$ from 0 to 255 do
    $(a, b) \leftarrow (0, 0)$
    $s = F(s, m_{l+256+i}, a, b)$
end for
```

2.5 The finalization

After the stage of encryption, the authentication tag T with bit length ℓ_{tag} is generated by the finalization, which is described by the following pseudo-code.

```
$(m_0, m_1, \cdots, m_{511}) \leftarrow (0, \cdots, 0)$
for $i$ from $0$ to $511 - \ell_{tag}$ do
    $(a, b) \leftarrow (1, 1)$
    $s = F(s, m_i, a, b)$
end for
for $i$ from $0$ to $\ell_{tag} - 1$ do
    $T_i = g(L(s))$
    $(a, b) \leftarrow (1, 1)$
    $s = F(s, m_{512-\ell_{tag}+i}, a, b)$
end for
```

3 Slid Pairs of ACORN

Two distinct (Key, IV) pairs which generate an identical state up to a clock difference are said to be slidable. For ACORN-128, slid pairs which generate an identical state up to a clock difference in initialization or processing the associated data can generate an identical state without a clock difference in encryption for chosen associated data.

It is difficult to find slid pairs for ACORN-128 in a straight way. Instead, we try to search for slid pairs starting from 256-th round. Our observation is that the first 256 rounds mapping (Key, IV) to the state of 256th round is a bijection. Denote by $s = (0, 0, \cdots, 0, s_{37}, s_{38}, \cdots, s_{292})$ and $c = (0, 0, \cdots, 0, c_{37}, c_{38}, \cdots, c_{292})$ two states of 256th round of initialization. Denote by s^t the state at round $t+256$, and denote by s_i^t the i-th bit of s^t. For $0 \leq t_c < t_s \leq 1279$ the two states s^{t_s} and c^{t_c} are identical if and only if $s^{t_s - t_c + 1} = F(c, 1, 1, 1)$. For $t_s \geq 1280$, i.e., $t_s + 256 \geq 1536$, the latter is sufficient but not necessary. There exist s and c such that $s^{t_s - t_c + 1} = F(c, 1, 1, 1)$ if and only if there is s satisfying $s_0^{t_s - t_c} = 1$ and $s_i^{t_s - t_c + 1} = 0$ for $0 \leq i \leq 35$. Let $t = t_s - t_c + 1$. We can directly solve these equations for $38 \leq t \leq 257$. All these systems of equations are linear. Each system consists of 37 linear equations in 256 indeterminates and has therefore 2^{219} solutions. The total number of solutions for such s is around 2^{227}. The lower bound of the probability that a random (Key, IV) pair is slidable at the initialization is thus about 2^{-29}.

Now, we discuss the possibility for ACORN-128 that two distinct (Key, IV) pairs generate an identical state in the stage of processing the associated data up to a clock difference. As the previously mentioned, a state s in the form of $(0, 0, \cdots, 0, s_{37}, s_{38}, \cdots, s_{292})$ corresponds to a state of 256th round of initialization which corresponds to a pair of (Key, IV). If a (Key, IV) pair generate such state in the stage of processing the associated data, which occurs with probability 2^{-37}, then a slid pair can be obtained after setting the following 1280 bits of the associated data to be $(1, 0, 0, \cdots, 0)$. There always exist such states for chosen associated data up to 37 bits. Nevertheless, it seems very difficult to compute the exact relations for such slid pairs with unknown keys.

