FSE 2023 (Beijing and Kobe)

Cryptanalysis of Rocca and Feasibility of Its Security Claim

Akinori Hosoyamada†, Akiko Inoue‡, Ryoma Ito*, Tetsu Iwata⁑, Kazuhiko Minematsu‡, Ferdinand Sibleyras†, and Yosuke Todo†

†NTT, ‡ NEC, * NICT, ⁑ Nagoya University

Overview of Results

NTT

Rocca
(Sakamoto, et al. FSE 2022)

- AES-based AEAD for the use in the beyond 5G systems.
- Security claims:
  - 256-bit security against the key recovery and distinguishing attacks.
  - 128-bit security against the forgery attack.

Our Contributions

1. Breaking the security claim of Rocca.

We propose the key-recovery attack against Rocca with the complexity of 2^{128}.

2. Exploring the feasibility of the (original) security claim of Rocca.

We show the feasibility of unbalanced bit security for indistinguishability and forgery.

Specification of Rocca

Rocca

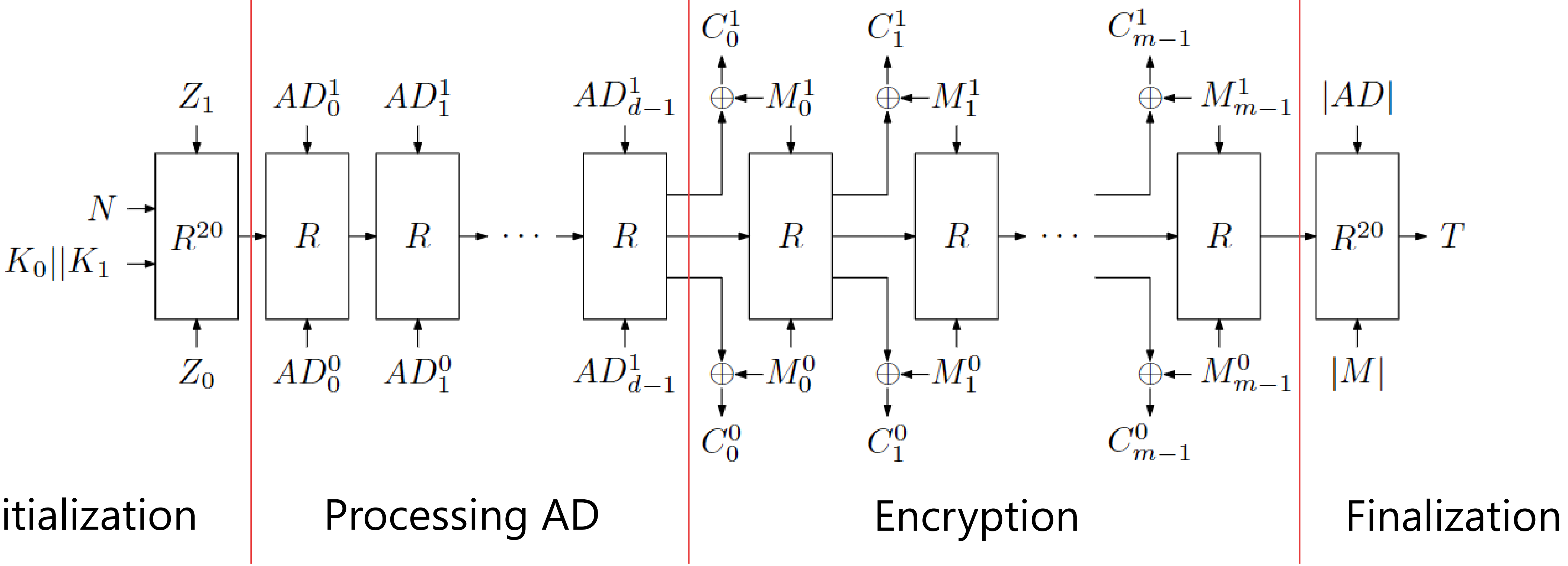
NTT

- AES-based AEAD (proposed at FSE 2022).
  - AES-NI and SIMD-friendly design.
  - Ultra high speed (over 100 Gbps)
  - Security claims (ToSC 2021.i2.1-30)
    - 256-bit security against key recovery and distinguishing attacks.
    - 128-bit security against forgery attack.
    - No claim : nonce misuse, related key, known key.

  - Modified claims (ePrint 2022/116, 20220421 ver.)
    - 256-bit security against key recovery attack.
    - 128-bit security against distinguishing and forgery attacks.
    - No claim : nonce misuse, related key, known key.

Structure of Rocca

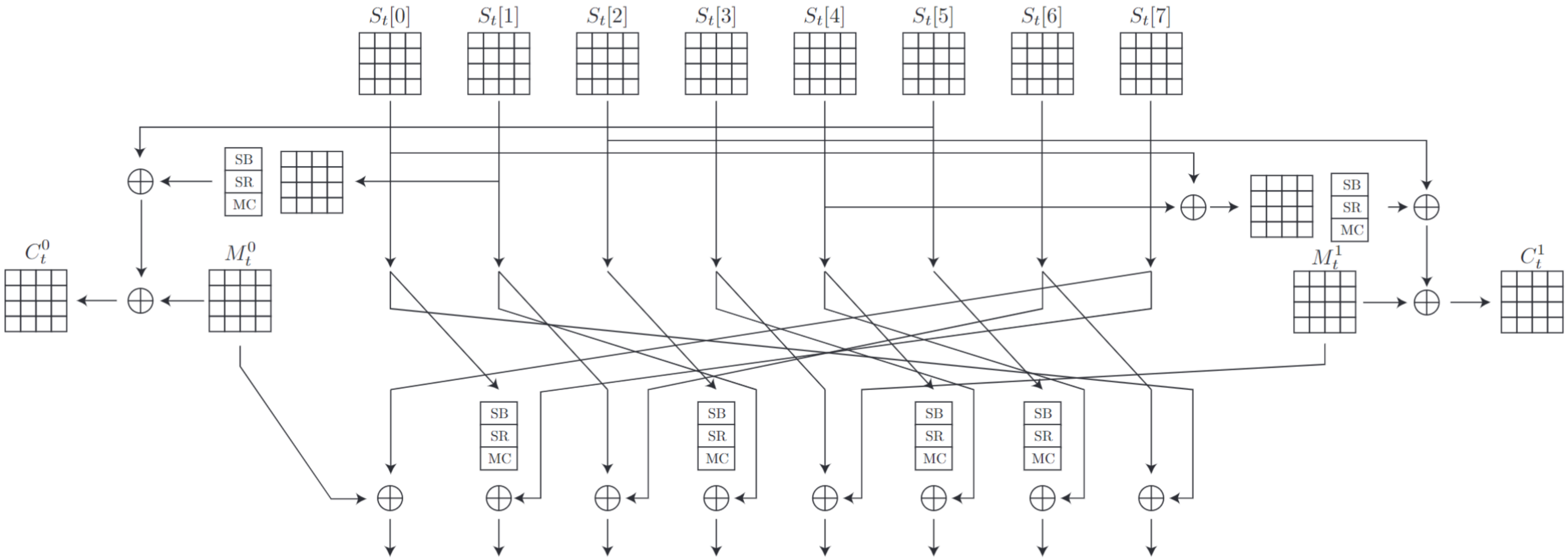
Rocca is permutation-based online AEAD.



