Security Proof for Romulus-T

Chun Guo [1], Tetsu Iwata [2], Mustafa Khairallah [3], Kazuhiko Minematsu [4], Thomas Peyrin [5]

[1] Shandong University, China
chun.guo@sdu.edu.cn
[2] Nagoya University, Japan
tetsu.iwata@nagoya-u.jp
[3] Seagate Research, Singapore
mustafa.khairallah@seagate.com
[4] NEC Corporation, Japan
k-minematsu@nec.com
[5] Nanyang Technological University, Singapore
thomas.peyrin@ntu.edu.sg

September 20, 2022

Abstract. Romulus consists of four modes, Romulus-N, Romulus-M, Romulus-T, and Romulus-H. Among them, Romulus-T is an AEAD mode designed for leakage-resilience, and it takes a variant of TEDT that slightly differs from the original proposal of Berti *et al.* (CHES 2020). We provide a formal support for its claimed single-user security of $(n - \log n)$ bits.

1 Introduction

Romulus is a submission to the NIST Lightweight Cryptography competition, and it consists of four modes, Romulus-N, Romulus-M, Romulus-T, and Romulus-H. Romulus-N is a nonce-based AE (NAE) mode, Romulus-M is a nonce misuse-resistant AE (MRAE) mode, Romulus-T is a leakage-resilient AE mode, and Romulus-H is a hash function. All these modes use a tweakable block cipher Skinny-128-384+, which basically consists of Skinny-128-384 [3] reduced to 40 rounds.

Romulus-T follows TEDT of Berti *et al.* [4], with slight differences in the nonce length, the maximum message length, and the definition of the hash function. For the hash function, Romulus-T adopts Romulus-H, which is the MDPH hashing mode [18] that combines a compression function by Hirose [12] with a domain extension scheme by Hirose *et al.* [13].

We provide a formal support for its claimed single-user security of $(n - \log n)$ bits [11], where $n = 128$. Concretely, we show the following results:

- First, in Theorem 1, we prove the $(n-\log n)$-bit CIML2 security in the "unbounded leakage" setting [5, 6] and in the ideal cipher model. The notion stands for *ciphertext integrity with misuse*-resistance *and (encryption and decryption) leakage*, and it captures a strong integrity adversary that observes leakages of the encryption and decryption oracles, and that can repeat nonces.
- Next, in Theorem 2, we prove the $(n-\log n)$-bit CCAm$ security in the ideal cipher model *without leakages*, and in the nonce-misuse resilience setting. The notion stands for *chosen-ciphertext security with misuse-resilience*, and it captures a strong confidentiality adversary that has the encryption and decryption oracles, and that can repeat nonces. We note that as we are dealing with misuse-resilience [2], we focus on the case where the nonces used for encrypting confidential messages are not reused.
- Finally, in Theorem 3, we prove $n/2$-bit CCAmL2 security in the ideal cipher model. The notion stands for *chosen-ciphertext security with misuse-resilience and leakage*, and it captures a strong confidentiality adversary that observes the leakages of the encryption and decryption oracles, and that can repeat nonces.

2 Preliminaries

Let $\{0,1\}^*$ be the set of all finite bit strings, including the empty string ε. For $X \in \{0,1\}^*$, let $|X|$ denote its bit length. Here $|\varepsilon| = 0$. For integer $n \geq 0$, let $\{0,1\}^n$ be the set of n-bit strings, and let $\{0,1\}^{\mathsf{L}_{\mathsf{enc}} n} = \bigcup_{i \in \{0,\ldots,n\}} \{0,1\}^i$, where $\{0,1\}^0 = \{\varepsilon\}$. If X is uniformly distributed over a set $\mathcal{X}$, we write $X \xleftarrow{\$} \mathcal{X}$. For two bit strings X and Y, $X \,\|\, Y$ is their concatenation. We also write this as XY if it is clear from the context. Let 0^i (1^i) be the string of i zero bits (i one bits), and for instance we write 10^i for $1 \,\|\, 0^i$. We write $\mathtt{msb}_i(X)$ (resp. $\mathtt{lsb}_i(X)$) to denote the i most (resp. least) significant bits of X. For $X \in \{0,1\}^*$, let $|X|_n = \max\{1, \lceil |X|/n \rceil\}$. Let $(X[1], \ldots, X[x]) \xleftarrow{n} X$ be the parsing of X into n-bit blocks . Here $X[1] \,\|\, X[2] \,\|\, \ldots \,\|\, X[x] = X$ and $x = |X|_n$. When $X = \varepsilon$, we have $X[1] \xleftarrow{n} X$ and $X[1] = \varepsilon$. Let $X \lll i$ denote the left rotation shift of X by i bits.

Padding. Romulus-T and the underlying hash function Romulus-H use an injective padding that is defined on the whole byte strings. In detail, for any $X \in \{0,1\}^*$ of length multiple of 8 (*i.e.*, byte string), let

$$\mathtt{ipad}_l(X) = \begin{cases} \varepsilon, & \text{if } X = \varepsilon, \\ X \,\|\, 0^{l-(|X| \bmod l)-8} \,\|\, c, & \text{if } X \neq \varepsilon \text{ and } |X| \bmod l = 0, \end{cases},$$

where $c = \mathtt{len}_8(Z)$ for some Z of $(|X| \bmod l)$ bits. Here, Z is interpreted as the empty string when $|X| \bmod l = 0$, and otherwise the last partial block obtained by parsing X into l-bit blocks. We remark that X is a byte string. When $|X| \bmod l = 0$, $\mathtt{ipad}_l$ appends 0^l to X. As a special case, when $X = \varepsilon$, $\mathtt{ipad}_l(X) = 0^l$. When $|X| \bmod l \neq 0$ the padding is interpreted as applying $\mathtt{ipad}_l$ to the last (partial) block. The injectivity is easily confirmed. We use $l = 128$ for Romulus-T and $l = 256$ for Romulus-H. The byte length encoding uses the last byte.

(Tweakable) Block Cipher. A tweakable block cipher (TBC) is a keyed function $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T_W} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K}$ is the key space, $\mathcal{T_W}$ is the tweak space, and $\mathcal{M} = \{0,1\}^n$ is the message space, such that for any $(K, T_w) \in \mathcal{K} \times \mathcal{T_W}$, $\widetilde{\mathsf{E}}(K, T_w, \cdot)$, or $\widetilde{\mathsf{E}}_K^{T_w}(\cdot)$ for short, is a permutation over $\mathcal{M}$. The decryption routine is written as $(\widetilde{\mathsf{E}}_K^{T_w})^{-1}(\cdot)$, where if $C = \widetilde{\mathsf{E}}_K^{T_w}(M)$ holds for some (K, T_w, M) we have $M = (\widetilde{\mathsf{E}}_K^{T_w})^{-1}(C)$. When $\mathcal{T_W}$ is singleton, it is essentially a block cipher.

A tweakable URP (TURP) with a tweak space $\mathcal{TW}$ and a message space $\mathcal{X}$, $\widetilde{\mathsf{P}} : \mathcal{TW} \times \mathcal{X} \to \mathcal{X}$, is a random tweakable permutation with uniform distribution over $\mathrm{Perm}(\mathcal{TW}, \mathcal{X})$. The decryption is written as $\mathsf{P}^{-1}(*)$ for URP and $(\widetilde{\mathsf{P}}^{-1})^{T_w}(*)$ for TURP given tweak T_w. An ideal TBC $\widetilde{\mathsf{E}} : \mathcal{K} \times \mathcal{T_W} \times \mathcal{M} \to \mathcal{M}$ is a TBC sampled uniformly at random from all TBCs with key space $\mathcal{K}$, tweak space $\mathcal{T_W}$ and plaintext space $\mathcal{M}$. In this case, $\widetilde{\mathsf{E}}_K^{T_w}$ is a random permutation of $\mathcal{M}$ for each $(K, T_w) \in \mathcal{K} \times \mathcal{T_W}$ even if the key K is *public*.

Definition 1. *A nonce-based authenticated encryption (NAE) is a tuple $\Pi = (\mathcal{E}, \mathcal{D})$. For key space $\mathcal{K}$, nonce space $\mathcal{N}$, message space $\mathcal{M}$ and associated data (AD) space $\mathcal{A}$, the encryption algorithm $\mathcal{E}$ takes a key $K \in \mathcal{K}$ and a tuple (N, A, M) of a nonce $N \in \mathcal{N}$, an AD $A \in \mathcal{A}$, and a plaintext $M \in \mathcal{M}$ as input, and returns a ciphertext $C \in \mathcal{M}$ and a tag $T \in \mathcal{T}$. Typically, $\mathcal{T} = \{0,1\}^\tau$ for a fixed, small τ. The decryption algorithm $\mathcal{D}$ takes $K \in \mathcal{K}$ and the tuple (N, A, C, T) as input, and returns $M \in \mathcal{M}$ or the reject symbol $\perp$. The corresponding encryption and decryption oracles are written as $\mathcal{E}_K$ and $\mathcal{D}_K$.*

An NAE scheme usually assumes each nonce in encryption queries to be distinct. However, our security definitions consider the case that nonces may be reused (misused) in encryption queries.

Security Definitions: Black-box. For non-leaking security, we follow the nonce-misuse resilience model of Ashur *et al.* [2, 4]. In detail, Ashur *et al.*'s idea is to divide adversarial encryption queries into *nonce-respecting challenge* and *nonce-reusing non-challenge* ones, and only require (confidentiality and integrity) security among challenge queries. For the concrete formalism, we follow the extension CCAm\$ notion of Berti *et al.* [4].

Definition 2 (CCAm\$ **Advantage**)**.** *Given a nonce-based authenticated encryption* $\mathsf{AEAD} = (\mathcal{E}, \mathcal{D})$*, the* chosen ciphertext misuse resilience advantage *of an adversary* $\mathcal{A}$ *against* AEAD *is*

$$\mathbf{Adv}^{\mathsf{CCAm\$}}_{\mathsf{AEAD}}(\mathcal{A}) := \left| \Pr\left[\mathcal{A}^{\mathcal{E}_K, \mathcal{E}_K, \mathcal{D}_K, \widetilde{\mathsf{E}}, \widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathcal{E}_K, \$, \perp, \widetilde{\mathsf{E}}, \widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\right] \right|,$$

where the probability is taken over the key $K \leftarrow \mathcal{K}$, over $\mathcal{A}$'s random tape and the ideal TBC $\widetilde{\mathsf{E}}$ and where:

- $\mathcal{E}_K(N, A, M)$*: outputs* $\mathcal{E}_K(N, A, M)$*;*
- $\$(N, A, M)$ *outputs and associates a fresh random pair $(C, T) \xleftarrow{\$} \mathcal{C}_{|M|} \times \mathcal{T}$ to fresh input, and the associated C otherwise;*
- $\mathcal{D}_K(N, A, C, T)$ *outputs $\mathcal{D}_K(N, A, C, T)$ if (N, A, C, T) is not an oracle answer to an encryption query (N, A, M) for some M, and $\perp$ otherwise;*
- $\perp(N, A, C, T)$ *outputs* $\perp$*;*
- *(i) nonce N cannot be used both in query to $O_1(N, *, *)$ and $O_2(N, *, *)$; (ii) $O_2(*, *, *)$ is nonce-respecting; (iii) if (C, T) is returned by $O_1(N, A, M)$ or $O_2(N, A, M)$ query, $O_3(N, A, C, T)$ is forbidden.*

Security Definitions: Leakage. For leakage security, we follow Berti *et al.*'s [4] *ciphertext integrity with misuse*-resistance *and (encryption & decryption) leakage (*CIML2*)* and *chosen-ciphertext security with misuse-*resilience *and leakage (*CCAmL2*)*. We adopt the single-user version of the muCIML2 definition of Berti *et al.* [4].

Definition 3 (CIML2 **Advantage**). *Given a nonce-based authenticated encryption* $\mathsf{AEAD} = (\mathcal{E}, \mathcal{D})$ *with leakage function pair* $\mathsf{L} = (\mathsf{L}_{\mathsf{enc}}, \mathsf{L}_{\mathsf{dec}})$*, the ciphertext integrity advantage with misuse-*resistance *and leakage of an adversary* $\mathcal{A}$ *against* AEAD *is*

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{AEAD},\mathsf{L}}(\mathcal{A}) := \left| \Pr\left[\mathcal{A}^{\mathcal{LE}_K, \mathcal{LD}_K, \widetilde{\mathsf{E}}, \widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathcal{LE}_K, \mathcal{LD}^{\perp}_K, \widetilde{\mathsf{E}}, \widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\right] \right| ,$$

where the probability is taken over the key $K \leftarrow \mathcal{K}$*, over* $\mathcal{A}$*'s random tape and the ideal TBC* $\widetilde{\mathsf{E}}$*, and where:*

- $\mathcal{LE}_K(N, A, M)$*: outputs the ciphertext* $\mathcal{E}_K(N, A, M)$ *and the corresponding leakage* $\mathsf{L}_{\mathsf{enc}}(K, N, A, M)$*;*
- $\mathcal{LD}_K(N, A, C, T)$*: outputs* $\big(\mathcal{D}_K(N, A, C, T), \mathsf{L}_{\mathsf{dec}}(K, N, A, C, T)\big)$*;*
- $\mathcal{LD}^{\perp}_K(*, *, *, *)$*: computes* $\mathsf{leak}_{\mathsf{d}} \leftarrow \mathsf{L}_{\mathsf{dec}}(K, N, A, C, T)$ *and if* C *is an output of some leaking encryption query* (N, A, M) *for some* M *outputs* $(M, \mathsf{leak}_{\mathsf{d}})$*, else outputs* $(\perp, \mathsf{leak}_{\mathsf{d}})$*.*
- $\mathcal{A}$ *is forbidden to make trivial decryption queries, i.e., query* $\mathcal{D}_K(N, A, C)$ *such that the action* $\mathcal{E}_K(N, A, M) \rightarrow C$ *happens before.*

We adopt the single-user version of the muCCAmL2 definition of Berti *et al.* [4].

Definition 4 (CCAmL2 **Advantage**). *Given an authenticated encryption* $\mathsf{AEAD} = (\mathcal{E}, \mathcal{D})$ *with leakage function pair* $\mathsf{L} = (\mathsf{L}_{\mathsf{enc}}, \mathsf{L}_{\mathsf{dec}})$*, the* chosen-ciphertext advantage with misuse-resilience and leakage *of an adversary* $\mathcal{A}$ *against* AEAD *is*

$$\mathbf{Adv}^{\mathsf{CCAmL2}}_{\mathsf{AEAD},\mathsf{L}}(\mathcal{A}) := \left| \Pr\left[\mathsf{PrivK}^{\mathsf{CCAmL2},0}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}} \Rightarrow 1\right] - \Pr\left[\mathsf{PrivK}^{\mathsf{CCAmL2},1}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}} \Rightarrow 1\right] \right|,$$

where the security game $\mathsf{PrivK}^{\mathsf{CCAmL2},b}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}}$ *is defined in Figure 1.*

$\mathsf{PrivK}^{\mathsf{CCAmL2},b}_{\mathcal{A},\mathsf{AEAD},\mathsf{L},u}$ is the output of the following experiment:

Initialization: generates $K \leftarrow \mathcal{K}$ and sets $\mathcal{E}_{ch}, \mathcal{E} \leftarrow \emptyset$.

Leaking encryption queries: $\mathcal{A}^{\mathsf{L}}$ gets adaptive access to $\mathcal{LE}(\cdot, \cdot, \cdot)$,
$\mathcal{LE}(N, A, M)$ outputs $\perp$ if $(N, *, *) \in \mathcal{E}_{ch}$, else computes $C \leftarrow \mathcal{E}_K(N, A, M)$ and $\mathsf{leak}_{\mathsf{e}} \leftarrow \mathsf{L}_{\mathsf{enc}}(K, N, A, M)$, updates $\mathcal{E} \leftarrow \mathcal{E} \cup \{N\}$ and finally returns $(C, \mathsf{leak}_{\mathsf{e}})$.

