Bit-Wise Analysis for Forgery Attacks on AES-Based AEAD Schemes

Takuro Shiraya[1], Kosei Sakamoto[2], and Takanori Isobe[1(✉)]

[1] University of Hyogo, Kobe, Japan
takanori.isobe@ai.u-hyogo.ac.jp
[2] Mitsubishi Electric Corporation, Kamakura, Japan

Abstract. We examine the security of AES-based authenticated encryption schemes, including the AEGIS family, Tiaoxin-346, Rocca and Rocca-S. Existing studies evaluated the security against forgery attacks, focusing on state collisions in the encryption phase. These studies estimated the lower bounds for the number of active S-boxes by a byte-wise search. However, this approach might underestimate these bounds, as it potentially include invalid characteristics. In this paper, we conduct a bit-wise evaluation of the AEGIS family, Tiaoxin-346, Rocca, and Rocca-S against forgery attacks based on state collision by Boolean satisfiability problem (SAT) tools. This approach enables us to derive tighter bounds for the minimum number of active S-boxes. Besides, for AEGIS-128L, Tiaoxin-346, and Rocca, we incorporate values of differential distribution tables of S-boxes to obtain the exact differential characteristics probability, which directly lead to actual forgery attacks on AEGIS-128L, Tiaoxin-346, and Rocca. These results reveal that AEGIS-128L cannot claim 256-bit security for forgery attacks, even with a 256-bit tag. Furthermore, for the first time, we perform a security evaluation against forgery attacks exploiting tag collisions in the tag generation phase.

Keywords: AEAD · Forgery attack · differential characteristics probability · SAT solver

1 Introduction

1.1 Background

At SAC 2013, Wu and Preneel proposed an AES-based Authenticated Encryption with an Associated Data (AEAD) scheme called AEGIS-128/128L/256, designed to a high-speed encryption in software [23]. To realize high-speed encryption, the AEGIS family utilizes the AES New Instructions (AES-NI) [4,8], a particular instruction set for single instruction multiple data (SIMD). The AEGIS family was submitted to the CAESAR competition [1], and AEGIS-128 was selected as the final portfolio for high-performance applications. AEGIS-128L/256 has been submitted as an Internet Draft to the RFC [5], featuring the introduction of a 256-bit tag. Nikolić proposed an efficient AEAD scheme called

K. Minematsu and M. Mimura (Eds.): IWSEC 2024, LNCS 14977, pp. 3–22, 2024.
https://doi.org/10.1007/978-981-97-7737-2_1

Tiaoxin-346 using AES-NI in 2014 [15], which was chosen as a third-round candidate in the CAESAR competition. At FSE 2022, Sakamoto et al. proposed an AES-based AEAD scheme called Rocca [16,17] for B5G systems. At ESORICS 2023, Ravi et al. proposed Rocca-S, an AEAD scheme for 6G [2], which supports a 256-bit tag. These ciphers consist of four phases. In the initialization phase, a key and nonce are loaded into a state. An associated data is used to update the state in the authenticated data phase. In the encryption phase, a plaintext is loaded into the state, and a ciphertext is generated. In the finalization phase, a tag is generated.

Forgery attacks are a powerful form of attack against AEAD. In recent years, automatic methods have been utilized to search for distinguishers in cryptanalysis. One such method is based on mixed-integer linear programming (MILP), as proposed by Mouha et al. [14]. This method aims to estimate the lower bound on the number of active S-boxes. Another significant method in automatic search is based on the Boolean satisfiability problem (SAT) or its extension called satisfiability modulo theories (SMT). Sun et al. have proposed an SAT-based automatic search tool for differential characteristics that efficiently evaluates the optimal differential characteristics [20,21].

1.2 Existing Work

Minaud constructed linear biases in the keystream of AEGIS-256 and showed that it is possible to recover information from partially known encrypted plaintext, regardless of the keys involved [13]. Eichlseder et al. proposed improved keystream approximations for the AEGIS family and proved upper bounds for the squared correlation contribution of any suitable linear characteristic [7]. At FSE 2022, Liu et al. showed distinguishing and key recovery attacks for the encryption phase of AEGIS-128 and Tiaoxin-346 by exploiting some algebraic properties in a class of weak keys [12]. Hosoyamada et al. conducted a key-recovery attack on Rocca, showing that despite the designers' claims of 256-bit security, it actually possesses only 128-bit security [9]. This issue has been fixed by introducing a key forward operation in the initialization phase [17]. In this paper, we consider this variant as Rocca. Derbez et al. assessed the key commitment security of the AEGIS family, considering various existing frameworks, and culminated in developing an $O(1)$ attack applicable to all variants of AEGIS [6].

Regarding forgery attacks, existing research focuses on a class of forgery attacks that exploit state collisions by introducing differences during the encryption phase. These roughly estimate the upper bounds of differential characteristics probability for state collisions in the encryption phase by a byte-wise active S-box search [2,15,17,22,23]. The byte-wise estimation potentially underestimates the lower bounds for the number of active S-boxes due to the inclusion of invalid differential characteristics, as this evaluation cannot cover bit-level behaviors. Especially, the estimated bounds for the AEGIS family are particularly rough, even as the IETF considers their standardization. To our

Table 1. Summary of forgery attacks based on state collision in the encryption phase.

Target	Tag Size	Probability	Bounds	Reference
AEGIS-128L	128/256 bits	2^{-216}	Exact	**Our** (Fig. 3)
Tiaoxin-346	128 bits	2^{-180}	Exact	**Our**
Rocca	128 bits	2^{-150}	Exact	**Our** (Fig. 4)

knowledge, no bit-wise evaluation of forgery attacks based on state collisions has been conducted during the encryption phase for AEGIS, Tiaoxin-346, Rocca, and Rocca-S.

1.3 Our Contribution

In this paper, we conduct a bit-wise evaluation against forgery attacks based on state collisions using the Boolean satisfiability problem (SAT) tools [20,21]. This enables us to derive more accurate bounds of AES-based AEAD schemes. Specifically, we estimate the minimum number of active S-boxes by considering bit-level transitions of differential characteristics to exclude invalid characteristics of existing byte-wise searches. Besides, for AEGIS-128L, Tiaoxin-346, and Rocca, we incorporate differential distribution tables of S-boxes, i.e. take the actual differential probabilities via S-box operations into consideration to derive the exact differential characteristics probability. These directly lead to actual forgery attacks on AEGIS-128L, Tiaoxin-346, and Rocca.

Furthermore, for the first time, we perform a security evaluation against forgery attacks that exploit a tag collision in the finalization phase. This assumes that the adversary introduces differences into the plaintext, which canceled out during the finalization phase. Then, we show that forgery attacks are feasible on reduced variants in the finalization of target ciphers. Our results reveal the security margin of the finalization phases. Our contributions are summarized as follows.