Rocca uses 128-bit nonce, 256-bit key, and output 128-bit tag.
The round function absorbs two 128-bit blocks, in total, 256 bits.

Round function of Rocca

- AES-NI friendly design



The round function consists of the AES round function and XOR.
AES-NI accelerates the implementation.

Unbalanced security claims of Rocca

- Rocca claims unbalanced bit-security level.

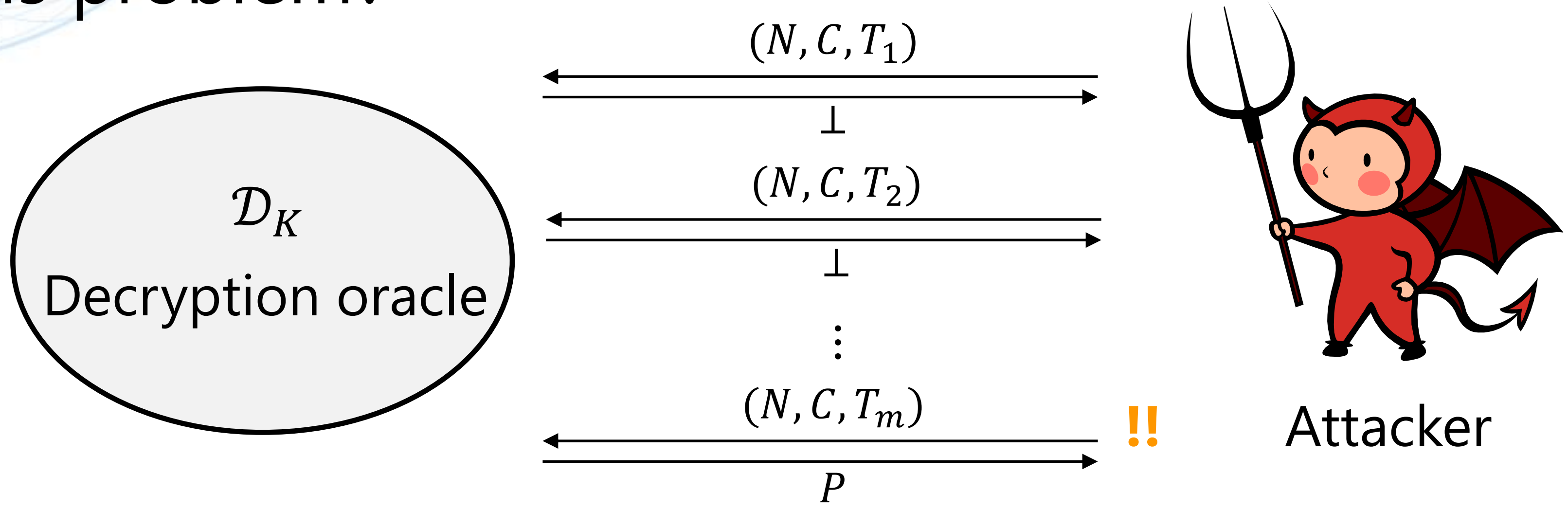
1. 256-bit key recovery & 128-bit tag.
   - We propose the key-recovery attack with the complexity of 2^{128}.
2. 256-bit IND security & 128-bit tag.
   - We consider the security definition capturing such an unbalanced security claim and show the (in)feasibility of this claim.

Attack Exploiting Decryption Oracle

256-bit key recovery security and 128-bit tag

- Rocca uses
  - a 256-bit key to claim the 256-bit security against key recovery.
  - a 128-bit tag to claim the 128-bit security against forgery.

