Practical Committing Attacks against Rocca-S

Ryunosuke Takeuchi, Yosuke Todo, and Tetsu Iwata

[1] Nagoya University, Nagoya, Japan
takeuchi.ryunosuke.u2@s.mail.nagoya-u.ac.jp, tetsu.iwata@nagoya-u.jp
[2] NTT Social Informatics Laboratories, Musashino, Japan
yosuke.todo@ntt.com

Abstract. This note shows practical committing attacks against Rocca-S, an authenticated encryption with associated data scheme designed for 6G applications. Previously, the best complexity of the attack was 2^{64} by Derbez et al. in ToSC 2024(1)/FSE 2024. We show that the committing attack against Rocca by Takeuchi et al. in ToSC 2024(2)/FSE 2025 can be applied to Rocca-S, where Rocca is an earlier version of Rocca-S. We show a concrete test vector of our attack. We also point out a committing attack that exploits equivalent keys.

Keywords: Rocca-S · Committing security · Equivalent keys · Practical attack.

1 Introduction

Rocca-S is a nonce-based authenticated encryption with associated data (AEAD) scheme designed for 6G applications [ABC+23], and is currently considered for standardization [NFI24]. It is mentioned in an early version of the Internet-Draft, `draft-nakano-rocca-s-03`, that Rocca-S provides 128-bit key-committing security [NFI23a], while this claim was withdrawn in latter versions without a reason [NFI23b,NFI24,ABC+23]. In ToSC 2024(1)/FSE 2024, Derbez et al. analyzed the security of Rocca-S in terms of committing security [DFI+24]. They focused on the strong notion called the FROB (full robustness) setting [FOR17], and showed a FROB attack with 2^{64} complexity. The complexity is below the generic complexity of 2^{128}, while carrying out the attack in practice is a non-trivial goal. Rocca-S is based on an earlier AEAD scheme called Rocca [SLN+21]. Hosoyamada et al. presented the security evaluation of Rocca in terms of key recovery [HII+22] to break the security claim by the designers, and Takeuchi et al. extended the analysis to cover committing attacks [TTI24] to show a practical FROB attack against Rocca.

In this note, we show that the FROB attack against Rocca in [TTI24] can be applied to Rocca-S. In particular, we show an example test vector of the FROB attack against Rocca-S, thereby practically breaking its committing security in the strongest security notion. We also point out the version of Rocca-S in the Internet-Draft [NFI24] has a class of equivalent keys, allowing trivial FROB attacks.

In this note, to make it succinct, we follow exactly the same notation used in [TTI24], and we omit the description of Rocca-S, which can be found in [NFI24,ABC+23].

2 Overview of the Attack

Let Enc be the encryption function of Rocca-S, and let $(C,T) = \mathrm{Enc}_K(N, A, M)$, where K is a 256-bit key, N is a 128-bit nonce, A is associated data (AD), M is a message, C is a ciphertext, and T is a 256-bit tag. Let Dec be the decryption function. We write $\mathrm{Dec}_K(N, A, C, T) = M$ or $\mathrm{Dec}_K(N, A, C, T) = \perp$, where $\perp$ denotes rejection. The goal of the FROB attack is to output (K, N, A), (K', N', A'), and (C, T) such that $\mathrm{Dec}_K(N, A, C, T) \neq \perp$, $\mathrm{Dec}_{K'}(N', A', C, T) \neq \perp$, $K \neq K'$, and $N = N'$.

We adopt the attack in [TTI24, Sect. 5.2] to Rocca-S, and our attack is presented in Algorithm 1. Given any (K, K') and (N, N') with $K \neq K'$ and $N = N'$, we have two internal states S_0 and S'_0 after the initialization of Rocca-S, where $S_0, S'_0 \in (\{0,1\}^{128})^7$. That is, S_0 and S'_0 consist of 7 blocks, where one block corresponds to 128 bits, and let us write $S_0 = S_0[0] \,\|\, \cdots \,\|\, S_0[6]$, which we abbreviate to $S_0 = S_0[0..6]$. We