Leaking decryption queries: $\mathcal{A}^{\mathsf{L}}$ gets adaptive access to $\mathcal{LD}(\cdot, \cdot, \cdot)$,
$\mathcal{LD}(N, A, C)$ outputs $\perp$ if $(N, A, C) \in \mathcal{E}_{ch}$, else computes $M \leftarrow \mathcal{D}_K(N, A, C)$ and $\mathsf{leak}_{\mathsf{d}} \leftarrow \mathsf{L}_{\mathsf{dec}}(K, N, A, C)$ and returns $(M, \mathsf{leak}_{\mathsf{d}})$;

Challenge queries: on possibly many occasions $\mathcal{A}^{\mathsf{L}}$ submits $(N_{ch}, A_{ch}, M^0, M^1)$,
If M^0 and M^1 have different (block) length or $N_{ch} \in \mathcal{E}$ or $(N_{ch}, *, *) \in \mathcal{E}_{ch}$, returns $\perp$; Else computes $C^b \leftarrow \mathcal{E}_K(N_{ch}, A_{ch}, M^b)$ and $\mathsf{leak}^b_{\mathsf{e}} \leftarrow \mathsf{L}_{\mathsf{enc}}(K, N_{ch}, A_{ch}, M^b)$, updates $\mathcal{E}_{ch} \leftarrow \mathcal{E}_{ch} \cup \{(N_{ch}, A_{ch}, C^b)\}$ and finally returns $(C^b, \mathsf{leak}^b_{\mathsf{e}})$;

Decryption challenge leakage queries: $\mathcal{A}^{\mathsf{L}}$ gets adaptive access to $\mathsf{L}_{\mathsf{decch}}(\cdot, \cdot, \cdot)$,
$\mathsf{L}_{\mathsf{decch}}(N_{ch}, A_{ch}, C^b)$ computes and outputs $\mathsf{leak}^b_{\mathsf{d}} \leftarrow \mathsf{L}_{\mathsf{dec}}(k, N_{ch}, A_{ch}, C^b)$ if $(N_{ch}, A_{ch}, C^b) \in \mathcal{E}_{ch}$; Else it outputs $\perp$;

Finalization: $\mathcal{A}^{\mathsf{L}}$ outputs a guess bit b' which is defined as the output of the game.

Fig. 1: The $\mathsf{PrivK}^{\mathsf{CCAmL2},b}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}}$ game.

Balls-in-bin Lemma. We'll rely on a balls-in-bin lemma from [20, Appendix A] presented as follows.

Lemma 1 (Balls-in-Bin). *Consider throwing a ball into a bin that is chosen independently uniformly at random from* $2^n \geq 8$ *bins. Then the probability that, after throwing* σ *balls with* $8 \leq \sigma \leq 2^n$*, any bin contains* $2\log_2 \sigma$ *balls or more, is less than* $\frac{1}{2^n}$*.*

3 Specifications for Romulus-T

The specification of Romulus-T is depicted in Figures 2 and 3. Several optimizations have been performed over the original TEDT specification in order to make implementations more efficient and more streamlined with other Romulus members. The first modification is using Romulus-H, this helps implementations in two aspects:

(1) It absorbs 2 blocks of messages for every 2 calls to the TBC, making the performance faster than comparable hash functions.
(2) Since the Hir compression function shares the same tweakey input between the two calls, the compression function can be easily parallized in both software and hardware.

Similar parallilizability arguments apply to the key-stream generation (the inner for loop of Figure 2), where both calls to the TBC can be parallilized, with only one different byte in their respective tweakeys. The counters used in the key-stream generation are the same counters used in Romulus-N, and the padding used in the hash function is a combination between the lightweight padding used in Romulus-H and padding the counter value used during key-stream generation.

Besides, since these two parts of the scheme can be implemented without masking, their performance can be very competitive compared to schemes that require uniform masking. Unprotected Skinny hardware and software implementations have been shown to be quite competitive in terms of their performance and energy consumption [17, 15, 7, 3, 1].

When it comes to the protected TBC calls; the key derivation function and the tag generation/verification function, they are built using protected TBC implementations. Protected Skinny implementations have been shown to have significant advantage over other types of primitives, as the tweakey (the majority of its state) can be either unprotected or cheaply protected, and some parts of it (almost half the state) is public and does not need any direct protection. This is shown in research work on masked TBC-based modes [19] or benchmarks on Romulus-N [22, 14]. While masked implementations TBCs suffer from a slow-down due to their latency, the protected TBC is only called twice, amortizing any performance penalty, a feature shared only by a handful of candidates; ISAP and some Ascon implementations. We also refer the reader to the excellent study by Verhamme *et al.* on ISAP, Ascon and Romulus-T [23].

Algorithm $\textsf{Romulus-T}.\mathcal{E}[\widetilde{\mathsf{E}}]_K(N, A, M)$

if $M = \epsilon$ **then** $C \leftarrow \epsilon$
else
  $(M[1], \ldots, M[m]) \xleftarrow{n} M$
  $S \leftarrow \widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}(N)$
  for $i = 1$ **to** $m - 1$
    $C[i] \leftarrow M[i] \oplus \widetilde{\mathsf{E}}_S^{(0^n,64,\pi(i-1))}(N)$
    $S \leftarrow \widetilde{\mathsf{E}}_S^{(0^n,65,\pi(i-1))}(N)$
  end for
  $z \leftarrow \widetilde{\mathsf{E}}_S^{(0^n,64,\pi(m-1))}(N)$
  $C[m] \leftarrow M[m] \oplus \mathtt{lsb}_{|M[m]|}(z)$
  $C \leftarrow C[1] \,\|\, \ldots \,\|\, C[m]$
$U \leftarrow \mathtt{ipad}^*(A) \| \mathtt{ipad}^*(C) \| N \| \pi(|C|_n)$
$H \leftarrow \textsf{Romulus-H}[\widetilde{\mathsf{E}}](U)$
$(L, R) \xleftarrow{n} H$
$T \leftarrow \widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}(L)$
return (C, T)

Algorithm $\textsf{Romulus-T}.\mathcal{D}[\widetilde{\mathsf{E}}]_K(N, A, C, T)$

$U \leftarrow \mathtt{ipad}^*(A) \| \mathtt{ipad}^*(C) \| N \| \pi(|C|_n)$
$H \leftarrow \textsf{Romulus-H}[\widetilde{\mathsf{E}}](U)$
$(L, R) \xleftarrow{n} H$
$L' \leftarrow \left(\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}\right)^{-1}(T)$
if $L \neq L'$ **then return** $\perp$
else if $M = \epsilon$ **then return** ϵ
else
  $(C[1], \ldots, C[m]) \xleftarrow{n} C$
  $S \leftarrow \widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}(N)$
  for $i = 1$ **to** $m - 1$
    $M[i] \leftarrow C[i] \oplus \widetilde{\mathsf{E}}_S^{(0^n,64,\pi(i-1))}(N)$
    $S \leftarrow \widetilde{\mathsf{E}}_S^{(0^n,65,\pi(i-1))}(N)$
  end for
  $z \leftarrow \widetilde{\mathsf{E}}_S^{(0^n,64,\pi(m-1))}(N)$
  $M[m] \leftarrow C[m] \oplus \mathtt{lsb}_{|C[m]|}(z)$
  $M \leftarrow M[1] \,\|\, \ldots \,\|\, M[m]$
return M

Algorithm $\textsf{Romulus-H}[\widetilde{\mathsf{E}}](M)$

$L \leftarrow 0^n,\ R \leftarrow 0^n$
$(M[1], \ldots, M[m]) \xleftarrow{2n} \mathtt{ipad}_{2n}(M)$
for $i = 1$ **to** $m - 1$
  $(L, R) \leftarrow \textsf{Hir}[\widetilde{\mathsf{E}}](L, R, M[i])$
$Y \leftarrow \textsf{Hir}[\widetilde{\mathsf{E}}](\varphi(L, R), M[m])$
return Y

Algorithm $\textsf{Hir}[\widetilde{\mathsf{E}}](L, R, M)$

$L' \leftarrow \widetilde{\mathsf{E}}_R^M(L) \oplus L$
$R' \leftarrow \widetilde{\mathsf{E}}_R^M(L \oplus 1) \oplus L \oplus 1$
return (L', R')

Algorithm $\varphi(L, R)$

$L' \leftarrow L \oplus 2$
return (L', R)

Fig. 2: The Romulus-T leakage-resilient AEAD modes. $\pi : \{0, ..., 2^{56} - 2\} \rightarrow \{0, 1\}^{n-8}$ is a bijective mapping, and please refer to the specification document [11] for the concrete instantiation of π. The red underlined statements are recommended for side-channel secure implementations.

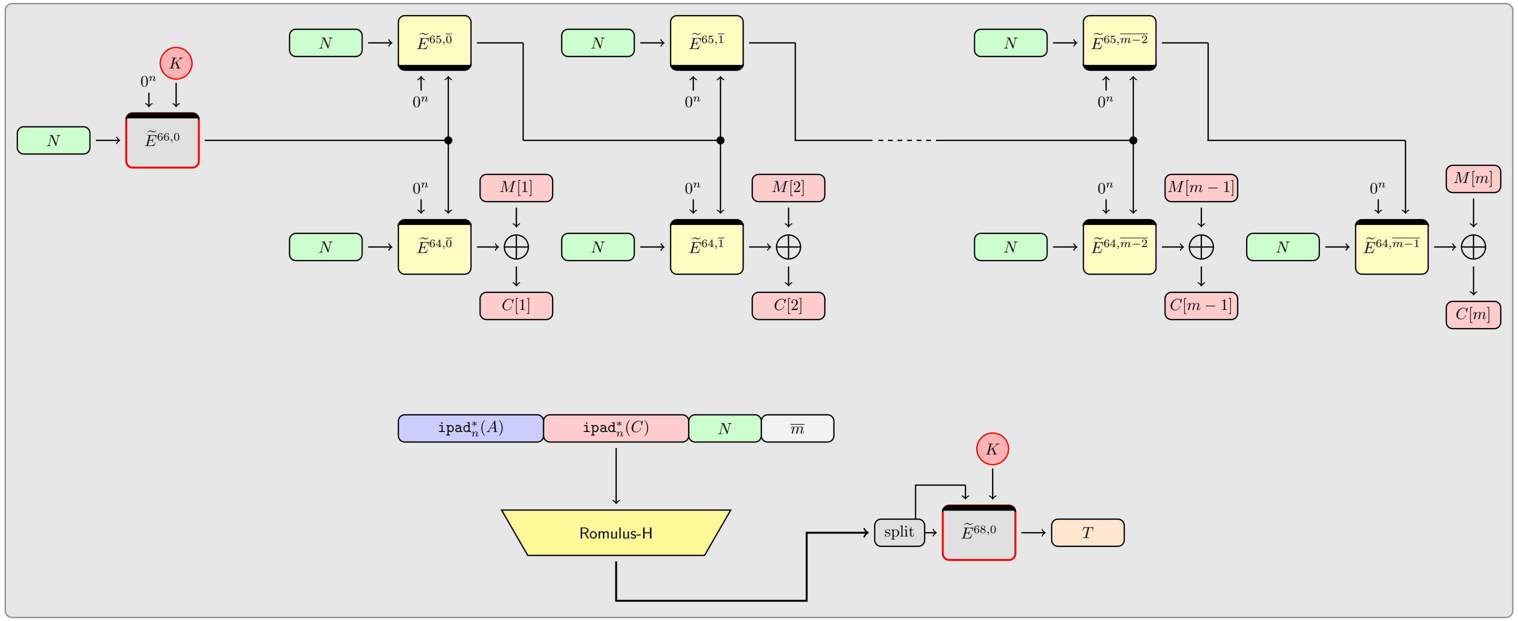

Fig. 3: The Romulus-T leakage-resilient AE mode. The red-circled TBC calls are the Key-Derivation Function (KDF) and Tag Generation Function (TGF): for side-channel security they need heavy protection to be "leak-free", while the other TBC calls can be leaking. Note that the value 0 in the tweak input of the TGF and KDF is to be understood as 0^{56}, not as $\overline{0}$. This shows the encryption when the last message block has full n bits, otherwise we chop the TBC output. See Figure 2.

4 CIML2 Security

In this section we establish the leakage resilient integrity of Romulus-T. We prove the CIML2 security in the "unbounded leakage" setting [5, 6] and in the ideal cipher model.[1] Formally, we assume that all the intermediate values completely leak except the master key (i.e., K of $\widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}$ and $\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}$) remains secret. This means that every internal call to the TBC $\widetilde{\mathsf{E}}_K^{T_w}(X) \to Y$ or $(\widetilde{\mathsf{E}}_K^{T_w})^{-1}(Y) \to X$ completely leaks $\{K, T_w, X, Y\}$, and every internal action $C \leftarrow Y \oplus M$ completely leaks $\{Y, M, C\}$. We denote this family of leakage functions by L^*.

Since we analyze Romulus-T in the ideal TBC model, we can prove information theoretic security and adversarial power is solely characterized by the number of queries. To simplify notations, we define

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T},\mathsf{L}^*}(q_e, q_d, \sigma_m, \sigma_a, q_{\widetilde{\mathsf{E}}}) := \max \left\{ \mathbf{Adv}^{\mathsf{muCIML2}}_{\mathcal{A},\mathsf{TEDT},\mathsf{L}^*,u} \right\},$$

where the maximum is taken over all CIML2 adversaries making q_e leaking encryption queries, q_d leaking decryption queries, $q_{\widetilde{\mathsf{E}}}$ ideal TBC queries and has σ_a n-bit blocks in the queried associated data and σ_m n-bit blocks in the queried messages (including the incomplete last blocks).

Then, our main result is as follows. This claim implies the claimed $n - \log_2 n$-bit integrity security [11, Sect. 4.4]. In addition, the security is kept against full nonce misuse and full leakages (except that the AEAD key K does not leak).

Theorem 1. *Assume that $n \geq 3$ and leakage L^* is "unbounded" as above. Then, in the ideal TBC model, when $Q := 3\sigma_m + \sigma_a + 6(q_e + q_d) + q_{\widetilde{\mathsf{E}}} \leq 2^n/8$ it holds*

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T128},\mathsf{L}^*}(q_e, q_d, \sigma_m, \sigma_a, q_{\widetilde{\mathsf{E}}}) \leq \frac{12Q + 2nq_d}{2^n} + \frac{8Q(q_e + q_d)}{2^{2n}}. \tag{1}$$

The proofs are available in Appendix B.

As long as we carefully protect the aforementioned master key (K in $\widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}$ and $\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}$), the bounds are asymptotically optimal $O\left(\frac{\sigma_a+\sigma_m+q_{\widetilde{\mathsf{E}}}+nq_d}{2^n}\right)$. Concretely, when $n = 128$, integrity is up to $\sigma \gg 2^{120}$ blocks, $q_{\widetilde{\mathsf{E}}} \gg 2^{120}$ offline computations (derived from the condition $3\sigma_m + \sigma_a + 6(q_e + q_d) + q_{\widetilde{\mathsf{E}}} \leq 2^n/8$) and $q_d \approx 2^{120}$ (due to the term $2nq_d/2^n$) decryption queries.

5 Black-box Security

In this section we prove CCAm$ security for Romulus-T in the ideal cipher model, *without leakages.* To simplify the notations, we define

$$\mathbf{Adv}^{\mathsf{CCAm\$}}_{\mathsf{Romulus\text{-}T}}(q_e, q_m, q_d, \sigma_m, \sigma_a, q_{\widetilde{\mathsf{E}}}) := \max \left\{ \mathbf{Adv}^{\mathsf{CCAm\$}}_{\mathsf{TEDT},\mathcal{A},u} \right\},$$

where the maximum is taken over all CCAm$ adversaries making q_e challenge encryption queries, q_m non-challenge encryption queries, q_d decryption queries, $q_{\widetilde{\mathsf{E}}}$ ideal TBC queries and has σ_a n-bit blocks in all queried associated data and σ_m n-bit blocks in all queried messages (including the incomplete last blocks). The following claim implies the claimed $n - \log_2 n$-bit privacy security [11, Sect. 4.4]. In addition, the security is kept in the nonce-misuse resilience setting (but no leakage here).

Theorem 2. *When $n \geq 3$, in the ideal TBC model, when $Q := 3\sigma_m + \sigma_a + 6(q_e + q_d + q_m) + q_{\widetilde{\mathsf{E}}} \leq 2^n/8$ it holds*

$$\mathbf{Adv}^{\mathsf{CCAm\$}^*}_{\mathsf{Romulus\text{-}T}}(q_e, q_m, q_d, \sigma_m, \sigma_a, q_{\widetilde{\mathsf{E}}}) \leq \frac{(4n + 16)Q + nq_e + 2nq_d}{2^n} + \frac{8Q(q_e + q_d)}{2^{2n}}. \tag{2}$$

The proof is available in Appendix C.

