Security Analysis of Rocca-S

Patrick Derbez[1]
Pierre-Alain Fouque[1]
André Schrottenloher[2]

[1]Univ Rennes
[2]Inria Rennes

Context

- **Rocca** is an authenticated encryption scheme for beyond-5G applications, designed in 2021 **[SLNKI21]**
- **Rocca-S** is an updated version of Rocca **[ABC+23]**
- Rocca-S has been submitted for standardization at the IETF

In Feb. 2023 our team did a third-party security analysis of Rocca-S for KDDI research.

Sakamoto, Liu, Nakano, Kiyomoto, Isobe, "Rocca: An Efficient AES-based Encryption Scheme for Beyond 5G", IACR Trans. Symmetric Cryptol. 2021

Nakano, Fukushima, Isobe, "Encryption algorithm Rocca-S", IETF draft standard (2023)

Anand, Banik, Caforio, Fukushima, Isobe, Kiyomoto, Liu, Nakano, Sakamoto, Takeuchi, "An Ultra-High Throughput AES-based Authenticated Encryption Scheme for 6G: Design and Implementation", ESORICS 2023

Design

Rocca: nonce-based AEAD

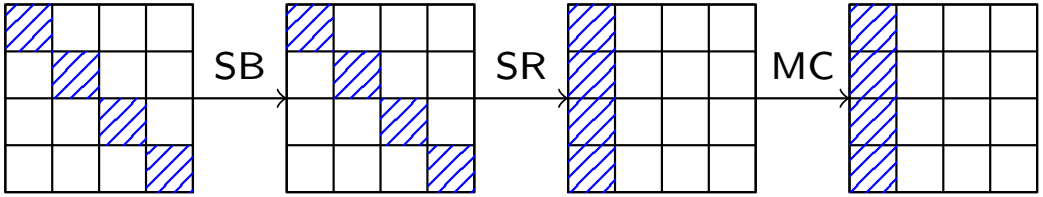
M, A (plaintext + associated data) $\longrightarrow \boxed{E_{K,N}} \longrightarrow$ $C, T = E_{K,N}(M, A)$ ciphertext **and** authentication tag

- M is encrypted ($\rightarrow$ ciphertext C) and authenticated ($\rightarrow$ tag T)
- A is authenticated but not encrypted
- N **shall not be reused**

Internal state

Similarly to AEGIS, Rocca-S is AES-based and uses a large internal state.

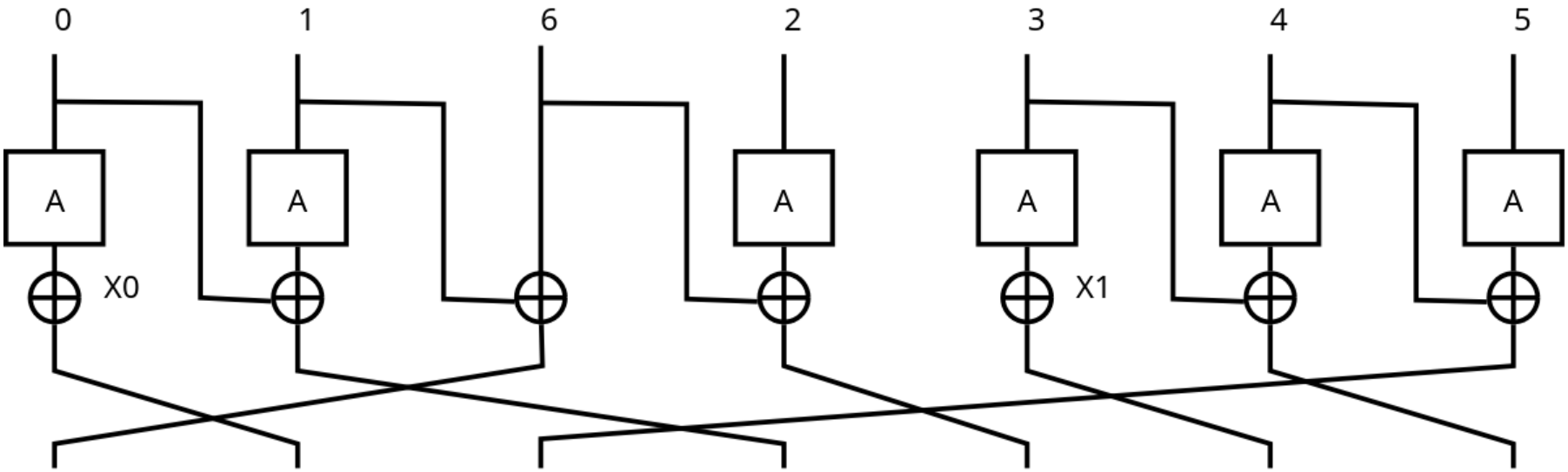
- AES state: 4×4 matrix of bytes
- AES round: $A = MC \circ SR \circ SB$