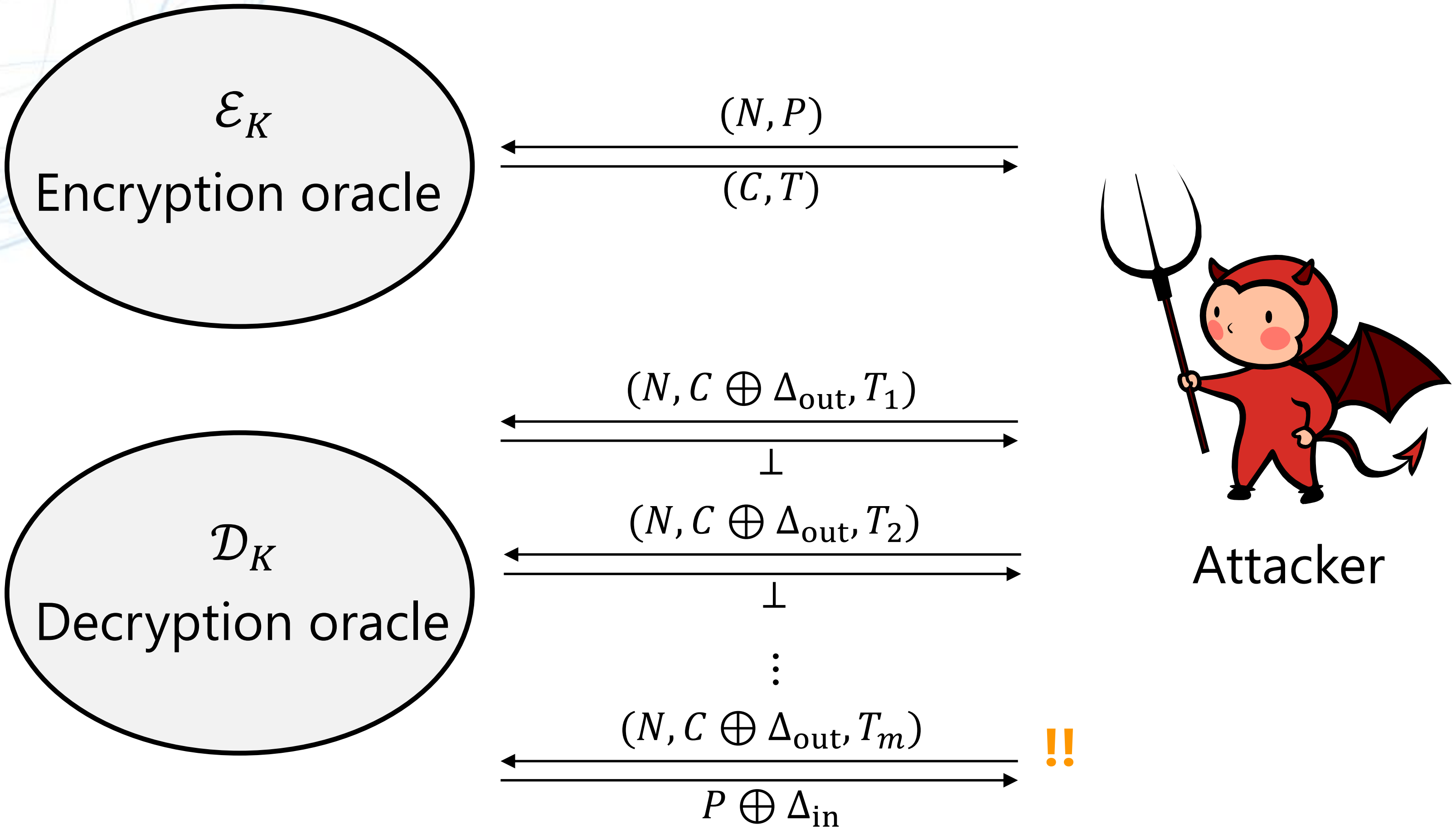
- What is problem?



  - Attacker can query 2^{128} (N, C, T_i) and get P with lower than 2^{128} complexity.

Getting valid PC pairs with the same nonce

- An attacker can get a valid pair (N, P, C, T) and $(N, P \oplus \Delta_{\mathrm{in}}, C \oplus \Delta_{\mathrm{out}}, T)$ with a complexity of 2^{128} chosen Δ_{out}.



What do we learn from this?

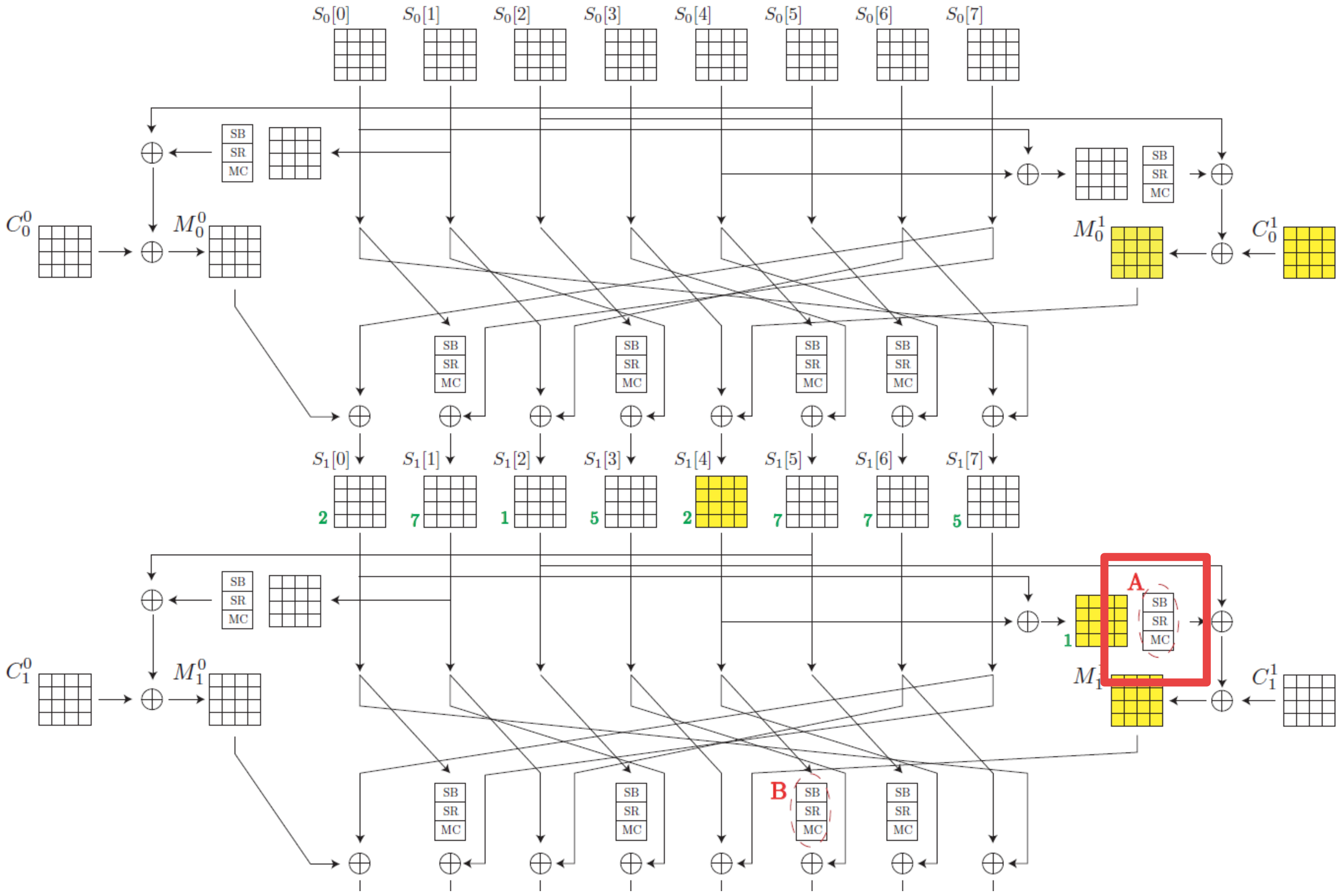
NTT

- Nonce-respecting scenario.
  - We can eliminate the attack using the PC pair with the same nonce.
- Unbalanced bit security.
  - An attacker can collect such a pair with a complexity of 2^{τ}.
  - When $\tau < \kappa$, we must care about such attacks.
- We can't eliminate the attack exploiting multiple PC pairs with the same nonce!!