The bounds are almost the same as Theorem 1. When $n = 128$, CCA misuse-resilience is up to $\sigma_a + \sigma_m \approx 2^{120}$ blocks, $q_{\widetilde{\mathsf{E}}} \approx 2^{120}$ offline computations and $q_d \approx 2^{120}$ decryption queries. While it seems that the bounds are not affected by the q_m non-challenge queries, these queries actually affect σ which, in turn, affects the bound.

6 CCAmL2 Security

We now detail the leakage-resistant CCA security of Romulus-T. The leakage model, assumptions and proof approaches all follow [4, Sect. 6], and changes are mostly notational. Though, to be conservative, we present the details in this section.

[1] This is inevitable, since we are using $\widetilde{\mathsf{E}}$ for (keyless) hashing.

6.1 Modeling Leakage Functions

We model the leakage as *probabilistic* efficient functions manipulating and/or computing (partially) secret values. In Romulus-T, each computation of $\widetilde{\mathsf{E}}$ (resp. $\oplus$) comes with some additional (internal) information given by $\mathsf{L}_{\widetilde{\mathsf{E}}}$ (resp. $\mathsf{L}_{\oplus}$). We split the leakage trace resulting from the leaking execution of the TBC $\widetilde{\mathsf{E}}$ between its *input* and *output* parts: if $\widetilde{\mathsf{E}}_K^{T_w}(X) \to Y$, $\mathsf{L}_{\widetilde{\mathsf{E}}}(K, T_w, X, Y) := (\mathsf{L}_{\widetilde{\mathsf{E}}}^{in}(K, T_w; X), \mathsf{L}_{\widetilde{\mathsf{E}}}^{out}(K, T_w; Y))$ with semicolon.

Since the leakage functions $\mathsf{L}_{\widetilde{\mathsf{E}}}^{in}$, $\mathsf{L}_{\widetilde{\mathsf{E}}}^{out}$, and $\mathsf{L}_{\oplus}$ are probabilistic (which is indeed likely in practice), measuring p times the leakage from the same computation would *not* result in completely identical traces. Therefore, we write $[\mathsf{L}_{\oplus}]^p$ for the vector of p leakages of $\oplus$ (and use similar notations for the other operations). Because of the plentiful possible uses of $\widetilde{\mathsf{E}}$, we will next denote its input-output leakage function pair as $(\mathsf{L}^{in}, \mathsf{L}^{out})$ for simplicity.

To prevent "future computation attacks" [10] in the ideal cipher model, we assume oracle-free leakage functions [24]: they cannot make any call to $\widetilde{\mathsf{E}}$, since it is natural for an implementation not to evaluate computations that are unrelated to its current state. Therefore, we will say that the leakage function associated to $\widetilde{\mathsf{E}}$ is oracle-free, if $\mathcal{Q}(\mathsf{L}_{\widetilde{\mathsf{E}}}^{in}) = \mathcal{Q}(\mathsf{L}_{\widetilde{\mathsf{E}}}^{out}) = \emptyset$, where $\mathcal{Q}(\mathsf{L}_{\widetilde{\mathsf{E}}}^{*})$ is the transcript of queries and answers made by $\mathsf{L}_{\widetilde{\mathsf{E}}}^{*}$ to $\widetilde{\mathsf{E}}$ when $\mathsf{L}_{\widetilde{\mathsf{E}}}^{*}$ is evaluated on its inputs.

To achieve confidentiality, the leakages have to be somewhat "bounded". To this end, we essential use the same leakage assumptions as [4, Sect. 6], i.e., the probability that an adversary recovers an ephemeral key before it is being refreshed should be small, and the leakages due to XORs are also somewhat bounded. Formally, define

$$\mathbf{Adv}^{\mathsf{2\text{-}up}[q]}(\mathcal{A}) := \Pr_{\widetilde{\mathsf{E}}, s_1}\Big[s_2 \leftarrow \widetilde{\mathsf{E}}_{s_1}^{T_w}(P_A), z \leftarrow \widetilde{\mathsf{E}}_{s_1}^{T_w}(P_B), \mathcal{G} \leftarrow \mathcal{A}^{\widetilde{\mathsf{E}}}(s_2, z, \mathsf{leak}) : s_1 \in \mathcal{G}\Big],$$

where $|\mathcal{G}| = q$, and $\mathcal{A}$'s input leak is a list of leakages depending on values T_w, P_A, P_B, s_0 specified by $\mathcal{A}$:

$$\mathsf{leak} = \Big[\mathsf{L}^{out}(s_0, T_w; s_1), \mathsf{L}^{in}(s_1, T_w; P_A), \mathsf{L}^{out}(s_1, T_w; s_2), \mathsf{L}^{in}(s_1, T_w; P_B), \mathsf{L}^{out}(s_1, T_w; z)\Big]^p. \tag{3}$$

We further define

$$\mathbf{Adv}^{\mathsf{2\text{-}up}[q]}(p, q_{\widetilde{\mathsf{E}}}, t) := \max\Big\{\mathbf{Adv}^{\mathsf{2\text{-}up}[q]}(\mathcal{A})\Big\}, \tag{4}$$

where the maximum is taken over all adversaries that repeat their measurements p times, makes $q_{\widetilde{\mathsf{E}}}$ $\widetilde{\mathsf{E}}$-queries, and runs in time t.

Regarding leakages resulting from XORing the random (looking) block stream with the message blocks in Romulus-T, we define

$$\mathbf{Adv}^{\mathsf{LORL2}}(\mathcal{A}) := \left|\Pr_{\widetilde{\mathsf{E}}, z}\Big[c^0 \leftarrow z \oplus m^0 : \mathcal{A}^{\widetilde{\mathsf{E}}}(c^0, \mathsf{leak}_0) \Rightarrow 1\Big] - \Pr_{\widetilde{\mathsf{E}}, z}\Big[c^1 \leftarrow z \oplus m^1 : \mathcal{A}^{\widetilde{\mathsf{E}}}(c^1, \mathsf{leak}_1) \Rightarrow 1\Big]\right|, \tag{5}$$

where leak_b again depends on values T_w, s specified by $\mathcal{A}$:

$$\mathsf{leak}_b = \Big([\mathsf{L}^{out}(s, T_w; z)]^p, \mathsf{L}_{\oplus}(z, m^b), [\mathsf{L}_{\oplus}(z, c^b)]^{p-1}\Big). \tag{6}$$

We also define

$$\mathbf{Adv}^{\mathsf{LORL2}}(p, q_{\widetilde{\mathsf{E}}}, t) := \max_{\mathcal{A}}\Big\{\mathbf{Adv}_T^{\mathsf{LORL2}}(\mathcal{A})\Big\}. \tag{7}$$

6.2 CCAmL2 Analysis of Romulus-T

We define the leakage function $\mathsf{L} = (\mathsf{L}_{\mathsf{enc}}, \mathsf{L}_{\mathsf{dec}})$ of Romulus-T as:

- $\mathsf{L}_{\mathsf{enc}}$, the leakages generated during the encryption:
  - $\mathsf{L}^{in}(K, T_w; X)$ & $\mathsf{L}^{out}(K, T_w; Y)$ generated by internal calls to $\widetilde{\mathsf{E}}(K, T_w; X) \to Y$ (excluding KDF-calls $\widetilde{\mathsf{E}}_K^{(0^n, 66, 0^{n-8})}$ and TGF-calls $\widetilde{\mathsf{E}}_K^{(R, 68, 0^{n-8})}$, which are again modeled as leak-free),
  - $\mathsf{L}_{\oplus}(a, b)$ generated by the internal actions $a \oplus b$,
  - all the intermediate values involved in the computations of the hash functions (i.e., hash functions are non-protected, and leak everything).
- $\mathsf{L}_{\mathsf{dec}}$, the above that are generated during the decryption.

Define

$$\mathbf{Adv}_{\mathsf{Romulus\text{-}T}, \mathsf{L}}^{\mathsf{CCAmL2}}(q_e, q_m, q_d, p-1, q_{\widetilde{\mathsf{E}}}, \sigma_a, \sigma_m, t) := \max\Big\{\mathbf{Adv}_{\mathsf{Romulus\text{-}T}, \mathsf{L}}^{\mathsf{CCAmL2}}(\mathcal{A})\Big\},$$

where the maximum is taken over all CCAmL2 adversaries making q_e challenge encryption queries, q_m non-challenge encryption queries, q_d decryption queries, $p-1$ challenge decryption leakage queries to $\mathsf{L}_{\mathsf{decch}}$, $q_{\widetilde{\mathsf{E}}}$ ideal TBC queries and has σ_a n-bit blocks in *all* queried associated data and σ_m n-bit blocks in all queried messages (including the incomplete last blocks). The following theorem supports the $n/2$-bit leakage confidentially claim in [11, Sect. 4.4.1] (see the Interpretation below).

Theorem 3. *Assume that $n \geq 3$ and the* Romulus-T *implementation has leakage functions* $\mathsf{L} = (\mathsf{L}_{\mathsf{enc}}, \mathsf{L}_{\mathsf{dec}})$ *defined above, where* L^{in}, L^{out}, $\mathsf{L}_{\oplus}$ *satisfy the assumptions specified by Eq. (3) and Eq. (5). Then, in the ideal TBC model, when* $Q := 3\sigma_m + \sigma_a + 6(q_e + q_d + q_m) + q_{\widetilde{\mathsf{E}}} \leq 2^n/8$ *it holds*

$$\mathbf{Adv}^{\mathsf{CCAmL2}}_{\mathsf{Romulus\text{-}T,L}}(q_e, q_m, q_d, p-1, q_{\widetilde{\mathsf{E}}}, \sigma_a, \sigma_m, t) \leq \frac{25Q + 4nq_d}{2^n} + \frac{16Q(q_d + q_e + q_m)}{2^{2n}} + \sigma_m \cdot \mathbf{Adv}^{\mathit{LORL2}}(p, q^*, t^*) + 2\sigma_m \cdot \mathbf{Adv}^{\mathit{2\text{-}up}[q^*]}(p, q^*, t^*). \quad (8)$$

where $\mathbf{Adv}^{\mathit{2\text{-}up}[q^*]}$ *and* $\mathbf{Adv}^{\mathit{LORL2}}$ *are defined in Eq. (4) and Eq. (7) respectively,* $q^* = 2q_{\widetilde{\mathsf{E}}} + 4\sigma + 6(q_e + q_d + q_m)$, $t^* = O(t + p\sigma t_l)$, *and* t_l *is the total time to evaluate* L^{in} *and* L^{out}.

Interpretation. The term $\sigma_m \cdot \mathbf{Adv}^{\mathsf{LORL2}}(p, q^*, t^*)$ corresponds to the reduction to the "minimal" message manipulation. On the other hand, the term $2\sigma_m \cdot \mathbf{Adv}^{\mathsf{2\text{-}up}[q^*]}(p, q^*, t^*)$ captures the hardness of side-channel key recovery, and it is roughly of some birthday type, namely

$$O\left(\sigma_m \cdot \frac{q_{\widetilde{\mathsf{E}}} + \sigma_m + t}{c \cdot 2^n}\right) = O\left(\frac{(q_{\widetilde{\mathsf{E}}} + \sigma_m + t)\sigma_m}{c \cdot 2^n}\right),$$

for some parameter c that depends on the concrete conditions. Yet, it is typically assumed that with such a small data complexity (only 3 relevant leakage traces), the value of c should be very small [16, 21].

References

1. Adomnicai, A., Peyrin, T.: Fixslicing AES-like Ciphers New bitsliced AES speed records on ARM-Cortex M and RISC-V. IACR Trans. Cryptogr. Hardw. Embed. Syst. 2021(1), 402–425 (2021)
2. Ashur, T., Dunkelman, O., Luykx, A.: Boosting authenticated encryption robustness with minimal modifications. In: Advances in Cryptology—Crypto 2017, Part III. pp. 3–33. LNCS, Springer (2017)
3. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 123–153. Springer (2016)
4. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.X.: TEDT: a leakage-resistant AEAD mode 2020(1), 256–320 (2019), https://tches.iacr.org/index.php/TCHES/article/view/8400
5. Berti, F., Koeune, F., Pereira, O., Peters, T., Standaert, F.X.: Ciphertext integrity with misuse and leakage: Definition and efficient constructions with symmetric primitives. pp. 37–50. ACM Press (2018)
6. Berti, F., Pereira, O., Peters, T., Standaert, F.X.: On leakage-resilient authenticated encryption with decryption leakages 2017(3), 271–293 (2017)
7. Caforio, A., Balli, F., Banik, S.: Energy analysis of lightweight aead circuits. Cryptology ePrint Archive, Paper 2020/607 (2020), https://eprint.iacr.org/2020/607, https://eprint.iacr.org/2020/607
8. Chen, S., Steinberger, J.P.: Tight security bounds for key-alternating ciphers. In: Advances in Cryptology—Eurocrypt 2014. LNCS, vol. 8441, pp. 327–350. Springer (2014)
9. Dai, Y., Steinberger, J.P.: Tight security bounds for multiple encryption. IACR Cryptol. ePrint Arch. p. 96 (2014), http://eprint.iacr.org/2014/096
10. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th Annual Symposium on Foundations of Computer Science (FOCS). pp. 293–302. IEEE (2008)
11. Guo, C., Iwata, T., Khairallah, M., Minematsu, K., Peyrin, T.: Designers/submitters (in alphabetical order)
12. Hirose, S.: Some plausible constructions of double-block-length hash functions. In: Fast Software Encryption (FSE). pp. 210–225. LNCS, Springer (2006)
13. Hirose, S., Park, J.H., Yun, A.: A simple variant of the Merkle-Damgård scheme with a permutation. In: Advances in Cryptology—Asiacrypt. pp. 113–129. LNCS, Springer (2007)
14. Khairallah, M., Bhasin, S.: Hardware Implementations of Romulus: Exploring Nonce Misuse Resistance and Boolean Masking. NIST Lightweight Cryptography Workshop 2022
15. Khairallah, M., Peyrin, T., Chattopadhyay, A.: Preliminary hardware benchmarking of a group of round 2 nist lightweight aead candidates. Cryptology ePrint Archive, Paper 2020/1459 (2020), https://eprint.iacr.org/2020/1459, https://eprint.iacr.org/2020/1459
16. Mangard, S., Oswald, E., Popp, T.: Power Analysis Attacks - Revealing the Secrets of Smart Cards. Springer (2007)
17. Mohajerani, K., Haeussler, R., Nagpal, R., Farahmand, F., Abdulgadir, A., Kaps, J.P., Gaj, K.: Fpga benchmarking of round 2 candidates in the nist lightweight cryptography standardization process: Methodology, metrics, tools, and results. Cryptology ePrint Archive, Paper 2020/1207 (2020), https://eprint.iacr.org/2020/1207, https://eprint.iacr.org/2020/1207
18. Naito, Y.: Optimally indifferentiable double-block-length hashing without post-processing and with support for longer key than single block. pp. 65–85. LNCS, Springer (2019)
19. Naito, Y., Sasaki, Y., Sugawara, T.: Secret can be public: Low-memory aead mode for high-order masking. Cryptology ePrint Archive, Paper 2022/812 (2022), https://eprint.iacr.org/2022/812, https://eprint.iacr.org/2022/812
20. Peyrin, T., Seurin, Y.: Counter-in-Tweak: Authenticated Encryption Modes for Tweakable Block Ciphers. Version 20160524:153228. Cryptology ePrint Archive, Report 2015/1049 (2015)

21. Pietrzak, K.: A leakage-resilient mode of operation. In: Advances in Cryptology—Eurocrypt. pp. 462–482. LNCS, Springer (2009)
22. Steinbauer, T., Nagpal, R., Primas, R., Mangard, S.: Tvla on selected nist lwc finalists https://cryptography.gmu.edu/athena/LWC/Reports/TUGraz/TUGraz_Report_HW_5_candidates_RUB.pdf
23. Verhamme, C., Cassiers, G., Standaert, F.: Analyzing the Leakage Resistance of the NIST's Lightweight Crypto Standardization Process Finalists. NIST Lightweight Cryptography Workshop 2022 (2022), https://csrc.nist.gov/csrc/media/Events/2022/lightweight-cryptography-workshop-2022/documents/papers/analyzing-the-leakageresistance-of-the-nist-lwc-standardization-process-finalists.pdf
24. Yu, Y., Standaert, F.X., Pereira, O., Yung, M.: Practical leakage-resilient pseudorandom generators. In: ACM Conf. on Computer and Communications Security (CCS). pp. 141–151. ACM Press (2010)

A Proof Preparations

A.1 Injectivity of Padding

We first show the injectivity of the RTpad scheme.