Forgery Attacks Based on State Collisions. As shown in Table 1, our bit-wise approach significantly improves the upper bounds of differential characteristics probability of these ciphers. Especially, we significantly improve the upper bounds of differential characteristics probability for AEGIS-128/128L/256. Our results indicate that AEGIS-128/256 could claim a 256-bit forgery security by differential attacks if it supports a 256-bit tag. Additionally, these confirm that Rocca-S achieves 256-bit security for forgery attacks.

For AEGIS-128L, Tiaoxin-346, and Rocca, we succeeded in deriving the exact differential characteristics probability, which directly leads to actual forgery attacks. More precisely, forgery attacks are feasible with time complexity of 2^{216}, 2^{180} and 2^{150} for AEGIS-128L, Tiaoxin-346, and Rocca, respectively. These results reveal that these cannot claim 256-bit security for forgery attacks even with a 256-bit tag.

Table 2. Summary of forgery attacks based on tag collision in the finalization phase.

Target	Tag Size	Attacked Round (Full)	Time
AEGIS-128	128 bits	2 (6)	2^{125}
AEGIS-128L	128/256 bits	2/3 (6/6)	$2^{72}/2^{158}$ (Fig. 5)
AEGIS-256	128 bits	2 (6)	2^{125}
Tiaoxin-346	128 bits	3 (1+20)	2^{36}
Rocca	128 bits	4 (20)	2^{125}
Rocca-S	256 bits	4 (16)	2^{214}

Forgery Attacks Based on Tag Collisions. Table 2 shows that forgery attacks exploiting tag collision are feasible to 2/3/2, 3, 4, and 4-round in the finalization phase of AEGIS-128/128L/256, Tiaoxin-346, Rocca, and Rocca-S, respectively. On the other hand, we find that the finalization phase of AEGIS-128/128L/256, Tiaoxin-346, Rocca, and Rocca-S are secure against forgery attacks based on tag collision after 3/4/3, 4, 5, and 5 rounds, respectively. As far as we know, these are the first evaluation results for tag collision attacks in the finalization phase.

2 Preliminaries

In this section, we first describe forgery attacks. Then, we explain differential characteristics, the security evaluation using the automatic method, and the specifications of AEGIS-128/128L/256, Tiaoxin-346, Rocca, and Rocca-S, respectively.

2.1 Forgery Attacks

The goal of the forgery attacks is to generate the same tag when different messages are input. It has been shown in [15] that the forgery attack is a main threat to the constructions like Tiaoxin-346 and AEGIS as only one-round updates are used to absorb each block of associated data and plaintext.

To proceed with the forgery attacks, we request the encryption of some messages, nonce, and the associated data. The contents of the message, the nonce, and the associated data are not of concern to us. If an internal collision occurs during the cipher operations under these conditions, it becomes possible to forge the tag. We utilize differential characteristics to implement the forgery attacks. The evaluation method of this research is described in Sect. 4.3, while the differential characteristics are explained in Sect. 2.2.

2.2 Differential Characteristics

In this paper, We consider that based on the AES round function, we must regard only an S-box in AES as a non-linear function. In general, differential propagation can be probabilistic only when the differences pass a non-linear function.

Therefore, the differential probability decreases only when the differences pass an S-box. The S-box with a non-zero input difference is called an "active S-box." Basically, when all S-boxes are independent of each other, we can estimate the differential probability of the entire round function by the product of the differential probability of all active S-boxes. We can apply this method to our round function because, for differential propagation, all S-boxes are independent of each other [11,16]. Let DP_{F_R} and DP_s be the differential probabilities of the whole round function and S-box, respectively. We can calculate DP_{F_R} as follows.

$$DP_{F_R} = \prod_{i=1}^{n} DP_s, \tag{1}$$

where n is the number of active S-boxes in this differential characteristic, which indicates a certain differential propagation. DP_{F_R} is equivalent to the probability of an internal collision in a certain round of F_R.

When evaluating security against an internal collision on t rounds of F_R, the maximum differential probability must be evaluated such that the differences in states at t rounds will all be 0. This can be calculated by searching for the differential characteristics with the minimum number of active S-boxes among all the differential characteristics. Conversely, the maximum differential probability of F_R can be estimated by searching for the lower bound for the number of active S-boxes. Let DP_{F_Rmax} and DP_{smax} be the maximum differential probabilities of the differential characteristics with the minimum number of active S-boxes on F_R and the S-box, respectively. DP_{F_Rmax} can be calculated as follows.

$$DP_{F_Rmax} = \prod_{i=1}^{m} DP_{smax}, \tag{2}$$

where m is the lower bound of the number of active S-boxes.

2.3 Automatic Search Tools for Differential Cryptanalysis

The automatic search method showed incredible performances in the search for various distinguishers in cryptanalysis. The first category of automatic search is based on mixed integer linear programming (MILP). Another important automatic search is based on the Boolean satisfiability problem (SAT) or the more general extension called satisfiability modulo theories (SMT). Let's consider an example of security evaluation against differential attacks using an active S-box. In the search using MILP, binary variables are assigned to the input and output of each operation, and the differential propagation of each operation is represented in a linear form. Minimizing the number of active S-boxes in the objective function, the lower bound for the active S-boxes is derived. In the search using SAT, binary variables are assigned to the input and output of each operation, and the differential propagation of each operation is represented in CNF. By adding a CNF to minimize the number of active S-boxes, the SAT problem is repeatedly solved to derive the lower bound for the active S-boxes. In this paper, we adopt the SAT method.

3 Our Targets

We explain the encryption phase and the finalization phase, as the other phases are not involved in our evaluation.

3.1 AEGIS Family

A family of AEGIS, including AEGIS-128/128L/256, consists of four phases: initialization, processing the authenticated data, encryption, and finalization [23].

AEGIS-128. The input of the round function $R(S, X_r)$ consists of the state S and one block (X_r). The round function of AEGIS-128 is given as follows:

$$S'[0] = \mathrm{A}(S[4], S[0] \oplus X_r), \quad S'[1] = \mathrm{A}(S[0], S[1]), \quad S'[2] = \mathrm{A}(S[1], S[2]),$$
$$S'[3] = \mathrm{A}(S[2], S[3]), \quad S'[4] = \mathrm{A}(S[3], S[4]).$$

Let A be one AES round function, $\mathrm{A}(X, K)$ are defined as follows:

$$\mathrm{A}(X, K) = (\mathrm{MixColumns} \circ \mathrm{ShiftRows} \circ \mathrm{SubBytes}(X)) \oplus K.$$

Encryption Phase. Let *msglen* be the length of plaintext in bits, and the number of 128-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{128} \rceil$. Let P_i and C_i $(0 \leq i \leq v-1)$ be the 128-bit plaintext/ciphertext block, respectively. The data X_r inserted in r rounds is expressed as $X_r = P_r$. The ciphertext C_i is expressed as follows:

$$C_i = P_i \oplus S[1] \oplus S[4] \oplus (S[2] \& S[3]), \qquad (0 \leq i \leq v-1).$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let *adlen* be the length of the associated data, *tmp* is expressed as $tmp = S[3] \oplus (adlen || msglen)$, where *adlen* and *msglen* are expressed as 64-bit integers. In the finalization phase, 6 iterations of the round function $R(S, tmp)$ are applied to the state S. After 6 iterations of the round function, the 128-bit tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \oplus S[4].$$

