Security Evaluation of Romulus

Jooyoung Lee

KAIST, Daejeon, Korea
hicalf@kaist.ac.kr

Executive Summary

The Romulus Modes. Romulus is a submission to the NIST lightweight competition, currently in the final round [GIK+19]. Romulus consists of three authenticated encryption (AE) modes and one hash function, all using the Skinny tweakable block cipher as the underlying primitive. In particular, the AE modes include a nonce-based AE mode Romulus-N, a nonce misuse-resistant AE mode Romulus-M, and a leakage-resilient AE mode Romulus-T.

The goal of this report is to evaluate the security of Romulus-N and Romulus-M. Their security proof has already been given in [IKMP20], so we will revisit and verify their security claims. In [IKMP20], the authors proposed three versions for each of Romulus-N and Romulus-M, while Romulus-N and Romulus-M at the final round share the exact same specifications as Romulus-N1 and Romulus-M1 (from version 1.2 and [IKMP20]), respectively, except that the number of Skinny-128/384 rounds is reduced from 56 to 40.

The Scope of Evaluation. Our evaluation is focused on the provable security of Romulus-N and Romulus-M, where we assume the security of the underlying tweakable block cipher in the standard model. More precisely, when a key is chosen uniformly at random and kept secret, the keyed tweakable block cipher is assumed to behave like an independent random permutation for each tweak. Such an ideal counterpart of a tweakable block cipher is called a tweakable uniform random permutation.

The security of Romulus-N and Romulus-M is proved in the nonce-respecting model in terms of privacy and authenticity. For Romulus-M, we will also analyze its nonce-misuse resistance. In this case, its security bound depends on the maximum number of repetitions of a nonce.

Summary of Security Evaluation. In this evaluation, we confirm the provable security of Romulus-N and Romulus-M as given in the NIST submission version 1.3. Specifically, when the underlying n-bit tweakable cipher is modeled as a tweakable uniform random permutation, we confirm the following statements.

- Romulus-N is perfectly secure in terms of privacy as long as nonces are not repeated.
- Romulus-N is unforgeable up to 2^τ verification(decryption) queries as long as nonces are not repeated, where the tag is truncated to τ bits.
- Romulus-M is perfectly secure in terms of privacy as long as nonces are not repeated. The privacy of Romulus-M is also guaranteed when a nonce is repeated at most r times, and when $r\sigma_e$ is small in front of 2^n, where σ_e denotes the total number of effective blocks in encryption.
- Romulus-M is unforgeable up to 2^n verification(decryption) queries as long as nonces are not repeated. The authenticity of Romulus-M is also guaranteed

when a nonce is repeated at most r times, and when rq_e and rq_d are all small in front of 2^n, where q_e and q_d denote the number of encryption queries and the number of verification(decryption) queries, respectively.

In the original proof, the output of each block cipher evaluation is viewed as a random variable, and it is lazily sampled since each tweak defines an independent random permutation. In this way, the probability of certain bad events (such as collisions in chaining variables) which breaks the randomness of the ciphertexts and the tags is carefully upper bounded. On the other hand, our proof is mainly based on the H-coefficient technique, leading to slight improvement in the coefficients of the security bounds.

Conclusion. In this evaluation, we proved the security of Romulus-N and Romulus-M; the best attack on any of these modes implies a chosen-plaintext attack (CPA) in the single-key setting against the underlying tweakable block cipher. So unless the tweakable block cipher is broken by CPA adversaries in the single-key setting, Romulus indeed maintains the claimed n-bit security. To evaluate the security of Romulus, with the standard model proof, we can focus on the security evaluation of the underlying primitive. The provable security of Romulus-N and Romulus-M is a clear advantage over any scheme with security proofs in non-standard models.

Table of Contents

1 Preliminaries 5
1.1 Notation 5
1.2 Security Notions 5
2 Specification of the Romulus Modes 7
2.1 Romulus-N 9
2.2 Romulus-M 9
3 Security of Romulus-N in the Nonce-respecting Setting 10
3.1 Proof of Privacy 14
3.2 Proof of Authenticity 14
3.3 Summary 17
4 Security of Romulus-M in the Nonce-respecting Setting 18
4.1 Proof of Privacy 18
4.2 Proof of Authenticity 18
4.3 Summary 21
5 Security of Romulus-M in the Nonce-misuse Setting 22
5.1 Proof of Privacy 22
5.2 Proof of Authenticity 25
5.3 Summary 28

1 Preliminaries

1.1 Notation

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 an integer $n \geqslant 0$, let $\{0,1\}^n$ be the set of n-bit strings, and let $\{0,1\}^{\leqslant n} = \bigcup_{i=0,\ldots,n} \{0,1\}^i$, where $\{0,1\}^0 = \{\varepsilon\}$. Let $[\![n]\!] = \{1,\ldots,n\}$ and $[\![n]\!]_0 = \{0,1,\ldots,n-1\}$. Let $|X|_n = \max\{1, \lceil |X|/n \rceil\}$. For positive integers i and j, let $(i)_j$ denote $i \cdot (i-1) \cdot \cdots \cdot (i-(j-1))$.

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 $1 \,\|\, 0^i$. When $|X| = |Y|$, the bitwise XOR of X and Y is denoted by $X \oplus Y$. For a binary string X such that $|X| \geqslant x$, we write $\mathtt{lmt}_x(X)$ (resp. $\mathtt{rmt}_x(X)$) to denote the leftmost (resp. rightmost) x bits of X. By convention, for an integer $X \in [\![2^c]\!]_0$, we assume a standard integer-to-binary encoding, i.e., an integer $\sum_{i=0}^{c-1} x_i 2^i$ for $x_i \in \{0,1\}$ is encoded to $(x_{c-1} \ldots x_1 x_0) \in \{0,1\}^c$. For example, $X \oplus 1$ denotes $X \oplus 0^{c-1}1$.

Galois Field. An element a in the Galois field $\mathrm{GF}(2^n)$ will be interchangeably represented as an n-bit string $a_{n-1} \ldots a_1 a_0$, a formal polynomial $a_{n-1}\mathtt{x}^{n-1} + \cdots + a_1\mathtt{x} + a_0$, or an integer $\sum_{i=0}^{n-1} a_i 2^i$.

Matrix. Let G be an $n \times n$ binary matrix defined over $\mathrm{GF}(2)$. For $X \in \{0,1\}^n$, let $G(X)$ denote the matrix-vector multiplication over $\mathrm{GF}(2)$, where X is interpreted as a column vector. We may write $G \cdot X$ instead of $G(X)$. Let I denote the $n \times n$ identity matrix over $\mathrm{GF}(2)$.

1.2 Security Notions

Tweakable Block Cipher. A tweakable block cipher (TBC) is a keyed function $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$, where $\mathcal{K}$ is the key space, $\mathcal{T}$ is the tweak space, and $\mathcal{M} = \{0,1\}^n$ is the message space, such that for any $(K,T) \in \mathcal{K} \times \mathcal{T}$, $\widetilde{E}(K,T,\cdot)$ is a permutation over $\mathcal{M}$. We interchangeably write $\widetilde{E}(K,T,M)$ or $\widetilde{E}_K(T,M)$ or $\widetilde{E}_K^T(M)$.

The security of $\widetilde{E}$ is defined by the indistinguishability from an ideal object, called a tweakable uniform random permutation (TURP), denoted by $\widetilde{\mathsf{P}}$; it is a set of independent uniform random permutations (URPs) over $\mathcal{M}$ indexed by tweak $T \in \mathcal{T}$. We will consider an adversary $\mathcal{A}$ making only chosen-plaintext, chosen-tweak queries to $\widetilde{E}$ (since the encryption and the decryption algorithms of Romulus use only forward queries to the underlying tweakable block cipher). The advantage of $\mathcal{A}$ breaking the security of $\widetilde{E}$ is defined as

$$\mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left| \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\widetilde{E}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\widetilde{\mathsf{P}}(\cdot,\cdot)} \Rightarrow 1\right] \right|.$$

A (q,t)-adversary against the security of $\widetilde{E}$ is an algorithm making at most q encryption queries and running in time at most t. We define $\mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(q,t)$ as the maximum of $\mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(\mathcal{A})$ over all (q,t)-adversaries against $\widetilde{E}$.

Nonce-based AE Scheme. Given four non-empty sets $\mathcal{K}$, $\mathcal{N}$, $\mathcal{A}$ and $\mathcal{M}$ (all being subsets of $\{0,1\}^*$) and tag length τ, a nonce-based authenticated encryption (AE) scheme is a tuple

$$\Pi = (\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \Pi.\mathsf{Enc}, \Pi.\mathsf{Dec}),$$

where $\Pi.\mathsf{Enc}$ and $\Pi.\mathsf{Dec}$ are called the encryption and decryption algorithms of Π, respectively. The encryption algorithm $\Pi.\mathsf{Enc}$ takes as input a key $K \in \mathcal{K}$, a nonce $N \in \mathcal{N}$, an associated data $A \in \mathcal{A}$, and a message $M \in \mathcal{M}$, and outputs a ciphertext $C \in \{0,1\}^*$ and a tag $T \in \{0,1\}^\tau$ such that $|C| = |M|$. The decryption algorithm $\Pi.\mathsf{Dec}$ takes as input a tuple $(K, N, A, C, T) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \{0,1\}^* \times \{0,1\}^\tau$, and outputs either a message $M \in \mathcal{M}$ or a special symbol $\perp$. We require that

$$\Pi.\mathsf{Dec}(K, N, A, \Pi.\mathsf{Enc}(K, N, A, M)) = M$$

for any tuple $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$. We will write $\Pi.\mathsf{Enc}_K(N, A, M)$ and $\Pi.\mathsf{Dec}_K(N, A, C, T)$ to denote $\Pi.\mathsf{Enc}(K, N, A, M)$ and $\Pi.\mathsf{Dec}(K, N, A, C, T)$, respectively.

The privacy of Π is measured by the indistinguishability of the encryption oracle $\Pi.\mathsf{Enc}_K$ from Rand, where Rand returns an independent random string of length $|M| + \tau$ on any (distinct) input (N, M). The advantage of $\mathcal{A}$ breaking the privacy of Π is defined as

$$\mathbf{Adv}^{\mathtt{priv}}_{\Pi}(\mathcal{A}) \stackrel{\text{def}}{=} \left| \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\Pi.\mathsf{Enc}_K} \Rightarrow 1\right] - \Pr\left[\mathcal{A}^{\mathsf{Rand}} \Rightarrow 1\right] \right|.$$

When Π is based on a smaller primitive, say a tweakable block cipher, an (r, q, σ, t)-adversary against the privacy of Π is an algorithm making at most q encryption queries with at most r repetitions of a nonce, and running in time at most t, where the total number of effective blocks (i.e., the total number of tweakable block cipher calls) is at most σ. We define $\mathbf{Adv}^{\mathtt{priv}}_{\Pi}(r, q, \sigma, t)$ as the maximum of $\mathbf{Adv}^{\mathtt{priv}}_{\Pi}(\mathcal{A})$ over all (r, q, σ, t)-adversaries $\mathcal{A}$ against the privacy of Π.