Slid Pairs with a Same Key. Next we consider the case for a same key with distinct IVs. We call a key is slidable if there are two distinct IV and IV$'$ such that (Key, IV) and (Key, IV$'$) generate an identical state up to a clock difference. Denote $\mathcal{I} = \{0, 61, 107, 154, 193, 230, 289\}$. The initialization gives a one-to-one correspondence between the key and the 128 bits $s_{37}, s_{38}, \cdots, s_{164}$

of the state s at 256th round. That is, an identical key gives $c_i = s_i$, $37 \le i \le 164$. Note that for $t \le 1280$, $F(s^{t-1}, 0, 1, 1) = s^t = F(c, 1, 1, 1)$ if and only if $s_i^{t-1} = c_i$ for $i \notin \mathcal{I}$ and $s_i^{t-1} = c_i + 1$ for $i \in \mathcal{I}$. Since for $165 \le i \le 292$ each equation contains a unique linear indeterminate c_i, the system $s^t = F(c, 1, 1, 1)$ has solutions if and only if the following system

$$s_i^{t-1} = \begin{cases} 1, & i = 0; \\ 0, & 1 \le i \le 36; \\ s_i, & 37 \le i \le 164 \text{ and } i \ne 61, 107, 154; \\ s_i + 1, & i = 61, 107, 154, \end{cases} \tag{2}$$

has solutions. The above system consists of 165 equation in 256 indeterminates. Since s_i^{t-1} is linear on s for $i + t - 1 < 293$, the system (2) is linear and has solutions for $38 \le t \le 129$. In particular, s_i^{t-1} is linear on $s_{37}, s_{38}, \cdots, s_{164}$ for $i + t - 1 < 165$. The number of solutions for $s_{37}, s_{38}, \cdots, s_{164}$ is 2^{t-38} for $38 \le t \le 129$. In the other words, there are at least $2^{92} - 1$ keys which are slidable.

As a matter of fact, there are more slid keys. For $t > 1280$, $F(s^{t-1}, m_{t-1}, 1, 1) = s^t = F(c, 1, 1, 1)$ if and only if $s_i^{t-1} = c_i$ for $i \notin \mathcal{I}$ and $s_i^{t-1} = c_i + m_{t-1} + 1$ for $i \in \mathcal{I}$. Similarly, the system $s^t = F(c, 1, 1, 1)$ has solutions if and only if the following system

$$s_i^{t-1} = \begin{cases} m_{t-1} + 1, & i = 0; \\ 0, & 1 \le i \le 36; \\ s_i, & 37 \le i \le 164 \text{ and } i \ne 61, 107, 154; \\ s_i + m_{t-1} + 1, & i = 61, 107, 154, \end{cases} \tag{3}$$

has solutions. For a fixed key, the system (3) holds with a probability 2^{-165}. Assume that each bit of the state is sufficiently random after initialization. The probability becomes one over the space of 128-bit IV and 37-bit associated data. Therefore, each key has a probability 1 to be slidable.

4 State Recovering Attacks on ACORN

In the state recovering attacks, given a keystream z of length n an attacker attempts to recover the internal state of the cipher. In the related-key state recovering attacks, given keystreams of some lengths generated by related keys an attacker attempts to recover the internal state of the cipher, where the values of these keys are initially unknown but some mathematical relationship connecting the keys is known to or can be controlled by the attacker. In this section, we analyze the cipher ACORN in both points of view.

4.1 The case of a single (Key, IV) pair

In this section, we describe state recovering attacks on ACORN-128 using a single (Key, IV) pair, which takes advantage of guess-and-determine method.

Denoted by s the internal state of ACORN-128. The keystream bit function is

$$g(x) = x_{12} + x_{154} + x_{61}x_{193} + x_{61}x_{235} + x_{193}x_{235}, \tag{4}$$

where $x = L(s)$. By (1), we have $x_{193} = s_{193} + s_{160} + s_{154}$ and $x_{235} = s_{235}$. If we guess the values of x_{193} and x_{235}, then $g(x)$ is linear on x and on s. In other words, given the rightmost 139 bits $s_{154}, s_{155}, \cdots, s_{292}$ of the state, the keystream bit function $g(L(s))$ is linear on the other 154 bits $s_0, s_1, \cdots, s_{153}$ which are consisted of the leftmost three LFSRs, and $g(L(s)) = z$ is always linear if guessing the values of the feedback bits u's, see also Fig.1. One can obtain 58 linear equations after guessing $s_{154}, s_{155}, \cdots, s_{292}$. A trivial attack is setting up another 96 linear equations by guessing the values of the first 48 feedback bits u's, where $u = f(L(s))$'s are also linear on s. Then 154 linear equations on 154 variables are set up, and therefore the state can be recovered in $c \cdot 2^{139+48}$ with data complexity of 106 bits, where $c \approx 2^{13}$ is the time complexity of solving this system of 154 linear equations.