AEGIS-128L. The input of the round function $R(S, X_{r,a}, X_{r,b})$ consists of the state S and two blocks $(X_{r,a}, X_{r,b})$. The round function of AEGIS-128L is given as follows:

$$S'[0] = \mathrm{A}(S[7], S[0] \oplus X_{r,a}), \quad S'[1] = \mathrm{A}(S[0], S[1]),$$
$$S'[2] = \mathrm{A}(S[1], S[2]), \quad S'[3] = \mathrm{A}(S[2], S[3]),$$
$$S'[4] = \mathrm{A}(S[3], S[4] \oplus X_{r,b}), \quad S'[5] = \mathrm{A}(S[4], S[5]),$$
$$S'[6] = \mathrm{A}(S[5], S[6]), \quad S'[7] = \mathrm{A}(S[6], S[7]).$$

Encryption Phase. Let $msglen$ be the length of plaintext in bits, the number of 256-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{256} \rceil$. Let $P_i = P_i^0 || P_i^1$ and $C_i = C_i^0 || C_i^1 (0 \leq i \leq v-1)$ be the 256-bit plaintext/ciphertext block, respectively. The data $X_r = X_{r,a} || X_{r,b}$ inserted in r rounds is expressed as $X_{r,a} = P_i^0, X_{r,b} = P_i^1$. The ciphertext C_i is expressed as follows:

$$C_i^0 = P_i^0 \oplus S[1] \oplus S[6] \oplus (S[2] \& S[3]),$$
$$C_i^1 = P_i^1 \oplus S[2] \oplus S[5] \oplus (S[6] \& S[7]), \qquad (0 \leq i \leq v-1).$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let $adlen$ be the length of the associated data, tmp is expressed as $tmp = S[2] \oplus (adlen || msglen)$, where $adlen$ and $msglen$ are expressed as 64-bit integers. In the finalization phase, 6 iterations of the round function $R(S, tmp, tmp)$ are applied to the state S. After 6 iterations of the round function, the 128-bit tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \oplus S[4] \oplus S[5] \oplus S[6] \oplus S[7].$$

In the finalization phase of RFC's Draft [5], 7 iterations of the round function are applied to the state S. If the tag size is 128 bits, the tag is generated using the same method as described in the proposed paper. Otherwise, if the tag size is 256 bits, the tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \;||\; S[4] \oplus S[5] \oplus S[6] \oplus S[7].$$

AEGIS-256. The input of the round function $R(S, X_r)$ consists of the state S and one block (X_r). The round function of AEGIS-256 is given as follows:

$$S'[0] = \mathrm{A}(S[5], S[0] \oplus X_r), \quad S'[1] = \mathrm{A}(S[0], S[1]), \quad S'[2] = \mathrm{A}(S[1], S[2]),$$
$$S'[3] = \mathrm{A}(S[2], S[3]), \quad S'[4] = \mathrm{A}(S[3], S[4]), \quad S'[5] = \mathrm{A}(S[4], S[5]).$$

Encryption Phase. Let $msglen$ be the length of plaintext in bits, and the number of 128-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{128} \rceil$. Let P_i and C_i $(0 \leq i \leq v-1)$ be the 128-bit plaintext/ciphertext block, respectively. The data X_r inserted in r rounds is expressed as $X_r = P_r$. The ciphertext C_i is expressed as follows:

$$C_i = P_i \oplus S[1] \oplus S[4] \oplus S[5] \oplus (S[2] \& S[3]), \qquad (0 \leq i \leq v-1).$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let $adlen$ be the length of the associated data, tmp is expressed as $tmp = S[3] \oplus (adlen || msglen)$, where $adlen$ and $msglen$ are expressed as 64-bit integers. In the finalization phase, 6 iterations of the round function $R(S, tmp)$ are applied to the state S. After 6 iterations of the round function, the 128-bit tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \oplus S[4] \oplus S[5].$$

3.2 Tiaoxin-346

Tiaoxin-346 consists of four phases: initialization, processing associated data, encryption, and finalization/tag production [15]. The input of the round function $R(T_3, T_4, T_6, X_{r,0}, X_{r,1}, X_{r,2})$ consists of the state (T_3, T_4, T_6) and three blocks $(X_{r,0}, X_{r,1}, X_{r,2})$. The round function of Tiaoxin-346 is given as follows:

$$
\begin{aligned}
&T'_3[0] = \mathrm{A}(T_3[2], T_3[0]) \oplus X_{r,0}, && T'_3[1] = \mathrm{A}(T_3[0], \mathrm{const_a}), && T'_3[2] = T_3[1],\\
&T'_4[0] = \mathrm{A}(T_4[3], T_4[0]) \oplus X_{r,1}, && T'_4[1] = \mathrm{A}(T_4[0], \mathrm{const_a}), && T'_4[2] = T_4[1],\\
&T'_4[3] = T_4[2], && T'_6[0] = \mathrm{A}(T_6[5], T_6[0]) \oplus X_{r,2}, &&\\
&T'_6[1] = \mathrm{A}(T_6[0], \mathrm{const_a}), && T'_6[2] = T_6[1], && T'_6[3] = T_6[2],\\
&T'_6[4] = T_6[3], && T'_6[5] = T_6[4]. &&
\end{aligned}
$$

Encryption Phase. Let $msglen$ be the length of plaintext in bits, and the number of 256-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{256} \rceil$. Let $P_i = P_i^0 || P_i^1$ and $C_i = C_i^0 || C_i^1$ $(0 \leq i \leq v-1)$ be the 256-bit plaintext/ciphertext block, respectively. The data X_r inserted in r rounds is expressed as $X_{r,0} = P_i^0, X_{r,1} = P_i^1, X_{r,2} = P_i^0 \oplus P_i^1$. The ciphertext C_i is expressed as follows:

$$
\begin{aligned}
C_i^0 &= T_3[0] \oplus T_3[2] \oplus T_4[1] \oplus (T_6[3] \& T_4[3]),\\
C_i^1 &= T_6[0] \oplus T_4[2] \oplus T_3[1] \oplus (T_6[5] \& T_3[2]), \qquad (0 \leq i \leq v-1).
\end{aligned}
$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let $adlen$ be the length of the associated data, 1 iteration of the round function $R(T_3, T_4, T_6, adlen, msglen, adlen \oplus msglen)$ is applied to the state S. Then, 20 iterations of the round function $R(T_3, T_4, T_6, \mathrm{const_b}, \mathrm{const_a}, \mathrm{const_b})$ are applied to the state S. After 20 iterations of the round function, the 128-bit tag T is generated as follows:

$$
\begin{aligned}
T = {} & T_3[0] \oplus T_3[1] \oplus T_3[2] \oplus T_4[0] \oplus T_4[1] \oplus T_4[2] \oplus T_4[3]\\
& \oplus T_6[0] \oplus T_6[1] \oplus T_6[2] \oplus T_6[3] \oplus T_6[4] \oplus T_6[5].
\end{aligned}
$$