The authenticity of Π is measured by the adversarial advantage of finding any successful forgery via queries to $\Pi.\mathsf{Enc}_K$ and $\Pi.\mathsf{Dec}_K$. Precisely, the advantage of $\mathcal{A}$ breaking the authenticity of Π is defined as

$$\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}) \stackrel{\text{def}}{=} \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\Pi.\mathsf{Enc}_K, \Pi.\mathsf{Dec}_K} \text{ forges } \right],$$

where $\mathcal{A}$ forges if it receives a value $M' \neq \perp$ from $\Pi.\mathsf{Dec}_K$. Here, to prevent trivial wins, if $(C, T) \leftarrow \Pi.\mathsf{Enc}_K(N, A, M)$ is obtained earlier, $\mathcal{A}$ cannot query (N, A, C, T) to $\Pi.\mathsf{Dec}_K$. An $(r, q_e, q_d, \sigma_e, \sigma_d, t)$-adversary against the authenticity of Π is an algorithm making at most q_e encryption queries with at most r

repetitions of a nonce and at most q_d decryption queries, and running in time at most t, where the total number of effective blocks in encryption (resp. decryption queries) is at most σ_e (resp. σ_d). We define $\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(r, q_e, q_d, \sigma_e, \sigma_d, t)$ as the maximum of $\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A})$ over all $(r, q_e, q_d, \sigma_e, \sigma_d, t)$-adversaries $\mathcal{A}$ against the authenticity of Π.

Nonce-respecting vs. Nonce-misuse Adversaries. When $r = 1$, we say that $\mathcal{A}$ (against either the privacy or the authenticity of Π) is *nonce-respecting*, otherwise $\mathcal{A}$ is said *nonce-misusing*. However, the adversary is allowed to repeat nonces when it makes decryption queries. When we consider only nonce-respecting adversaries, we will simply drop the parameter r, writing $\mathbf{Adv}^{\mathtt{priv}}_{\Pi}(q, \sigma, t)$ and $\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(q_e, q_d, \sigma_e, \sigma_d, t)$. Furthermore, when we consider information theoretic security, we will drop the parameter t.

MAC-Security Against a Single Verification Query. When we analyze the authenticity of a nonce-based AE schem Π, we can assume $q_d = 1$, and use the following lemma.

Lemma 1. *If there exists a function δ of q_e, σ, t, and potentially r, such that, for any $((r), q_e, 1, \sigma, t)$-adversary $\mathcal{A}$ against Π,*

$$\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}) \leqslant \delta((r), q_e, \sigma, t),$$

then, for any $((r), q_e, q_d, \sigma, t)$-adversary $\mathcal{A}'$ against Π, one has

$$\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}') \leqslant q_d \cdot \delta((r), q_e, \sigma, t).$$

Proof. Given a $((r), q_e, q_d, \sigma, t)$-adversary $\mathcal{A}'$ against Π, one can use it as a subroutine to construct a $((r), q_e, 1, \sigma, t)$-adversary $\mathcal{A}$ against Π as follows:

- $\mathcal{A}$ chooses j uniformly at random from $\{1, \ldots, q_d\}$;
- $\mathcal{A}$ faithfully relays each encryption query made by $\mathcal{A}'$ to its encryption oracle; if $\mathcal{A}$ receives (C, T) from the oracle as the answer to this query, then $\mathcal{A}$ sends (C, T) to $\mathcal{A}'$;
- $\mathcal{A}$ relays $\mathcal{A}'$'s j-th decryption query (N', M', C', T') to its decryption oracle, and for any other decryption query, $\mathcal{A}$ sends $\perp$ to $\mathcal{A}'$.

Then it is easy to see that

$$\mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}) \geqslant \frac{1}{q_d} \cdot \mathbf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}'). \qquad \square$$

2 Specification of the Romulus Modes

For the description of Romulus-N and Romulus-M, we will use the following parameters: n for nonce length and message block length, k for key length, d

for counter bit length, and τ for tag length, where n is a multiple of 8. While we fix $\tau = n$, a tag forRomulus-N can be truncated if needed, at the cost of decreased security against forgery. As actual values for these parameters, the Romulus modes use $n = k = 128$ and $d = 56$. Both of Romulus-N and Romulus-M are based on a tweakable block cipher and a state update function ρ.

Padding. For Romulus-N and Romulus-M, AD blocks and message blocks are both of length multiples of 8. For $X \in \{0,1\}^{\leqslant l}$ of length multiple of 8 (i.e, byte string), let

$$\mathtt{pad}_l(X) = \begin{cases} X & \text{if } |X| = l, \\ X \,\|\, 0^{l-|X|-8} \,\|\, \mathtt{len}_8(X), & \text{if } 0 \leqslant |X| < l, \end{cases}$$

where $\mathtt{len}_8(X)$ denotes the one-byte encoding of the byte-length of X, assuming that $l < 256$ bytes. Here, $\mathtt{pad}_l(\varepsilon) = 0^l$. When $l = 128$ (as used for Romulus-N and Romulus-M), $\mathtt{len}_8(X)$ has 16 variations (i.e., byte length 0 to 15), and it is encoded to the last 4 bits of $\mathtt{len}_8(X)$ (for example, $\mathtt{len}_8(11) = \mathtt{00001011}$).

Parsing. For $X \in \{0,1\}^*$, 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 $x = |X|_n$. Note that $|X[x]| < n$ if $|X|$ is not a multiple of n. When $X = \varepsilon$, we have $X[1] \xleftarrow{n} X$ and $X[1] = \varepsilon$. Note in particular that $|\varepsilon|_n = 1$.

Tweakable Block Cipher. The Romulus modes are based on a tweakable block cipher

$$\widetilde{E} : \mathcal{K} \times \overline{\mathcal{T}} \times \mathcal{M} \to \mathcal{M}$$

where $\mathcal{K} = \{0,1\}^k$, $\mathcal{M} = \{0,1\}^n$, and $\overline{\mathcal{T}} = \mathcal{T} \times \mathcal{B} \times \mathcal{D}$. Here, $\mathcal{T} = \{0,1\}^n$, $\mathcal{D} = [\![2^d - 1]\!]_0$, and $\mathcal{B} = [\![256]\!]_0$. $\mathcal{T}$ will be used to process nonces or AD blocks, $\mathcal{D}$ will be used for counters, while $\mathcal{B}$ is for domain separation. For a counter value $i \in \mathcal{D}$, we will write $\bar{i}$ to denote the i-th clocking of the counter as a part of the tweak (e.g. see Figure 1).

State Update Function. For an $n \times n$ binary matrix A and for $i = 0, \ldots, n$, let $A^{(i)}$ denote an $n \times n$ matrix that is equal to A except the $(i+1)$-th to n-th rows, which are set to all zero. So $A^{(0)}$ is the zero matrix and $A^{(n)} = A$. When n is a multiple of 8, A is called *sound* if A is regular (full-rank) and $A^{(i)} + I$ is regular for all $i = 8, 16, \ldots, n$.

The state update function ρ is defined using the following sound matrix.

$$G = \begin{pmatrix} G_s & 0 & 0 & \ldots & 0 \\ 0 & G_s & 0 & \ldots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & \ldots & 0 & G_s & 0 \\ 0 & \ldots & 0 & 0 & G_s \end{pmatrix},$$

where 0 represents the 8×8 zero matrix, and

$$G_s = \begin{pmatrix} 0\,1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,0\,1 \\ 1\,0\,0\,0\,0\,0\,0\,1 \end{pmatrix}.$$

It is easy to see that G is sound since G_s is regular.

The state update function $\rho : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ is defined as

$$\rho(S, M) = (S', C),$$

where $C = M \oplus G(S)$ and $S' = S \oplus M$. As the inverse of ρ with respect to its second parameter, $\rho^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ is defined as

$$\rho^{-1}(S, C) = (S', M),$$

where $M = C \oplus G(S)$ and $S' = S \oplus M$. So, for any $(S, M) \in \{0,1\}^n \times \{0,1\}^n$, if $\rho(S, M) = (S', C)$, then $\rho^{-1}(S, C) = (S', M)$.

2.1 Romulus-N

The specification of the NAE mode Romulus-N is shown in Figure 1, while Figure 3 gives a more graphical representation. To encrypt (N, A, M) under key K, in Romulus-N, we first hash $A = (A[1], \ldots, A[a])$ into S in line 7 in Figure 1, where $A[1], A[3], A[5], \ldots$ are injected into the state with ρ, and $A[2], A[4], A[6], \ldots$ are processed with the TBC. We then use nonce N to compute $S = \widetilde{E}_K^{(N, w_A, \overline{a})}(S)$, that could be seen as the nonce-dependent MAC value of A. Then M is processed with ρ to generate C, where we keep using the TBC that takes N as a part of the input. The tag T is generated as $T = G(\widetilde{E}_K^{(N, w_M, \overline{m})}(S))$ from the final state S after the process of M.

2.2 Romulus-M

The specification of the MRAE mode Romulus-M is shown in Figure 2; we first hash both A and M into the state S in line 19 or 21, and then the tag T is computed as $T = G(\widetilde{E}_K^{(N, w, \overline{a+m})}(S))$ in line 23 (or 24). The tag generation follows the same process as Romulus-N to hash both A and M, and the encryption part of M is similar to Romulus-N. Figure 4 gives a more graphical representation of Romulus-M.