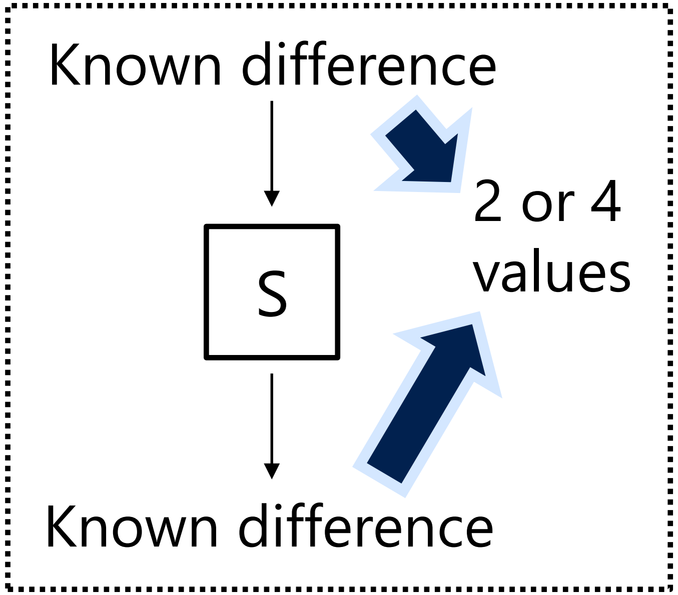
In the case of Rocca

- The security claim is the nonce-respecting only.
- The designers don't consider the attack using the PC pairs with the same nonce.

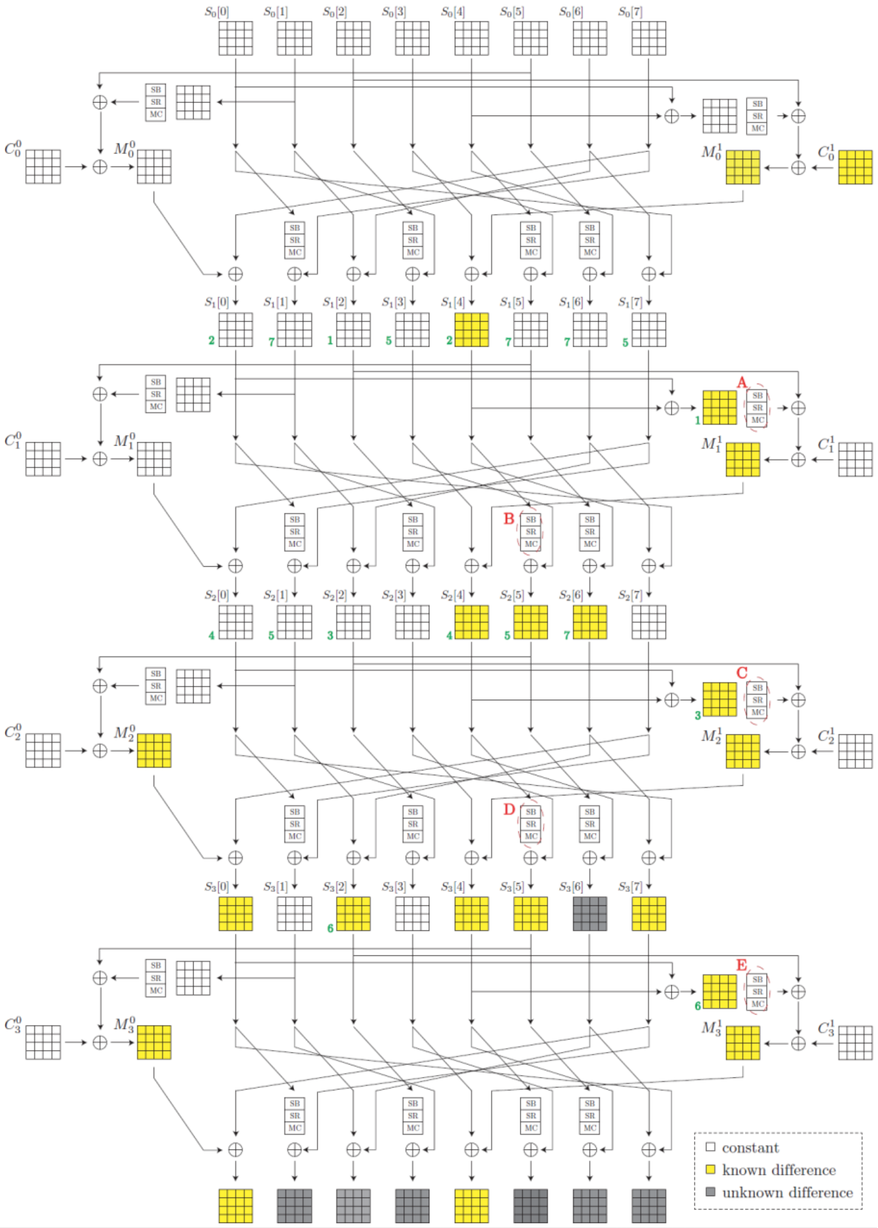
Attack using a PC pair with the same nonce



Add difference!!