Lemma 2. *For* $\mathsf{RTpad}(A,N,C) = \mathit{ipad}_{2n}\big(\mathit{ipad}_n^*(A)\|\mathit{ipad}_n^*(C)\|N\|\pi(|C|_8)\big)$*, it holds* $\mathsf{RTpad}(A,N,C) \neq \mathsf{RTpad}(A',N',C')$ *for any two distinct triples* (A,N,C) *and* (A',N',C').

Proof. For simplicity, let $V := \mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $V' := \mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$.

For clearness, we first show the injectivity of ipad_{2n}, i.e., $\mathsf{ipad}_{2n}(V) \neq \mathsf{ipad}_{2n}(V')$ as long as $V \neq V'$. We distinguish two cases:

- Case 1: $|V|_8 \neq |V'|_8$. It further consists of two subcases:
  - Subcase 1.1: $|V|_8 \neq |V'|_8 \mod 2n$. Then $\mathsf{ipad}_{2n}\big(V\big) \neq \mathsf{ipad}_{2n}\big(V'\big)$ since the padded fields are distinct;
  - Subcase 1.2: $|V|_8 = |V'|_8 \mod 2n$. Then it has to be $|V|_8 = |V'|_8 + 2\ell n$ for some integer ℓ, and $\mathsf{ipad}_{2n}\big(V\big)$ and $\mathsf{ipad}_{2n}\big(V'\big)$ have different number of blocks.
- Case 2: $|V|_8 = |V'|_8$, though $V \neq V'$. Then $\mathsf{ipad}_{2n}\big(V\big)$ and $\mathsf{ipad}_{2n}\big(V'\big)$ have different prefixes.

It remains to prove $\mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8) \neq \mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$:

- Case 1: $N \neq N'$. Then $\mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$ have different suffixes $N\|\pi(|C|_8)$ and $N'\|\pi(|C'|_8)$;
- Case 2: $N = N'$, but $C \neq C'$. It further consists of two subcases:
  - Subcase 2.1: $|C| \neq |C'|$. Then again, $\mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$ have different suffixes $N\|\pi(|C|_8)$ and $N'\|\pi(|C'|_8)$;
  - Subcase 2.2: $|C| = |C'|$. Then it necessarily holds $\mathsf{ipad}_n^*(C) \neq \mathsf{ipad}_n^*(C')$, and thus $\mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$ have different suffixes $\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$.
- Case 3: $N = N'$ and $C = C'$. Then it must be $A \neq A'$. In this case, the suffixes $\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$ are the same, but $\mathsf{ipad}_n^*(A)\|\mathsf{ipad}_n^*(C)\|N\|\pi(|C|_8)$ and $\mathsf{ipad}_n^*(A')\|\mathsf{ipad}_n^*(C')\|N'\|\pi(|C'|_8)$ have different prefixes $\mathsf{ipad}_n^*(A)$ and $\mathsf{ipad}_n^*(A')$.

The above complete the analysis. □

A.2 Number of Underlying TBC Calls

Consider a security game $\mathcal{A}^{\mathsf{Game}(\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}],\widetilde{\mathsf{E}})}$ for any notion. Assume that (also see Theorems 1, 2 and 3):

- $\mathcal{A}$ makes q queries to the (keyed) encryption and decryption oracles instantiated with $\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}]$, and
- The processed ADs A consist of σ_a blocks of n bits (including the incomplete last blocks), and
- The processed messages M consist of σ_m blocks of n bits (including the incomplete last blocks).

For each triple (A,C,N), when $|\mathsf{ipad}_n^*(A)|_n \leq |A|_n+1$, $|\mathsf{ipad}_n^*(C)|_n \leq |C|_n+1$, and the equalities hold when A or C does not has incomplete last blocks. In addition, the hash $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]\big(\mathsf{RTpad}(A,N,C)\big)$ makes at most $|A|_n+|C|_n+5$ queries to $\widetilde{\mathsf{E}}$:

- When $|\mathsf{ipad}_n^*(A)|_n + |\mathsf{ipad}_n^*(C)|_n + |N|_n + 1 = |A|_n + |C|_n + 4$ is even, the padded input $\mathsf{RTpad}(A,N,C)$ has $\frac{|A|_n+|C|_n+4}{2}$ blocks of $2n$ bits;
- When $|\mathsf{ipad}_n^*(A)|_n + |\mathsf{ipad}_n^*(C)|_n + |N|_n + 1 = |A|_n + |C|_n + 4$ is odd, the padded input $\mathsf{RTpad}(A,N,C)$ has $\frac{|A|_n+|C|_n+5}{2}$ blocks of $2n$ bits.

Therefore, it holds:

- The hash and tag generation makes at most $\sum |\mathsf{ipad}_n^*(A)|_n + \sum |\mathsf{ipad}_n^*(C)|_n + (5+1)q = \sigma_a + \sigma_m + 6q$ TBC calls, and
- The encryption pass makes $2m$ TBC calls (including the initial calls using tweaks $(0^n, 66, 0^{n-8})$) to process the σ_m message blocks.

By these, define the "number of blocks" σ as

$$\sigma := 3\sigma_m + \sigma_a + 6q.$$

Then, in the security game $\mathcal{A}^{\mathsf{Game}(\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}]_K,\widetilde{\mathsf{E}})}$, the ideal TBC $\widetilde{\mathsf{E}}$ receives at most

$$Q := \sigma + q_{\widetilde{\mathsf{E}}} = 3\sigma_m + \sigma_a + 6q + q_{\widetilde{\mathsf{E}}}. \tag{9}$$

queries in total. Subsequent analyses will replace q with other notations that depend on the context.

A.3 Properties of Hirose Compression Function

Recall that $\mathsf{Hir}[\widetilde{\mathsf{E}}]$ is the Hirose compression function based on the ideal TBC $\widetilde{\mathsf{E}}$. Note that any adversary $\mathcal{A}$ against $\mathsf{Hir}[\widetilde{\mathsf{E}}]$ can be normalized to an adversary $\mathcal{A}'$ that only makes pairs of Hirose "matching" queries: $\mathcal{A}'$ runs $\mathcal{A}$, and

- each time $\mathcal{A}$ makes a forward query $\widetilde{\mathsf{E}}_K^{T_w}(X)$, $\mathcal{A}'$ makes a query $\widetilde{\mathsf{E}}_K^{T_w}(X \oplus \theta) \to Y'$ right after relaying $\widetilde{\mathsf{E}}_K^{T_w}(X) \to Y$, and
- each time $\mathcal{A}$ makes a backward query $(\widetilde{\mathsf{E}}_K^{T_w})^{-1}(Y)$, $\mathcal{A}'$ makes a query $\widetilde{\mathsf{E}}_K^{T_w}(X \oplus \theta) \to Y'$ right after relaying $(\widetilde{\mathsf{E}}_K^{T_w})(Y) \to X$.

Therefore, we could concentrate on adversaries that only make such pairs of "matching" queries. In this vein, the function $\mathsf{Hir}[\widetilde{\mathsf{E}}]$ is collision resistant [12, Theorem 4].

Lemma 3. *For any $\mathcal{A}$ making Q (unordered) pairs of matching queries to $\widetilde{\mathsf{E}}$ with $1 \le Q \le 2^n/4$, it holds*

$$\Pr\big[((L,R,M),(L',R',M')) \leftarrow \mathcal{A}^{\widetilde{\mathsf{E}}} : \mathsf{Hir}[\widetilde{\mathsf{E}}](L,R,M) = \mathsf{Hir}[\widetilde{\mathsf{E}}](L',R',M')\big] \le 3\Big(\frac{Q}{2^{n-1}}\Big)^2 \le \frac{6Q}{2^n}.$$

We also need multi-collision resistance bound of Davies-Meyer. A similar result has been proven by Berti et al. [4, Lemma 2]. However, Berti et al.'s result was only proven for chopped Davies-Meyer.

Lemma 4. *For any adversary $\mathcal{A}$ making at most $2Q \le 2^n/2$ queries to $\widetilde{\mathsf{E}}$ and any integer λ, it holds*

$$\Pr\big[((K_1,T_{w1},X_1),\ldots,(K_\lambda,T_{w\lambda},,X_\lambda)) \leftarrow \mathcal{A}^{\widetilde{\mathsf{E}}} : \widetilde{\mathsf{E}}_{K_1}^{T_{w1}}(X_1) \oplus X_1 = \ldots = \widetilde{\mathsf{E}}_{K_\lambda}^{T_{w\lambda}}(X_\lambda) \oplus X_\lambda\big] \le \frac{(4Q)^\lambda}{\lambda! 2^{(\lambda-1)n}}.$$

In particular, when $\lambda = n+1$, $n \ge 2$ and $Q \le 2^n/8$, it holds

$$\Pr\big[((K_1,T_{w1},X_1),\ldots,(K_{n+1},T_{wn+1},,X_{n+1})) \leftarrow \mathcal{A}^{\widetilde{\mathsf{E}}} : \widetilde{\mathsf{E}}_{K_1}^{T_{w1}}(X_1) \oplus X_1 = \ldots = \widetilde{\mathsf{E}}_{K_{n+1}}^{T_{wn+1}}(X_{n+1}) \oplus X_{n+1}\big] \le \frac{Q}{2^n}.$$

Proof. The statement on chopped Davies-Meyer was given in [4, Lemma 2]. For completeness, we present a complete argument. Consider any λ $\widetilde{\mathsf{E}}$ queries $\widetilde{\mathsf{E}}_{K_1}^{T_{w1}}(X_1) = Y_1, \ldots, \widetilde{\mathsf{E}}_{K_\lambda}^{T_{w\lambda}}(X_\lambda) = Y_\lambda$ listed according the order they were made, and let $Z_\lambda = X_\lambda \oplus Y_\lambda$ for each i. Then, since $2Q \le 2^n/2$, we have

1 The event $Z_2 = Z_1$ is equivalent with $X_2 \oplus Y_2 = X_1 \oplus Y_1$. Wlog assume that the query $\widetilde{\mathsf{E}}_{K_2}^{T_{w2}}(X_2) = Y_2$ occurs after $\widetilde{\mathsf{E}}_{K_1}^{T_{w1}}(X_1) = Y_1$. If $\widetilde{\mathsf{E}}_{K_2}^{T_{w2}}(X_2) = Y_2$ was due to a forward query $\widetilde{\mathsf{E}}_{K_2}^{T_{w2}}(X_2) \to Y_2$, then Y_2 is uniformly distributed in at least $2^n - 2Q$ possibilities, and the probability to have $X_2 \oplus Y_2 = X_1 \oplus Y_1$ is at most $1/(2^n - 2Q) \le 2/2^n$; if $\widetilde{\mathsf{E}}_{K_2}^{T_{w2}}(X_2) = Y_2$ was due to a backward query $(\widetilde{\mathsf{E}}_{K_2}^{T_{w2}})^{-1}(Y_2) \to X_2$, then X_2 is uniformly distributed in at least $2^n - 2Q$ possibilities, and the probability to have $X_2 \oplus Y_2 = X_1 \oplus Y_1$ remains at most $1/(2^n - 2Q) \le 2/2^n$. Therefore, it always holds $\Pr[Z_2 = Z_1] \le \frac{1}{2^n - 2Q} \le \frac{2}{2^n}$;

2 Similarly to $\Pr[Z_2 = Z_1]$, $\Pr[Z_3 = Z_1] \le \frac{1}{2^n-2Q} \le \frac{2}{2^n}$, ..., $\Pr[Z_\lambda = Z_1] \le \frac{1}{2^n-2Q} \le \frac{2}{2^n}$.

Thus in total we have

$$\Pr[\lambda \text{ collisions}] \le \binom{2Q}{\lambda} \cdot \Big(\frac{2}{2^n}\Big)^{\lambda-1} \le \frac{(4Q)^\lambda}{\lambda! 2^{(\lambda-1)n}}.$$

When $\lambda = n+1$ and $n \ge 2$ and $Q \le 2^n/8$ (thus $8Q/2^n \le 1$), we further have

$$\begin{aligned}
&\Pr\big[((K_1,T_{w1},X_1),\ldots,(K_{n+1},T_{wn+1},,X_{n+1})) \leftarrow \mathcal{A}^{\widetilde{\mathsf{E}}} : \widetilde{\mathsf{E}}_{K_1}^{T_{w1}}(X_1) \oplus X_1 = \ldots = \widetilde{\mathsf{E}}_{K_{n+1}}^{T_{wn+1}}(X_{n+1}) \oplus X_{n+1}\big] \\
&\le \frac{(4Q)^\lambda}{\lambda! 2^{(\lambda-1)n}} \le \frac{1}{2 \times (n+1)!} \times \Big(\frac{8Q}{2^n}\Big)^{n+1} \le \frac{1}{2 \times 3!} \times \frac{8Q}{2^n} \\
&\le \frac{Q}{2^n}
\end{aligned}$$

as claimed.

A.4 Idealizing Romulus-T

Our proofs will frequently employ idealized Romulus-T scheme, in which the TBC calls using the master key are replaced by a (secret) tweakable random permutation. In detail, in the real scheme Romulus-T$[\widetilde{\mathsf{E}}]$, we introduce a random tweakable permutation $\widetilde{\mathsf{P}}$ that is independent from $\widetilde{\mathsf{E}}$, and replace all calls to $\widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}(X)/\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}(X)$ and $(\widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})})^{-1}(Y)/(\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})})^{-1}(T)$ by calls to a random tweakable permutation $\widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(X)/\widetilde{\mathsf{P}}^{(R,68,0^{n-8})}(X)$ and $(\widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})})^{-1}(Y)/(\widetilde{\mathsf{P}}^{(R,68,0^{n-8})})^{-1}(T)$. Denote the obtained idealized scheme by Romulus-T$[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]$. It is easy to see that, there exists a distinguisher $\mathcal{D}$ that has access to either $(\widetilde{\mathsf{E}}_K^*,\widetilde{\mathsf{E}})$ or $(\widetilde{\mathsf{P}},\widetilde{\mathsf{E}})$ and makes at most $2q$ queries to $\widetilde{\mathsf{E}}_K^*/\widetilde{\mathsf{P}}$ and Q (see Eq. (9)) queries to $\widetilde{\mathsf{E}}$, such that

$$\begin{aligned}
&\left|\Pr\left[\mathcal{A}^{\mathsf{Game}\left(\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}],\widetilde{\mathsf{E}}\right)}\Rightarrow 1\right]-\Pr\left[\mathcal{A}^{\mathsf{Game}\left(\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}],\widetilde{\mathsf{E}}\right)}\Rightarrow 1\right]\right|\\
\le&\left|\Pr\left[\mathcal{D}^{\widetilde{\mathsf{E}}_K^*,\widetilde{\mathsf{E}}}\Rightarrow 1\right]-\Pr\left[\mathcal{D}^{\widetilde{\mathsf{P}},\widetilde{\mathsf{E}}}\Rightarrow 1\right]\right|\\
\le&\frac{Q}{2^n}. \qquad (10)
\end{aligned}$$

The latter bound $Q/2^n$ follows from [9, Theorem 6].

B Proof of Theorem 1

Since we assume leak-freeness of the TBC calls $\widetilde{\mathsf{E}}_K^{(0^n,66,0^{n-8})}(X)/\widetilde{\mathsf{E}}_K^{(R,68,0^{n-8})}(X)$ using the master AEAD key K, we are able to idealize the schemes using the results in Appendix A.4:

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}]}(\mathcal{A})-\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}(\mathcal{A})\le\frac{Q}{2^n} \qquad (11)$$

when the adversarial power of $\mathcal{A}$ is as assumed in Theorem 1 and $Q:=3\sigma_m+\sigma_a+6(q_e+q_d)+q_{\widetilde{\mathsf{E}}}$.