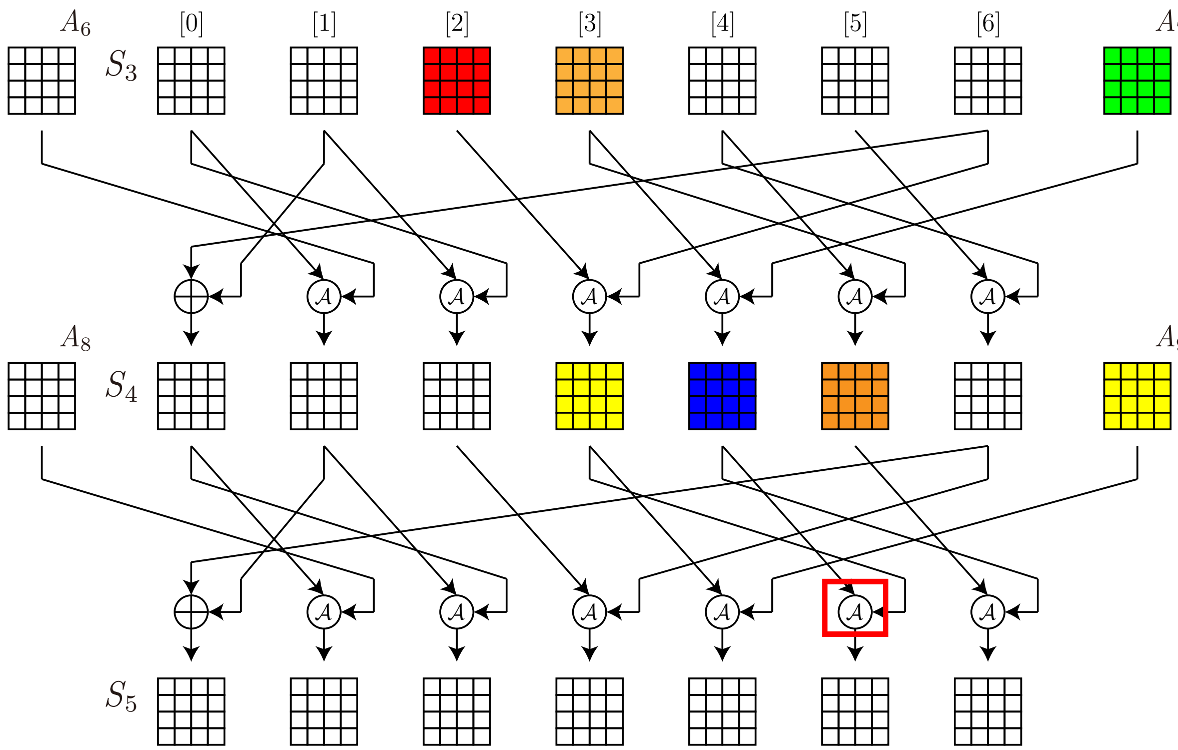

Fig. 1. The last two rounds of the attack

use the similar notation for S'_0 and other internal states as well. We also write, e.g., $S_3[0,1,4..6]$ to mean $S_3[0] \| S_3[1] \| S_3[4] \| S_3[5] \| S_3[6]$, and $S_4[3,5]$ to mean $S_4[3] \| S_4[5]$.

We have two known internal states S_0 and S'_0, and the overall approach is to absorb $A = (A_0, \ldots, A_9)$ into S_0 to have S_5 and $A' = (A'_0, \ldots, A'_9)$ into S'_0 to have S'_5 so that $S_5 = S'_5$ holds. Once this holds, for any message M, the ciphertexts C and C' computed from S_5 and S'_5 are the same, and the tags T and T' computed from S_5, S'_5, and M are also the same, giving the FROB attack.

For given S_0 and S'_0, Algorithm 1 returns $A = (A_0, \ldots, A_9)$ and $A' = (A'_0, \ldots, A'_9)$ such that $S_5 = S'_5$ holds. We first fix $A_0, A_6, A'_6, A_8, A'_8$, and A_9 with $A_6 = A'_6$ and $A_8 = A'_8$ arbitrarily (line 1, 2). We then choose A'_0 randomly (line 3), and compute $A_1, \ldots, A_5$ and $A'_1, \ldots, A'_5$ so that $S_3[0,1,4..6] = S'_3[0,1,4..6]$ holds (line 5), i.e., $A_0, \ldots, A_5$ and $A'_0, \ldots, A'_5$ make 5 out of 7 blocks of S_3 and S'_3 collide. This can be done by following the equation (and the corresponding equations for $A'_1, \ldots, A'_5$) below:

$$
\begin{aligned}
A_3 &= \mathcal{A}(\mathcal{A}(S_0[2]) \oplus S_0[6]) \oplus \mathcal{A}^{-1}(S_3[5] \oplus \mathcal{A}(\mathcal{A}(S_0[1]) \oplus S_0[0]) \oplus \mathcal{A}(S_0[5]) \oplus S_0[4]) \\
A_1 &= \mathcal{A}(S_0[3]) \oplus \mathcal{A}^{-1}(\mathcal{A}(S_0[2]) \oplus S_0[6] \oplus \mathcal{A}^{-1}(S_3[6] \oplus \mathcal{A}(\mathcal{A}(S_0[2]) \oplus S_0[6]) \oplus A_3)) \\
A_2 &= \mathcal{A}(\mathcal{A}(S_0[4]) \oplus S_0[3]) \oplus \mathcal{A}(S_0[3]) \oplus A_1 \oplus \mathcal{A}(S_0[6] \oplus S_0[1]) \oplus S_3[0] \\
A_4 &= \mathcal{A}(\mathcal{A}(S_0[5]) \oplus S_0[4] \oplus \mathcal{A}(S_0[0]) \oplus A_0) \oplus S_3[1] \\
A_5 &= \mathcal{A}(\mathcal{A}(\mathcal{A}(S_0[1]) \oplus S_0[0]) \oplus \mathcal{A}(S_0[5]) \oplus S_0[4]) \oplus S_3[4]
\end{aligned}
$$

Figure 1 shows the remaining part of the attack, where $S_3[2]$ and $S_3[3]$ have differences, and the other 5 blocks of S_3 do not have differences. After one round, $S_4[3]$, $S_4[4]$, and $S_4[5]$ have differences. It is easy to cancel out the difference in $S_5[4]$ by using A_9 such that $\Delta A_9 = \mathcal{A}(S_4[3]) \oplus \mathcal{A}(S'_4[3])$ (line 16). We aim to cancel out the difference in $S_5[5]$ and $S_5[6]$ at the same time by choosing A_7.

Algorithm 1 Collision from two different states

Input: S_0, S'_0
Output: $A_0, \ldots, A_9$, $A'_0, \ldots, A'_9$ such that $S_5 = S'_5$
1: Choose arbitrary A_0, A_6, A_8, and A_9.
2: Set $A'_6 \leftarrow A_6$ and $A'_8 \leftarrow A_8$.
3: **while** $S_5 \neq S'_5$ **do**
4: Choose A'_0 randomly
5: Obtain $A_1, \ldots, A_5$ and $A'_1, \ldots, A'_5$ satisfying $S_3[0, 1, 4..6] = S'_3[0, 1, 4..6]$
6: Compute $S_4[3, 5]$ and $S'_4[3, 5]$
7: $\Delta I = \mathcal{A}(S_4[5]) \oplus \mathcal{A}(S'_4[5])$
8: $\Delta O = \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(S_4[3] \oplus S'_4[3])$
9: **for** $(i, j) \in \{0, 1, 2, 3\} \times \{0, 1, 2, 3\}$ **do**
10: **if** $\Delta I_{i,j} \xrightarrow{\mathrm{Sb}} \Delta O_{i,j}$ is possible **then**
11: Pick an input x s.t. $\mathrm{Sb}(x) \oplus \mathrm{Sb}(x \oplus \Delta I_{i,j}) = \Delta O_{i,j}$
12: $A_{7,i,j} = \mathcal{A}(S_3[3])_{i,j} \oplus x$
13: $A'_{7,i,j} = \mathcal{A}(S'_3[3])_{i,j} \oplus x \oplus \Delta I_{i,j}$
14: **end if**
15: **end for**
16: $A'_9 = A_9 \oplus \mathcal{A}(S_4[3]) \oplus \mathcal{A}(S'_4[3])$
17: Compute S_5 and S'_5
18: **end while**

The condition to succeed in the attack is

$$S_5[5] = \mathcal{A}(S_4[4]) \oplus S_4[3] = \mathcal{A}(S'_4[4]) \oplus S'_4[3]\,,$$
$$S_5[6] = \mathcal{A}(S_4[5]) \oplus S_4[4] = \mathcal{A}(S'_4[5]) \oplus S'_4[4]\,.$$