Alternatively, we guess 139 bits of the state and 27 feedback bits (the first ones), and obtain $58 + 2 \cdot 27 = 112$ linear equations and 49 quadratic equations. We verify that there are 42 equations each of which approximates a linear one with a probability 0.75. The rest 154 bits of the state can therefore be recovered with a probability $0.75^{42} \approx 2^{-17.4}$ by solving a system of approximately linear equations, in which 127 keystream bits are used. In other words, applying guess-and-determine technique the state of ACORN-128 can be recovered in $c \cdot 2^{183.4}$ operations with data complexity of continuous $2^{17.4}+126 \approx 2^{17.4}$ bits. This attack is faster than the previous one by a factor of 12, while the data complexity is much larger.

Improved Guess and Determine Attacks. Next we describe a combination-based guess and determine attack on ACORN-128, which improve time complexity of the previous attack by a factor of about 2^{20}.

The idea is guessing the values of combinations of the state bits rather than single bits. The keystream bit function (4) can be rewritten as

$$g(x) = x_{12} + x_{61} + x_{154} + (x_{61} + x_{193})(x_{61} + x_{235}), \tag{5}$$

where $x = L(s)$. We can see that the keystream bit $g(x)$ approximately $x_{12} + x_{61} + x_{154}$ with a probability 0.75 assuming that $x_{61} + x_{193}$ and $x_{61} + x_{235}$ are independent. Denote by s^t the state at time t and $x^t = L(s^t)$. Note that s^t_i and x^t_i are linear on s for $t + i < 293$. This fact implies that $x^t_{12} + x^t_{61} + x^t_{154}$ is linear on s for $t < 139 = 293 - 154$ and its nonlinear part is the same as that of s^t_{154} for $t < 232 = 293 - 61$. Then we focus on the nonlinear part of s^t_{154}. In the encryption stage, the feedback bit function f can be rewritten as

$$f(x) = 1 + x_0 + x_{23} + x_{66} + x_{107} + (x_{23} + x_{160})(x_{23} + x_{244}) + (x_{66} + x_{111})x_{230}, \tag{6}$$

where $x = L(s)$. For $t < 49 = 293 - 244$, the feedback bit function $f(x^t)$ at time t is quadratic on s. For $0 \leq i \leq 292$ and $0 \leq t < 33$, $s^{293-i+t}_i$ and $x^{293-i+t}_i$ are quadratic on s and has the same quadratic terms as $f(x^t)$. Particularly, for $0 \leq t < 33$ the nonlinear part of s^{139+t}_{154} is $(x^t_{23} + x^t_{160})(x^t_{23} + x^t_{244}) + (x^t_{66} + x^t_{111})x^t_{230}$, and so does s^{58+t}_{235}. As mentioned above, we guess the values of $x^t_{23} + x^t_{160}$ and x^t_{230} for $t < 24$ and obtain $2 \cdot 24$ linear equations from the guessed values and 24 approximately linear equations from $z_{139+t} = x^{139+t}_{12} + x^{139+t}_{61} + x^{139+t}_{154}$ each of which holds with probability 0.75. We guess the values of $x^t_{61} + x^t_{193}$ for $t < 58$ and obtain $2 \cdot 58$ linear equations from the guessed values and $z_t = g(x^t)$. Additionally, we guess the values of $x^{58+t}_{61} + x^{58+t}_{193}$ for $t < 24$ and obtain $2 \cdot 24$ linear equations from the guessed values and $z_{58+t} = g(x^{58+t})$. Together with the rest free $139 - 58 - 24 = 57$ approximately linear equations from $z_t = g(x^t)$ with $82 \leq t < 139$, we have set up a probabilistic linear system of $2 \cdot 58 + 5 \cdot 24 + 57 = 293$ equations on 293 indeterminates with a total probability $0.75^{24+57} \approx 2^{-33.6}$. To sum up, we guess $58 + 3 \cdot 24 = 130$ values of linear combinations of the state, and recover the internal state in $c \cdot 2^{163.6}$ operations where $c \approx 2^{16}$, using around $2^{33.6}$ keystream bits.

The data complexity can be cut down at the cost of slightly increased time complexity, by applying a similar method. A set of attack scenarios for ACORN-128 is listed in Table 1.