3.3 Rocca

Rocca consists of four phases: initialization, processing the associated data, encryption, and finalization [16,17]. The input of the round function $R(S, X_{r,a}, X_{r,b})$ consists of the state S and two blocks $(X_{r,a}, X_{r,b})$. The round function of Rocca is given as follows:

$$
\begin{aligned}
&S'[0] = S[7] \oplus X_{r,a}, && S'[1] = \mathrm{A}(S[0], S[7]), && S'[2] = S[1] \oplus S[6],\\
&S'[3] = \mathrm{A}(S[2], S[1]), && S'[4] = S[3] \oplus X_{r,b}, && S'[5] = \mathrm{A}(S[4], S[3]),\\
&S'[6] = \mathrm{A}(S[5], S[4]), && S'[7] = S[0] \oplus S[6]. &&
\end{aligned}
$$

Encryption Phase. Let $msglen$ be the length of plaintext in bits, and the number of 256-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{256} \rceil$. Let $P_i = P_i^0 || P_i^1$ and $C_i = C_i^0 || C_i^1$ $(0 \leq i \leq v-1)$ be the 256-bit plaintext/ciphertext block, respectively. The data X_r inserted in r rounds is expressed as $X_{r,a} = P_i^0, X_{r,b} = P_i^1$. The ciphertext C_i is expressed as follows:

$$\begin{aligned} C_i^0 &= \mathrm{A}(S[1], S[5]) \oplus P_i^0, \\ C_i^1 &= \mathrm{A}(S[0] \oplus S[4], S[2]) \oplus P_i^1, \qquad (0 \leq i \leq v-1). \end{aligned}$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let $adlen$ be the length of the associated data, 20 iterations of the round function $R(S, adlen, msglen)$ are applied to the state S. After 20 iterations of the round function, the 128-bit tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \oplus S[4] \oplus S[5] \oplus S[6] \oplus S[7].$$

3.4 Rocca-S

Rocca-S consists of four phases: initialization, processing the associated data, encryption, and finalization [2]. The input of the round function $R(S, X_{r,a}, X_{r,b})$ consists of the state S and two blocks $(X_{r,a}, X_{r,b})$. The round function of Rocca-S is given as follows:

$$\begin{aligned} &S'[0] = S[6] \oplus S[1], \quad S'[1] = \mathrm{A}(S[0], X_{r,a}), \quad S'[6] = \mathrm{A}(S[1], S[0]), \\ &S'[3] = \mathrm{A}(S[2], S[6]), \quad S'[6] = \mathrm{A}(S[3], X_{r,b}), \quad S'[5] = \mathrm{A}(S[4], S[3]), \\ &S'[6] = \mathrm{A}(S[5], S[4]). \end{aligned}$$

Encryption Phase. Let $msglen$ be the length of plaintext in bits, the number of 256-bit plaintext blocks v is expressed as $v = \lceil \frac{msglen}{256} \rceil$. Let $P_i = P_i^0 || P_i^1$ and $C_i = C_i^0 || C_i^1$ $(0 \leq i \leq v-1)$ be the 256-bit plaintext/ciphertext block, respectively. The data X_r inserted in r rounds is expressed as $X_{r,a} = P_i^0, X_{r,b} = P_i^1$. The ciphertext C_i is expressed as follows:

$$\begin{aligned} C_i^0 &= \mathrm{A}(S[3] \oplus S[5], S[0]) \oplus P_i^0, \\ C_i^1 &= \mathrm{A}(S[4] \oplus S[6], S[2]) \oplus P_i^1, \qquad (0 \leq i \leq v-1). \end{aligned}$$

In the encryption phase, $v-1$ iterations of the round function are applied to the state S, and the ciphertext block C_i is generated.

Finalization Phase. Let $adlen$ be the length of the associated data, 16 iterations of the round function $R(S, adlen, msglen)$ are applied to the state S. After 16 iterations of the round function, the 256-bit tag T is generated as follows:

$$T = S[0] \oplus S[1] \oplus S[2] \oplus S[3] \;||\; S[4] \oplus S[5] \oplus S[6].$$

4 Methods of SAT-Aided Security Evaluations

In this section, we recall the pure SAT-based method to evaluate differential characteristics shown in Sun et al.'s work [20,21]. Then, we explain the tag- and state-collision-based forgery attacks.

4.1 Security Evaluation of SAT

The Boolean satisfiability problem (SAT) is the problem of determining whether there exists an evaluation for the binary variables such that the value of the given Boolean formula equals one. The SAT is formulated with Boolean variables, the operators AND($\wedge$), OR($\vee$), NOT($\neg$), and parentheses. Every Boolean formula can be converted into an equivalent formula that is in conjunctive normal form (CNF), which is a propositional formula of the form $\bigwedge_{i=0}^{n} \bigvee_{j=0}^{m_i} C_i j$, where each $C_i j (0 \leq i \leq n, 0 \leq j \leq m_i)$ is either an atomic formula, i.e., a variable or constant, or the negation of an atomic formula, and each disjunction $\bigvee_{j=0}^{m_i} C_i j$ is called a clause. In this study, we generate CNF using PySAT [10] and derive solutions using the parkissat-rs [24] and mallob-kicaliglu [18,19] solvers.

4.2 SAT-Based Automatic Search for Differential Characteristics

Since our targets are constructed by an S-box, matrix, and XOR operations, it is sufficient to describe these modeling methods. Note that the modeling of a matrix can be implemented by an XOR operation as a matrix is decomposed by multiple XOR operations.

- **XOR.** For a bit-wise XOR operation, s.t., $\alpha \oplus \beta = \gamma$. Differential propagation is valid over XOR if the following clauses hold

$$\left.\begin{aligned} \alpha \vee \beta \vee \overline{\gamma} = 1,\ \alpha \vee \overline{\beta} \vee \gamma = 1, \\ \overline{\alpha} \vee \beta \vee \gamma = 1,\ \overline{\alpha} \vee \overline{\beta} \vee \overline{\gamma} = 1. \end{aligned}\right\}$$

- **S-box.** Let $\boldsymbol{a} = (a_0, a_1, \ldots, a_{i-1})$, $\boldsymbol{b} = (b_0, b_1, \ldots, b_{i-1})$, and $\boldsymbol{p} = \sum_{i=0}^{j-1} p_i$ be the input and output differences of an i-bit S-box and boolean variables expressing the weight in an S-box where j is the maximum weight of the differential propagation, respectively. To express the differential propagation and its weight in an S-box, we construct the following Boolean formula:

$$f(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}) = \begin{cases} 1 & \textit{if } \Pr(\boldsymbol{a} \rightarrow \boldsymbol{b}) = 2^{-\boldsymbol{p}}, \\ 0 & \textit{otherwise.} \end{cases}$$