From the equations above, we have

$$S_4[3] \oplus S'_4[3] = \mathcal{A}(S_4[4]) \oplus \mathcal{A}(S'_4[4])\,,$$
$$S_4[4] \oplus S'_4[4] = \mathcal{A}(S_4[5]) \oplus \mathcal{A}(S'_4[5])\,.$$

Note that $S_4[5]$ and $S'_4[5]$ are fixed when we chose A_0 and A'_0. Therefore, $S_4[4] \oplus S'_4[4]$ is determined. Similarly, $S_4[3] \oplus S'_4[3]$ is also determined. We succeed in the attack if

$$\mathrm{SB}(S_4[4]) \oplus \mathrm{SB}(S'_4[4]) = \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(S_4[3] \oplus S'_4[3])$$

holds. We see that the input and output differences of the S-box are determined, but we can freely choose $S_4[4]$ and $S'_4[4]$ by controlling A_7 and A'_7. Therefore, when the differential transition from $\Delta I = S_4[4] \oplus S'_4[4] = \mathcal{A}(S_4[5]) \oplus \mathcal{A}(S'_4[5])$ (line 7) to $\Delta O = \mathrm{SR}^{-1} \circ \mathrm{MC}^{-1}(S_4[3] \oplus S'_4[3])$ (line 8) is possible, we can choose such $S_4[4]$ and $S'_4[4]$ (line 9–15).

The input and output differences of the S-box (highlighted in red in Fig. 1) are determined once we choose A_0 and A'_0 (and $A_1, \ldots, A_5, A'_1, \ldots, A'_5$ are fixed so that $S_3[0, 1, 4..6] = S'_3[0, 1, 4..6]$ holds). The probability that randomly chosen input/output differences are possible is about 1/2. Since there are 16 S-boxes, the probability that we can construct such A_7 and A'_7 is 2^{-16}. In our attack, we construct such (S_3, S'_3), and if it does not lead to a possible differential transition, we reconstruct different (S_3, S'_3) by randomly choosing A'_0 (line 4) until we have a pair having a possible transition. Therefore, the attack complexity is 2^{16}. We emphasize that the complexity is practical.

3 Test Vector

We present a test case for the FROB attack against Rocca-S by showing a concrete example of (K, N, A, M), (K', N', A', M'), and (C, T) such that $(C, T) = \mathrm{Enc}_K(N, A, M) = \mathrm{Enc}_{K'}(N', A', M')$, with the constraint that $K \neq K'$ and $N = N'$.

We define K, K', N, and N' as follows (written in hex in an array):

$$\begin{aligned}
K_0 &= \{\texttt{01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01}\}\\
K_1 &= \{\texttt{01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01, 01}\}\\
K_0' &= \{\texttt{01, 23, 45, 67, 89, AB, CD, EF, 01, 23, 45, 67, 89, AB, CD, EF}\}\\
K_1' &= \{\texttt{01, 23, 45, 67, 89, AB, CD, EF, 01, 23, 45, 67, 89, AB, CD, EF}\}\\
N = N' &= \{\texttt{02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02, 02}\}
\end{aligned}$$

The state after the initialization of Rocca-S becomes S_0 and S_0' as follows:

$$\begin{aligned}
S_0[0] &= \{\texttt{CE, 6C, C0, EE, 6D, 6E, 66, E5, CA, E1, FC, F9, 00, D7, 62, 73}\}\\
S_0[1] &= \{\texttt{B3, 3F, 7F, FE, B3, 90, 7B, 9D, F8, 51, 43, FD, 52, EE, CD, 03}\}\\
S_0[2] &= \{\texttt{36, FC, 93, FB, A3, 9D, FE, 04, 31, 2D, 63, 96, 9A, 5E, C9, 3D}\}\\
S_0[3] &= \{\texttt{A8, 33, A1, 83, 69, E4, 4B, 33, 60, C8, 9B, 18, 6B, 6A, 5A, DF}\}\\
S_0[4] &= \{\texttt{89, F5, D6, 8B, 9A, 75, 81, 0C, 2A, E6, B9, 37, 2B, BD, 1D, 00}\}\\
S_0[5] &= \{\texttt{ED, 41, E0, D9, CF, 08, 7C, 3D, 5A, 3D, 75, DA, 9C, 83, EE, 3B}\}\\
S_0[6] &= \{\texttt{EB, 6F, 33, 38, BF, D0, 36, 28, 2E, 9F, 8E, C0, D7, 79, A9, 2C}\}
\end{aligned}$$