We can thus focus on establishing unforgeability for the idealized schemes Romulus-T$[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]$. Denote by G_1 the game that captures the interaction between the CIML2 adversary $\mathcal{A}$ and Romulus-T$[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]$. Following Hirose [12], we normalize the game: for each $\widetilde{\mathsf{E}}$ query either made by $\mathcal{A}$, we assume the system makes its Hirose matching query immediately (see Sect. A.3). Denote G_2 the obtained normalized game. By Eq. (9), $\widetilde{\mathsf{E}}$ receives at most Q queries in G_1. Therefore, in G_2, the number of $\widetilde{\mathsf{E}}$ queries doesn't exceed $2Q$, and the number of (unordered) matching $\widetilde{\mathsf{E}}$ query pairs is at most Q. Clearly,

$$\Pr\left[\mathcal{A}\text{ forges in }\mathsf{G}_1\right]\le\Pr\left[\mathcal{A}\text{ forges in }\mathsf{G}_2\right],$$

and we divide the unforgeability argument for G_2 into two substeps in two paragraphs below: first, we define and bound several simple bad events that may occur during an execution of the game G_2; then, we show $\mathcal{A}$ is unable to forge in G_2 as long as none of these conditions is fulfilled.

B.1 Bad Events for Unforgeability

We keep a list

$$\mathcal{Q}_{\widetilde{\mathsf{E}}}=\left((K_1,T_{w1},X_1,Y_1),\ldots,(K_{q_{\widetilde{\mathsf{E}}}},T_{wq_{\widetilde{\mathsf{E}}}},X_{q_{\widetilde{\mathsf{E}}}},Y_{q_{\widetilde{\mathsf{E}}}})\right)$$

for the transcript of queries and responses to the ideal TBC $\widetilde{\mathsf{E}}$ that appeared during the execution of the game G_2, where the j-th tuple (K_j,T_{wj},X_j,Y_j) indicates either a forward query $\widetilde{\mathsf{E}}_{K_j}^{T_{wj}}(X_j)\to Y_j$ or a backward $(\widetilde{\mathsf{E}}_{K_j}^{T_{wj}})^{-1}(Y_j)\to X_j$. Based on $\mathcal{Q}_{\widetilde{\mathsf{E}}}$ and the definition of Romulus-H, we define

$$\mathcal{Q}_{\mathsf{H}}^*:=\left((U_1,L_1\|R_1),(U_2,L_2\|R_2),\ldots\right)$$

as the pairs of inputs and outputs of the hash Romulus-T$[\widetilde{\mathsf{E}}]$ that can be determined by the information in $\mathcal{Q}_{\widetilde{\mathsf{E}}}$, where $U_j\in\{0,1\}^*$ is the (variable size) hash input and $L_j,R_j\in\{0,1\}^n$ keep the output. Actually, $\mathcal{Q}_{\mathsf{H}}^*$ keeps the hash evaluations that might appear during the execution of the game G_2.

With these, we identify the following events during an execution of G_2:

- (B-1) *Hash collision*: there exist distinct hash records $(U,L\|R)\neq(U^*,L^*\|R^*)\in\mathcal{Q}_{\mathsf{H}}^*$ such that $L\|R=L^*\|R^*$;
- (B-2) *Multi hash semi-collision*: $\mu_R\ge n+1$, where

$$\mu_R:=\max_{r\in\{0,1\}^n}\left|\left\{(U,L\|R)\in\mathcal{Q}_{\mathsf{H}}^*:R=r\right\}\right|.$$

To bound the probabilities, we need to rely on a useful lemma on collision security of Romulus-H, which is stated and proven in the following paragraph.

Multi-semicollision Resistance of $\mathsf{Romulus\text{-}H}$**.** We will need the (multi-semi) collision resistance of $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$, which is formally stated as follows.

Lemma 5. *Consider any oracle machine $\mathcal{A}^{\widetilde{\mathsf{E}}}$ making Q (unordered) pairs of matching queries to the ideal TBC $\widetilde{\mathsf{E}}$, such that $Q \le 2^n/8$. Then, the probability to observe either of the following two events is at most $\frac{11Q}{2^n}$:*

1 *Collision on* $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$*: there exist two distinct pairs $(U, L\|R)$ and $(U^*, L^*\|R^*)$ in $\mathcal{Q}^*_{\mathsf{H}}$ such that $L\|R = L^*\|R^*$;*
2 *Multi-semicollision on* $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$*: $\mu_R \ge n+1$, where*

$$\mu_R := \max_{r\in\{0,1\}^n}\Big|\big\{(U, L\|R) \in \mathcal{Q}^*_{\mathsf{H}} : R = r\big\}\Big|.$$

Proof. We denote by G_1 the game that captures the interaction between the adversary $\mathcal{A}$ and $\widetilde{\mathsf{E}}$.

<u>*Bad events.*</u>. We define three bad events during the execution of G_1:

– (B-1): *Collision on compression function.* Formally, there exist two distinct pairs of $\widetilde{\mathsf{E}}$ query records $\big((R\|M, L, Y_1), (R\|M, L\oplus\theta, Y_2)\big)$ and $\big((R'\|M', L', Y_1'), (R'\|M', L'\oplus\theta, Y_2')\big)$ such that:
  - $L \oplus Y_1 = L' \oplus Y_1'$, and $L\oplus\theta\oplus Y_2 = L'\oplus\theta\oplus Y_2'$; or
  - $L \oplus Y_1 = L' \oplus\theta\oplus Y_2'$, and $L\oplus\theta\oplus Y_2 = L'\oplus Y_1'$.

– (B-2): *Hitting initial-vector.* there exist a pair of $\widetilde{\mathsf{E}}$ query records $\big((R\|M, L, Y_1), (R\|M, L\oplus\theta, Y_2)\big)$ such that $L\oplus Y_1 = L\oplus\theta\oplus Y_2 = 0^n$ (i.e., $\mathsf{Hir}[\widetilde{\mathsf{E}}](L\|R, M) = 0^{2n}$);

– (B-3): *n-collision.* there exist $n+1$ records $(R_1\|M_1, L_1, Y_1), \ldots, (R_{n+1}\|M_{n+1}, L_{n+1}, Y_{n+1}) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$ such that $L_1\oplus Y_1 = \ldots = L_{n+1}\oplus Y_{n+1}$.

Lemmas 3 and 4 immediately imply

$$\Pr\big[\text{(B-1)}\big] \le \frac{6Q}{2^n}, \qquad \Pr\big[\text{(B-3)}\big] \le \frac{Q}{2^n}.$$

For (B-2), consider any pair $\big((R'\|M', L', Y_1'), (R'\|M', L'\oplus\theta, Y_2')\big)$. Clearly, regardless of the directions of these two queries, it holds $\Pr[Y_1 = L \wedge Y_2 = L\oplus\theta] \le \frac{1}{(2^n-2Q)^2} \le \frac{4}{2^{2n}}$ when $2Q \le 2^n/2$, and thus

$$\Pr\big[\text{(B-2)}\big] \le \frac{4Q}{2^{2n}}.$$

A union bound yields

$$\Pr\big[\text{(B-1)} \vee \text{(B-2)} \vee \text{(B-3)}\big] \le \frac{11Q}{2^n}.$$

<u>*Claims on* $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$</u>. Then, conditioned on $\neg$(B-1) $\wedge$ $\neg$(B-2) $\wedge$ $\neg$(B-3), we argue that the two events on $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$ cannot occur, so that $\Pr\big[\text{(B-1)} \vee \text{(B-2)} \vee \text{(B-3)}\big]$ provides the bound.

We first consider event (1) (collision). For any two hash records $(U, L\|R)$ and $(U^*, L^*\|R^*)$, assume that *tail* is the maximum common suffix of U and U^*, i.e., $U = header\|v\|tail$, $U^* = header^*\|v^*\|tail$, $|v| = |v^*| = 2n$, $v \ne v^*$, and $|header|, |header^*|$, and $|tail|$ are multiples of $2n$. Then we distinguish two cases:

Case 1: either $header\|v$ **or** $header^*\|v^*$ **is empty.** Without loss of generality, we assume $header\|v$ is empty. Since U isn't empty, this means *tail* isn't empty. Then in $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U^*)$, the hash-chain value after absorbing v^* is different from the initial vector 0^{2n} by $\neg$(B-2). So the two "first-block" calls in absorbing *tail* in $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U)$ and $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U^*)$ are different. By $\neg$(B-1), this means the resulted hash-chain values are different. Then by iteratively applying $\neg$(B-1), it can be seen the "last-block calls" in $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U)$ and $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U^*)$ are different, and further $L\|R \ne L^*\|R^*$.

Case 2: neither $header\|v$ **or** $header^*\|v^*$ **is empty.** Then by $\neg$(B-1), the two hash-chain values after absorbing $v \ne v^*$ are different. If *tail* is empty, then as $v \ne v^*$ the "last-block calls" in $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U)$ and $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](U^*)$ are clearly different and further $L\|R \ne L^*\|R^*$; otherwise, $L\|R \ne L^*\|R^*$ follows by iteratively applying $\neg$(B-1).

We then consider event (2) (multi-semicollision). The above show that distinct hash inputs U and U^* necessarily result in distinct "last-block-calls". By this, any $n+1$ hash inputs $(U_1, \ldots, U_{n+1})$ necessarily result in n distinct "last-block-calls" denoted $\mathsf{Hir}[\widetilde{\mathsf{E}}](L_1, R_1, M_1), \ldots, \mathsf{Hir}[\widetilde{\mathsf{E}}](L_{n+1}, R_{n+1}, M_{n+1})$. By the definition of

$\mathsf{Hir}[\widetilde{\mathsf{E}}]$, it can be seen such a multi-semicollision correspond to an $(n+1)$-collision on the Davies-Meyer function. Concretely, assume that for

$$L'_1\|R'_1 = \mathsf{Hir}[\widetilde{\mathsf{E}}](L_1, R_1, M_1), \ldots, L'_{n+1}\|R'_{n+1} = \mathsf{Hir}[\widetilde{\mathsf{E}}](L_{n+1}, R_{n+1}, M_{n+1}),$$

it holds $R'_1 = \ldots = R'_{n+1}$, then it essentially holds

$$\widetilde{\mathsf{E}}_{R_1}^{M_1}(L_1 \oplus \theta) \oplus (L_1 \oplus \theta) = \ldots = \widetilde{\mathsf{E}}_{R_{n+1}}^{M_{n+1}}(L_{n+1} \oplus \theta) \oplus (L_{n+1} \oplus \theta),$$

contradicting $\neg$(B-3). These complete the proof. □

Probability of bad events. For simplicity let $\mathsf{Bad} = \text{(B-1)} \vee \text{(B-2)}$, then Lemma 5 implies

$$\Pr\big[\mathsf{Bad}\big] \le \frac{11Q}{2^n}. \tag{12}$$

B.2 Unforgeable unless Bad

Conditioned on $\neg\mathsf{Bad}$, we argue all non-trivial decryption queries (N, A, C, T) (see Definition 3) result in $\perp$ except with a bounded probability.

If decrypting (N, A, C, T) does not yield $\perp$, then right after this decryption finished, there exists a hash record $(\mathsf{RTpad}(A, N, C), L\|R) \in \mathcal{Q}_{\mathsf{H}}^*$ and a $\widetilde{\mathsf{P}}$ query $\widetilde{\mathsf{P}}^{R^*}(L^*) = T$ in the history, such that $R^* = R$ and $L^* = L$. This means at some time during the execution, the following query records exist in the history:

$$(T_w, K, X, Y_1) \in \mathcal{Q}_{\widetilde{\mathsf{E}}}^*,\ (T_w, K, X \oplus \theta, Y_2) \in \mathcal{Q}_{\widetilde{\mathsf{E}}}^*,\ \widetilde{\mathsf{P}}^{(R^*, 68, 0^{n-8})}(L^*) = T,$$

where $T_w\|K$ is the concatenation of the last block of the $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$ input $\mathsf{RTpad}(A, N, C)$ and an n-bit previous chain value, and $X \oplus Y_1 = L = L^*$ and $X \oplus \theta \oplus Y_2 = R = R^*$. We distinguish two cases as follows.

Case 1: The internal query for $\widetilde{\mathsf{P}}^{(R^*, 68, 0^{n-8})}(L^*) = T$ happens After the pair of $\widetilde{\mathsf{E}}$ queries. As $R^* = R$, we simplify the notation as $\widetilde{\mathsf{P}}^{(R, 68, 0^{n-8})}(L^*) = T$. We argue it cannot be a forward query $\widetilde{\mathsf{P}}^{(R, 68, 0^{n-8})}(L^*) \to T$. For this, assume otherwise, then it's due to an earlier encryption query $\mathcal{LE}_{\mathbf{K}}(N', A', M') \to (C', T)$, and that $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](\mathsf{RTpad}(A', N', C')) = L\|R$ (i.e., it collides with the hash evaluation $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](\mathsf{RTpad}(A, N, C)) = L\|R$ in question). Now,

- if $(N, A, C) = (N', A', C')$, then since we forbid trivial decryption queries, the tag produced by $\mathcal{LE}_{\mathbf{K}}(N', A', M')$ cannot be T, and hence cannot trigger the query $\widetilde{\mathsf{P}}^{(R, 68, 0^{n-8})}(L^*) \to T$;
- if $(N, A, C) \neq (N', A', C')$, then by Lemma 2 we have $\mathsf{RTpad}(A, N, C) \neq \mathsf{RTpad}(A', N', C')$, which further implies a hash collision and contradicts $\neg$(B-1).

In all, it has to be backward $(\widetilde{\mathsf{P}}^{(R, 68, 0^{n-8})})^{-1}(T) \to L^*$. During the execution of G_2, the number of such backward queries is at most q_d. Conditioned on $\neg$(B-2), the number of encountered hash records $(U^\dagger, L^\dagger\|R^\dagger) \in \mathcal{Q}_{\mathsf{H}}^*$ with $R^\dagger = R$ is at most n. This implies that the number of "target" $L^\dagger$ values is also at most n. For each such "target" $L^\dagger$ and each backward query $(\widetilde{\mathsf{P}}^{(R, 68, 0^{n-8})})^{-1}(T) \to L^*$, we have

$$\Pr[L^* = L^\dagger] \le \frac{1}{2^n - q_e - q_d} \le \frac{2}{2^n}.$$

Therefore,

$$\Pr\big[\text{Case 1} \mid \neg\mathsf{Bad}\big] \le \frac{2nq_d}{2^n}. \tag{13}$$

Case 2: The internal query for $\widetilde{\mathsf{P}}^{(R^*, 68, 0^{n-8})}(L^*) = T$ happens Before the pair of $\widetilde{\mathsf{E}}$ queries. We consider the query (T_w, K, X, Y_1) first. Regardless of its direction, either X or Y_1 is uniform in at least $2^n - 2Q$ possibilities. Thus, when $Q \le 2^n/4$, it holds

$$\Pr[X \oplus Y_1 = L^*] \le \frac{1}{2^n - 2Q} \le \frac{2}{2^n}.$$

Similarly,

$$\Pr[X \oplus \theta \oplus Y_2 = R^*] \le \frac{1}{2^n - 2Q} \le \frac{2}{2^n}.$$

Therefore, for each such triple of queries, the probability of collision is at most $\frac{4}{2^{2n}}$. We have at most $q_d + q_e$ choices for the $\widetilde{\mathsf{P}}$ record $\widetilde{\mathsf{P}}^{(R^*,68,0^{n-8})}(L^*) = T$, and $2Q$ choices for the (ordered) pair of $\widetilde{\mathsf{E}}$ queries. Therefore,

$$\Pr\big[\text{Case } 2\big] \leq \frac{8Q(q_d + q_e)}{2^{2n}}. \tag{14}$$

Note that these arguments are significantly simplified by the normalization of the game: without the normalization, the query for $\widetilde{\mathsf{P}}^{(R^*,68,0^{n-8})}(L^*) = T$ may happen between the two "matching" $\widetilde{\mathsf{E}}$ queries, giving rise to many additional cases.

B.3 Summarizing

Gathering Eqs. (12), (13) and (14) yields

$$\mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}],\mathsf{L}^*}(\mathcal{D}) \leq \frac{11Q + 2nq_d}{2^n} + \frac{8Q(q_d + q_e)}{2^{2n}}. \tag{15}$$

This plus the gap term Eq. (10) yield Theorem 1 (more precisely, Eq. (1)).