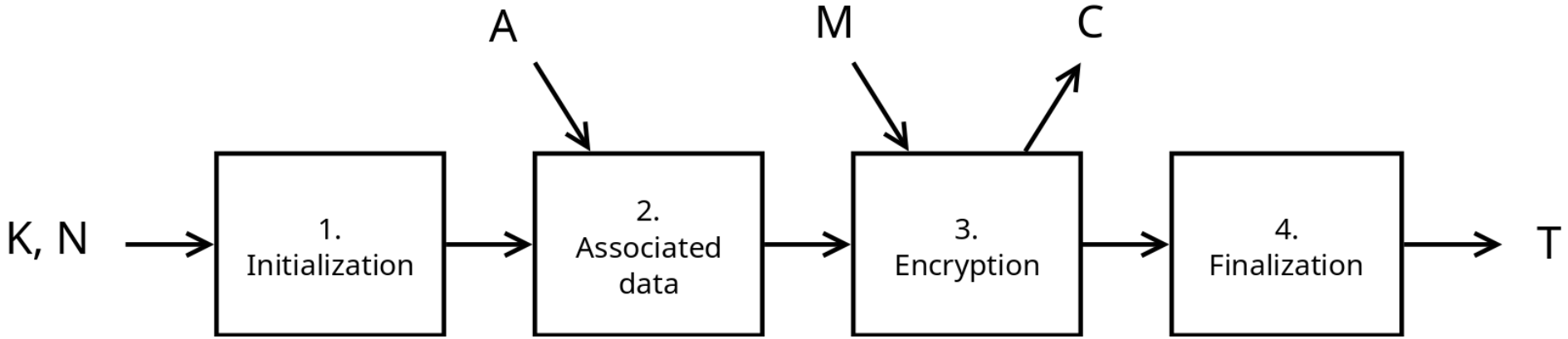
- Rocca-S state: $S = S[0] \| S[1] \| \cdots \| S[6]$ ($6 \times 128 = 896$ bits)

Round function

The round function R absorbs a 256-bit input $X_0 \| X_1$.



Parameters: $\mathbf{K} = \mathbf{K_0} \| \mathbf{K_1}$ (256-bit), 128-bit N, 256-bit T



- **Initialization:** load **K**, N, do 16 empty rounds, XOR keys to the state
- **AD:** process AD blocks by pairs: $S \leftarrow R(S, A_0, A_1)$
- **Encryption:** process message blocks by pairs M_0, M_1:

$$\begin{cases} C_0 = A(S[3] \oplus S[5]) \oplus S[0] \oplus M_0 \\ C_1 = A(S[4] \oplus S[6]) \oplus S[2] \oplus M_1 \\ S \leftarrow R(S, M_0, M_1) \end{cases}$$

- **Finalization:** apply 16 rounds $S \leftarrow R(S, |AD|, |M|)$, output:

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

Analysis

Rocca-S security

Chosen-plaintext queries:

Choose A, M, N, get C, T.

Verification queries:

Choose C, T, get "true" (and plaintext) if T is a valid tag.

Security goals:

- Key-recovery (adversary cannot find **K**): 256 bits
- Forgery (adversary cannot output new valid C, T): 192 bits
- Quantum: 128 bits for both

Rocca-S limitations

1. Nonce-reuse

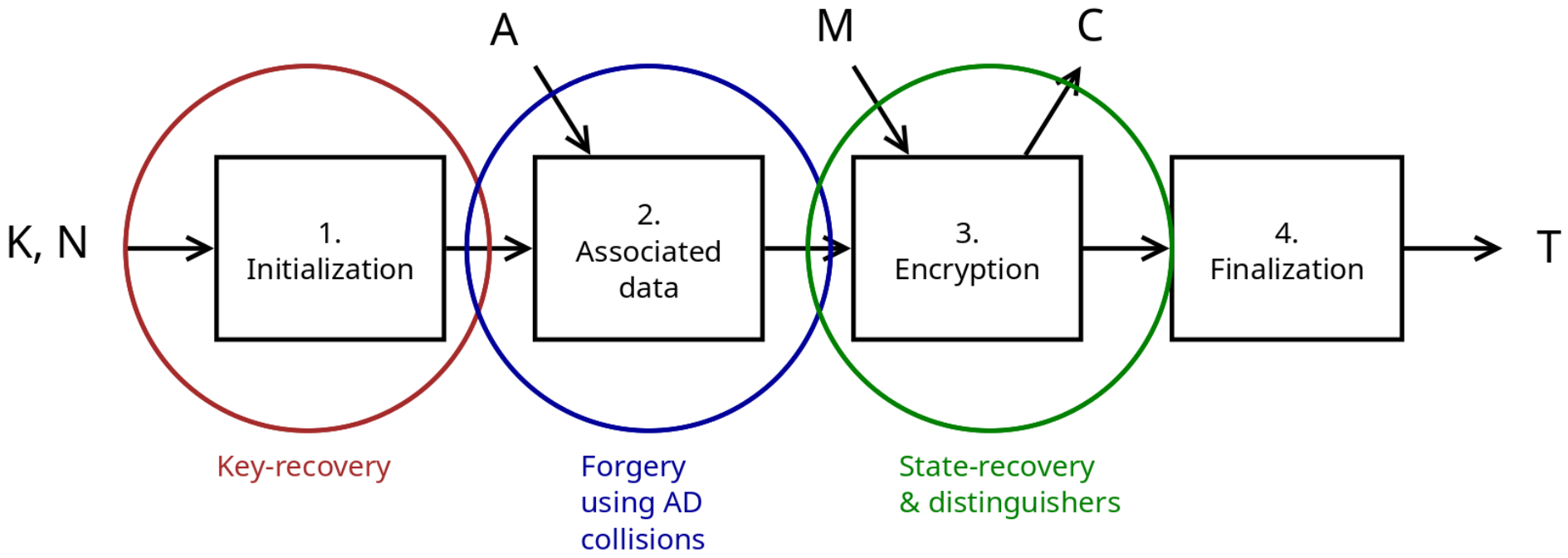
Rocca-S does not claim security if a nonce N is reused (like AEGIS)