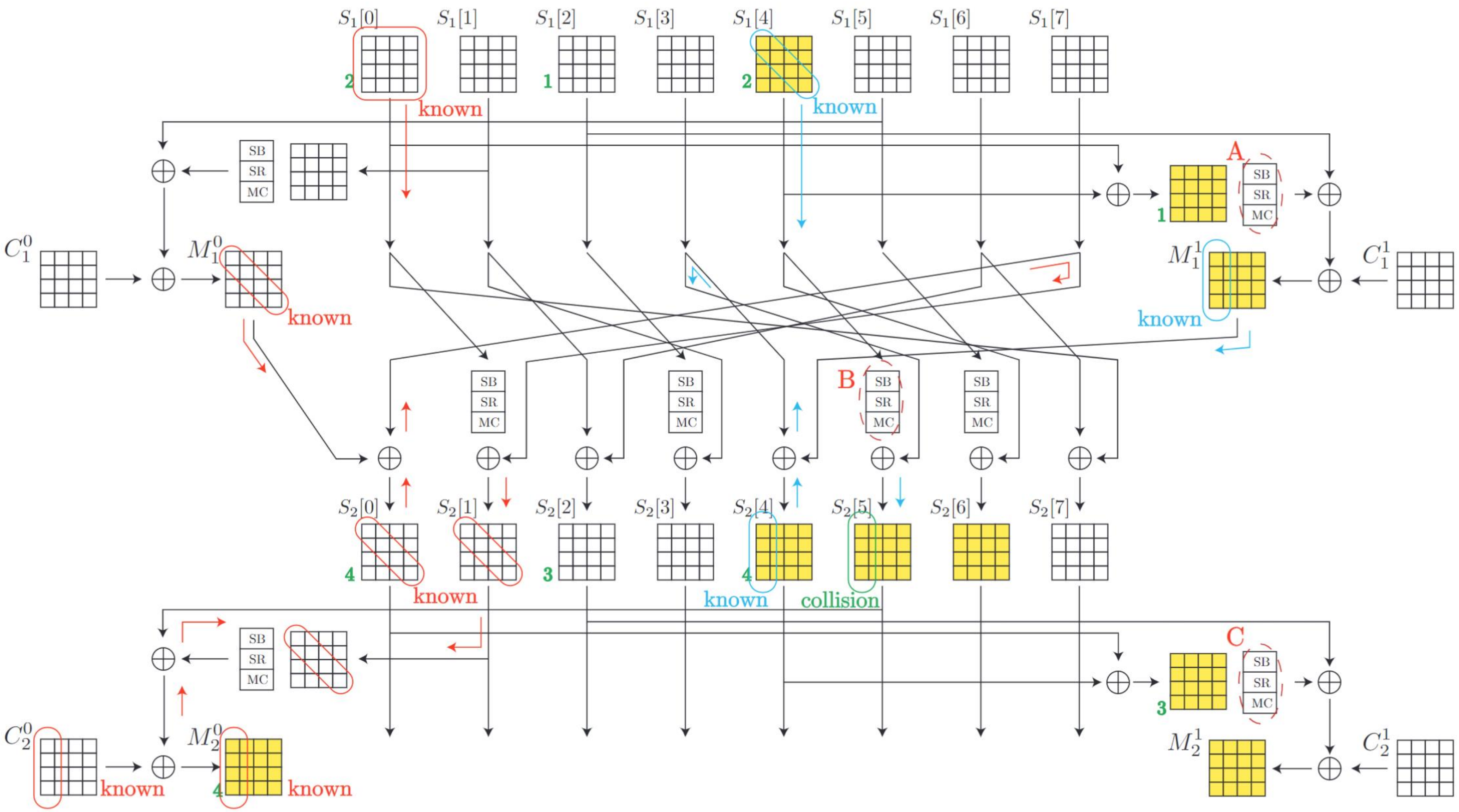


Attack using a PC pair with the same nonce



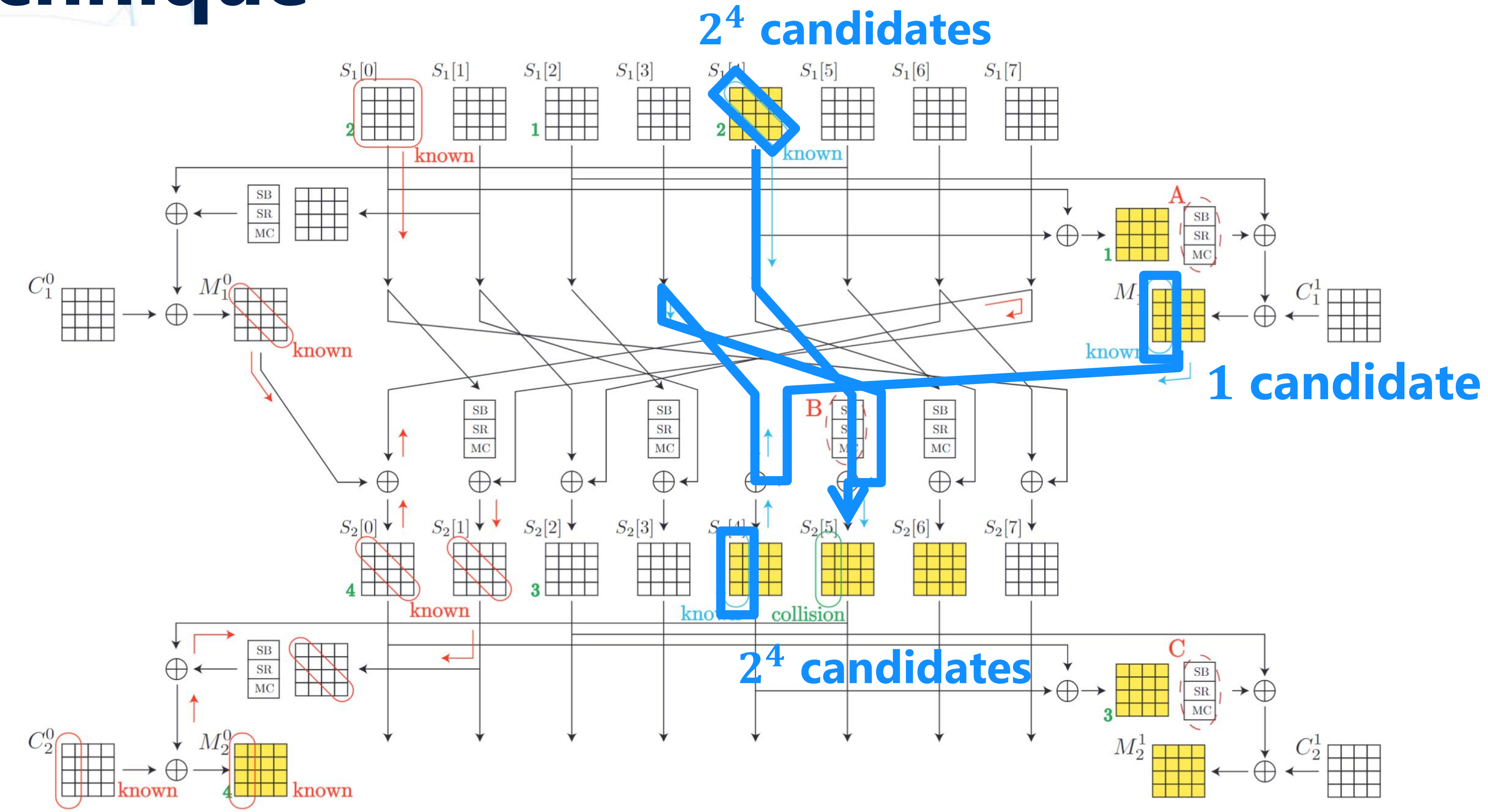
- Yellow bytes have **known** difference.
- We know input/output differences in five AES round functions (**A, B, C, D, and E**).
  - #candidates of each byte is reduced to (almost) 2 with high probability.
  - #candidates of each 128-bit state is reduced to 2^{16} with high probability.
- Step-by-step straightforward procedure recovers the whole internal state with a complexity of 2^{64} using only one PC pair.