$$\begin{aligned}
S_0'[0] &= \{\texttt{85, E6, 2C, 94, 39, B1, 22, D2, C0, 03, 9C, 9F, 67, 01, 22, 8E}\}\\
S_0'[1] &= \{\texttt{98, 44, A4, 27, 01, 83, 22, 2F, 10, FA, 5D, 40, A9, FA, 00, 87}\}\\
S_0'[2] &= \{\texttt{FA, F7, D0, 65, 08, DC, C7, A0, 56, F7, FB, F4, A0, 37, 6F, 74}\}\\
S_0'[3] &= \{\texttt{51, 5A, 5F, 17, 75, 60, F4, 97, F4, BF, AF, 10, 27, 12, 7C, C9}\}\\
S_0'[4] &= \{\texttt{DC, 9E, 4E, F6, 27, AC, EE, 70, 39, A9, 8B, FE, 96, EB, E6, 44}\}\\
S_0'[5] &= \{\texttt{7A, 93, FE, 3A, E3, A1, 69, 21, 64, FB, 7F, 44, 51, 41, 45, C3}\}\\
S_0'[6] &= \{\texttt{64, 4C, F7, 45, D5, DB, 7A, C7, D0, C4, D1, 26, 24, 4F, E6, DE}\}
\end{aligned}$$

Then, the associated data A and A' to make $S_3[0, 1, 4..6]$ collide are as follows:

$$\begin{aligned}
A_0 &= \{\texttt{00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00}\}\\
A_1 &= \{\texttt{35, 05, AF, 52, BD, 5D, 12, E1, 77, 7A, 43, D3, 77, 55, 4F, 64}\}\\
A_2 &= \{\texttt{15, D6, EE, 57, C9, CC, A8, B0, 5B, 8E, F0, 0E, 30, 38, E7, FA}\}\\
A_3 &= \{\texttt{22, 40, D5, D1, B3, CD, 12, 66, 80, 5B, E6, 77, 99, 9B, 68, 09}\}\\
A_4 &= \{\texttt{E7, F7, 08, 19, E4, 75, 1E, E2, FD, 64, 38, BB, 0D, 03, 3C, 7C}\}\\
A_5 &= \{\texttt{2F, A5, 9D, C6, 6B, 08, EF, 58, 62, 52, 1E, CA, 66, 6E, C9, 4E}\}\\
A_6 &= \{\texttt{00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00}\}\\
A_7 &= \{\texttt{08, 13, 8D, 6A, E8, 70, C5, 75, 97, 13, E2, 7C, BB, C4, C4, 73}\}\\
A_8 &= \{\texttt{00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00}\}\\
A_9 &= \{\texttt{00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00}\}
\end{aligned}$$