  A set A, which contains all the vectors satisfying $f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = 0$, is expressed as follows:

$$A = \{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in \mathbb{F}_2^{2i+j} \mid f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = 0\},$$

  where $\boldsymbol{x} = (x_0, x_1, \ldots, x_{i-1})$, $\boldsymbol{y} = (y_0, y_1, \ldots, y_{i-1})$, and $\boldsymbol{z} = (z_0, z_1, \ldots, z_{j-1})$. We need to exclude the propagation expressed A because it is equivalent to a set of invalid propagation patterns as follows:

$$\bigvee_{c=0}^{i-1}(a_c \oplus x_c) \vee \bigvee_{d=0}^{i-1}(b_d \oplus y_d) \vee \bigvee_{e=0}^{j-1}(p_e \oplus z_e) = 1,\ (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \in A. \tag{3}$$

These clauses exactly extract the differential propagation with the corresponding weight in an i-bit S-box. We can convert Eq. (3) to

$$g(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{p}) = \bigwedge_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \in \mathbb{F}_2^{2i+j}} \left(f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \vee \bigvee_{c=0}^{i-1}(a_c \oplus x_c) \vee \bigvee_{d=0}^{i-1}(b_d \oplus y_d) \vee \bigvee_{e=0}^{j-1}(p_e \oplus z_e) \right).$$

This equation is called the *product-of-sum* of g. We can reduce the number of clauses in g by several tools, such as Espresso logic minimizer[1]. For the modeling of an S-box to count the number of S-boxes, we simply replace $\boldsymbol{p}$ to a, expressing whether an S-box is active.

– **Boolean cardinality constraints.** Lastly, we need to give an objective function to search the lower bounds for the number of Active S-boxes/the exact differential characteristics probability. Such a function can be implemented by Boolean cardinality constraints. In SAT, it is necessary to model the problem of searching the lower bounds for the number of Active S-boxes/the exact differential characteristics probability, and we utilize Boolean cardinality constraints. Boolean cardinality constraints put numerical restrictions on the number of propositional variables that are allowed to be true at the same time. The following constitute a typical construct of the Boolean cardinality constraints,

$$\sum_{i=1}^{n} x_i \leq k,$$

where $(x_1, ..., x_n)$ are Boolean variables (0 or 1), and k define the maximum number of variables. In searching for the lower bounds for the number of Active S-boxes and the exact differential characteristics probability, the variable x_i corresponds to the binary variable $a/\boldsymbol{p}$ as previously described in the context of S-box modeling for AS/DCP. We utilized the better encoding method proposed by Bailleux et al. [3], which is implemented in the CardEnc module from the PySAT for Boolean cardinality constraints.

4.3 Our Analysis of Forgery Attacks

We consider two types of approaches for forgery attacks. In the following, we explain details of our evaluations.

State Collision. The first one exploits state collisions in which the adversary inserts differences into a plaintext, causing state collisions during the encryption phase, as shown in Fig. 1. This approach is the same as existing work [22] and designer's evaluations [2,15,17,23].

In this setting, using the modeling explained in Sect. 4.2, we estimate the lower bounds for the number of active S-boxes by considering the bit-level behaviors of differentials during the search. Additionally, by exploiting the properties

[1] https://ptolemy.berkeley.edu/projects/embedded/pubs/downloads/espresso/index.htm.

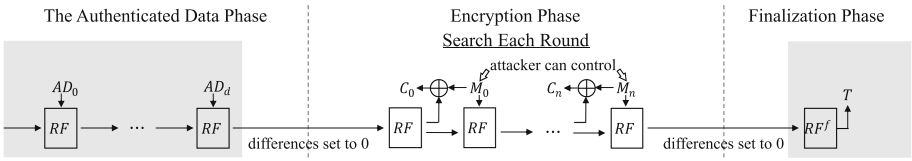

Fig. 1. Overview of forgery attacks based on state collisions.

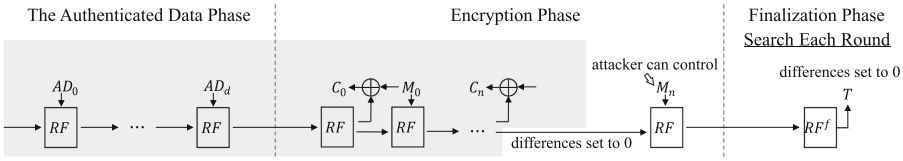

Fig. 2. Overview of forgery attacks based on tag collisions.

of the differential distribution table of the S-box, we find optimal differential characteristic probabilities instead of the upper bounds.

Tag Collision. The second approach exploits tag collisions in which the adversary inserts differences into a plaintext in the last round of the encryption phase so that a state collision occurs in the finalization phase, as shown in Fig. 2. This approach has not been explored in existing work.

In this setting, we also aim to estimate the lower bounds for the number of active S-boxes and, furthermore, derive optimal differential characteristic probabilities, leading to tag collisions.

Table 3. The differential characteristics probability for forgery attacks based on state collisions in the encryption phase (lower bounds for the number of active S-boxes) $[-\log_2]$.

Target	Tag Size (bits)	4R	5R	6R	7R	8R	9R	10R	Reference
AEGIS-128	128 bits			156 (26)	156 (26)	156 (26)	156 (26)	156 (26)	[23]
		–	–	336 (56)	336 (56)	336 (56)	336 (56)	336 (56)	Section 5.1
AEGIS-128L	128/256 bits		150 (25)	150 (25)	150 (25)	150 (25)	150 (25)	150 (25)	[23]
		–	210 (35)	210 (35)	210 (35)	210 (35)	210 (35)	210 (35)	Section 5.1
		–	216	216	216	216	216	216	Section 5.2
AEGIS-256	128 bits	156(26)	156(26)	156(26)	156(26)	156(26)	156(26)	156(26)	[23]
		–	–	–	420 (70)	420 (70)	420 (70)	420 (70)	Section 5.1
Tiaoxin-346	128 bits				180 (30)	180 (30)	180 (30)	180 (30)	[15]
		–	–	–	180 (30)	180 (30)	180 (30)	180 (30)	Section 5.1
		–	–	–	180	180	180	180	Section 5.2
Rocca	128 bits	144 (24)	144 (24)	144 (24)	144 (24)	144 (24)	144 (24)	144 (24)	[17]
		216 (36)	150 (25)	150 (25)	150 (25)	144 (24)	144 (24)	144 (24)	[22]
		234 (39)	216 (36)	180 (30)	150 (25)	150 (25)	150 (25)	150 (25)	Section 5.1
					150	150	150	150	Section 5.2
Rocca-S	256 bits	276 (46)	276 (46)	276 (46)	276 (46)	276 (46)	276 (46)	276 (46)	[2]
		300 (50)	276 (46)	276 (46)	276 (46)	276 (46)	276 (46)	276 (46)	Section 5.1

5 Results of Forgery Attacks Based on State Collisions

In this section, we show results of bit-level analysis for forgery attacks based on state collisions with a comparison to existing results. In Sect. 5.1, we estimate the minimum number of active S-boxes by considering bit-level transitions of differential characteristics to exclude invalid characteristics of byte-wise searches. In Sect. 5.2, we incorporate differential distribution tables of S-boxes for AEGIS-128L, Tiaoxin-346, and Rocca. Due to computational complexity issues, it was not feasible to obtain these results for AEGIS-128/256 and Rocca-S.