MitM technique



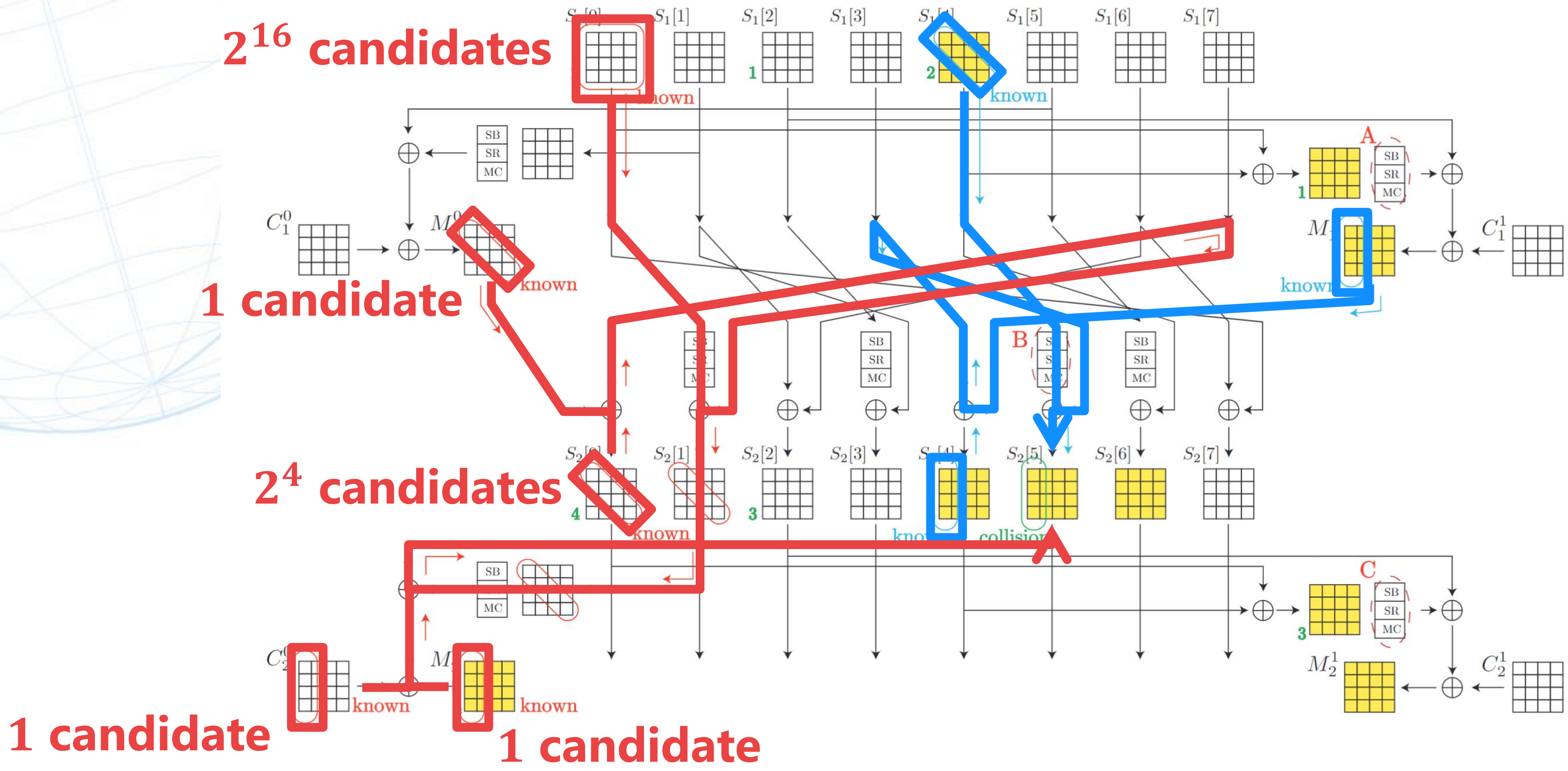
We compute each columns of $S_2[5]$.

MitM technique



$2^4 \times 2^4 \times 1 = 2^8$ candidates in the blue line.