Algorithm Romulus-N.Enc$_K(N, A, M)$

1. $S \leftarrow 0^n$
2. $(A[1], \ldots, A[a]) \xleftarrow{n} A$
3. **if** $|A[a]| < n$ **then** $w_A \leftarrow 26$ **else** 24
4. $A[a] \leftarrow \mathtt{pad}_n(A[a])$
5. **for** $i = 1$ **to** $\lfloor a/2 \rfloor$
6. $\quad (S, \eta) \leftarrow \rho(S, A[2i-1])$
7. $\quad S \leftarrow \widetilde{E}_K^{(A[2i], 8, \overline{2i-1})}(S)$
8. **end for**
9. **if** $a \bmod 2 = 0$ **then** $V \leftarrow 0^n$ **else** $A[a]$
10. $(S, \eta) \leftarrow \rho(S, V)$
11. $S \leftarrow \widetilde{E}_K^{(N, w_A, \overline{a})}(S)$
12. $(M[1], \ldots, M[m]) \xleftarrow{n} M$
13. **if** $|M[m]| < n$ **then** $w_M \leftarrow 21$ **else** 20
14. **for** $i = 1$ **to** $m - 1$
15. $\quad (S, C[i]) \leftarrow \rho(S, M[i])$
16. $\quad S \leftarrow \widetilde{E}_K^{(N, 4, \overline{i})}(S)$
17. **end for**
18. $M'[m] \leftarrow \mathtt{pad}_n(M[m])$
19. $(S, C'[m]) \leftarrow \rho(S, M'[m])$
20. $C[m] \leftarrow \mathtt{lmt}_{|M[m]|}(C'[m])$
21. $S \leftarrow \widetilde{E}_K^{(N, w_M, \overline{m})}(S)$
22. $(\eta, T) \leftarrow \rho(S, 0^n)$
23. $C \leftarrow C[1] \,\|\, \ldots \,\|\, C[m-1] \,\|\, C[m]$
24. **return** (C, T)

Algorithm Romulus-N.Dec$_K(N, A, C, T)$

1. $S \leftarrow 0^n$
2. $(A[1], \ldots, A[a]) \xleftarrow{n} A$
3. **if** $|A[a]| < n$ **then** $w_A \leftarrow 26$ **else** 24
4. $A[a] \leftarrow \mathtt{pad}_n(A[a])$
5. **for** $i = 1$ **to** $\lfloor a/2 \rfloor$
6. $\quad (S, \eta) \leftarrow \rho(S, A[2i-1])$
7. $\quad S \leftarrow \widetilde{E}_K^{(A[2i], 8, \overline{2i-1})}(S)$
8. **end for**
9. **if** $a \bmod 2 = 0$ **then** $V \leftarrow 0^n$ **else** $A[a]$
10. $(S, \eta) \leftarrow \rho(S, V)$
11. $S \leftarrow \widetilde{E}_K^{(N, w_A, \overline{a})}(S)$
12. $(C[1], \ldots, C[m]) \xleftarrow{n} C$
13. **if** $|C[m]| < n$ **then** $w_C \leftarrow 21$ **else** 20
14. **for** $i = 1$ **to** $m - 1$
15. $\quad (S, M[i]) \leftarrow \rho^{-1}(S, C[i])$
16. $\quad S \leftarrow \widetilde{E}_K^{(N, 4, \overline{i})}(S)$
17. **end for**
18. $\widetilde{S} \leftarrow (0^{|C[m]|} \,\|\, \mathtt{rmt}_{n-|C[m]|}(G(S)))$
19. $C'[m] \leftarrow \mathtt{pad}_n(C[m]) \oplus \widetilde{S}$
20. $(S, M'[m]) \leftarrow \rho^{-1}(S, C'[m])$
21. $M[m] \leftarrow \mathtt{lmt}_{|C[m]|}(M'[m])$
22. $S \leftarrow \widetilde{E}_K^{(N, w_C, \overline{m})}(S)$
23. $(\eta, T^*) \leftarrow \rho(S, 0^n)$
24. $M \leftarrow M[1] \,\|\, \ldots \,\|\, M[m-1] \,\|\, M[m]$
25. **if** $T^* = T$ **then return** M **else** $\perp$

Algorithm $\rho(S, M)$

1. $C \leftarrow M \oplus G(S)$
2. $S' \leftarrow S \oplus M$
3. **return** (S', C)

Algorithm $\rho^{-1}(S, C)$

1. $M \leftarrow C \oplus G(S)$
2. $S' \leftarrow S \oplus M$
3. **return** (S', M)

Fig. 1: The Romulus-N nonce-based AE mode. Lines of [**if (statement) then** $X \leftarrow x$ **else** x'] are shorthand for [**if (statement) then** $X \leftarrow x$ **else** $X \leftarrow x'$]. The dummy variable η is always discarded.

3 Security of Romulus-N in the Nonce-respecting Setting

In this section, we prove the security of Romulus-N against nonce-respecting adversaries. Let Romulus*-N denote the AE mode obtained from Romulus-N by replacing the underlying keyed tweakable block cipher $\widetilde{E}$ by a truly random tweakable permutation

$$\widetilde{\mathsf{P}} : \overline{\mathcal{T}} \times \mathcal{M} \to \mathcal{M}$$

with $\overline{\mathcal{T}} = \mathcal{T} \times \mathcal{B} \times \mathcal{D}$. So one can view $\widetilde{\mathsf{P}}$ itself as the secret key of Romulus*-N. From now on, we will focus on the security of Romulus*-N.

For a fixed parameter τ such that $1 \leqslant \tau \leqslant n - 1$, let

$$\mathsf{trunc}_\tau : \{0,1\}^n \to \{0,1\}^\tau$$

be a function that takes τ bits of the input in any way (e.g., the leftmost τ bits of an n-bit input). Let

$$\rho_C^{-1}(S) \stackrel{\text{def}}{=} S \oplus \mathtt{pad}_n(C) \oplus \mathtt{lmt}_{|C|}(G(S)) \,\|\, 0^{n-|C|}$$

Algorithm Romulus-M.Enc$_K(N, A, M)$

1. $S \leftarrow 0^n$
2. $(X[1], \ldots, X[a]) \xleftarrow{n} A$
3. $(X[a+1], \ldots, X[a+m]) \xleftarrow{n} M$
4. $z \leftarrow |X[a+m]|$
5. $w \leftarrow 48$
6. **if** $|X[a]| < n$ **then** $w \leftarrow w \oplus 2$
7. **if** $|X[a+m]| < n$ **then** $w \leftarrow w \oplus 1$
8. **if** $a \bmod 2 = 0$ **then** $w \leftarrow w \oplus 8$
9. **if** $m \bmod 2 = 0$ **then** $w \leftarrow w \oplus 4$
10. $X[a] \leftarrow \mathtt{pad}_n(X[a])$
11. $X[a+m] \leftarrow \mathtt{pad}_n(X[a+m])$
12. $x \leftarrow 40$
13. **for** $i = 1$ **to** $\lfloor (a+m)/2 \rfloor$
14. $\quad (S, \eta) \leftarrow \rho(S, X[2i-1])$
15. $\quad$ **if** $i = \lfloor a/2 \rfloor + 1$ **then** $x \leftarrow x \oplus 4$
16. $\quad S \leftarrow \widetilde{E}_K^{(X[2i], x, \overline{2i-1})}(S)$
17. **end for**
18. **if** $a \bmod 2 = m \bmod 2$ **then**
19. $\quad (S, \eta) \leftarrow \rho(S, 0^n)$
20. **else**
21. $\quad (S, \eta) \leftarrow \rho(S, X[a+m])$
22. $S \leftarrow \widetilde{E}_K^{(N, w, \overline{a+m})}(S)$
23. $(\eta, T) \leftarrow \rho(S, 0^n)$
24. **if** $M = \epsilon$ **then return** (ϵ, T)
25. $S \leftarrow T$
26. **for** $i = 1$ **to** m
27. $\quad S \leftarrow \widetilde{E}_K^{(N, 36, \overline{i-1})}(S)$
28. $\quad (S, C[i]) \leftarrow \rho(S, X[a+i])$
29. **end for**
30. $C[m] \leftarrow \mathtt{lmt}_z(C[m])$
31. $C \leftarrow C[1] \,\|\, \ldots \,\|\, C[m-1] \,\|\, C[m]$
32. **return** (C, T)

Algorithm Romulus-M.Dec$_K(N, A, C, T)$

1. **if** $C = \epsilon$ **then** $M \leftarrow \epsilon$
2. **else**
3. $\quad S \leftarrow T$
4. $\quad (C[1], \ldots, C[m]) \xleftarrow{n} C$
5. $\quad z \leftarrow |C[m]|$
6. $\quad C[m] \leftarrow \mathtt{pad}_n(C[m])$
7. $\quad$ **for** $i = 1$ **to** m
8. $\quad\quad S \leftarrow \widetilde{E}_K^{(N, 36, \overline{i-1})}(S)$
9. $\quad\quad (S, M[i]) \leftarrow \rho^{-1}(S, C[i])$
10. $\quad$ **end for**
11. $\quad M[m] \leftarrow \mathtt{lmt}_z(M[m])$
12. $\quad M \leftarrow M[1] \,\|\, \ldots \,\|\, M[m-1] \,\|\, M[m]$
13. $S \leftarrow 0^n$
14. $(X[1], \ldots, X[a]) \xleftarrow{n} A$
15. $(X[a+1], \ldots, X[a+m]) \xleftarrow{n} M$
16. $w \leftarrow 48$
17. **if** $|X[a]| < n$ **then** $w \leftarrow w \oplus 2$
18. **if** $|X[a+m]| < n$ **then** $w \leftarrow w \oplus 1$
19. **if** $a \bmod 2 = 0$ **then** $w \leftarrow w \oplus 8$
20. **if** $m \bmod 2 = 0$ **then** $w \leftarrow w \oplus 4$
21. $X[a] \leftarrow \mathtt{pad}_n(X[a])$
22. $X[a+m] \leftarrow \mathtt{pad}_n(X[a+m])$
23. $x \leftarrow 40$
24. **for** $i = 1$ **to** $\lfloor (a+m)/2 \rfloor$
25. $\quad (S, \eta) \leftarrow \rho(S, X[2i-1])$
26. $\quad$ **if** $i = \lfloor a/2 \rfloor + 1$ **then** $x \leftarrow x \oplus 4$
27. $\quad S \leftarrow \widetilde{E}_K^{(X[2i], x, \overline{2i-1})}(S)$
28. **end for**
29. **if** $a \bmod 2 = m \bmod 2$ **then**
30. $\quad (S, \eta) \leftarrow \rho(S, 0^n)$
31. **else**
32. $\quad (S, \eta) \leftarrow \rho(S, X[a+m])$
33. $S \leftarrow \widetilde{E}_K^{(N, w, \overline{a+m})}(S)$
34. $(\eta, T^*) \leftarrow \rho(S, 0^n)$
35. **if** $T^* = T$ **then return** M **else** $\perp$

Algorithm $\rho(S, M)$

1. $C \leftarrow M \oplus G(S)$
2. $S' \leftarrow S \oplus M$
3. **return** (S', C)

Algorithm $\rho^{-1}(S, C)$

1. $M \leftarrow C \oplus G(S)$
2. $S' \leftarrow S \oplus M$
3. **return** (S', M)

Fig. 2: The Romulus-M misuse-resistant AE mode. The dummy variable η is always discarded. Note that in the case of empty message, no encryption call has to be performed in the encryption part.

for $C \in \{0,1\}^{\leqslant n}$. In particular, $\rho_C^{-1}(S) = S \oplus C \oplus G(S)$ for $C \in \{0,1\}^n$. Since G is sound, ρ_C^{-1} is a permutation over $\{0,1\}^n$ for any $C \in \{0,1\}^{\leqslant n}$. We also define

$$\rho_C(S) \stackrel{\text{def}}{=} S \oplus C$$

for $C \in \{0,1\}^n$.