$$
\begin{aligned}
A'_0 &= \{\mathtt{DF}, \mathtt{64}, \mathtt{9F}, \mathtt{EF}, \mathtt{0A}, \mathtt{74}, \mathtt{2B}, \mathtt{5C}, \mathtt{17}, \mathtt{20}, \mathtt{3A}, \mathtt{13}, \mathtt{F6}, \mathtt{CE}, \mathtt{40}, \mathtt{DB}\} \\
A'_1 &= \{\mathtt{72}, \mathtt{C3}, \mathtt{41}, \mathtt{D3}, \mathtt{B6}, \mathtt{AA}, \mathtt{68}, \mathtt{6B}, \mathtt{48}, \mathtt{5B}, \mathtt{4C}, \mathtt{A1}, \mathtt{EF}, \mathtt{E3}, \mathtt{02}, \mathtt{F2}\} \\
A'_2 &= \{\mathtt{B4}, \mathtt{7B}, \mathtt{9F}, \mathtt{F4}, \mathtt{08}, \mathtt{53}, \mathtt{8F}, \mathtt{28}, \mathtt{34}, \mathtt{9B}, \mathtt{79}, \mathtt{FE}, \mathtt{DE}, \mathtt{13}, \mathtt{00}, \mathtt{D2}\} \\
A'_3 &= \{\mathtt{03}, \mathtt{F8}, \mathtt{37}, \mathtt{3C}, \mathtt{37}, \mathtt{7E}, \mathtt{8A}, \mathtt{6F}, \mathtt{E2}, \mathtt{74}, \mathtt{07}, \mathtt{4A}, \mathtt{43}, \mathtt{E6}, \mathtt{F4}, \mathtt{7E}\} \\
A'_4 &= \{\mathtt{F5}, \mathtt{91}, \mathtt{F8}, \mathtt{33}, \mathtt{43}, \mathtt{99}, \mathtt{3F}, \mathtt{9C}, \mathtt{F9}, \mathtt{52}, \mathtt{14}, \mathtt{C8}, \mathtt{A9}, \mathtt{3D}, \mathtt{6E}, \mathtt{77}\} \\
A'_5 &= \{\mathtt{8F}, \mathtt{0E}, \mathtt{72}, \mathtt{19}, \mathtt{8B}, \mathtt{64}, \mathtt{F6}, \mathtt{FD}, \mathtt{FA}, \mathtt{77}, \mathtt{8C}, \mathtt{85}, \mathtt{C9}, \mathtt{AB}, \mathtt{50}, \mathtt{FC}\} \\
A'_6 &= \{\mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}\} \\
A'_7 &= \{\mathtt{D1}, \mathtt{BF}, \mathtt{60}, \mathtt{56}, \mathtt{14}, \mathtt{FD}, \mathtt{63}, \mathtt{15}, \mathtt{A0}, \mathtt{5E}, \mathtt{33}, \mathtt{94}, \mathtt{1E}, \mathtt{6C}, \mathtt{A6}, \mathtt{CB}\} \\
A'_8 &= \{\mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}, \mathtt{00}\} \\
A'_9 &= \{\mathtt{B2}, \mathtt{94}, \mathtt{04}, \mathtt{D1}, \mathtt{47}, \mathtt{A0}, \mathtt{1A}, \mathtt{4F}, \mathtt{EA}, \mathtt{89}, \mathtt{98}, \mathtt{E9}, \mathtt{2A}, \mathtt{5C}, \mathtt{F0}, \mathtt{69}\}
\end{aligned}
$$

Let the messages $M = (M_0, M_1)$ and $M' = (M'_0, M'_1)$ be the following values:

$$
\begin{aligned}
M_0 = M'_0 &= \{\mathtt{FE}, \mathtt{DC}, \mathtt{BA}, \mathtt{98}, \mathtt{76}, \mathtt{54}, \mathtt{32}, \mathtt{10}, \mathtt{FE}, \mathtt{DC}, \mathtt{BA}, \mathtt{98}, \mathtt{76}, \mathtt{54}, \mathtt{32}, \mathtt{10}\} \\
M_1 = M'_1 &= \{\mathtt{FE}, \mathtt{DC}, \mathtt{BA}, \mathtt{98}, \mathtt{76}, \mathtt{54}, \mathtt{32}, \mathtt{10}, \mathtt{FE}, \mathtt{DC}, \mathtt{BA}, \mathtt{98}, \mathtt{76}, \mathtt{54}, \mathtt{32}, \mathtt{10}\}
\end{aligned}
$$

Then, we have $(C, T) = \mathrm{Enc}_K(N, A, M) = \mathrm{Enc}_{K'}(N', A', M')$, where $C = (C_0, C_1)$ and

$$
\begin{aligned}
C_0 &= \{\mathtt{4C}, \mathtt{F6}, \mathtt{03}, \mathtt{97}, \mathtt{FE}, \mathtt{1B}, \mathtt{8E}, \mathtt{81}, \mathtt{20}, \mathtt{AC}, \mathtt{A3}, \mathtt{6A}, \mathtt{AE}, \mathtt{B7}, \mathtt{70}, \mathtt{21}\}\,, \\
C_1 &= \{\mathtt{0A}, \mathtt{AF}, \mathtt{51}, \mathtt{71}, \mathtt{C0}, \mathtt{78}, \mathtt{EC}, \mathtt{8A}, \mathtt{A1}, \mathtt{D7}, \mathtt{16}, \mathtt{D8}, \mathtt{72}, \mathtt{6D}, \mathtt{F4}, \mathtt{7E}\}\,, \\
T &= \{\mathtt{DA}, \mathtt{1B}, \mathtt{2D}, \mathtt{6F}, \mathtt{3D}, \mathtt{7F}, \mathtt{FA}, \mathtt{7F}, \mathtt{16}, \mathtt{4F}, \mathtt{DD}, \mathtt{CA}, \mathtt{0A}, \mathtt{25}, \mathtt{5D}, \mathtt{66}, \\
&\qquad \mathtt{F1}, \mathtt{42}, \mathtt{20}, \mathtt{10}, \mathtt{05}, \mathtt{3A}, \mathtt{D2}, \mathtt{84}, \mathtt{95}, \mathtt{C4}, \mathtt{1F}, \mathtt{C5}, \mathtt{33}, \mathtt{B3}, \mathtt{3E}, \mathtt{11}\}\,.
\end{aligned}
$$