MitM technique

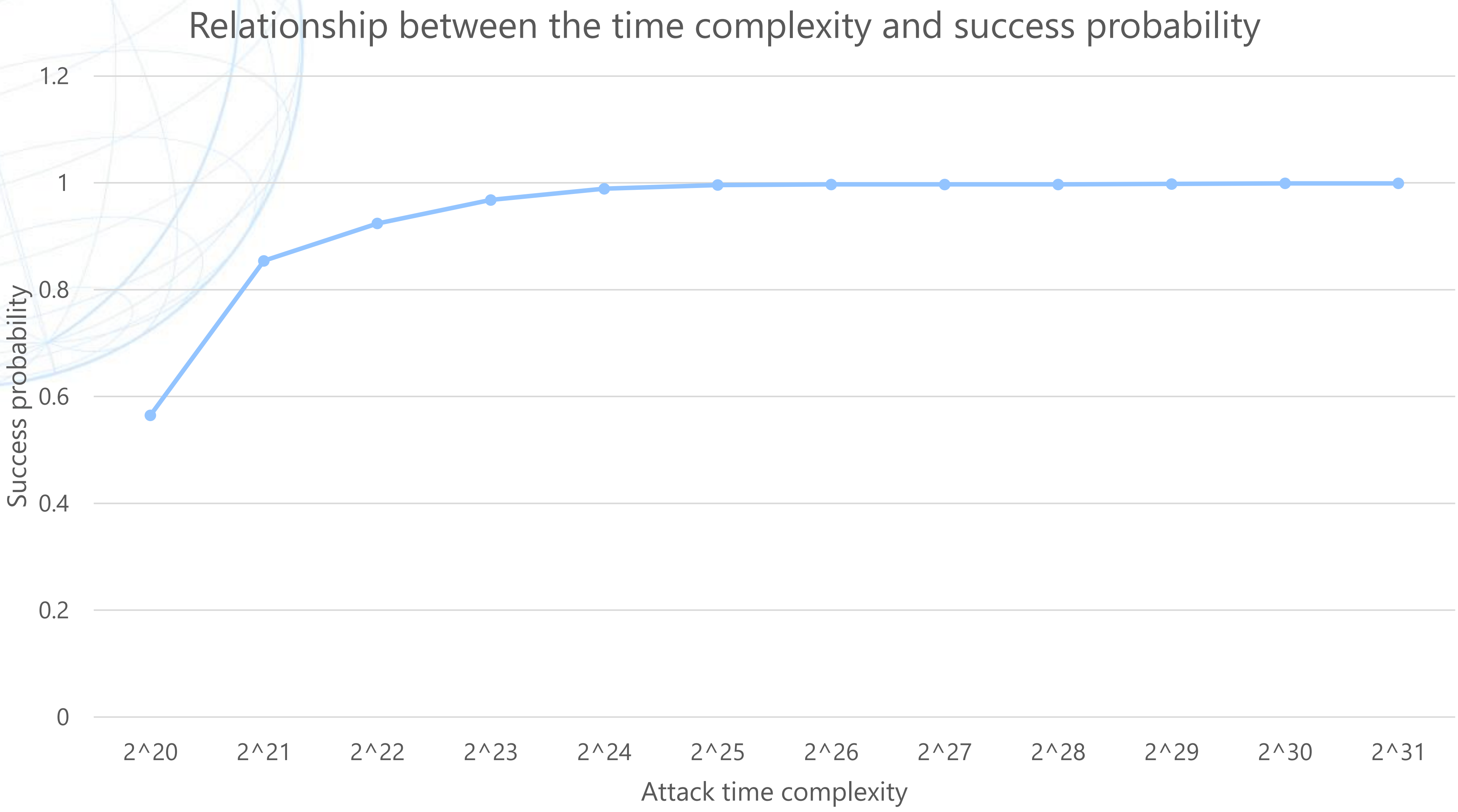


$2^4 \times 2^4 \times 1 = 2^8$ candidates in the blue line.

$2^{16} \times 1 \times 2^4 \times 1 \times 1 = 2^{20}$ candidates in the red line.

32-bit matching!!

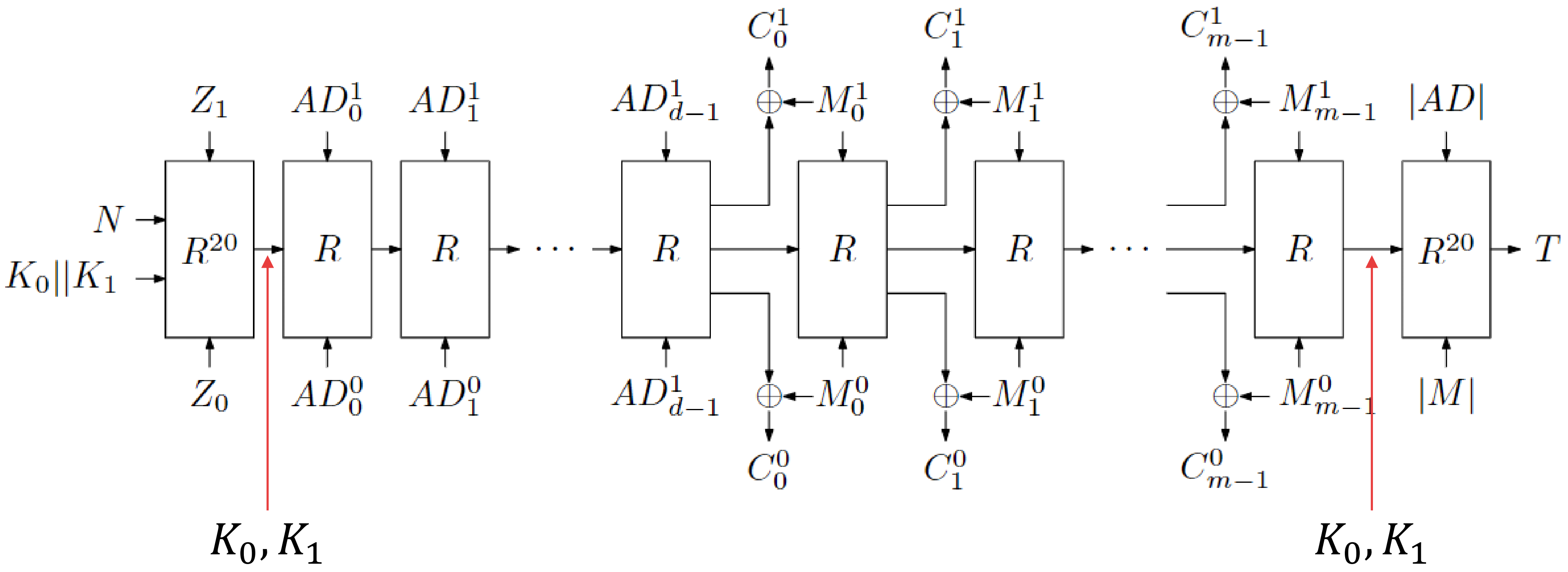
Experimental results



Possible countermeasure

NTT

- Involve the secret key after/before of the initialization/finalization.



This countermeasure never prevents the internal state recovery attack.
However, even if it's recovered, the countermeasure makes the key recovery attack (and trivial universal forgery attack) non-trivial.

On Validity of the Security Claim

Unbalanced security claims of Rocca

- Rocca claims unbalanced bit-security level.

1. 256-bit key recovery & 128-bit tag.
   - We propose the key-recovery attack with the complexity of 2^{128}.