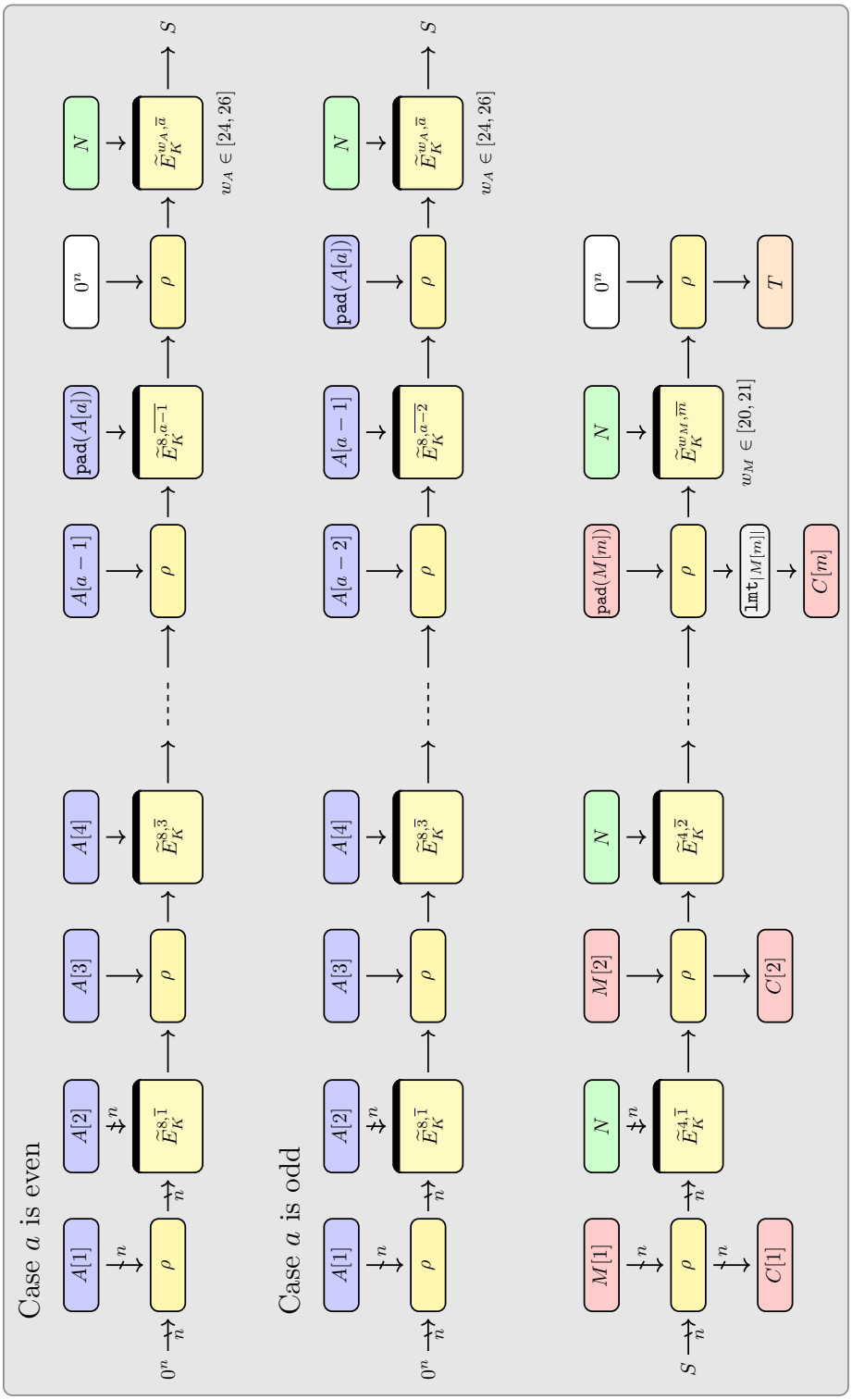

Fig. 3: The Romulus-N nonce-based AE mode. (Top) process of AD with an even number of AD blocks. (Middle) process of AD with an odd number of AD blocks. (Bottom) Encryption.

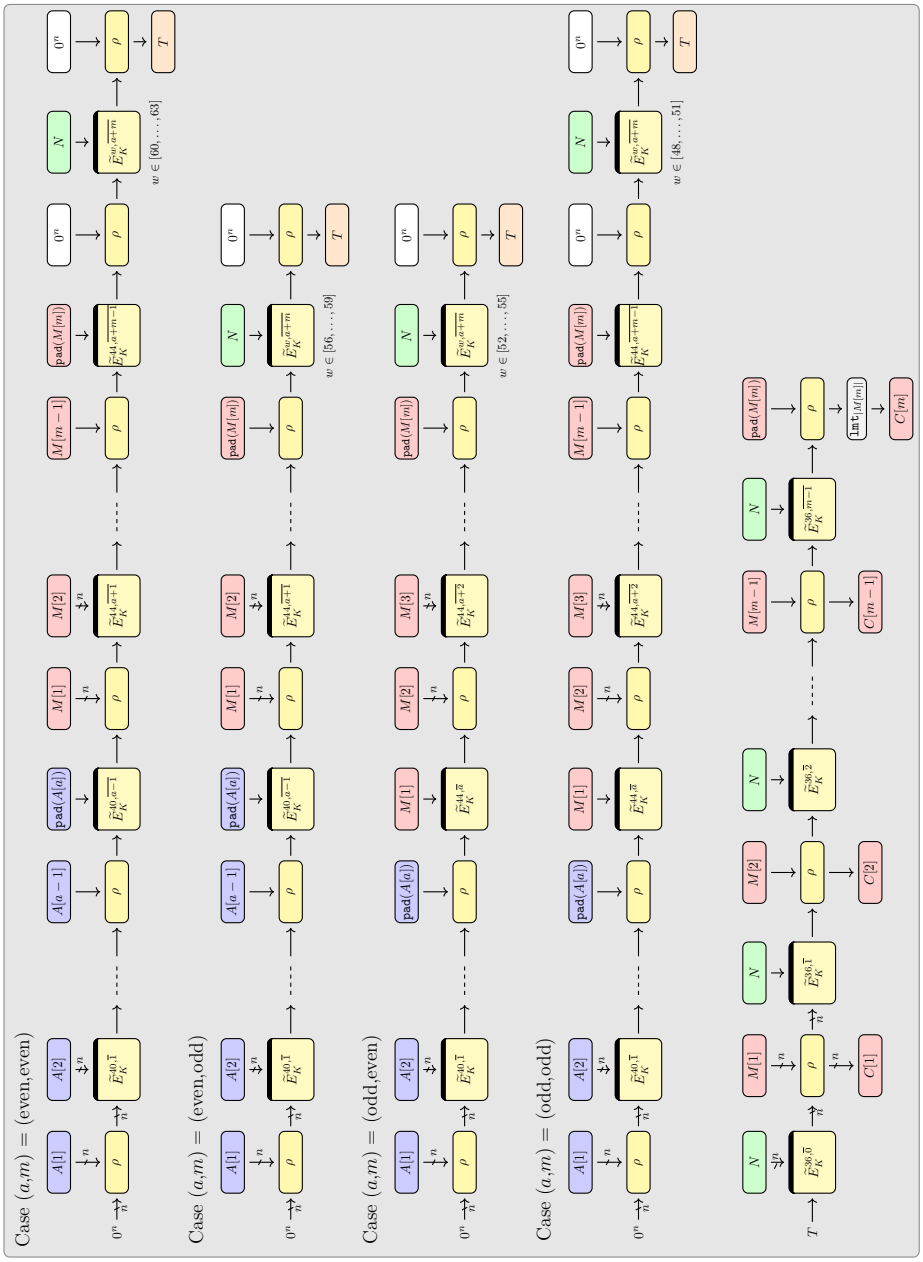

Fig. 4: The Romulus-M misuse-resistant AE mode. (Top) process of AD with an even/even, even/odd, odd/even, odd/odd number of AD blocks and M blocks, respectively. (Bottom) Encryption.

3.1 Proof of Privacy

The i-th ciphertext block of an encryption query with nonce N is defined as $C[i] = M[i] \oplus G(S)$ with the last block truncated if needed, and the tag is defined as $T = G(S)$ for some S, where G is regular and $S = \widetilde{\mathsf{P}}^{N,x,y}(S')$ for some x, y and S'. If the adversary is nonce-respecting, then tweaks (x, y, N) are all distinct throughout the game. If $\widetilde{\mathsf{P}}$ is evaluated by lazy sampling, then ciphertext blocks $C[i]$ and tag T will be all uniform and independent at random (no matter how many queries are made).[1] Therefore, for any adversary $\mathcal{A}$ against the privacy of Romulus⋆-N, we have

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}^\star\text{-}\mathsf{N}}(\mathcal{A}) = 0.$$

3.2 Proof of Authenticity

We will assume that $q_d = 1$, and then use Lemma 1. Furthermore, without loss of generality, we can assume that the single verification query is made at the end of the game (after all the encryption queries have been made). Given a $(1, q_e, 1, \sigma, t)$-adversary $\mathcal{A}$ against the authenticity of an AE mode Romulus⋆-N, we can slightly modify it to obtain a $(1, q_e, 1, \sigma, t)$-adversary $\mathcal{B}$ distinguishing the real world $(\mathsf{Romulus}^\star\text{-}\mathsf{N}.\mathsf{Enc}_{\widetilde{\mathsf{P}}}, \mathsf{Romulus}^\star\text{-}\mathsf{N}.\mathsf{Dec}_{\widetilde{\mathsf{P}}})$ and the ideal world $(\mathsf{Rand}, \mathsf{Rej})$ such that

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{N}}(\mathcal{A}) \leqslant \left|\Pr\left[\mathcal{B}^{\mathsf{Romulus}^\star\text{-}\mathsf{N}.\mathsf{Enc}_{\widetilde{\mathsf{P}}},\mathsf{Romulus}^\star\text{-}\mathsf{N}.\mathsf{Dec}_{\widetilde{\mathsf{P}}}} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\mathsf{Rand},\mathsf{Rej}} \Rightarrow 1\right]\right|,$$

where Rand returns an independent random string of length $|M| + \tau$ on any (distinct) input (N, M) and Rej returns $\perp$ for the (single) decryption query. In order to upper bound $\mathcal{B}$'s distinguishing advantage, we can use Patarin's coefficient-H technique [Pat08]. At the end of the game, $\mathcal{B}$ will have sets of query-response pairs

$$\begin{cases} \tau_e = ((N_1, A_1, M_1, C_1, T_1), \ldots, (N_{q_e}, A_{q_e}, M_{q_e}, C_{q_e}, T_{q_e})), \\ \tau_v = (N', A', M', T', b'), \end{cases}$$

where $b' \in \{\top, \perp\}$, and it holds that $b' = \perp$ in the ideal world. In the real world, $\mathcal{B}$ is given additional information $\widetilde{\mathsf{P}}^{(\cdot,8,\cdot)}(\cdot)$ (used to encrypt associate data) for free. In the ideal world, $\mathcal{B}$ is given a uniform tweakable permutation $\widetilde{\mathsf{P}}^{(\cdot,8,\cdot)}(\cdot)$ which is independent of Rand and Rej. Overall, $\mathcal{B}$ obtains a *transcript* $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,8,\cdot)})$.