C Proof of Theorem 2

C.1 CCA to CPA

Note that in the misuse resilience setting, schemes which achieve both CPA confidentiality and authenticity also achieve CCA confidentiality [2]:

$$\begin{aligned}
\mathbf{Adv}^{\mathsf{CCAm\$}^*}_{\mathsf{AEAD}}(\mathcal{A}) &= \Big|\Pr\big[\mathcal{A}^{\mathcal{E}_K,\mathcal{E}_K,\mathcal{D}_K,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big] - \Pr\big[\mathcal{A}^{\mathcal{E}_K,\$,\perp,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big]\Big| \\
&\leq \underbrace{\Big|\Pr\big[\mathcal{A}^{\mathcal{E}_K,\mathcal{E}_K,\mathcal{D}_K,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big] - \Pr\big[\mathcal{A}^{\mathcal{E}_K,\mathcal{E}_K,\perp,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big]\Big|}_{\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{AEAD}}(\mathcal{A}):\ \mathsf{INT\text{-}CTXT}\text{ advantage of }\mathcal{A}\text{ on }\mathsf{AEAD}} \\
&\quad + \underbrace{\Big|\Pr\big[\mathcal{A}^{\mathcal{E}_K,\mathcal{E}_K,\perp,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big] - \Pr\big[\mathcal{A}^{\mathcal{E}_K,\$,\perp,\widetilde{\mathsf{E}},\widetilde{\mathsf{E}}^{-1}} \Rightarrow 1\big]\Big|}_{\text{defined as }\mathbf{Adv}^{\mathsf{CPAm\$}}_{\mathsf{AEAD}}(\mathcal{A})}.
\end{aligned} \tag{16}$$

Clearly, $\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\mathsf{Romulus\text{-}T}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}}(\mathcal{A})$. Therefore, we focus on the CPA advantage $\mathbf{Adv}^{\mathsf{CPAm\$}}_{\mathsf{Romulus\text{-}T}}(\mathcal{A})$—switching to the CPA setting greatly simply the setting as well as the notations.

C.2 CPAm$ Security of Romulus-T

In a similar vein to Appendix B, we could focus on CPA security of the idealized schemes $\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]$, with the gap due to Eq. (10) in mind:

$$\mathbf{Adv}^{\mathsf{CPAm\$}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}]}(\mathcal{A}) - \mathbf{Adv}^{\mathsf{CPAm\$}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}(\mathcal{A}) \leq \frac{Q}{2^n} \tag{17}$$

when the adversarial power of $\mathcal{A}$ is as assumed in Theorem 2, and $Q := 3\sigma_m + \sigma_a + 6(q_e + q_d + q_m) + q_{\widetilde{\mathsf{E}}}$.

To bound $\mathbf{Adv}^{\mathsf{CPAm\$}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}(\mathcal{A})$, we will employ Patarin's H-coefficient technique.

Transcripts. In the CPAm$ setting, we summarize the adversarial queries to the ideal TBC $\widetilde{\mathsf{E}}$ in the set $\mathcal{Q}_{\widetilde{\mathsf{E}}} = \big((K_1, T_{w1}, X_1, Y_1), \ldots, (K_{q_{\widetilde{\mathsf{E}}}}, T_{wq_{\widetilde{\mathsf{E}}}}, X_{q_{\widetilde{\mathsf{E}}}}, Y_{q_{\widetilde{\mathsf{E}}}})\big)$. During the interaction, we reveal all the $\widetilde{\mathsf{E}}$ queries internally made by $\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]$ and the $\widetilde{\mathsf{E}}$ queries underlying the non-challenge encryption queries (i.e., queries to the first encryption oracle). These queries also give rise to records of the form (K, T_w, X, Y). To make a distinction, we denote by $\mathcal{Q}^*_{\widetilde{\mathsf{E}}}$ the union of these records and the adversarial query transcript $\mathcal{Q}_{\widetilde{\mathsf{E}}}$. Thus we have $\big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}\big| \leq Q$ which is as defined by Eq. (9). Following Sect. B, we also keep the list $\mathcal{Q}^*_{\mathsf{H}} = \big((U_1, L_1\|R_1), (U_2, L_2\|R_2), \ldots\big)$ for the appeared inputs and outputs of the hash $\mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}]$.

Besides, the list

$$\mathcal{Q}_e = \Big((N_1, A_1, M_1, C_1, T_1), \ldots, (N_{q_e}, A_{q_e}, M_{q_e}, C_{q_e}, T_{q_e})\Big)$$

summarizes the queries to the challenge (second) encryption oracle, indicating that the j-th challenge encryption query (N_j, A_j, M_j) gives rise to (C_j, T_j). For any pair of indices $(j, \ell) \in \{1, ..., q_e\} \times \{1, ..., m_j\}$ that pinpoints a message block, let $Y_j[\ell] = M_j[\ell] \oplus C_j[\ell]$. Recall that we've switched to the CPA setting, so there is no "decryption query transcript".

In all, we define the *transcript* as

$$\mathcal{Q} = (\mathcal{Q}^*_{\mathsf{H}}, \mathcal{Q}_e, \mathcal{Q}^*_{\widetilde{\mathsf{E}}}).$$

H-coefficient lemma. With respect to some fixed distinguisher $\mathcal{A}$, a transcript $\mathcal{Q}$ is called *attainable* if there exist oracles $(\widetilde{\mathsf{E}}, \widetilde{\mathsf{P}})$ such that the interaction of $\mathcal{A}$ with the ideal scheme $\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}, \widetilde{\mathsf{P}}]$ yields $\mathcal{Q}$. We denote Θ the set of attainable transcripts. In all the following, we denote T_{re}, resp. T_{id}, the probability distribution of the transcript $\mathcal{Q}$ induced by the real world, resp. the ideal world (note that these two probability distributions depend on the distinguisher). By extension, we use the same notation to denote a random variable distributed according to each distribution.

With the above, the main lemma of H-coefficient technique is as follows.

Lemma 6. *Fix a distinguisher $\mathcal{A}$. Let $\Theta = \Theta_{Good} \cup \Theta_{Bad}$ be a partition of the set of attainable transcripts Θ. Assume that there exists ε_1 such that for any $\mathcal{Q} \in \Theta_{Good}$, one has*

$$\frac{\Pr[T_{\mathsf{re}} = \mathcal{Q}]}{\Pr[T_{\mathsf{id}} = \mathcal{Q}]} \geq 1 - \varepsilon_1,$$

and that there exists ε_2 such that $\Pr[T_{\mathsf{id}} \in \Theta_{Bad}] \leq \varepsilon_2$. Then $\mathbf{Adv}(\mathcal{A}) \leq \varepsilon_1 + \varepsilon_2$.

A proof could be found in [8].

Given a set $\mathcal{Q}_{\widetilde{\mathsf{E}}}$ and an ideal TBC $\widetilde{\mathsf{E}}$, we say that $\widetilde{\mathsf{E}}$ *extends* $\mathcal{Q}_{\widetilde{\mathsf{E}}}$, denoted $\widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}$, if $\widetilde{\mathsf{E}}_K^{T_w}(X) = Y$ for all $(K, T_w, X, Y) \in \mathcal{Q}_{\widetilde{\mathsf{E}}}$. It's easy to see that for any attainable transcript $\mathcal{Q} = (\mathcal{Q}_{\mathsf{H}}^*, \mathcal{Q}_e, \mathcal{Q}_{\widetilde{\mathsf{E}}}^*)$, the interaction of $\mathcal{A}$ with oracles $(\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}}, \widetilde{\mathsf{P}}], \widetilde{\mathsf{E}})$ produces $\mathcal{Q}$ if and only if $\widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}$ and the encryption oracle responds consistently with $\mathcal{Q}_e$.

Wlog assume that $|M_j[\ell]| = n$ for any message block $M_j[\ell]$. Then, in the ideal world, all the blocks in $C_1, \ldots, C_{q_e}$ are uniformly distributed in $\{0,1\}^n$. Therefore,

$$\Pr\big[T_{\mathsf{id}} = \mathcal{Q}\big] = \Pr\big[\widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^*\big] \times \left(\frac{1}{2^n}\right)^{q_e + \sum_{i=1}^{q_e} m_i}. \tag{18}$$

Bad Transcripts. With the above, a transcript $\mathcal{Q}$ is bad if one of the following is fulfilled:

– (B-1) $\mu_Y \geq 2\log_2 \sigma_m$, where

$$\mu_Y := \max_{Y \in \{0,1\}^n} \big|\{(j,\ell) : j \in \{1,...,q_e\},\ \{1,...,m_j\},\ M_j[\ell] \oplus C_j[\ell] = Y\}\big|.$$

– (B-2)*Multi hash semi-collision*: $\mu_R \geq n+1$, where

$$\mu_R := \max_{r \in \{0,1\}^n} \Big|\big\{(U, L\|R) \in \mathcal{Q}_{\mathsf{H}}^* : R = r\big\}\Big|.$$

– (B-3) there exists two distinct encryption queries $(N_j, A_j, M_j, C_j, T_j)$ and $(N_{j'}, A_{j'}, M_{j'}, C_{j'}, T_{j'})$ with the corresponding hash records $(\mathsf{RTpad}(A_j, N_j, C_j), L_j\|R_j) \neq (\mathsf{RTpad}(A_{j'}, N_{j'}, C_{j'}), L_{j'}\|R_{j'}) \in \mathcal{Q}_{\mathsf{H}}^*$ satisfying either of the follows:
  - (B-31)*hash collision*: $L_j\|R_j = L_{j'}\|R_{j'}$, or
  - (B-32)*contradiction*: $R_j = R_{j'}$ and $T_j = T_{j'}$.

For (B-1), in the ideal world $C_j[\ell]$ and thus $M_j[\ell] \oplus C_j[\ell]$ is uniform. In addition, $\sum_{j=1}^{q_e} m_j = \sigma_m \ll 2^n$. Hence, Lemma 1 implies

$$\Pr\big[\text{(B-1)}\big] = \Pr\big[\mu_Y \geq 2\log_2 \sigma_m\big] \leq \frac{1}{2^n}.$$

By Lemma 5, we have

$$\Pr\big[\text{(B-2)} \vee \text{(B-31)}\big] \leq \frac{11Q}{2^n}.$$

Conditioned on $\neg$(B-2), for any $(N_j, A_j, M_j, C_j, T_j) \in \mathcal{Q}_e$ with $(\mathsf{RTpad}(A_j, N_j, C_j), L_j\|R_j) \in \mathcal{Q}_{\mathsf{H}}^*$, the number of $(N_{j'}, A_{j'}, M_{j'}, C_{j'}, T_{j'}) \in \mathcal{Q}_e$ with $(\mathsf{RTpad}(A_{j'}, N_{j'}, C_{j'}), L_{j'}\|R_{j'}) \in \mathcal{Q}_{\mathsf{H}}^*$ is at most $n-1$. For each pair of such indices (j, j'), the tags T_j and $T_{j'}$ are uniform in the ideal world, and thus $\Pr[T_j = T_{j'}] = \frac{1}{2^n}$. Since we have at most q_e choices for j, it holds

$$\Pr\big[\text{(B-32)} \mid \text{(B-2)}\big] \leq \frac{(n-1)q_e}{2^n}.$$

In all,

$$\Pr\big[T_{\mathsf{id}} \in \Theta_{\mathsf{Bad}}\big] \leq \frac{1}{2^n} + \frac{11Q}{2^n} + \frac{(n-1)q_e}{2^n} \leq \frac{11Q}{2^n} + \frac{nq_e}{2^n}. \tag{19}$$

Ratio of Probabilities of Good Transcripts. Consider an arbitrary good transcript $\mathcal{Q}$. For any $(N_j, A_j, M_j, C_j, T_j) \in \mathcal{Q}_e$, the initial session key $S_0^{(j)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_j)$ is uniform. With this observation, we define a predicate $\mathsf{BadKD}(\widetilde{\mathsf{P}})$ to capture the "none-freshness" of this key.

Formally, $\mathsf{BadKD}(\widetilde{\mathsf{P}})$ is fulfilled, if there exists a record $(N_j, A_j, M_j, C_j, T_j) \in \mathcal{Q}_e$ such that the key $S_0^{(j)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_j)$ satisfies one of the follows:

- $\big(S_0^{(j)}, (0^n, 65, \pi(0)), N_j, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$, or
- $\big(S_0^{(j)}, (0^n, 64, \pi(0)), N_j, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$, or
- $\big(S_0^{(j)}, (0^n, 64, \pi(0)), \star, M_j[1] \oplus C_j[1]\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$.

For a pair $(N, \ell) \in \{0,1\}^n \times \{0, 1, ...\}$, we define an auxiliary set of keys

$$\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[N, \ell] := \Big\{ S : \big(S, (0^n, 65, \pi(\ell)), N, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}} \text{ or } \big(S, (0^n, 64, \pi(\ell)), N, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}} \Big\}. \tag{20}$$

In addition, for a key stream block $Y \in \{0,1\}^n$, define

$$\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[Y, \ell]^{-1} := \big\{ S : \big(S, (0^n, 64, \pi(\ell)), \star, Y\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}} \big\}. \tag{21}$$

Conditioned on the values of

$$S_0^{(1)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_1), \ldots, S_0^{(j-1)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_{j-1}),$$

the key $S_0^{(j)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_j)$ is uniform in at least $2^n - q_e - q_m$ possibilities, since it must be the first (and unique) time the nonce N_j appears in encryption queries. Therefore, when $q_e + q_m \le Q/2 \le 2^n/2$, we have

$$\begin{aligned}
\Pr_{\widetilde{\mathsf{P}}}\big[\mathsf{BadKD}(\widetilde{\mathsf{P}})\big] &\le \sum_{j=1}^{q_e} \frac{\big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[N_j, 0]\big|}{2^n - q_e - q_m} + \sum_{j=1}^{q_e} \frac{\big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[Y_j[1], 0]^{-1}\big|}{2^n - q_e - q_m} \\
&\le \sum_{j=1}^{q_e} 2 \frac{\big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[N_j, 0]\big| + \big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[Y_j[1], 0]^{-1}\big|}{2^n}.
\end{aligned} \tag{22}$$

We then analyze the q_e encryption queries in turn, and define a sequence of bad predicates

$$\begin{aligned}
&\mathsf{BadE}_1^{(1)}, \mathsf{BadE}_2^{(1)}, \ldots, \mathsf{BadE}_{m_1-1}^{(1)}, \\
&\ldots \\
&\mathsf{BadE}_1^{(q_e)}, \mathsf{BadE}_2^{(q_e)}, \ldots, \mathsf{BadE}_{m_{q_e}-1}^{(q_e)}.
\end{aligned} \tag{23}$$

As will be seen, each predicate concerns with the encryption of a specific plaintext block. Formally, for $1 \le j \le q_e$, consider the j-th query $(N_j, A_j, M_j, C_j, T_j)$, and let

$$S_0^{(j)} = \widetilde{\mathsf{P}}^{(0^n,66,0^{n-8})}(N_j),\ S_1^{(j)} = \widetilde{\mathsf{E}}_{S_0^{(j)}}^{(0^n,65,\pi(0))}(N_j),\ S_2^{(j)} = \widetilde{\mathsf{E}}_{S_1^{(j)}}^{(0^n,65,\pi(1))}(N_j), ..., S_{m_j-1}^{(j)} = \widetilde{\mathsf{E}}_{S_{m_j-2}^{(j)}}^{(0^n,65,\pi(m_j-2))}(N_j)$$

be the derived intermediate values. Then for $1 \le \ell \le m_j - 1$, $\mathsf{BadE}_\ell^{(j)}(\widetilde{\mathsf{E}})$ is fulfilled, if any of the following conditions is fulfilled:

- $\mathsf{BadE}_\ell^{(j)}$-(C-1): $\big(S_\ell^{(j)}, (0^n, 65, \pi(\ell)), N_j, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$ or $\big(S_\ell^{(j)}, (0^n, 64, \pi(\ell)), N_j, \star\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$, or $\big(S_\ell^{(j)}, (0^n, 64, \pi(\ell)), \star, Y_j[\ell+1]\big) \in \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$;
- $\mathsf{BadE}_\ell^{(j)}$-(C-2): $S_\ell^{(j)} = Y_j[\ell]$ (recall that $Y_j[\ell] = M_j[\ell] \oplus C_j[\ell]$).

It is not hard to see that, conditioned on $\widetilde{\mathsf{E}} \vdash \mathcal{Q}^*_{\widetilde{\mathsf{E}}}$ and $\neg\mathsf{BadKD}(\widetilde{\mathsf{P}})$ and $\neg\mathsf{BadE}_{\ell-1}^{(j)}(\widetilde{\mathsf{E}}) \wedge \ldots \wedge \neg\mathsf{BadE}_1^{(1)}(\widetilde{\mathsf{E}})$, the value $S_\ell^{(j)}$ is uniform in at least $2^n - Q$ possibilities.[2] Therefore,

$$\Pr\big[\mathsf{BadE}_\ell^{(j)}\text{-(C-1)} \vee \mathsf{BadE}_\ell^{(j)}\text{-(C-2)}\big] \le \frac{\Big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[N_j, \ell]\Big| + \Big|\mathcal{Q}^*_{\widetilde{\mathsf{E}}}[Y_j[\ell+1], \ell]^{-1}\Big| + 1}{2^n - Q}.$$

[2] This proof didn't normalize $\mathcal{A}$, and thus the number of $\widetilde{\mathsf{E}}$ queries remains Q.