2. **256-bit IND security & 128-bit tag.**
   - **We consider the security definition capturing such an unbalanced security claim and show the (in)feasibility of this claim.**

Security Claim of Rocca

NTT

Security claims. Rocca provides 256-bit security against key-recovery and distinguishing attacks and 128-bit security against forgery attacks in the nonce-respecting setting. We do not claim its security in the related-key and known-key settings.

Rocca doesn't satisfy this unbalanced security claim because a key-recovery attack of complexity 2^{128} exists.

Still, the following question is of theoretical interest:

- **Is 256-bit indistinguishability achievable for any AEAD with relatively short, 128-bit tags?**

High IND security with short tag

- AEAD users (non cryptographers) may truncate the tag without careful consideration.
  - Intuitively, the tag truncation only affects the forgery security.
  - AEAD user must truncate it due to the narrow bandwidth or storage restriction.

- Can we truncate the tag of AEAD without too much impact on the indistinguishability security?
  - If the security is only ensured under the unified AE security, the answer is no.

Security Notion 1: Unified AE Security

NTT

Real

Enc_K Dec_K

v.s.

Ideal

$\$$ $\perp$

Standard setting when considering security of AEADs against CCAs

- Distinguishable with 2^t queries (t : tag length)
(by querying all tags to Dec oracle for a fixed N and C)

- The unbalanced security claim is unachievable for **any** AEAD.

Security Notion 2: IND-CCA

NTT

Real

Enc_K Dec_K

v.s.

Ideal

$\$$ Dec_K

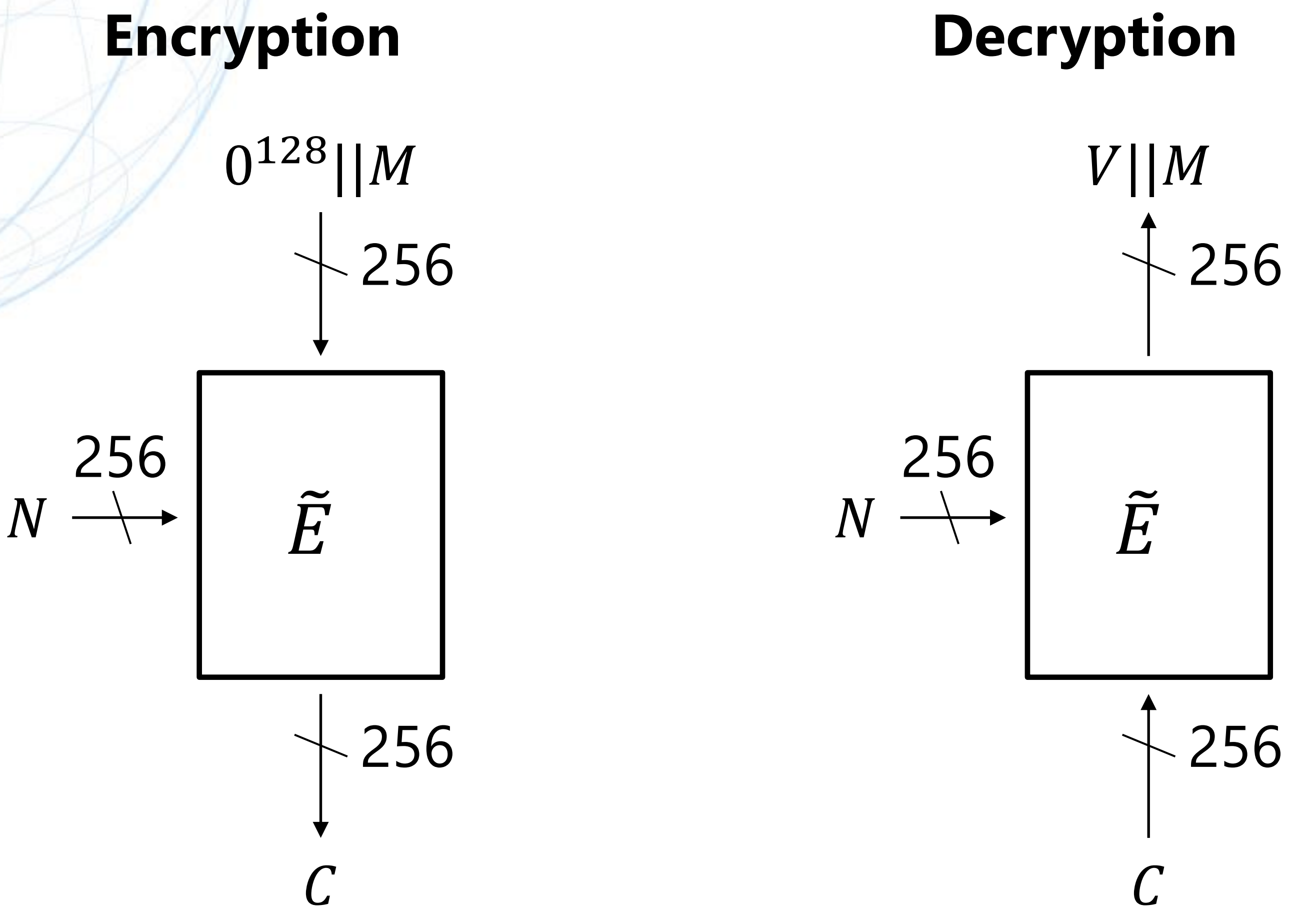
Infeasibility result on IND-CCA:

- Online AEADs cannot achieve more than t-bit IND-CCA-security [Kha22]

[Kha22] Mustafa Khairallah. Security of COFB against chosen ciphertext attacks. IACRTrans. Symmetric Cryptol., 2022(1):138–157, 2022.

Feasibility results : Encode-then-Encipher

- $\tilde{E}$ is a TBC with a 256-bit input, 256-bit output, and 256-bit tweak.



Check if $V = 0^{128}$ or not.

Conclusion

- Attack
  - We break the key-recovery security claim of Rocca.
    - The attack requires 2^{128} complexity.
  - The attack is practical when the nonce is misused or RUP.
  - We can say that Rocca's security level is tag length rather than key length.
- Validity of the security claim
  - Discussing unbalanced security is meaningful.
  - It's out of focus of the unified AE security, and we need to consider others.
  - Achieving the IND-CCA security is difficult in the online AEAD.
  - Encode-then-encipher is feasible solution, but far from the practical.
    - More practical solution is open question.