4.2 Using two related (Key, IV) pairs

In this section, we investigate state recovering attacks on ACORN-128 using two related (Key, IV) pairs. In this context, we assume that these two related (Key, IV) pairs generate an identical

Table 1. Attack scenarios for a single (Key, IV)

Guessed Values	Linear Eqs.	Appr. Eqs.	Time Complexity	Data Complexity
$58 + 3 \cdot 24$	212	81	$c \cdot 2^{163.6}$	$2^{33.6}$
$58 + 3 \cdot 25$	216	81	$c \cdot 2^{165}$	2^{18}
$58 + 3 \cdot 26$	220	81	$c \cdot 2^{166.3}$	165

state in the encryption stage at the same time or up to a clock difference. The possibilities of slid pairs and slid keys of ACORN-128 have been discussed in Section 3.

In the attacks, we apply differential-and-algebraic technique combining with the previous observations. Denote by $s = (s_0, s_1, \cdots, s_{292})$ the identical state. We choose a pair of plain-texts to be encrypted by s, and observe the difference Δz of keystreams z and z'. In the attacks, we choose a pair of 49-bit plain-texts $(p_0, p_1, \cdots, p_{48})$ and $(p_0 + 1, p_1 + 1, \cdots, p_{48} + 1)$ to be encrypted by s.

We can directly obtain 42 linear equations from Δz_{58+i}, $0 \le i \le 41$. To set up more linear or approximately linear equations, we can substitute the differential equation obtained from Δz_{58+i} in the equation with respect to z_{58+i}. Note that $x^t_{235} = s^t_{235} = l(s^{t-58}) + u_{t-58} + p_{t-58}$ where $l(s^{t-58}) = s^{t-58}_{234} + s^{t-58}_{239}$ and $u_{t-58} = f(s^{t-58})$. To observe how the nonlinear terms of $f(x)$ affect the nonlinearity of keystream-bit function, we substitute $x^t_{235} = l(s^{t-58}) + u_{t-58} + p_{t-58}$ in keystream-bit function $g(x^t)$ at time $58 \le t \le 99$, cf. (5),

$$\begin{aligned} g(x^t) &= x^t_{12} + x^t_{61} + x^t_{154} + (x^t_{61} + x^t_{193})(x^t_{61} + x^t_{235}) \\ &= x^t_{12} + x^t_{61} + x^t_{154} + (x^t_{61} + x^t_{193})(x^t_{61} + l(s^{t-58}) + u_{t-58} + p_{t-58}) \\ &\quad \text{(since introducing a difference on } p_{t-58} \text{ gives } \Delta z_t = x^t_{61} + x^t_{193}) \\ &= x^t_{12} + x^t_{61} + x^t_{154} + \Delta z_t(x^t_{61} + l(s^{t-58}) + u_{t-58} + p_{t-58}). \end{aligned} \tag{7}$$

As discussed in Section 4.1, the state-update function of ACORN shows that for $0 \le t \le 292 - i$, the state bit s^t_i is a linear combination of $s_0, s_1, \cdots, s_{292}$, and so does x^t_i. For $58 \le t \le 99$, x^t_{12}, x^t_{61}, x^t_{154}, s^{t-58}_{234} and s^{t-58}_{239} are linear combinations of $s_0, s_1, \cdots, s_{292}$, and therefore after substituting the differential equation $\Delta z_t = x^t_{61} + x^t_{193}$ the keystream-bit function $g(s^t)$ has the same nonlinear terms as $u_{t-58} = f(s^{t-58})$ if $\Delta z_t = 1$ and otherwise is linear on s. It also explains why we can obtain 42 linear equations from $\Delta z_t = s^t_{61} + s^t_{193}$ for $58 \le t \le 99$ without guessing any values.