5.1 Lower Bounds for the Number of Active S-Boxes

Table 3 shows the lower bounds for the number of active S-boxes, considering bit-wise differential transitions, while existing work focuses on byte-wise truncated characteristics. By using the maximum differential probability of the S-box, namely 2^{-6} as the differential probability of each active S-box, we estimate the upper bounds of the differential characteristic probabilities.

AEGIS-128/128L/256. For the encryption phase of AEGIS-128, AEGIS-128L, and AEGIS-256, we identify differential characteristics that lead to state collisions after 6, 5, and 7 rounds, respectively. As a result, we significantly improve the upper bounds compared to the results provided by the designers. Our findings suggest that AEGIS-128 and AEGIS-256 could claim 256-bit forgery security by differential attacks, provided they support a 256-bit tag.

Tiaoxin-346. For the encryption phase of Tiaoxin-346, we identify differential characteristics that lead to state collisions after 7 rounds. According to Table 1, our evaluation is the same as the byte-wise evaluation by the designer [15]. Thus, unlike AEGIS-128 and AEGIS-256, our results show that Tiaoxin-346 cannot claim 256-bit forgery security, even if supporting a 256-bit tag.

Rocca. For the encryption phase of Rocca, we identify differential characteristics that lead to state collisions after 4 rounds. According to Table 1, our results improve the bounds by a byte-wise evaluation [17]. Our results also show that Rocca cannot claim 256-bit forgery security, even if supporting a 256-bit tag.

Rocca-S. For the encryption phase of Rocca-S, we identify differential characteristics that lead to state collisions after 4 rounds. According to Table 1, our evaluation matches with the byte-wise evaluation by the designer [2].

5.2 Exact Differential Characteristics Probability

Table 3 also shows the exact bounds of differential characteristic probabilities for AEGIS-128L, Tiaoxin-346, and Rocca by exploiting the properties of the differential distribution table of the S-box, namely properly choosing the probability of 2^{-6} or 2^{-7} in each S-box.

Table 4. The differential characteristics probability for forgery attacks based on tag collisions in the finalization phase (lower bounds for the number of active S-boxes) $[-\log_2]$.

Target	Tag Size	1R	2R	3R	4R	5R	6R	Reference
AEGIS-128	128 bits	48 (8)	114 (19)	144 (24)	210 (35)	288 (48)	348 (58)	Section 6.1
		52	125	158	–	–	–	Section 6.2
AEGIS-128L	128/256 bits	12 (2)	72 (12)	144 (24)	210 (35)	288 (48)	360 (60)	Section 6.1
		12	72	158	–	–	–	Section 6.2
AEGIS-256	128 bits	48 (8)	114 (19)	144 (24)	210 (35)	288 (48)	360 (60)	Section 6.1
		52	125	158	–	–	–	Section 6.2
Tiaoxin-346	128 bits	12 (2)	24 (4)	36 (6)	198 (33)	258 (43)	300 (50)	Section 6.1
		12	24	36	203 or more	–	–	Section 6.2
Rocca	128 bits	12 (2)	12 (2)	114 (19)	114 (19)	186 (31)	354 (59)	Section 6.1
		12	12	125	125	190	–	Section 6.2
Rocca-S	256 bits	48 (8)	114 (19)	186 (31)	210 (35)	450 (75)	–	Section 6.1
		52	125	198	214	437 or more	–	Section 6.2

AEGIS-128L. For the encryption phase of AEGIS-128L, we find optimal differential characteristics for forgery attacks after 5 rounds. Our results reveal that forgery attacks are possible with a time complexity of 2^{216}. The differential characteristic for the 5-round forgery attack is shown in Fig. 3.

Tiaoxin-346. For the encryption phase of Tiaoxin-346, we find optimal differential characteristics for forgery attacks after 7 rounds. Our results reveal that forgery attacks are possible with a time complexity of 2^{180}.

Rocca. For the encryption phase of Rocca, we find optimal differential characteristics for forgery attacks after 7 rounds. Our results reveal that forgery attacks are possible with a time complexity of 2^{150}. The differential characteristic for the 7-round forgery attack is shown in Fig. 4.

6 Results of Forgery Attacks Based on Tag Collisions

In this section, we show the results of a bit-level search for forgery attacks exploiting tag collisions. In Sect. 6.1, we estimate the minimum number of active S-boxes by considering bit-level transitions of differential characteristics, which lead to tag collisions. In Sect. 6.2, we utilize differential distribution tables of S-boxes to accurately derive the exact probabilities of differential characteristics for tag collisions.

6.1 Lower Bounds for the Number of Active S-Boxes

Table 4 shows the lower bounds for the number of active S-boxes, which lead to tag collisions. These can be converted into the upper bounds for the differential characteristics probability for each round.

AEGIS-128/128L/256. As the tag length of AEGIS-128/128L/256 is 128/128 or 256/128 bit [23], the lower bounds for the number of active S-boxes should be 22/22 or 43/22 or more in the finalization phase, respectively. According to Table 4, for the finalization phase of AEGIS-128/128L/256, the estimated number of rounds required to be secure against forgery attacks based on tag collisions is estimated as 3/5/3 rounds, respectively.

Tiaoxin-346. As the tag length of Tiaoxin-346 is 128 bits [15], the lower bounds for the number of active S-boxes should be 22 or more in the finalization phase. According to Table 4, for the finalization phase of Tiaoxin-346, the estimated number of rounds required to be secure against forgery attacks based on tag collisions is estimated as 4 rounds.

Rocca. As the tag length of Rocca is 128 bits [16,17], the lower bounds for the number of active S-boxes should be 22 or more in the finalization phase. According to Table 4, for the finalization phase of Rocca, the estimated number of rounds required to be secure against forgery attacks based on tag collisions is estimated as 5 rounds.

Rocca-S. As the tag length of Rocca-S is 256 bits [2], the lower bounds for the number of active S-boxes should be 43 or more in the finalization phase. According to Table 4, for the finalization phase of Rocca-S, the estimated number of rounds required to be secure against forgery attacks based on tag collisions is estimated as 5 rounds.

6.2 Exact Differential Characteristics Probability

Table 4 shows the exact differential characteristics probability for forgery attacks based on tag collisions for each round.