4 FROB Attack from Equivalent Keys

We describe a committing attack that uses equivalent keys. The key length of the version of Rocca-S in [ABC+23] is fixed to 256 bits, while in the version in the Internet-Draft [NFI24], the key length can be 128, 192, or 256 bits. In [NFI24, Sect. 2.3.8], the key is padded before running the encryption/decryption procedure, i.e., for key $K \in \{0,1\}^{128} \cup \{0,1\}^{192} \cup \{0,1\}^{256}$, the padding $\mathsf{pad} : \{0,1\}^{128} \cup \{0,1\}^{192} \cup \{0,1\}^{256} \to \{0,1\}^{256}$ is applied on K before it is used, and the encryption works as $(C, T) = \mathrm{Enc}_{\mathsf{pad}(K)}(N, A, M)$, and the decryption works as $\mathrm{Dec}_{\mathsf{pad}(K)}(N, A, C, T) = M$ or $\mathrm{Dec}_{\mathsf{pad}(K)}(N, A, C, T) = \perp$.

There are two methods of padding in [NFI24]. One is to use a zero padding and the other one is to use a key derivation function (KDF). The zero padding works as

$$
\mathsf{pad}(K) = \begin{cases} K \,\|\, 0^{128} & \text{if } |K| = 128\,, \\ K \,\|\, 0^{64} & \text{if } |K| = 192\,, \\ K & \text{if } |K| = 256\,. \end{cases}
$$

This means that the encryption with a 128-bit key K is equivalent to that with a 192-bit key $K \,\|\, 0^{64}$, which is also equivalent to that with a 256-bit key $K \| 0^{128}$, since $\mathsf{pad}(K) = \mathsf{pad}(K \| 0^{64}) = \mathsf{pad}(K \| 0^{128}) = K \| 0^{128}$, and the key length does not affect the encryption/decryption process after the padding. This forms a large set of equivalent keys, and this also allows trivial FROB attacks. For instance, for any $K \in \{0,1\}^{128}$, (K, N, A, M), $(K \,\|\, 0^{64}, N, A, M)$, and $(K \,\|\, 0^{128}, N, A, M)$ all give the same output (C, T) for any nonce N, associated data A, and message M.

The KDF padding uses a key derivation function $\mathsf{KDF} : \{0,1\}^* \to \{0,1\}^{256}$ and works as

$$
\mathsf{pad}(K) = \begin{cases} \mathsf{KDF}(K) & \text{if } |K| = 128\,, \\ \mathsf{KDF}(K) & \text{if } |K| = 192\,, \\ K & \text{if } |K| = 256\,. \end{cases}
$$

Note that it is reasonable to assume that KDF returns independent outputs for different input lengths. We now observe that KDF is not used if the key is already 256 bits. This means that the encryption with a 128-bit key K is equivalent to that with a 256-bit key $K' = \mathsf{KDF}(K)$, since $\mathsf{pad}(K) = \mathsf{pad}(K') = \mathsf{KDF}(K)$. There exists a large set of equivalent keys, allowing trivial FROB attacks. For instance, for any $K \in \{0,1\}^{128}$, (K, N, A, M) and (K', N, A, M) with $K' = \mathsf{KDF}(K)$ give the same output (C, T) for any nonce N, associated data A, and message M.

The issue is that the padding is non-injective and the key length is not involved in encryption/decryption once the key is padded. This could be avoided by using an injective padding, and/or by including the key length into the initialization and finalization steps.