For $t \ge 100$, the state bit s^t_{193} has nonlinear terms on s and a difference may be brought by the plain-text, so the case is different. For $100 \le t \le 132$, we have $\Delta(x^t_{12} + x^t_{61} + x^t_{154}) = 0$ and $\Delta x^t_{193} = 1$ and thus

$$\begin{aligned} \Delta z_t &= (x^t_{61} + x^t_{193})(x^t_{61} + x^t_{235}) + (x^t_{61} + x^t_{193} + \Delta x^t_{193})(x^t_{61} + x^t_{235} + \Delta x^t_{235}) \\ &= (x^t_{61} + x^t_{193})(x^t_{61} + x^t_{235}) + (x^t_{61} + x^t_{193} + 1)(x^t_{61} + x^t_{235} + \Delta x^t_{235}) \\ &= x^t_{61} + x^t_{235} + (x^t_{61} + x^t_{193} + 1)\Delta x^t_{235}. \end{aligned} \tag{8}$$

Substituting the above equation in $z_t = g(s^t)$ gives

$$\begin{aligned} g(x^t) &= x^t_{12} + x^t_{61} + x^t_{154} + (x^t_{61} + x^t_{193})(\Delta z_t + (x^t_{61} + x^t_{193} + 1)\Delta x^t_{235}) \\ &= x^t_{12} + (\Delta z_t + 1)x^t_{61} + x^t_{154} + \Delta z_t x^t_{193}, \end{aligned} \tag{9}$$

where $x^t_{12}, x^t_{61}, x^t_{154}$ are linear on s and x^t_{193} is quadratic on s which has the same nonlinear part as $f(x^{t-100})$. The equations (8) and (9) also apply for $t = 134, 135, 136$, and similar properties apply to $x^t_{12}, x^t_{61}, x^t_{154}$ and x^t_{193}. For $139 \le t \le 148$, we have $\Delta(x^t_{12} + x^t_{61} + x^t_{154}) = 1$ and $\Delta x^t_{193} = 1$, and obtain

$$g(x^t) = x^t_{12} + x^t_{61} + x^t_{154} + (x^t_{61} + x^t_{193})(\Delta z_t + 1 + (x^t_{61} + x^t_{193} + 1)\Delta x^t_{235})$$

$$= x^t_{12} + \Delta z_t x^t_{61} + x^t_{154} + (\Delta z_t + 1)x^t_{193}, \tag{10}$$

where x^t_{12}, x^t_{61} are linear on s and x^t_{154} is quadratic on s which has the same nonlinear part as $f(x^{t-139})$.

As mentioned in Section 4.1, for $0 \leq t < 33$ the nonlinear part of s^{139+t}_{154} is $(x^t_{23} + x^t_{160})(x^t_{23} + x^t_{244}) + (x^t_{66} + x^t_{111})x^t_{230}$, and so does s^{100+t}_{193}. We guess the values of $x^t_{23} + x^t_{160}$ and x^t_{230} for $0 \leq t \leq 24$ and obtain $2{\cdot}25$ linear equations from the guessed values and $2{\cdot}25$ linear equations from (7) and (9). Also, we guess the values of $x^t_{61} + x^t_{193}$ for $0 \leq t \leq 57$ and obtain $2 \cdot 58$ linear equations from the guessed values and $z_t = g(x^t)$. For the rest 38 equations as (7) or (9), the probability that at least half of the 38 values $\Delta z_t + c_t$'s equal to zeros is 0.56. That is, we can obtain at least 19 linear equations with probability 0.56. Together with the rest free 18 approximately linear equations from $z_t = g(x^t)$ with $150 \leq t < 139 + 25$ or $t = 133, 137, 138$ each of which holds with probability 0.75, we have obtain a probabilistic linear system of $42 + 4 \cdot 25 + 2 \cdot 58 + 19 + 18 = 295$ equations on 293 indeterminates. We set up a linear system of 293 equations with a total probability $0.56 \cdot 0.75^{16} \approx 0.56 \cdot 2^{-6.64}$. There are $\binom{18}{16} \approx 2^{7.26} > 2^{6.64}$ choices of 16 equations from 18 equations. Therefore the internal state can be recovered in $c \cdot 2^{58+2\cdot25+6.64} = c \cdot 2^{114.64}$ operations with data complexity of a pair of at most 164 keystream bits and with a success probability 0.56, where $c \approx 2^{16}$.