AEGIS-128/128L/256. For the finalization phase of AEGIS-128/128L/256, we find optimal differential characteristics up to 3/3/3 rounds, respectively. For AEGIS-128/128L/256, forgery attacks based on tag collisions are feasible with 2/2/2 rounds, respectively. The optimal differential characteristic for the 2 rounds of tag collisions is shown in Fig. 5.

Tiaoxin-346. For the finalization phase of Tiaoxin-346, we find optimal differential characteristics up to 3 rounds. For Tiaoxin-346, a forgery attack based on tag collisions is feasible with 3 rounds.

Rocca. According to Table 4, for the finalization phase of Rocca, we find optimal differential characteristics up to 5 rounds. For Rocca, a forgery attack based on tag collisions is feasible with 4 rounds.

Rocca-S. According to Table 4, for the finalization phase of Rocca-S, we find optimal differential characteristics up to 4 rounds. For Rocca-S, the maximum number of rounds that can be attacked in forgery attacks based on tag collisions is 4 rounds.

7 Conclusion

In this paper, we conducted a bit-wise evaluation of the AEGIS family, Tiaoxin-346, Rocca, and Rocca-S, against forgery attacks based on state collision and tag collision. We utilized the Boolean satisfiability problem (SAT) tools to obtain exact lower bounds for the number of active S-boxes. Moreover, we derived the optimal differential characteristics in both the encryption phase and the finalization phase. As a result, we obtained the lower bounds for the number of active S-boxes in certain rounds for each target and derived the probability of optimal differential characteristics for the first time.

Acknowledgments. This research was in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan. This work was also supported by JSPS KAKENHI Grant Number JP24H00696.

A Details of Differential Characteristics for Forgery Attacks

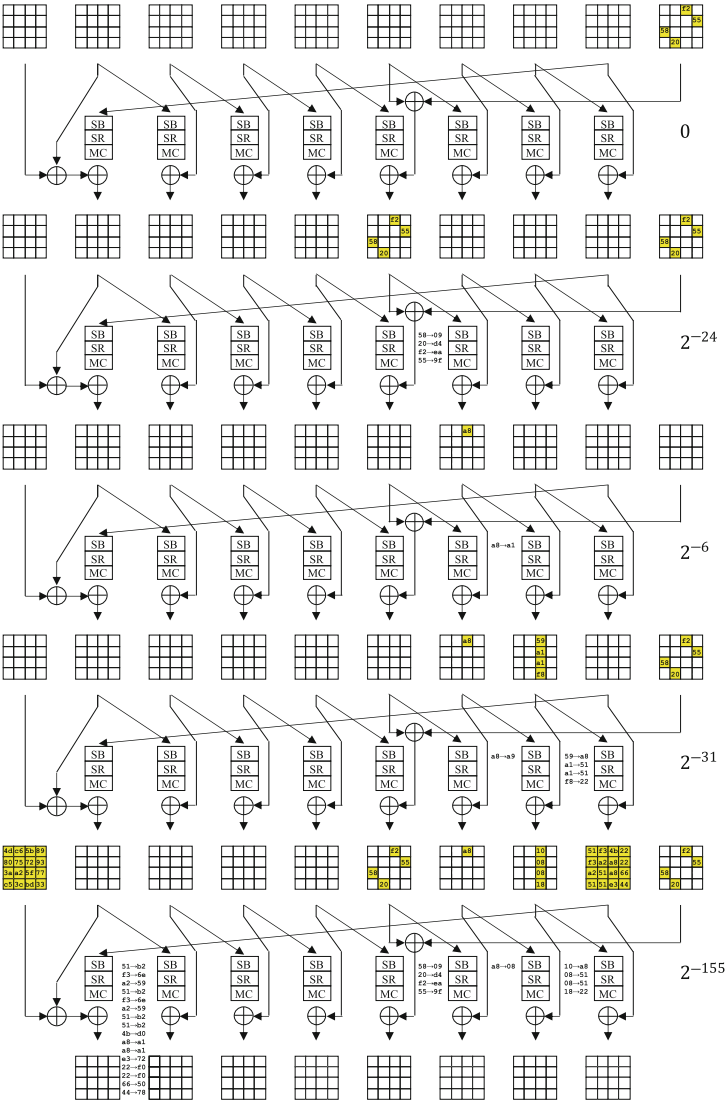

Fig. 3. Optimal differential characteristic for 5-rounds of AEGIS-128L for the forgery attack based on a state collision.

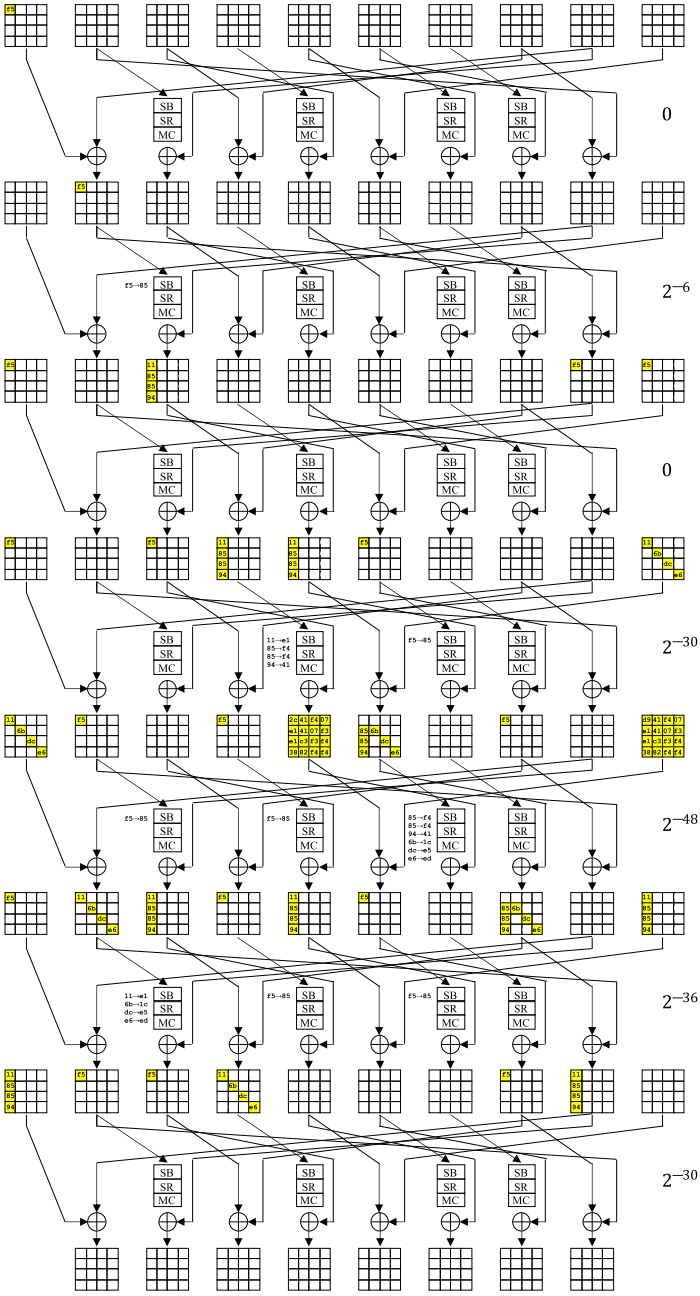

Fig. 4. Optimal differential characteristic for 7-rounds of Rocca for the forgery attack based on a state collision.