A transcript τ is called *attainable* if the probability to obtain the transcript in the ideal world is non-zero, and let Θ be the set of all attainable transcripts. We let Θ_{ideal} and Θ_{real} denote the probability distributions of the transcript in the ideal world and in the real world, respectively. Based on these notations, we restate the Coefficient-H technique [Pat08] as follows.

[1] There will be an inherent limitation on the number of possible queries due to the tweak space and the maximum input length.

Lemma 2. *For a distinguisher $\mathcal{B}$, let $\Theta = \mathsf{GoodT} \cup \mathsf{BadT}$ be a partition of the set of all attainable transcripts. If there exist ϵ_1 and ϵ_2 such that for any $\tau \in \mathsf{GoodT}$,*

$$\frac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} \geqslant 1 - \epsilon_1,$$

and $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{BadT}] \leqslant \epsilon_2$, then the distinguishing advantage of $\mathcal{B}$ is bounded by $\epsilon_1 + \epsilon_2$.

Since $\widetilde{\mathsf{P}}^{(\cdot,8,\cdot)}$ is public, one can determine the input to $\widetilde{\mathsf{P}}^{(N_i,w_{A_i},a_i)}$ (resp. $\widetilde{\mathsf{P}}^{(N',w_{A'},a')}$), denoted X_i (resp. X'), where a_i (resp. a') denotes the number of AD blocks for the i-th encryption query (resp. the unique decryption query). We are now ready to define a bad transcript. A transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,8,\cdot)})$ is defined to be *bad* if

$$(X_i, N_i, w_{A_i}, a_i, C_i) = (X', N', w_{A'}, a', C')$$

for some $i = 1, \ldots, q_e$. Otherwise τ is called *good*. We first show that, in the ideal world, the probability of obtaining a bad transcript is small.

Lemma 3. $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{BadT}] \leqslant \dfrac{2}{2^n}$.

Proof. Since $\mathcal{B}$ is nonce-respecting, there is a unique index i such that

$$(N_i, w_{A_i}, a_i, C_i) = (N', w_{A'}, a', C').$$

Furthermore, we have $A_i \neq A'$ since $C_i = C'$ and $\mathcal{B}$ does not make a redundant query. Suppose that a' is even. Note that

$$X_i = \widetilde{\mathsf{P}}^{(\mathtt{pad}(A_i[a']),8,a'-1)} \circ \rho_{A_i[a'-1]} \circ \cdots \circ \widetilde{\mathsf{P}}^{(A_i[2],8,1)} \circ \rho_{A_i[1]}(0^n),$$
$$X' = \widetilde{\mathsf{P}}^{(\mathtt{pad}(A'[a']),8,a'-1)} \circ \rho_{A'[a'-1]} \circ \cdots \circ \widetilde{\mathsf{P}}^{(A'[2],8,1)} \circ \rho_{A'[1]}(0^n).$$

Let k denote the last index where the AD blocks of A_i and A' are different. So $A_i[k] \neq A'[k]$ while $A_i[j] = A'[j]$ for $j = k+1, \ldots, a'(= a_i)$. In order for the collision $X_i = X'$ to happen, it should be the case that $k > 1$, and one of the following two cases should hold (according to the parity of k).

Case 1: When k is even,

$$\widetilde{\mathsf{P}}^{(A_i[k],8,k-1)}(S_i) = \widetilde{\mathsf{P}}^{(A'[k],8,k-1)}(S')$$

for some (not necessarily distinct) S_i and S'. (When $k = a'$, $A_i[k]$ and $A'[k]$ should be replaced by $\mathtt{pad}(A_i[k])$ and $\mathtt{pad}(A'[k])$, respectively.)

Case 2: When k is odd,

$$\widetilde{\mathsf{P}}^{(A_i[k-1],8,k-2)}(S_i) \oplus \widetilde{\mathsf{P}}^{(A'[k-1],8,k-2)}(S') = A_i[k] \oplus A'[k] \ (\neq 0^n)$$

for some (not necessarily distinct) S_i and S'.

The first case happens with probability $\frac{1}{2^n}$, while the second case happens with probability $\frac{1}{2^n-1}$ over the randomness of $\widetilde{\mathsf{P}}$ in the ideal world. By applying a similar argument to the case that a' is odd, we have

$$\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{BadT}] \leqslant \frac{1}{2^n-1} \leqslant \frac{2}{2^n}. \qquad \square$$

We next show that the ratio of the interpolation probabilities is close to one.

Lemma 4. $\dfrac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} \geqslant 1 - \dfrac{2}{2^\tau}.$

Proof. We fix a good transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,8,\cdot)})$. An important observation is that the probabilities of obtaining $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,8,\cdot)})$ are the same in the ideal and the real worlds; every tweak of $\widetilde{\mathsf{P}}$ used in the message encryption part, say $(N_i, \star, j)$ with $\star \neq 8$, is fresh, and one can evaluate $\widetilde{\mathsf{P}}^{(N_i,\star,j)}$ by lazy sampling. In this way, all the ciphertext blocks and tags will be chosen independently at random. Now we will upper bound the probability that $b' = \top$ in the real world given $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,8,\cdot)})$.

We write $(C'[1], \ldots, C'[m']) \xleftarrow{n} C'$ for some m', and for $j = 1, \ldots, m'$, let

$$\mathsf{D}_j \stackrel{\mathrm{def}}{=} \rho^{-1}_{C'[m']} \circ \cdots \circ \widetilde{\mathsf{P}}^{(N',4,j+1)} \circ \rho^{-1}_{C'[j+1]} \circ \widetilde{\mathsf{P}}^{(N',4,j)} \circ \rho^{-1}_{C'[j]}.$$

We need to upper bound the probability that

$$\mathsf{trunc}_\tau \circ G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')} \circ \mathsf{D}_1 \circ \widetilde{\mathsf{P}}^{(N',w_{A'},a')}(X') = T'.$$

We distinguish two cases as follows.

Case 1: $(N', X', w_{A'}, a') \neq (N_i, X_i, w_{A_i}, a_i,)$ for all $i = 1, \ldots, q_e$. Once we fix all the evaluations of $\widetilde{\mathsf{P}}^{(N',4,\cdot)}$ (appearing in D_1) and $\widetilde{\mathsf{P}}^{(N',w_{C'},m')}$, the number of preimages of T' under $\mathsf{trunc}_\tau \circ G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')} \circ \mathsf{D}_1$ is $2^{n-\tau}$ since $G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')} \circ \mathsf{D}_1$ is a permutation. Since $\widetilde{\mathsf{P}}^{(N',w_{A'},a')}(\cdot)$ has been evaluated at most once during the encryption phase, the probability that $\mathsf{trunc}_\tau \circ G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')} \circ \mathsf{D}_1 \circ \widetilde{\mathsf{P}}^{(N',w_{A'},a')}(X') = T'$ is upper bounded by $\frac{2^{n-\tau}}{2^n-1} \leqslant \frac{2}{2^\tau}$.

Case 2: $(N', X', w_{A'}, a') = (N_i, X_i, w_{A_i}, a_i,)$ and $C_i \neq C$ for some $1 \leqslant i \leqslant q_e$ (which is unique).

Case 2-1: $(w_{C'}, c') = (w_{C_i}, c_i)$. Let k denote the first index where the ciphertext blocks of C_i and C' are different. So $C_i[k] \neq C'[k]$ while $C_i[j] = C'[j]$ for $j = 1, \ldots, k-1$. Then two inputs to $\widetilde{\mathsf{P}}^{(N',4,k)}$, say S_i and S', are different for the i-th query and the decryption query. Once

we fix $\widetilde{\mathsf{P}}^{(N',4,j)}(\cdot)$ for $j \geqslant k+1$, then the number of preimages of T' under $\mathsf{trunc}_\tau \circ G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')} \circ \mathsf{D}_{k+1}$ is $2^{n-\tau}$. Over the random sampling of $\widetilde{\mathsf{P}}^{(N',4,k)}(S')$, the probability that it becomes one of the preimages is upper bounded by $\frac{2^{n-\tau}}{2^n-1} \leqslant \frac{2}{2^\tau}$.

Case 2-2: $(w_{C'}, c') \neq (w_{C_i}, c_i)$. Once we fix all the evaluations of $\widetilde{\mathsf{P}}^{(N',4,\cdot)}$ and $\widetilde{\mathsf{P}}^{(N',w_{A'},a')}$, we can fix

$$X'' \stackrel{\text{def}}{=} \mathsf{D}_1 \circ \widetilde{\mathsf{P}}^{(N',w_{A'},a')}(X'),$$

while the number of preimages of T' under $\mathsf{trunc}_\tau \circ G$ is $2^{n-\tau}$. Since $\widetilde{\mathsf{P}}^{(N',w_{C'},c')}(\cdot)$ has never been evaluated before, the probability that $\mathsf{trunc}_\tau \circ G \circ \widetilde{\mathsf{P}}^{(N',w_{C'},m')}(X'') = T'$ is upper bounded by $\frac{2^{n-\tau}}{2^n} \leqslant \frac{1}{2^\tau}$.

Therefore, the probability that $b' = \perp$ in the real world is lower bounded by $1 - \frac{2}{2^\tau}$, which completes the proof. □

By Lemma 1, 2, 3 and 4, we have

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{N}}(q_e, q_d, \sigma_e, \sigma_d, t) \leqslant \frac{2q_d}{2^n} + \frac{2q_d}{2^\tau}.$$

3.3 Summary

Any (q_e, σ_e, t)-adversary $\mathcal{A}$ against the privacy of Romulus-N can be viewed as an adversary against Romulus*-N such that

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus\text{-}N}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}^\star\text{-}\mathsf{N}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e, t + O(\sigma_e)).$$

Similarly, any $(q_e, q_d, \sigma_e, \sigma_d, t)$-adversary $\mathcal{A}$ against the authenticity of Romulus-N can be viewed as an adversary against Romulus*-N such that

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus\text{-}N}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{N}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d)).$$

With this observation, we obtain the following theorem.