An alternative strategy is adapting the guess according to the values of Δz_t's. This strategy takes advantage of the information of differences of keystream bits and thus makes the attack more effective. Here, we only discuss the average case. According to (7) and (9), for $58 \leq t \neq 133 \leq 136$ the substituted equation $z_t = g(s^t)$ is linear on s when $\Delta z_t = 0$. According to (10), for $139 \leq t \leq 148$ the substituted equation $z_t = g(s^t)$ is approximately linear on s when $\Delta z_t = 1$. Assume that there are equal numbers of ones and zeros for Δz_t. Then we obtain 39 linear equations for $58 \leq t \neq 133 \leq 136$ such that $\Delta z_t = 0$, and five approximately linear equations for $139 \leq t \leq 148$ such that $\Delta z_t = 1$ whose nonlinear parts are the same as $f(x^{t-139})$. We guess the values of $x^t_{23} + x^t_{160}$ and x^t_{230} for linearization of the five nonlinear equations, and obtain 25 linear equations, ten of which are from (7) and (9). Assuming that half of the ten equations have been derived from $\Delta z_t = 0$, we actually get 20 new linear equations. Assume that quarter numbers of t's such that $\Delta z_t = \Delta z_{t+42} = 1$. For $68 \leq t \neq 91 \leq 94$, there are around seven t's such that $\Delta z_t = \Delta z_{t+42} = 1$. We guess the values of $x^t_{23} + x^t_{160}$ and x^t_{230} for such t's, and obtain $4 \cdot 7$ linear equations and 7 approximately linear equations $z_t = g(s^t)$ for $t + 139$ each of which holds with a probability 0.75 according to (5). Additionally, we guess the values of $x^t_{23} + x^t_{160}$ and x^t_{230} for another 11 t's with $10 \leq t \leq 32$, and obtain $3 \cdot 11$ linear equations and 11 approximately linear equations $z_t = g(s^t)$ for $t + 139$ each of which holds with a probability 0.75. As before, we also guess the values of $x^t_{61} + x^t_{193}$ for $0 \leq t \leq 57$ and obtain $2{\cdot}58$ linear equations from (5). Totally, we set up $39{+}20{+}4{\cdot}7{+}3{\cdot}11{+}2{\cdot}58{+}42 = 278$ linear equations and $7{+}11 = 18$ approximately linear equations each of which holds with a probability 0.75. Note that we only need 15 approximately linear equations, and the total probability is $0.75^{15} \approx 2^{-6.23}$. Since $\binom{18}{15} > 2^{6.23}$, the internal state can be recovered in $c \cdot 2^{58+2\cdot(5+7+11)+6.23} = c \cdot 2^{110.23}$ operations with data complexity of a pair of at most 172 keystream bits.

Compared with Exhaustive Search. An exhaustive key search of ACORN-128 is $\mu 2^{128}$, where μ is the time of initialization and processing the associated data of the cipher that includes at least 2048 clocks to be done before the first keystream bits are produced. An average time required for one clock of the cipher is depended on the implementation. We believe that a conservative value for the coefficient μ is larger than 2^{10}, and an exhaustive search would require at least 2^{138} operations. This means that our attacks are faster than an exhaustive search by a factor of around 2^{10}.

5 Conclusion

In this paper, we have described some weaknesses of the authenticated cipher ACORN. The existing of pair pairs of ACORN-128 leads to related-key state recovering attacks and chosen-IV state recovering attacks with time complexity $c \cdot 2^{114.64}$ and with success probability 56%, where $c \approx 2^{16}$. An alternative and faster attack can be mounted for the average case. The attacks will also be more efficient if more (Key, IV) pairs can be used. Our results disclose some weaknesses of ACORN-128, even though the attacks do not threaten the real-life usage of the cipher. The results also explain why the cipher is insecure if one key and IV pair is used to encrypt more than one secret message. The existing of pair pairs of ACORN-128 with a extremely high probability is due to that the state-update function in initialization is the same as when processing the associated data. A possible countermeasure is using different constants to separate these two stages.

Acknowledgement

The authors are grateful to Hongjun Wu for his helpful clarification on the design of ACORN. The authors would also like to thank Shaoyu Du, Tianze Wang and Wenhao Wang for their helpful discussions.

References

1. Hongjun Wu. ACORN: A Lightweight Authenticated Cipher. Submission to CAESAR (2014)
2. Deike Priemuth-Schmid, Alex Biryukov. Slid Pairs in Salsa20 and Trivium. INDOCRYPT 2008, LNCS 5365, pp. 1–14. Springer (2008)
3. Eli Biham. New Types of Cryptanalytic Attacks Using Related Keys. Journal of Cryptology 7(4): 229–246 (1994)
4. Andrey Bogdanov, M. C. Mertens, Christof Paar, Jan Pelzl, Andy Rupp. A Parallel Hardware Architecture for Fast Gaussian Elimination over $GF(2)$. In: FCCM 2006, Proceedings of the 14th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Washington, DC, USA, pp. 237–248. IEEE Computer Society, Los Alamitos (2006)