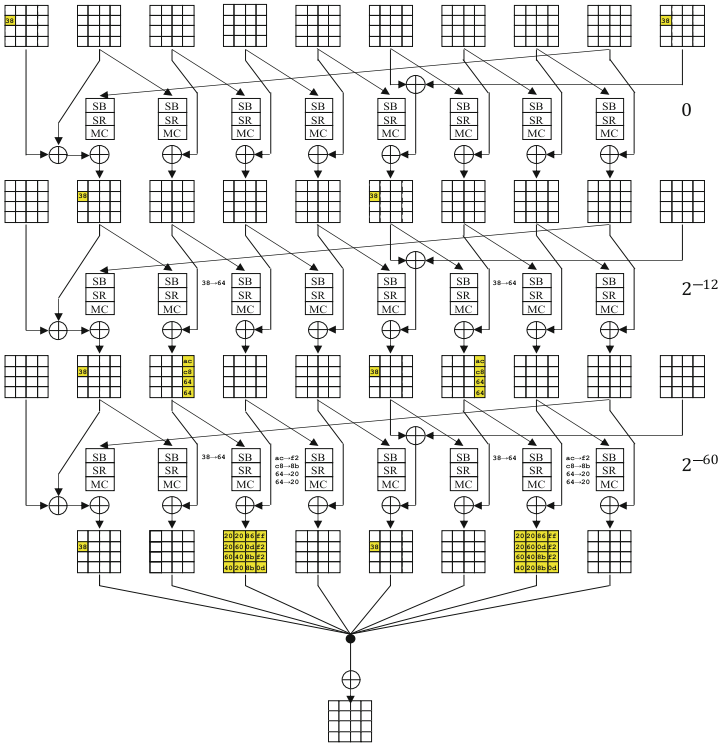

Fig. 5. Optimal differential characteristic for 2-rounds for forgery attack based on tag collision of AEGIS-128L.

References

1. CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness (2018). https://competitions.cr.yp.to/caesar.html
2. Anand, R., et al.: An ultra-high throughput AES-based authenticated encryption scheme for 6G: design and implementation. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds.) ESORICS 2023. LNCS, vol. 14344, pp. 229–248. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-50594-2_12
3. Bailleux, O., Boufkhad, Y.: Efficient CNF encoding of Boolean cardinality constraints. In: Rossi, F. (ed.) CP 2003. LNCS, vol. 2833, pp. 108–122. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45193-8_8
4. Intel Corporation: Intel® Intrinsics Guide (2024). https://software.intel.com/sites/landingpage/IntrinsicsGuide/. Accessed 03 July 2024
5. Denis, F., Lucas, S.: The AEGIS Family of Authenticated Encryption Algorithms. Internet-Draft draft-irtf-cfrg-aegis-aead-10, Internet Engineering Task Force (2024). Work in Progress
6. Derbez, P., Fouque, P., Isobe, T., Rahman, M., Schrottenloher, A.: Key committing attacks against AES-based AEAD schemes. IACR Trans. Symmetric Cryptol. **2024**(1), 135–157 (2024)

7. Eichlseder, M., Nageler, M., Primas, R.: Analyzing the linear keystream biases in AEGIS. IACR Trans. Symmetric Cryptol. **2019**(4), 348–368 (2019)
8. Gueron, S.: Intel Advanced Encryption Standard (AES) New Instructions Set (2010)
9. Hosoyamada, A., et al.: Cryptanalysis of Rocca and feasibility of its security claim. IACR Trans. Symmetric Cryptol. **2022**(3), 123–151 (2022)
10. Ignatiev, A., Morgado, A., Marques-Silva, J.: PySAT: a Python toolkit for prototyping with SAT oracles. In: Beyersdorff, O., Wintersteiger, C.M. (eds.) SAT 2018. LNCS, vol. 10929, pp. 428–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94144-8_26
11. Jean, J., Nikolić, I.: Efficient design strategies based on the `AES` round function. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 334–353. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_17
12. Liu, F., Isobe, T., Meier, W., Sakamoto, K.: Weak keys in reduced AEGIS and tiaoxin. IACR Trans. Symmetric Cryptol. **2021**(2), 104–139 (2021)
13. Minaud, B.: Linear biases in AEGIS keystream. In: Joux, A., Youssef, A. (eds.) SAC 2014. LNCS, vol. 8781, pp. 290–305. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13051-4_18
14. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C.-K., Yung, M., Lin, D. (eds.) Inscrypt 2011. LNCS, vol. 7537, pp. 57–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34704-7_5
15. Nikolic, I.: Tiaoxin-346. Submission to the CAESAR competition (2014)
16. Sakamoto, K., Liu, F., Nakano, Y., Kiyomoto, S., Isobe, T.: Rocca: an efficient AES-based encryption scheme for beyond 5G. IACR Trans. Symmetric Cryptol. **2021**(2), 1–30 (2021)
17. Sakamoto, K., Liu, F., Nakano, Y., Kiyomoto, S., Isobe, T.: Rocca: an efficient AES-based encryption scheme for beyond 5G (full version). IACR Cryptology ePrint Archive, p. 116 (2022)
18. Sanders, P., Schreiber, D.: Decentralized online scheduling of malleable NP-hard jobs. In: Cano, J., Trinder, P. (eds.) Euro-Par 2022. LNCS, vol. 13440, pp. 119–135. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-12597-3_8
19. Schreiber, D., Sanders, P.: Scalable SAT solving in the cloud. In: Li, C.-M., Manyà, F. (eds.) SAT 2021. LNCS, vol. 12831, pp. 518–534. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80223-3_35
20. Sun, L., Wang, W., Wang, M.: More accurate differential properties of LED64 and Midori64. IACR Trans. Symmetric Cryptol. **2018**(3), 93–123 (2018)
21. Sun, L., Wang, W., Wang, M.: Accelerating the search of differential and linear characteristics with the SAT method. IACR Trans. Symmetric Cryptol. **2021**(1), 269–315 (2021)
22. Takeuchi, N., Sakamoto, K., Isobe, T.: On optimality of the round function of Rocca. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **106**(1), 45–53 (2023)
23. Wu, H., Preneel, B.: AEGIS: a fast authenticated encryption algorithm. In: Lange, T., Lauter, K., Lisoněk, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 185–201. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43414-7_10
24. Zhang, X., Chen, Z., Cai, S.: ParKissat-RS (2022). https://github.com/songfu1983/ParKissat-RS