Theorem 1. *For* Romulus-N*, we have*

$$\begin{aligned}
\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus\text{-}N}}(q_e, \sigma_e, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e, t + O(\sigma_e)), \\
\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus\text{-}N}}(q_e, q_d, \sigma_e, \sigma_d, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d)) \\
&\quad + \frac{2q_d}{2^n} + \frac{2q_d}{2^\tau}.
\end{aligned}$$

4 Security of Romulus-M in the Nonce-respecting Setting

In this section, we prove the security of Romulus-M against nonce-respecting adversaries. Let Romulus*-M denote the AE mode obtained from Romulus-M by replacing the underlying keyed tweakable block cipher $\widetilde{E}$ by a truly random tweakable permutation

$$\widetilde{\mathsf{P}} : \overline{\mathcal{T}} \times \mathcal{M} \to \mathcal{M}$$

with $\overline{\mathcal{T}} = \mathcal{T} \times \mathcal{B} \times \mathcal{D}$. So one can view $\widetilde{\mathsf{P}}$ itself as the secret key of Romulus*-M. From now on, we will focus on the security of Romulus*-M.

4.1 Proof of Privacy

The i-th ciphertext block of an encryption query with nonce N is defined as $C[i] = M[i] \oplus G(S)$ with the last block truncated if needed, and the tag is defined as $T = G(S)$ for some S, where G is regular and $S = \widetilde{\mathsf{P}}^{N,x,y}(S')$ for some x, y and S'. If the adversary is nonce-respecting, then tweaks (x, y, N) are all distinct throughout the game. If $\widetilde{\mathsf{P}}$ is evaluated by lazy sampling, then ciphertext blocks $C[i]$ and tag T will be all uniform and independent at random. Therefore, for any adversary $\mathcal{A}$ against the privacy of Romulus*-M, we have

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(\mathcal{A}) = 0.$$

4.2 Proof of Authenticity

We will assume that $q_d = 1$, and then use Lemma 1. We can also assume that the single verification query is made at the end of the game after all the encryption queries have been made. Furthermore, we will give $\mathcal{A}$ direct access to an oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$. So $\mathcal{A}$ can now compute $Z = \widetilde{\mathsf{P}}^{(N,36,i)}(X)$ for any (N, i, X) of its choice. This only increases the advantage of $\mathcal{A}$.

We observe that for an encryption query (N_i, A_i, M_i), $\mathcal{A}$ can compute C_i from T_i and the oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$, and that for a decryption query (N', A', C', T'), $\mathcal{A}$ can compute M' from T', C', and the oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$. With this observation, we modify the game as follows:

- For an encryption query (N_i, A_i, M_i), $\mathcal{A}$ only receives T_i.
- Instead of making a decryption query (N', A', C', T'), $\mathcal{A}$ makes a (single) verification query of the form (N', A', M', T').

With this modification, we can focus on the analysis of the MAC part of Romulus*-M, denoted Romulus$^\star_{\mathsf{mac}}$-M.

Given a $(q_e, 1, \sigma_e, \sigma_d, t)$-adversary $\mathcal{A}$ against the authenticity of Romulus$^\star_{\mathsf{mac}}$-M, we can slightly modify it to obtain a $(q_e, 1, \sigma_e, \sigma_d, t)$-adversary $\mathcal{B}$ distinguishing

the real world ($\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M.Enc}_{\widetilde{\mathsf{P}}}, \mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M.Dec}_{\widetilde{\mathsf{P}}}$) and the ideal world ($\mathsf{Rand}, \mathsf{Rej}$) such that

$$\begin{aligned}&\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}}(\mathcal{A})\\ &\qquad\leqslant \left|\Pr\left[\mathcal{B}^{\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M.Enc}_{\widetilde{\mathsf{P}}},\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M.Dec}_{\widetilde{\mathsf{P}}}}\Rightarrow 1\right]-\Pr\left[\mathcal{B}^{\mathsf{Rand},\mathsf{Rej}}\Rightarrow 1\right]\right|,\end{aligned}$$

where Rand returns an independent random n-bit string on any (distinct) input (N, M) and Rej returns $\perp$ for the (single) decryption query. In order to upper bound $\mathcal{B}$'s distinguishing advantage, we can use Patarin's coefficient-H technique [Pat08]. At the end of the game, $\mathcal{B}$ will have sets of query-response pairs

$$\begin{cases}\tau_e = ((N_1, A_1, M_1, T_1), \ldots, (N_{q_e}, A_{q_e}, M_{q_e}, T_{q_e})),\\ \tau_v = (N', A', M', T', b'),\end{cases}$$

where $b' \in \{\top, \perp\}$, and it holds that $b' = \perp$ in the ideal world. In the real world, $\mathcal{B}$ is given additional information $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ for free at the end of the game. In the ideal world, $\mathcal{B}$ is given two independent and uniform tweakable permutations $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$, which are independent of Rand and Rej. Overall, $\mathcal{B}$ obtains a *transcript*

$$\tau = \left(\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)\right).$$

A transcript τ is called *attainable* if the probability to obtain the transcript in the ideal world is non-zero, and let Θ be the set of all attainable transcripts. We let Θ_{ideal} and Θ_{real} denote the probability distributions of the transcript in the ideal world and in the real world, respectively. Based on these notations, we will use Lemma 2.

Since $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ are made public, one can determine the input to $\widetilde{\mathsf{P}}^{(N_i,w_i,a_i+m_i)}$ (resp. $\widetilde{\mathsf{P}}^{(N',w',a'+m')}$), denoted X_i (resp. X'), where a_i (resp. a') denotes the number of AD blocks for the i-th encryption query (resp. the unique decryption query) and m_i (resp. m') denotes the number of message blocks for the i-th encryption query (resp. the unique decryption query). Then a transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$ is defined to be *bad* if

$$(X_i, N_i, w_i, a_i + m_i) = (X', N', w', a' + m')$$

for some $i = 1, \ldots, q_e$. Otherwise τ is called *good*. We first show that, in the ideal world, the probability of obtaining a bad transcript is small.

Lemma 5. $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}] \leqslant \dfrac{2}{2^n}$.

Proof. Since $\mathcal{B}$ is nonce-respecting, there is a unique index i such that

$$(N_i, w_i, a_i + m_i) = (N', w', a' + m').$$

By renaming the AD and the message blocks, let

$$W \stackrel{\text{def}}{=} (W[1], \ldots, W[a' + m']) \leftarrow (A_i[1], \ldots, \mathtt{pad}(A_i[a_i]), M_i[1], \ldots, \mathtt{pad}(M_i[m_i])),$$
$$W' \stackrel{\text{def}}{=} (W'[1], \ldots, W'[a' + m']) \leftarrow (A'[1], \ldots, \mathtt{pad}(A'[a']), M'[1], \ldots, \mathtt{pad}(M'[m'])).$$

Since $w_i = w'$, we have

$$a_i \bmod 2 = a' \bmod 2.$$

Suppose that a_i and a' are all even. We distinguish two cases.

Case 1: $W = W'$. Assuming that $\mathcal{B}$ makes no redundant query, we have $a_i \neq a'$. Let $a = \max\{a_i, a'\}$. Since a is even, it should be the case that

$$\widetilde{\mathsf{P}}^{(W[a],44,a-1)}(U) = \widetilde{\mathsf{P}}^{(W'[a],40,a-1)}(V)$$

for some U and V, since otherwise one would have $X_i \neq X'$. This event will happen with probability $\frac{1}{2^n}$ (over the random choice of $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$) since $\widetilde{\mathsf{P}}$ uses different tweaks.

Case 2: $W \neq W'$. Let k denote the last index where the blocks of W and W' are different.

Case 2-1: $k < \max\{a_i, a'\}$. Similarly to the analysis of Case 1, collision $X_i = X'$ requires

$$\widetilde{\mathsf{P}}^{(W[a'],44,a'-1)}(U) = \widetilde{\mathsf{P}}^{(W'[a'],40,a'-1)}(V)$$

for some U and V, and the probability of this event is $\frac{1}{2^n}$.

Case 2-2: $k \geqslant \max\{a_i, a'\}$. If k is even, then it should be the case that

$$\widetilde{\mathsf{P}}^{(W[k],44,k-1)}(U) = \widetilde{\mathsf{P}}^{(W'[k],44,k-1)}(V)$$

for some U and V. This event will happen with probability $\frac{1}{2^n}$ since $\widetilde{\mathsf{P}}$ uses different tweaks. If k is odd, then it should be the case that

$$\widetilde{\mathsf{P}}^{(W[k-1],r,k-2)}(U) \oplus \widetilde{\mathsf{P}}^{(W'[k-1],s,k-2)}(V) = W[k] \oplus W'[k] \neq 0^n.$$

for some U, V and r, $s \in \{40, 44\}$. This event will happen with probability at most $\frac{1}{2^n - 1}$.

By applying a similar argument to the case that a_i and a' are all odd, we have

$$\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}] \leqslant \frac{1}{2^n - 1} \leqslant \frac{2}{2^n}. \qquad \square$$

We next show that the ratio of the interpolation probabilities is close to one.

Lemma 6. $\dfrac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} \geqslant 1 - \dfrac{2}{2^n}$.

Proof. We fix a good transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$. An important observation is that the probabilities of obtaining $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$ are the same in the ideal and the real worlds; every tweak of $\widetilde{\mathsf{P}}$ used to generate a tag T_i is fresh, and the tweakable permutation can be evaluated by lazy sampling. In this way, all the tags will be chosen independently at random. Now we will upper bound the conditional probability that $b' = \top$ in the real world given $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$.

Since τ is good, we have

$$(X_i, N_i, w_i, a_i + m_i) \neq (X', N', w', a' + m')$$

for all $i = 1, \ldots, q_e$. In particular, there exists at most one index i such that $(N_i, w_i, a_i + m_i) = (N', w', a' + m')$. Therefore, the probability that

$$\widetilde{\mathsf{P}}^{(N',w',a'+m')}(X') = T'$$

is upper bounded by $\frac{1}{2^n - 1} \leqslant \frac{2}{2^n}$. Therefore, the probability that $b' = \bot$ in the real world is lower bounded by $1 - \frac{2}{2^n}$, which completes the proof. □

By Lemma 1, 2, 5 and 6, we have

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(q_e, q_d, \sigma_e, \sigma_d, t) \leqslant \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star_{\mathsf{mac}}\text{-}\mathsf{M}}(\mathcal{A}) \leqslant \frac{4q_d}{2^n}.$$

4.3 Summary

Any (q_e, σ_e, t)-adversary $\mathcal{A}$ against the privacy of Romulus-M can be viewed as an adversary against Romulus*-M such that

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}\text{-}\mathsf{M}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(q_e, t + O(\sigma_e)).$$

Similarly, any $(q_e, q_d, \sigma_e, \sigma_d, t)$-adversary $\mathcal{A}$ against the authenticity of Romulus-M can be viewed as an adversary against Romulus*-M such that

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}\text{-}\mathsf{M}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d)).$$

With this observation, we obtain the following theorem.