Thus, when $Q \le 2^n/2$, we have

$$\begin{aligned}
&\Pr\Big[\mathsf{BadE}_{\ell}^{(j)}(\widetilde{\mathsf{E}}) \mid \neg\mathsf{BadE}_{\ell-1}^{(j)}(\widetilde{\mathsf{E}}) \wedge \ldots \wedge \neg\mathsf{BadE}_{1}^{(1)}(\widetilde{\mathsf{E}}) \wedge \neg\mathsf{BadKD}(\widetilde{\mathsf{P}}) \wedge \widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}\Big] \\
&\le 2\frac{\Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[N_j,\ell]\Big| + \Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[Y_j[\ell+1],\ell]^{-1}\Big| + 1}{2^n}.
\end{aligned}$$

For $1 \le j \le q_e$ and $1 \le \ell \le m_j - 1$, conditioned on $\neg\mathsf{BadE}_{\ell}^{(j)}(\widetilde{\mathsf{E}}) \wedge \neg\mathsf{BadE}_{\ell-1}^{(j)}(\widetilde{\mathsf{E}}) \wedge \ldots \wedge \neg\mathsf{BadE}_{1}^{(1)}(\widetilde{\mathsf{E}}) \wedge \neg\mathsf{BadKD}(\widetilde{\mathsf{P}}) \wedge \widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}$, it can be seen the value $Y^{\dagger} = \widetilde{\mathsf{E}}_{S_{\ell}^{(j-1)}}^{(0^n,64,\pi(\ell))}(N_j)$ is uniform in at least $2^n - Q$ possibilities, and *these possibilities include* $Y_j[\ell+1]$ *(due to* $\neg\mathsf{BadE}_{\ell}^{(j)}$*-(C-1):* $\big(S_{\ell}^{(j)},(0^n,64,\pi(\ell)),\star,Y_j[\ell+1]\big) \notin \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}$*)*. Therefore,

$$\Pr\big[Y^{\dagger} = Y_j[\ell+1]\big] \ge \frac{1}{2^n}. \tag{24}$$

The probabilities of the predicates, plus Eq. (22), accumulate to

$$\begin{aligned}
&\Pr\Big[\underbrace{\mathsf{BadE}_{m_{q_e}-1}^{(q_e)}(\widetilde{\mathsf{E}}) \vee \ldots \vee \mathsf{BadE}_{1}^{(1)}(\widetilde{\mathsf{E}}) \vee \neg\mathsf{BadKD}(\widetilde{\mathsf{P}})}_{=\mathsf{Bad}(\widetilde{\mathsf{P}},\widetilde{\mathsf{E}})} \mid \widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}\Big] \\
&\le \sum_{j=1}^{q_e}\sum_{\ell=0}^{m_i-1} \frac{2\Big(\Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[N_j,\ell]\Big| + \Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[Y_j[\ell+1],\ell]^{-1}\Big| + 1\Big)}{2^n}.
\end{aligned}$$

Since $N_1, N_2, ..., N_{q_e}$ are distinct, it holds $\sum_{j=1}^{q_e}\sum_{\ell=0}^{m_j-1}\Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[N_j,\ell]\Big| \le Q$. On the other hand,

$$\sum_{j=1}^{q_e}\sum_{\ell=0}^{m_j-1}\Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[Y_j[\ell+1],\ell]^{-1}\Big| \le \sum_{Y\in\{0,1\}^n}\ \sum_{(j,\ell)\in\{1,\ldots,q_e\}\times\{1,\ldots,m_j\}:Y_j[\ell]=Y}\Big|\mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}[Y,\ell]^{-1}\Big| \le \mu_Y Q. \tag{25}$$

Gathering the above yields

$$\begin{aligned}
&\Pr\Big[\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}].\mathcal{E}(N_j,A_j,M_j) = C_j \text{ for all } j\in\{1,\ldots,q_e\} \mid \widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}\Big] \\
&\ge \Pr\Big[\neg\mathsf{Bad}(\widetilde{\mathsf{E}}) \mid \widetilde{\mathsf{E}} \vdash \mathcal{Q}_{\widetilde{\mathsf{E}}}^{*}\Big]\left(\frac{1}{2^n}\right)^{\sum_{j=1}^{q_e} m_i} \\
&\ge \left(1 - \frac{2Q + 2\mu_Y Q + 2\sigma_m}{2^n}\right)\left(\frac{1}{2^n}\right)^{\sigma_m}.
\end{aligned} \tag{26}$$

It remains to analyze the produced tags. Let the hash query record corresponding to $(N_j, A_j, M_j, C_j, T_j)$ be $(U_j, L_j\|R_j)$. Therefore, the event that the q_e tags equal $T_1, ..., T_{q_e}$ is equivalent to q_e equalities as follows:

$$\widetilde{\mathsf{P}}^{(R_1,68,0^{n-8})}(L_1) = T_1, \ldots, \widetilde{\mathsf{P}}^{(R_{q_e},68,0^{n-8})}(L_{q_e}) = T_{q_e}.$$

For the first equality, it clearly holds $\Pr\big[\widetilde{\mathsf{P}}^{(R_1,68,0^{n-8})}(L_1) = T_1\big] = \frac{1}{2^n}$. For the j-th equality, $j \in \{1,\ldots,q_e\}$, we need to additionally consider the influence of "$\widetilde{\mathsf{P}}^{(R_{j'},68,0^{n-8})}(L_{j'}) = T_{j'}$ for $j' = 1,\ldots,j-1$". By ¬(B-3), $L_{j'}\|R_{j'} \ne L_j\|R_j$ and $T_{j'}\|R_{j'} \ne T_j\|R_j$ for any $j' < j$. By this, $\Pr\big[\widetilde{\mathsf{P}}^{(R_j,68,0^{n-8})}(L_j) = T_j\big] \ge \frac{1}{2^n}$, and thus

$$\Pr\big[\widetilde{\mathsf{P}}^{(R_j,68,0^{n-8})}(L_j) = T_j \text{ for } j = 1,\ldots,q_e\big] \ge \frac{1}{2^{q_e n}}. \tag{27}$$

Gathering Eqs. (18), (26) and (27), and with $Q \le 2^n/8 \Rightarrow \log_2 \sigma_m \le n-3$, we have

$$\begin{aligned}
\frac{\Pr[T_{\mathsf{re}} = \tau]}{\Pr[T_{\mathsf{id}} = \tau]} &\ge \left(1 - \frac{2Q + 2\mu_Y Q + 2\sigma_m}{2^n}\right)\left(\frac{1}{2^n}\right)^{q_e+\sigma_m} \Big/ \left(\frac{1}{2^n}\right)^{q_e+\sigma_m} \\
&\ge 1 - \frac{(4n-8)Q}{2^n}, \qquad (\mu_Y \le 2\log_2\sigma_m \le 2(n-3)).
\end{aligned}$$

This plus Eqs. (17) and (19) yield

$$\mathbf{Adv}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}^{\mathsf{CPAm\$}}(\mathcal{D}) \le \frac{(4n+4)Q}{2^n} + \frac{nq_e}{2^n}.$$

By Eq. (16), this plus the bound in Eq. (1) yields Eq. (2).

D Proof of Theorem 3

This proof closely follows [4, Sect. 6.5], and the changes are mainly notational. In detail, we first define the process of encrypting a single message of m blocks. We in particular define both the real and the ideal encryption processes: the real encryption $\mathsf{RESM}[\widetilde{\mathsf{E}}]$ "mimics" Romulus-T and queries $\widetilde{\mathsf{E}}$ for encrypting one message, while the ideal process just samples many random values for encrypting. We show that the two processes are indistinguishable, with the help of the non-invertible leakage assumption.

We then focus on the idealized process $(\$, \mathsf{L}_{ideal}(M))$, and show how to relate its eavesdropper advantage to the term defined by Eq. (5). The (leaking) eavesdropper advantage of $(\mathsf{RESM}, \mathsf{L}_{\mathsf{RESM}}(M))$ can be derived via:

$$\begin{aligned}
&\text{(leaking) eavesdropper advantage of the minimal operation } \mathbf{Adv}^{\mathsf{LORL2}} \\
\Rightarrow &\text{(leaking) eavesdropper advantage of the ideal } (\$, \mathsf{L}_{ideal}(M)) \\
\Rightarrow &\text{(leaking) eavesdropper advantage of } (\mathsf{RESM}, \mathsf{L}_{\mathsf{RESM}}(M)).
\end{aligned}$$

As the 3rd step, based on the leaking eavesdropper advantage of $(\mathsf{RESM}, \mathsf{L}_{\mathsf{RESM}}(M))$, we establish the CCAmL2 advantage for Romulus-T. Below we expose in detail.

The Ideal Single-Message Encryption Process. Formally, they are defined by the following pseudocode.

Description of $\mathsf{RESM}[\widetilde{\mathsf{E}}]$:

- Gen picks $S_0 \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N, M[1]\|\ldots\|M[m])$ proceeds in two steps:
  (1) Initializes an empty list leak for the leakage;
  (2) for $i = 1, \ldots, m$, computes $S_i \leftarrow \widetilde{\mathsf{E}}_{S_{i-1}}^{(0^n,65,\pi(i-1))}(N)$, $Y[i] \leftarrow \widetilde{\mathsf{E}}_{S_{i-1}}^{(0^n,64,\pi(i-1))}(N)$, and $C[i] \leftarrow Y[i] \oplus M[i]$, and adds the leakages $[\mathsf{L}^{in}(S_{i-1}, (0^n, 65, \pi(i-1)); N), \mathsf{L}^{out}(k_{i-1}, (0^n, 65, \pi(i-1)); S_i)]^p$, $[\mathsf{L}^{in}(k_{i-1}, (0^n, 64, \pi(i-1)); N), \mathsf{L}^{out}(k_{i-1}, (0^n, 64, \pi(i-1)); Y[i])]^p$, $\mathsf{L}_{\oplus}(Y[i], M[i])$, and $[\mathsf{L}_{\oplus}(Y[i], C[i])]^{p-1}$ to the list leak.

  $\mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N, M[1]\|\ldots\|M[m])$ eventually returns $C[1]\|...\|C[m]$.
  We define $\mathsf{LRESM}_{S_0}[\widetilde{\mathsf{E}}](N, M[1]\|\ldots\|M[m]) = (\mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N, M[1]\|\ldots\|M[m]), \mathsf{leak})$ for the list leak standing at the end of the above process.

Description of IESM (an ideal process independent from $\widetilde{\mathsf{E}}$):

- $S_0 \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{IESM}_{S_0}(N, M[1]\|\ldots\|M[m])$ proceeds in two steps:
  (1) Initializes an empty list leak for the leakage;
  (2) for $i = 1, \ldots, m$, samples $S_i \xleftarrow{\$} \{0,1\}^n$ and $Y[i] \xleftarrow{\$} \{0,1\}^n$ such that $S_i \neq Y[i]$, sets $C[i] \leftarrow Y[i] \oplus M[i]$, and adds the leakages $[\mathsf{L}^{in}(S_{i-1}, (0^n, 65, \pi(i-1)); N), \mathsf{L}^{out}(k_{i-1}, (0^n, 65, \pi(i-1)); S_i)]^p$, $[\mathsf{L}^{in}(k_{i-1}, (0^n, 64, \pi(i-1)); N), \mathsf{L}^{out}(k_{i-1}, (0^n, 64, \pi(i-1)); Y[i])]^p$, $\mathsf{L}_{\oplus}(Y[i], M[i])$, and $[\mathsf{L}_{\oplus}(Y[i], C[i])]^{p-1}$ to the list leak.

  $\mathsf{IESM}_{S_0}(N, M[1]\|\ldots\|M[m])$ eventually returns $C[1]\|\ldots\|C[m]$.
  We define $\mathsf{LIESM}_{S_0}(N, M[1]\|\ldots\|M[m]) = (\mathsf{IESM}_{S_0}(N, M[1]\|\ldots\|M[m]), \mathsf{leak})$ for the list leak standing at the end of the above process.

The real and ideal single-message encryption processes (with leakages) are indistinguishable. This is a notational adaptation of [4, Lemma 6].

Lemma 7. *For every m-block message M, every nonce N, and every distinguisher $\mathcal{D}^{\widetilde{\mathsf{E}}}$ that makes $q_{\widetilde{\mathsf{E}}}$ queries to $\widetilde{\mathsf{E}}$ and runs in time t, it holds*

$$\begin{aligned}
&\left|\Pr[\mathcal{D}^{\widetilde{\mathsf{E}}}(M, \mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N, M[1]\|\ldots\|M[m])) \Rightarrow 1] - \Pr[\mathcal{D}^{\widetilde{\mathsf{E}}}(M, \mathsf{IESM}_{S_0}(N, M[1]\|\ldots\|M[m])) \Rightarrow 1]\right| \\
\leq &m \cdot \mathbf{Adv}^{\mathit{2\text{-}up}[q_{\widetilde{\mathsf{E}}}]}(p, q_{\widetilde{\mathsf{E}}} + 2m, O(t + m \cdot p \cdot t_l))
\end{aligned}$$

where t_l is the total time needed for evaluating L^{in} and L^{out}.

From 1-Block to m-Block Advantage. We then show the eavesdropper advantage of $\mathsf{IESM}[\widetilde{\mathsf{E}}]$ encrypting an m-block message is related to the defined term $\mathbf{Adv}^{\mathsf{LORL2}}$. This is a notational adaptation of [4, Lemma 7].

Lemma 8. *For every pair of m-block messages M^0 and M^1 and every distinguisher $\mathcal{D}^{\widetilde{\mathsf{E}}}$ that makes $q_{\widetilde{\mathsf{E}}}$ queries to $\widetilde{\mathsf{E}}$ and runs in time t, it holds*

$$\left|\Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{IESM}_{S_0}(N,M^0)) \Rightarrow 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{IESM}_{S_0}(N,M^1)) \Rightarrow 1]\right| \le m\cdot \mathbf{Adv}^{\mathit{LORL2}}(p, q_{\widetilde{\mathsf{E}}}, O(t+m\cdot p\cdot t_l)) + \frac{m}{2^n},$$

where t_l is as defined in Lemma 7.

For simplicity, we define

$$\mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(p, q_{\widetilde{\mathsf{E}}}, t, m) := \max_{\mathcal{A}}\left\{\left|\Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{LRESM}_{S_0}[\widetilde{\mathsf{E}}](N,M^0)) \Rightarrow 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{LRESM}_{S_0}[\widetilde{\mathsf{E}}](N,M^1)) \Rightarrow 1]\right|\right\},$$

where the abbreviation $\mathsf{eavl2}$ stands for *eavesdropper security with encryption and decryption leakages*, and the maximal is taken over all adversaries making $q_{\widetilde{\mathsf{E}}}$ queries to $\widetilde{\mathsf{E}}$ and running in time t. Gathering Lemmas 7 and 8, we obtain upper bounds on the eavesdropper advantage of RESM (which is also the eavesdropper advantage of $\mathsf{Romulus\text{-}T}$, since RESM "mimics" $\mathsf{Romulus\text{-}T}$) stated in Lemma 9. This is a notational adaptation of [4, Lemma 8].

Lemma 9. *For every pair of m-block messages M^0 and M^1 and every eavesdropper adversary $\mathcal{A}^{\widetilde{\mathsf{E}}}$ that makes $q_{\widetilde{\mathsf{E}}}$ queries to $\widetilde{\mathsf{E}}$ and runs in time t, it holds*

$$\begin{aligned}\mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(\mathcal{A}) =& \left|\Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N,M^0)) \Rightarrow 1] - \Pr[\mathcal{A}^{\widetilde{\mathsf{E}}}(\mathsf{RESM}_{S_0}[\widetilde{\mathsf{E}}](N,M^1)) \Rightarrow 1]\right| \\ \le& \frac{m}{2^n} + m\cdot\mathbf{Adv}^{\mathit{LORL2}}(p, q_{\widetilde{\mathsf{E}}}, O(t+m\cdot p\cdot t_l)) + 2m\cdot\mathbf{Adv}^{\mathit{2\text{-}up}[q_{\widetilde{\mathsf{E}}}]}(p, q_{\widetilde{\mathsf{E}}}+2m, O(t+m\cdot p\cdot t_l)),\end{aligned}$$

where t_l is as defined in Lemma 7.