2. Superposition queries

Rocca-S is insecure against a quantum attacker doing superposition queries (Q2) (like similar schemes) **[BS23]**

Bonnetain, S., "Single-query quantum hidden shift attacks", ePrint 2023/1306

Overview of possible attacks



Security analysis 1: key-recovery

We tried the following:

- Guess-and-determine / MITM attacks
- Differential cryptanalysis of the initialization phase
- Algebraic / integral attacks on initialization

Case study: (truncated) differential analysis

- Introduce a difference in nonce (and key for RK setting)
- Propagate the difference through the initialization
- Observe an output difference

Estimate the probability of the transition by counting the **active** S-Boxes.

Security analysis 1: key-recovery (ctd.)

The propagation rules are encoded using MILP and the minimal number of active S-Boxes is determined.

Nb rounds	2	3	4	5	6	7	8
# S-Boxes (SK)	7	22	40	68			
# S-Boxes (RK)			13	30	36	53	

- The best probability of transition through an S-Box is 2^{-6}

$\Longrightarrow$ with 43 active S-Boxes, the probability is $\leq 2^{-256}$

Security analysis 2: state-recovery

Principle: find the internal state for a given K, N, A, M

- May allow to encrypt new messages $\implies$ break authenticity

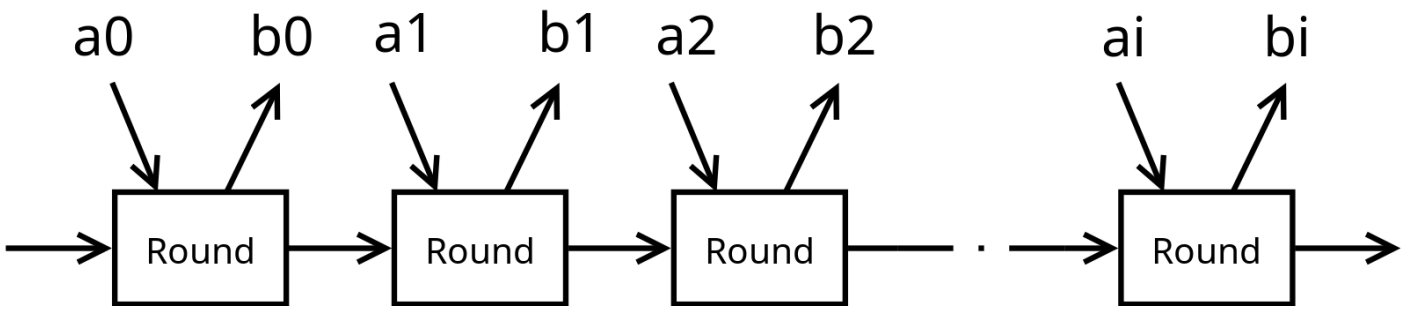
This problem can be reduced to solving a system of equations:
$C = f(S, M)$ (M, C known).

- Guess-and-determine / MITM inapplicable
- Algebraic attacks inapplicable

Security analysis 3: distinguishers

Linear distinguishers: choose linear masks for inputs a_i and outputs b_i such that: $\bigoplus_i (a_i \cdot M_i \oplus b_i \cdot C_i)$ is biased.

$\Longrightarrow$ large-data keystream distinguisher



- We adapted a MILP model from **[ENP19]**.
- Similarly to truncated differentials, one only counts the active S-Boxes
- We studied at most 7 rounds. Best results obtained for 4 rounds $\Longrightarrow$ 53 active S-Boxes $\Longrightarrow$ complexity above 2^{256}

Eichlseder, Nageler, Primas, "Analyzing the linear keystream biases in AEGIS". IACR Trans. Symmetric Cryptol., 2019

Security analysis 4: forgery using AD

Principle: introduce differences in AD which cancel out completely

- With some probability, creates a message N, A', M with same ciphertext & tag (forgery)

- We used the same MILP model for propagation
- The minimal number of active S-Boxes is 46: probability $2^{-276} < 2^{-256}$
- By instantiating the path by hand, we could obtain a small improvement (2^{-274})

Conclusion

- Confirms the security claims of the designers
- Confirms the high levels of security offered by Rocca-S (192 & 256 bits) for forgery and key-recovery

Thank you!