Theorem 2. *For* Romulus-M*, we have*

$$\begin{aligned}\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}\text{-}\mathsf{M}}(q_e, \sigma_e, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(q_e, t + O(\sigma_e)),\\ \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}\text{-}\mathsf{M}}(q_e, q_d, \sigma_e, \sigma_d, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\widetilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d))\\ &\quad + \frac{4q_d}{2^n}.\end{aligned}$$

5 Security of Romulus-M in the Nonce-misuse Setting

In this section, we prove the security of Romulus-M against nonce-misusing adversaries. Let Romulus*-M denote the AE mode obtained from Romulus-M by replacing the underlying keyed tweakable block cipher $\widetilde{E}$ by a truly random tweakable permutation (TURP)

$$\widetilde{\mathsf{P}} : \overline{\mathcal{T}} \times \mathcal{M} \rightarrow \mathcal{M}$$

with $\overline{\mathcal{T}} = \mathcal{T} \times \mathcal{B} \times \mathcal{D}$. So one can view $\widetilde{\mathsf{P}}$ itself as the secret key of Romulus*-M. From now on, we will focus on the security of Romulus*-M.

5.1 Proof of Privacy

Let $\mathcal{A}$ be an (r, q_e, σ, t)-adversary against the privacy of Romulus*-M. In order to upper bound $\mathcal{A}$'s distinguishing advantage, we will use Patarin's coefficient-H technique [Pat08]. At the end of the game, $\mathcal{A}$ will have a set of query-response pairs

$$\tau_e = ((N_1, A_1, M_1, T_1), \ldots, (N_{q_e}, A_{q_e}, M_{q_e}, T_{q_e})).$$

In the real world, $\mathcal{A}$ is given additional information $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ for free at the end of the game. In the ideal world, $\mathcal{A}$ is given two independent and uniform tweakable permutations $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$, which are independent of Rand. Overall, $\mathcal{A}$ obtains a *transcript*

$$\tau = \left(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)\right).$$

A transcript τ is called *attainable* if the probability to obtain the transcript in the ideal world is non-zero, and let Θ be the set of all attainable transcripts. We let Θ_{ideal} and Θ_{real} denote the probability distributions of the transcript in the ideal world and in the real world, respectively. Based on these notations, we will use Lemma 2.

Since $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ are made public, one can determine the input to $\widetilde{\mathsf{P}}^{(N_i,w_i,a_i+m_i)}$, denoted X_i, where a_i denotes the number of AD blocks for the i-th encryption query and m_i denotes the number of message blocks for the i-th encryption query. Then a transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$ is defined to be *bad* if one of the following conditions holds.

1. $\mathsf{BadT_1} \Leftrightarrow (X_i, N_i, w_i, a_i + m_i) = (X_j, N_j, w_j, a_j + m_j)$ for some i and j such that $1 \leqslant i < j \leqslant q_e$;
2. $\mathsf{BadT_2} \Leftrightarrow (N_i, T_i) = (N_j, T_j)$ for some i and j such that $1 \leqslant i < j \leqslant q_e$;
3. $\mathsf{BadT_3} \Leftrightarrow N_i = N_j$ and $G^{-1}(M_i[k] \oplus C_i[k]) \oplus M_i[k] = G^{-1}(M_j[k] \oplus C_j[k]) \oplus M_j[k]$ for some k, i and j such that $1 \leqslant k \leqslant \min\{m_i, m_j\}$ and $1 \leqslant i < j \leqslant q_e$.

If τ is not bad, then it will be called *good*. We first show that, in the ideal world, the probability of obtaining a bad transcript is small. Let

$$\mathsf{BadT} \stackrel{\text{def}}{=} \mathsf{BadT}_1 \cup \mathsf{BadT}_2 \cup \mathsf{BadT}_3.$$

Then we have the following lemma.

Lemma 7. $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}] \leqslant \dfrac{3(r-1)q_e}{2^n} + \dfrac{(r-1)\sigma}{2^n}$.

The proof is immediate from the following three lemmas.

Lemma 8. $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_1] \leqslant \dfrac{2(r-1)q_e}{2^n}$.

Proof. Since $\mathcal{B}$ makes at most r queries with the same nonce, the number of pairs (i, j) such that $i < j$ and

$$(N_i, w_i, a_i + m_i) = (N_j, w_j, a_j + m_j)$$

is at most $(r-1)q_e$. Once such a pair is fixed, let

$$W_i \stackrel{\text{def}}{=} (W_i[1], \ldots, W_i[a_i + m_i]) \leftarrow (A_i[1], \ldots, \mathtt{pad}(A_i[a_i]), M_i[1], \ldots, \mathtt{pad}(M_i[m_i])),$$
$$W_j \stackrel{\text{def}}{=} (W_j[1], \ldots, W_j[a_j + m_j]) \leftarrow (A_j[1], \ldots, \mathtt{pad}(A_j[a_j]), M_j[1], \ldots, \mathtt{pad}(M_j[m_j]))$$

by renaming the AD and the message blocks. Since $w_i = w_j$, we have

$$a_i \bmod 2 = a_j \bmod 2.$$

Suppose that a_i and a_j are all even. We distinguish two cases.

Case 1: $W_i = W_j$. Assuming that $\mathcal{B}$ makes no redundant query, we have $a_i \neq a_j$. Let $a = \max\{a_i, a_j\}$. Since a is even, it should be the case that

$$\widetilde{\mathsf{P}}^{(W[a],44,a-1)}(U) = \widetilde{\mathsf{P}}^{(W'[a],40,a-1)}(V)$$

for some U and V, since otherwise one would have $X_i \neq X_j$. This event will happen with probability $\frac{1}{2^n}$ (over the random choice of $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$) since $\widetilde{\mathsf{P}}$ uses different tweaks.

Case 2: $W_i \neq W_j$. Let k denote the last index where the blocks of W_i and W_j are different.

Case 2-1: $k < \max\{a_i, a_j\}$. Similarly to the analysis of Case 1, collision $X_i = X_j$ requires

$$\widetilde{\mathsf{P}}^{(W_i[a_j],44,a_j-1)}(U) = \widetilde{\mathsf{P}}^{(W_j[a_j],40,a_j-1)}(V)$$

for some U and V, and the probability of this event is $\frac{1}{2^n}$.

Case 2-2: $k \geqslant \max\{a_i, a_j\}$. If k is even, then it should be the case that

$$\widetilde{\mathsf{P}}^{(W_i[k],44,k-1)}(U) = \widetilde{\mathsf{P}}^{(W_j[k],44,k-1)}(V)$$

for some U and V. This event will happen with probability $\frac{1}{2^n}$ since $\widetilde{\mathsf{P}}$ uses different tweaks.

If k is odd, then it should be the case that

$$\widetilde{\mathsf{P}}^{(W_i[k-1],r,k-2)}(U) \oplus \widetilde{\mathsf{P}}^{(W_j[k-1],s,k-2)}(V) = W_i[k] \oplus W_j[k] \neq 0^n.$$

for some U, V and r, $s \in \{40, 44\}$. This event will happen with probability at most $\frac{1}{2^n-1}$.

By applying a similar argument to the case that a_i and a_j are all odd, we have

$$\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_1] \leqslant \frac{(r-1)q_e}{2^n-1} \leqslant \frac{2(r-1)q_e}{2^n}. \qquad \square$$

Lemma 9. $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_2] \leqslant \dfrac{(r-1)q_e}{2^n}$.

Proof. the number of pairs (i, j) such that $i < j$ and

$$(N_i, w_i, a_i + m_i) = (N_j, w_j, a_j + m_j)$$

is at most $(r-1)q_e$. Once such a pair is fixed, the probability that $T_i = T_j$ is $\dfrac{1}{2^n}$ since T_i and T_j are independently at random in the ideal world. Therefore, we have $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_2] \leqslant \dfrac{(r-1)q_e}{2^n}$. $\square$

Lemma 10. $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_3] \leqslant \dfrac{(r-1)\sigma}{2^n}$.

Proof. The number of possible choices for (j, k) such that $1 \leqslant j \leqslant q_e$ and $1 \leqslant k \leqslant m_j$ is σ. For each of such pairs, the number of indices i such that $i < j$ and $N_i = N_j$ is at most $r-1$. Once a triple (i, j, k) is fixed (satisfying the above properties), the probability that $G^{-1}(M_i[k] \oplus C_i[k]) \oplus M_i[k] = G^{-1}(M_j[k] \oplus C_j[k]) \oplus M_j[k]$ is $\dfrac{1}{2^n}$ since $C_i[k]$ and $C_j[k]$ are independently at random in the ideal world (assuming $k \leqslant m_i$). Therefore, we have $\Pr[\Theta_{\text{ideal}} \in \mathsf{BadT}_3] \leqslant \dfrac{(r-1)\sigma}{2^n}$. $\square$

Fix a good transcript $\tau = (\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$. The probability of obtaining $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ are the same in both worlds. Now consider the probability of obtaining τ_e in the real world. When $w \notin \{40, 44\}$, we will evaluate $\widetilde{\mathsf{P}}^{(\cdot,w,\cdot)}(\cdot)$ by lazy sampling. An important observation is that if τ is good, then every evaluation of $\widetilde{\mathsf{P}}^{(N,w,k)}(S)$ is fresh; it has not been determined by any previous call to $\widetilde{\mathsf{P}}$. More precisely,

1. every tag T is determined by a fresh query to $\widetilde{\mathsf{P}}$ since $(X_i, N_i, w_i, a_i + m_i)$ are all distinct;
2. every first ciphertext block $C_i[1]$ is determined by a fresh query to $\widetilde{\mathsf{P}}$ since (N_i, T_i) are all distinct;
3. for each $k > 1$, every k-th ciphertext block $C_i[k]$ is determined by a fresh query to $\widetilde{\mathsf{P}}$ since otherwise we would have a collision

$$G^{-1}(M_i[k] \oplus C_i[k]) \oplus M_i[k] = G^{-1}(M_j[k] \oplus C_j[k]) \oplus M_j[k]$$

for some i and j such that $1 \leqslant k \leqslant \min\{m_i, m_j\}$ and $1 \leqslant i < j \leqslant q_e$, which is impossible if τ is good.