Completing the CCAmL2 Proof. We now establish Theorem 3 with the help of Lemma 9.

First, recall from Definition 3 that a decryption query $\mathcal{D}_K(N,A,C)$ is *trivial* if the action $\mathcal{E}_K(N,A,M)\to C$ happens before. The leakages of trivial decryption queries may serve new information, thus requiring explicit treatments.

Then we step into the proof. For convenience, let G_0 be the game $\mathsf{PrivK}^{\mathsf{CCAmL2},0}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}}$, while G_0^* the game $\mathsf{PrivK}^{\mathsf{CCAmL2},1}_{\mathcal{A},\mathsf{AEAD},\mathsf{L}}$. The goal thus reduces to bounding $\left|\Pr[\mathsf{G}_0\Rightarrow 1]-\Pr[\mathsf{G}_0^*\Rightarrow 1]\right|$.

We proceed with the standard hybrid argument. For this, we first define two games G_1 and G_1^*: G_1, resp. G_1^*, is obtained from G_0, resp. G_0^*, by replacing all the KDF- and TGF-calls by calls to $\widetilde{\mathsf{P}}$. By Eq. (11), with $Q=3\sigma_m+\sigma_a+6(q_e+q_d+q_m)+q_{\widetilde{\mathsf{E}}}$, we have

$$\left|\Pr[\mathsf{G}_1\Rightarrow 1]-\Pr[\mathsf{G}_0\Rightarrow 1]\right| \le \frac{Q}{2^n} \text{ and } \left|\Pr[\mathsf{G}_1^*\Rightarrow 1]-\Pr[\mathsf{G}_0^*\Rightarrow 1]\right| \le \frac{Q}{2^n}. \tag{28}$$

Starting from G_1 and G_1^*, we define two more games G_2 and G_2^*: G_2, resp. G_2^*, is obtained from G_1, resp. G_1^*, by replacing the leaking decryption oracle $\mathcal{LD}$ (see Fig. 1) with the "always $\perp$" decryption oracle $\mathcal{LD}_K^{\perp}$ defined in Definition 3. It is easy to see

$$\begin{aligned}\left|\Pr[\mathsf{G}_2\Rightarrow 1]-\Pr[\mathsf{G}_1\Rightarrow 1]\right| &\le \mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}(\mathcal{A}) \le \frac{11Q+2nq_d}{2^n}+\frac{8Q(q_d+q_e+q_m)}{2^{2n}}, \\ \left|\Pr[\mathsf{G}_2^*\Rightarrow 1]-\Pr[\mathsf{G}_1^*\Rightarrow 1]\right| &\le \mathbf{Adv}^{\mathsf{CIML2}}_{\mathsf{Romulus\text{-}T}[\widetilde{\mathsf{E}},\widetilde{\mathsf{P}}]}(\mathcal{A}) \le \frac{11Q+2nq_d}{2^n}+\frac{8Q(q_d+q_e+q_m)}{2^{2n}}\end{aligned} \tag{29}$$

by adapting Eq. (14).

We then prove that

$$\begin{aligned}&\left|\Pr[\mathsf{G}_2\Rightarrow 1]-\Pr[\mathsf{G}_2^*\Rightarrow 1]\right| \\ \le\ & \underbrace{\sum_{i=1}^{q_e}\mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(p, q_{\widetilde{\mathsf{E}}}+Q, O(t+p\sigma_m t_l), m_i)}_{\le \frac{\sigma_m}{2^n}+\sigma_m\cdot\mathbf{Adv}^{\mathsf{LORL2}}(p,q_{\widetilde{\mathsf{E}}}+Q,O(t+p\sigma_m t_l))+2\sigma_m\cdot\mathbf{Adv}^{\mathsf{2\text{-}up}[q_{\widetilde{\mathsf{E}}}+Q]}(p,q_{\widetilde{\mathsf{E}}}+Q,O(t+p\sigma_m t_l))\ (\text{Lemma 9})}\ ,\end{aligned} \tag{30}$$

where m_i is the number of blocks in the ith challenge message, and $Q = 3\sigma_m + \sigma_a + 6(q_e + q_d + q_m) + q_{\widetilde{\mathsf{E}}}$ and t_l defined in Lemma 7. Gathering this and the gaps in Eqs. (28) and (29), we have

$$
\begin{aligned}
&\mathbf{Adv}^{\mathsf{CCAmL2}}_{\mathsf{Romulus\text{-}T,L}}(q_e, q_m, q_d, p-1, q_{\widetilde{\mathsf{E}}}, \sigma_a, \sigma_m, t) \\
&= \left|\Pr[\mathsf{G}_0 \Rightarrow 1] - \Pr[\mathsf{G}_0^* \Rightarrow 1]\right| \\
&\le 2 \times \Big(\frac{Q}{2^n} + \frac{11Q + 2nq_d}{2^n} + \frac{8Q(q_d + q_e + q_m)}{2^{2n}}\Big) + \\
&\quad \underbrace{\frac{\sigma_m}{2^n}}_{\le Q/2^n} + \sigma_m \cdot \mathbf{Adv}^{\mathsf{LORL2}}(p, q_{\widetilde{\mathsf{E}}} + Q, O(t + p\sigma_m t_l)) + 2\sigma_m \cdot \mathbf{Adv}^{\mathsf{2\text{-}up}[q_{\widetilde{\mathsf{E}}}+Q]}(p, q_{\widetilde{\mathsf{E}}} + Q, O(t + p\sigma_m t_l)) \\
&\le \frac{25Q + 4nq_d}{2^n} + \frac{16Q(q_d + q_e + q_m)}{2^{2n}} + \sigma_m \cdot \mathbf{Adv}^{\mathsf{LORL2}}(p, q^*, t^*) + 2\sigma_m \cdot \mathbf{Adv}^{\mathsf{2\text{-}up}[q^*]}(p, q^*, t^*). \qquad (31)
\end{aligned}
$$

These establish the claim Eq. (8).

To prove Eq. (30), we denote the q_e challenge tuples by (the suffix c stands for "challenge")

$$(Nc_1, Ac_1, Mc_1^0, Mc_1^1), \ldots, (Nc_{q_e}, Ac_{q_e}, Mc_{q_e}^0, Mc_{q_e}^1).$$

Then, we use q_e hops to replace $Mc_1^0, \ldots, Mc_{q_e}^0$ by $Mc_1^1, \ldots, Mc_{q_e}^1$ in turn, to show that G_2 can be transited to G_2^*. For convenience, we define $\mathsf{G}_{3,0} = \mathsf{G}_2$, and define a sequence of games

$$\mathsf{G}_{3,1}, \mathsf{G}_{3,2}, \ldots, \mathsf{G}_{3,q_e},$$

such that in the i-th system $\mathsf{G}_{3,i}$, the first i messages processed by the challenge encryption oracle are $Mc_1^0, \ldots, Mc_i^0$, while the remaining $q_e - i$ messages being processed are $Mc_{i+1}^1, \ldots, Mc_{q_e}^1$. It can be seen actually $\mathsf{G}_{3,q_e} = \mathsf{G}_2^*$.

We then show that for $i = 1, \ldots, q_e$, $\mathsf{G}_{3,i-1}$ and $\mathsf{G}_{3,i}$ are indistinguishable for $\mathcal{A}^{\widetilde{\mathsf{E}}}$. For this, from $\mathcal{A}^{\widetilde{\mathsf{E}}}$ we build an adversary $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$, such that $|\Pr[\mathsf{G}_{3,i-1} \Rightarrow 1] - \Pr[\mathsf{G}_{3,i} \Rightarrow 1]|$ is related to $\mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(\mathcal{A}_2^{\widetilde{\mathsf{E}}})$. In detail, initially, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ prepares a pair of tables $(PTable, PTable^{-1})$ to simulate the primitive $\widetilde{\mathsf{P}}$ via lazy sampling (recall that $\widetilde{\mathsf{P}}$ is a random tweakable permutation independent from $\widetilde{\mathsf{E}}$). Assume that entries in the tables are of the form $PTable(T_w, X) = Y$ and $PTable^{-1}(T_w, Y) = X$. It then runs $\mathcal{A}$, reacting as follows:

- Upon a query to $\widetilde{\mathsf{E}}$: simply relays.
- Upon a (non-challenge) encryption query (N_i, A_i, M_i) from $\mathcal{A}$,
  - if $((0^n, 66, 0^{n-8}), N_i) \notin PTable$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ samples an initial key $S_0^{(i)}$ such that $((0^n, 66, 0^{n-8}), S_0^{(i)}) \notin PTable^{-1}$, defines $PTable((0^n, 66, 0^{n-8}), N_i) \leftarrow S_0^{(i)}$ and $PTable^{-1}((0^n, 66, 0^{n-8}), S_0^{(i)}) \leftarrow N_i$, and then runs the encryption process $\mathsf{RESM}_{S_0^{(i)}}[\widetilde{\mathsf{E}}](N_i, M_i)$ to get the ciphertext C_i and leakages. $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ then computes $L_i \| R_i \leftarrow \mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](\mathsf{RTpad}(A_i, N_i, C_i))$ and $T_i \leftarrow PTable((R_i, 68, 0^{n-8}), L_i)$ (if $((R_i, 68, 0^{n-8}), L_i) \notin PTable$ then $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ defines $PTable((R_i, 68, 0^{n-8}), L_i)$ to a newly sampled value). For this entire process $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ has to make at most $3m_i + a_i + 6$ queries to $\widetilde{\mathsf{E}}$ with $m_i = |M_i|_n$ and $a_i = |A_i|_n$ (as analyzed in Appendix A.2) and spends $O(pm_it_l)$ time to evaluating the leakage functions. Finally, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ returns the results (C_i, T_i) and the leakages to $\mathcal{A}$;
  - if $((0^n, 66, 0^{n-8}), N_i) \in PTable$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ simply runs $\mathsf{RESM}_{S_0}(N_i, M_i)$ with $S_0 = PTable((0^n, 66, 0^{n-8}), N_i)$, calls $L_i \| R_i \leftarrow \mathsf{H}(\mathsf{RTpad}(A_i, N_i, C_i))$ and computes the tag $T_i \leftarrow PTable((R_i, 68, 0^{n-8}), R_i)$ on the obtained C_i, and returns (C_i, T_i) and the leakages to $\mathcal{A}$. The cost is similar to the above case.
- Upon a trivial decryption query (N_j, A_j, C_j, T_j) from $\mathcal{A}$ (cf. the beginning of this subsection for "trivial"), $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ simply runs the decryption $\mathsf{RESM}[\widetilde{\mathsf{E}}].\mathsf{Dec}_{S_0^j}(N_j, C_j)$ for $S_0^j = PTable((0^n, 66, 0^{n-8}), N_j)$, and relays the outputs to $\mathcal{A}$. The cost is similar to the encryption case.
- Upon a non-trivial decryption query (N_j, A_j, C_j, T_j) from $\mathcal{A}$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ computes $L_j \| R_j \leftarrow \mathsf{Romulus\text{-}H}[\widetilde{\mathsf{E}}](\mathsf{RTpad}(A_j, N_j, C_j))$. Then,
  - if $((R_j, 68, 0^{n-8}), T_j) \notin PTable^{-1}$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ samples L_j^* such that $((R_j, 68, 0^{n-8}), L_j^*) \notin PTable$, and sets $PTable((R_j, 68, 0^{n-8}), L_j) \leftarrow T_j$, $PTable^{-1}((R_j, 68, 0^{n-8}), T_j) \leftarrow L_j^*$;
  - if $((R_j, 68, 0^{n-8}), T_j) \in PTable^{-1}$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ just sets $L_j^* \leftarrow PTable^{-1}((R_j, 68, 0^{n-8}), T_j)$.

  Recall that $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ is mimicking "always $\perp$" decryption oracle $\mathcal{LD}_K^{\perp}$. Therefore, it returns $(\perp, L_j^*)$ to $\mathcal{A}$.
- Upon $\mathcal{A}$ submitting the j-th challenge tuple $(Nc_j, Ac_j, Mc_j^0, Mc_j^1)$, since the nonce Nc_j is fresh (by the restriction of $\mathsf{CCAmL2}$), it holds $((0^n, 66, 0^{n-8}), Nc_j) \notin PTable$. Therefore, depending on j, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ reacts as follows:

- When $j < i$, it encrypts Mc_j^0 and returns. In detail, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ samples $Sc_0^{(j)} \xleftarrow{\$} \{0,1\}^n$, sets table entries $PTable((0^n, 66, 0^{n-8}), Nc_j) \leftarrow Sc_0^{(j)}$ and $PTable^{-1}((0^n, 66, 0^{n-8}), Sc_0^{(j)}) \leftarrow Nc_j)$, and then runs $\mathsf{RESM}_{Sc_0^{(j)}}[\widetilde{\mathsf{E}}](Mc_j^0)$ to have the ciphertext Cc_j, performs the tag generation accordingly to produce Tc_j and returns (Cc_j, Tc_j) and the leakages to $\mathcal{A}^{\widetilde{\mathsf{E}}}$. The cost is similar to the non-challenge encryption queries.
- When $j = i$, it relays Mc_j^0 and Mc_j^1 to its eavesdropper challenger to obtain Cc_j^b and leakages leak_{enc} and $[\mathsf{leak}_{dec}]^{p-1}$, and then performs the tag generation accordingly to produce Tc_j and returns (Cc_j^b, Tc_j) to $\mathcal{A}$. Note that this means the relation $PTable((0^n, 66, 0^{n-8}), Nc_j) = S_0^{ch}$ is implicitly fixed, where S_0^{ch} is the secret key generated inside the eavesdropper challenger;
- When $j > i$, it simply encrypts Mc_j^1 and returns. The details are similar to the described case $j < i$.

– Upon $\mathcal{A}$ making the λ-th query to $\mathsf{L}_{\mathsf{decch}}(j)$ $(1 \le \lambda \le p-1)$,
  - When $j \ne i$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ performs the corresponding decryption and returns the obtained leakages to $\mathcal{A}$;
  - When $j = i$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ simply returns the λ-th leakage in the vector $[\mathsf{leak}_{dec}]^{p-1}$ as the answer.

It can be seen that the whole process is the same as either $\mathsf{G}_{3,i-1}$ or $\mathsf{G}_{3,i}$ depending on whether $b = 0$ or 1. By the remarks before, besides running $\mathcal{A}$, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ samples at most $2(q_m + q_e + q_d)$ random values (to emulate $\widetilde{\mathsf{P}}$) and internally processes $q_m + q_e + q_d - 1$ encryption/decryption queries (except for the query encrypted by the challenger). Therefore, $\mathcal{A}_2^{\widetilde{\mathsf{E}}}$ makes $Q = 3\sigma_m + \sigma_a + 6(q_e + q_d + q_m)$ additional queries to $\widetilde{\mathsf{E}}$, and evaluates the leakage functions for $2p\sigma_m$ times, resulting in $O(p\sigma_m t_l)$ added running time. Therefore, we have

$$\big|\Pr[\mathsf{G}_{3,i} \Rightarrow 1] - \Pr[\mathsf{G}_{3,i-1} \Rightarrow 1]\big| \le \mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(p, q_{\widetilde{\mathsf{E}}} + Q, O(t + p\sigma_m t_l), m_i).$$

This means

$$\begin{aligned}
\big|\Pr[\mathsf{G}_2^* \Rightarrow 1] - \Pr[\mathsf{G}_2 \Rightarrow 1]\big| &\le \big|\Pr[\mathsf{G}_{3,q_e} \Rightarrow 1] - \Pr[\mathsf{G}_{3,0} \Rightarrow 1]\big| \\
&\le \sum_{i=1}^{q_e} \Big(\big|\Pr[\mathsf{G}_{3,i} \Rightarrow 1] - \Pr[\mathsf{G}_{3,i-1} \Rightarrow 1]\big| \Big) \\
&\le \textstyle\sum_{i=1}^{q_e} \mathbf{Adv}^{\mathsf{eavl2}}_{\mathsf{RESM}}(p, q_{\widetilde{\mathsf{E}}} + Q, O(t + p\sigma_m t_l), m_i)
\end{aligned}$$

which is the claim in Eq. (30).