5 Conclusions

This note shows that Rocca-S is practically committing insecure. The cipher should not be used in applications where committing security is expected, e.g., in those analyzed in [GLR17,DGRW18,LGR21,ADG+22].

Acknowledgments. This work was supported in part by JSPS KAKENHI Grant Number JP24K07489.

References

[ABC+23] Ravi Anand, Subhadeep Banik, Andrea Caforio, Kazuhide Fukushima, Takanori Isobe, Shinsaku Kiyomoto, Fukang Liu, Yuto Nakano, Kosei Sakamoto, and Nobuyuki Takeuchi. An ultra-high throughput AES-based authenticated encryption scheme for 6G: Design and implementation. In Gene Tsudik, Mauro Conti, Kaitai Liang, and Georgios Smaragdakis, editors, *Computer Security - ESORICS 2023 - 28th European Symposium on Research in Computer Security, The Hague, The Netherlands, September 25-29, 2023, Proceedings, Part I*, volume 14344 of *Lecture Notes in Computer Science*, pages 229–248. Springer, 2023.

[ADG+22] Ange Albertini, Thai Duong, Shay Gueron, Stefan Kölbl, Atul Luykx, and Sophie Schmieg. How to abuse and fix authenticated encryption without key commitment. In Kevin R. B. Butler and Kurt Thomas, editors, *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pages 3291–3308. USENIX Association, 2022.

[DFI+24] Patrick Derbez, Pierre-Alain Fouque, Takanori Isobe, Mostafizar Rahman, and André Schrottenloher. Key committing attacks against AES-based AEAD schemes. *IACR Trans. Symmetric Cryptol.*, 2024(1):135–157, 2024.

[DGRW18] Yevgeniy Dodis, Paul Grubbs, Thomas Ristenpart, and Joanne Woodage. Fast message franking: From invisible salamanders to encryptment. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, volume 10991 of *Lecture Notes in Computer Science*, pages 155–186. Springer, 2018.

[FOR17] Pooya Farshim, Claudio Orlandi, and Razvan Rosie. Security of symmetric primitives under incorrect usage of keys. *IACR Trans. Symmetric Cryptol.*, 2017(1):449–473, 2017.

[GLR17] Paul Grubbs, Jiahui Lu, and Thomas Ristenpart. Message franking via committing authenticated encryption. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part III*, volume 10403 of *Lecture Notes in Computer Science*, pages 66–97. Springer, 2017.

[HII+22] Akinori Hosoyamada, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Mimematsu, Ferdinand Sibleyras, and Yosuke Todo. Cryptanalysis of Rocca and feasibility of its security claim. *IACR Trans. Symmetric Cryptol.*, 2022(3):123–151, 2022.

[LGR21] Julia Len, Paul Grubbs, and Thomas Ristenpart. Partitioning oracle attacks. In Michael D. Bailey and Rachel Greenstadt, editors, *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, pages 195–212. USENIX Association, 2021.

[NFI23a] Yuto Nakano, Kazuhide Fukushima, and Takanori Isobe. Encryption algorithm Rocca-S. Network Working Group, Internet-Draft, `https://datatracker.ietf.org/doc/draft-nakano-rocca-s/03/`, 2023.

[NFI23b] Yuto Nakano, Kazuhide Fukushima, and Takanori Isobe. Encryption algorithm Rocca-S. Network Working Group, Internet-Draft, `https://datatracker.ietf.org/doc/draft-nakano-rocca-s/04/`, 2023.

[NFI24] Yuto Nakano, Kazuhide Fukushima, and Takanori Isobe. Encryption algorithm Rocca-S. Network Working Group, Internet-Draft, `https://datatracker.ietf.org/doc/draft-nakano-rocca-s/05/`, 2024.

[SLN+21] Kosei Sakamoto, Fukang Liu, Yuto Nakano, Shinsaku Kiyomoto, and Takanori Isobe. Rocca: An efficient AES-based encryption scheme for beyond 5G. *IACR Trans. Symmetric Cryptol.*, 2021(2):1–30, 2021.

[TTI24] Ryunosuke Takeuchi, Yosuke Todo, and Tetsu Iwata. Key recovery, universal forgery, and committing attacks against revised Rocca: How finalization affects security. *IACR Trans. Symmetric Cryptol.*, 2024(2), 2024.