When every query to $\widetilde{\mathsf{P}}$ is fresh, its response is chosen uniformly at random from the set of size at most 2^n. So the probability of obtaining τ in the real world is not smaller than the probability of obtaining τ in the ideal world. Therefore, we have the following lemma.

Lemma 11. $\dfrac{\Pr[\Theta_{\text{real}} = \tau]}{\Pr[\Theta_{\text{ideal}} = \tau]} \geqslant 1.$

By Lemma 2, 7 and 11, we have

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(r, q_e, \sigma, t) \leqslant \frac{3(r-1)q_e}{2^n} + \frac{(r-1)\sigma}{2^n} \leqslant \frac{4r\sigma}{2^n}.$$

5.2 Proof of Authenticity

We will give $\mathcal{A}$ direct access to an oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$. So $\mathcal{A}$ can now compute $Z = \widetilde{\mathsf{P}}^{(N,36,i)}(X)$ for any (N, i, X) of its choice. This only increases the advantage of $\mathcal{A}$. We observe that for an encryption query (N_i, A_i, M_i), $\mathcal{A}$ can compute C_i from T_i and the oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$, and that for a decryption query (N', A', C', T'), $\mathcal{A}$ can compute M' from T', C', and the oracle $\widetilde{\mathsf{P}}^{(\cdot,36,\cdot)}(\cdot)$. With this observation, we modify the game as follows:

- For an encryption query (N_i, A_i, M_i), $\mathcal{A}$ only receives T_i.
- Instead of making a decryption query (N', A', C', T'), $\mathcal{A}$ makes a (single) verification query of the form (N', A', M', T').

With this modification, we can focus on the analysis of the MAC part of $\mathsf{Romulus}^\star\text{-}\mathsf{M}$, denoted $\mathsf{Romulus}^\star_{\mathsf{mac}}\text{-}\mathsf{M}$.

Given a (r, q_e, q_d, σ, t)-adversary $\mathcal{A}$ against the authenticity of $\mathsf{Romulus}^\star_{\mathsf{mac}}\text{-}\mathsf{M}$, we can slightly modify it to obtain a (r, q_e, q_d, σ, t)-adversary $\mathcal{B}$ distinguishing

the real world ($\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}.\mathsf{Enc}_{\widetilde{\mathsf{P}}}, \mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}.\mathsf{Dec}_{\widetilde{\mathsf{P}}}$) and the ideal world ($\mathsf{Rand}, \mathsf{Rej}$) such that

$$\begin{aligned}&\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}}(\mathcal{A})\\ &\qquad\leqslant \left|\Pr\left[\mathcal{B}^{\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}.\mathsf{Enc}_{\widetilde{\mathsf{P}}},\mathsf{Romulus}^{\star}_{\mathsf{mac}}\text{-}\mathsf{M}.\mathsf{Dec}_{\widetilde{\mathsf{P}}}} \Rightarrow 1\right] - \Pr\left[\mathcal{B}^{\mathsf{Rand},\mathsf{Rej}} \Rightarrow 1\right]\right|,\end{aligned}$$

where Rand returns an independent random n-bit string on any (distinct) input (N, M) and Rej returns $\perp$ for the (single) decryption query. In order to upper bound $\mathcal{B}$'s distinguishing advantage, we can use Patarin's coefficient-H technique [Pat08]. At the end of the game, $\mathcal{B}$ will have sets of query-response pairs

$$\begin{cases}\tau_e = ((N_1, A_1, M_1, T_1), \ldots, (N_{q_e}, A_{q_e}, M_{q_e}, T_{q_e})),\\ \tau_v = ((N'_1, A'_1, M'_1, T'_1, b'_1), \ldots, (N'_{q_d}, A'_{q_d}, M'_{q_d}, T'_{q_d}, b'_{q_d}))\end{cases}$$

where $b'_j \in \{\top, \perp\}$, and it holds that $b'_j = \perp$ in the ideal world. In the real world, $\mathcal{B}$ is given additional information $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ for free at the end of the game. In the ideal world, $\mathcal{B}$ is given two independent and uniform tweakable permutations $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$, which are independent of Rand and Rej. Overall, $\mathcal{B}$ obtains a *transcript*

$$\tau = \left(\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)\right).$$

A transcript τ is called *attainable* if the probability to obtain the transcript in the ideal world is non-zero, and let Θ be the set of all attainable transcripts. We let Θ_{ideal} and Θ_{real} denote the probability distributions of the transcript in the ideal world and in the real world, respectively. Based on these notations, we will use Lemma 2.

Since $\widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot)$ and $\widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot)$ are made public, one can determine the input to $\widetilde{\mathsf{P}}^{(N_i,w_i,a_i+m_i)}$ (resp. $\widetilde{\mathsf{P}}^{(N'_j,w'_j,a'_j+m'_j)}$), denoted X_i (resp. X'_j), where a_i (resp. a'_j) denotes the number of AD blocks for the i-th encryption query (resp. the j-th decryption query) and m_i (resp. m'_j) denotes the number of message blocks for the i-th encryption query (resp. the j-th decryption query). Then a transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$ is defined to be *bad* if

$$(X_i, N_i, w_i, a_i + m_i) = (X'_j, N'_j, w'_j, a'_j + m'_j)$$

for some $i \in \{1, \ldots, q_e\}$ and $j \in \{1, \ldots, q_d\}$, or

$$(X_i, N_i, w_i, a_i + m_i) = (X_j, N_j, w_j, a_j + m_j)$$

for some i and j such that $1 \leqslant i < j \leqslant q_e$. Otherwise τ is called *good*. We first show that, in the ideal world, the probability of obtaining a bad transcript is small.

Lemma 12. $\Pr[\Theta_{\mathrm{ideal}} \in \mathsf{BadT}] \leqslant \dfrac{2rq_e}{2^n} + \dfrac{2rq_d}{2^n}$.

Proof. With a similar argument to the proof of Lemma 7, the probability that $(X_i, N_i, w_i, a_i + m_i) = (X'_j, N'_j, w'_j, a'_j + m'_j)$ for some $i \in \{1, \ldots, q_e\}$ and $j \in \{1, \ldots, q_d\}$ is upper bounded by

$$\frac{2rq_d}{2^n}.$$

On the other hand, with the same argument as the proof of Lemma 8, the probability that $(X_i, N_i, w_i, a_i + m_i) = (X_j, N_j, w_j, a_j + m_j)$ for some i and j such that $1 \leqslant i < j \leqslant q_e$ is upper bounded by

$$\frac{2(r-1)q_e}{2^n},$$

which completes the proof. □

We next show that the ratio of the interpolation probabilities is close to one.

Lemma 13. $\dfrac{\Pr[\Theta_{\mathrm{real}} = \tau]}{\Pr[\Theta_{\mathrm{ideal}} = \tau]} \geqslant 1 - \dfrac{2q_d}{2^n}$.

Proof. We fix a good transcript $\tau = (\tau_e, \tau_v, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$. An important observation is that the probabilities of obtaining $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$ are the same in the ideal and the real worlds; every tweak of $\widetilde{\mathsf{P}}$ used to generate a tag T_i is fresh, and the tweakable permutation can be evaluated by lazy sampling. In this way, all the tags will be chosen independently at random. Now we will upper bound the probability that $b' = \top$ in the real world given $(\tau_e, \widetilde{\mathsf{P}}^{(\cdot,40,\cdot)}(\cdot), \widetilde{\mathsf{P}}^{(\cdot,44,\cdot)}(\cdot))$.

Since τ is good, we have

$$(X_i, N_i, w_i, a_i + m_i) \neq (X_j, N_j, w_j, a_j + m_j)$$

for all $i \in \{1, \ldots, q_e\}$ and $j \in \{1, \ldots, q_d\}$. For each $j \in \{1, \ldots, q_d\}$, there are at most r indices $i \in \{1, \ldots, q_e\}$ such that $N_i = N_j$. So the probability that

$$\widetilde{\mathsf{P}}^{(N_j, w_j, a_j + m_j)}(X_j) = T_j$$

is upper bounded by $\frac{1}{2^n - r} \leqslant \frac{2}{2^n}$ (if $r \leqslant 2^{n-1}$). Therefore, the probability that $b_j = \bot$ for some $j \in \{1, \ldots, q_d\}$ in the real world is lower bounded by $1 - \frac{2q_d}{2^n}$, which completes the proof. □

By Lemma 2, 12 and 13, we have

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star\text{-}\mathsf{M}}(r, q_e, q_d, \sigma_e, \sigma_d, t) = \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus}^\star_{\mathsf{mac}}\text{-}\mathsf{M}}(\mathcal{A}) \leqslant \frac{2rq_e}{2^n} + \frac{4rq_d}{2^n}$$

when $r \leqslant 2^{n-1}$.

5.3 Summary

Any (r, q_e, σ_e, t)-adversary $\mathcal{A}$ against the privacy of Romulus-M can be viewed as an adversary against Romulus*-M such that

$$\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus^\star\text{-}M}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e, t + O(\sigma_e)).$$

Similarly, any $(r, q_e, q_d, \sigma_e, \sigma_d, t)$-adversary $\mathcal{A}$ against the authenticity of Romulus-M can be viewed as an adversary against Romulus*-M such that

$$\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \leqslant \mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus^\star\text{-}M}}(\mathcal{A}) + \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d)).$$

With this observation, we obtain the following theorem.

Theorem 3. *If $1 \leqslant r \leqslant 2^{n-1}$, then we have*

$$\begin{aligned}
\mathbf{Adv}^{\mathtt{priv}}_{\mathsf{Romulus\text{-}M}}(r, q_e, \sigma_e, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e, t + O(\sigma_e)) + \frac{4r\sigma_e}{2^n}, \\
\mathbf{Adv}^{\mathtt{auth}}_{\mathsf{Romulus\text{-}M}}(r, q_e, q_d, \sigma_e, \sigma_d, t) &\leqslant \mathbf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(q_e + q_d, t + O(\sigma_e) + O(\sigma_d)) \\
&\quad + \frac{2rq_e + 4rq_d}{2^n}.
\end{aligned}$$

References

[GIK+19] Chun Guo, Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Romulus v1.3. Submission to NIST Lightweight Cryptography Project, 2019.

[IKMP20] Tetsu Iwata, Mustafa Khairallah, Kazuhiko Minematsu, and Thomas Peyrin. Duel of the Titans: The Romulus and Remus Families of Lightweight AEAD Algorithms. *IACR Trans. Symmetric Cryptol.*, 2020(1):43–120, 2020.

[Pat08] Jacques Patarin. The "Coefficients H" Technique. In *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2008.