Matching Attacks on Romulus-M

Makoto Habu[1], Kazuhiko Minematsu[2], and Tetsu Iwata[1]

[1] Nagoya University, Nagoya, Japan
habu.makoto@f.mbox.nagoya-u.ac.jp, tetsu.iwata@nagoya-u.jp
[2] NEC Corporation, Kanagawa, Japan
k-minematsu@nec.com

Abstract. This paper considers a problem of identifying matching attacks against Romulus-M, one of the ten finalists of NIST Lightweight Cryptography standardization project. Romulus-M is provably secure, i.e., there is a theorem statement showing the upper bound on the success probability of attacking the scheme as a function of adversaries' resources. If there exists an attack that matches the provable security bound, then this implies that the attack is optimal, and that the bound is tight in the sense that it cannot be improved. We show that the security bounds of Romulus-M are tight for a large class of parameters by presenting concrete matching attacks.

Keywords: Lightweight cryptography · Authenticated encryption with associated data · Provable security · Romulus-M · Tightness · Matching attack

1 Introduction

An authenticated encryption with associated data (AEAD) scheme is a symmetric key primitive used for securing data in terms of privacy and authenticity simultaneously. NIST holds Lightweight Cryptography standardization project[3] to select an international standard scheme for AEAD and hashing for constrained devices. In March 2021, NIST selected a total of ten finalists, and we consider one of the schemes called Romulus [3,5]. More precisely, the Romulus family of an AEAD scheme consists of Romulus-N, Romulus-M, Romulus-T, and Romulus-H [3]. Romulus-N is for nonce-based AEAD, Romulus-M is for nonce misuse-resistant AEAD, Romulus-T is for leakage-resilient AEAD, and Romulus-H is for hashing, and our focus is Romulus-M. They all use a tweakable block cipher (TBC) [8,9] as the underlying primitive, and they specifically use SKINNY [1] in their specifications [3]. The provable security results are presented in [5] for Romulus-N and Romulus-M, and there is also a third party proof on these schemes by Jooyoung Lee [7]. A provable security analysis on Romulus-T (a.k.a. TEDT mode) is presented in [2] and that on Romulus-H (a.k.a. MDPH) is in [11,4].

The provable security of a symmetric key scheme refers to a theorem statement showing the upper bound on the success probability of attacking the

[3] https://csrc.nist.gov/projects/lightweight-cryptography

scheme, where the upper bound is expressed as a function of adversaries' resources. Examples of the resources include the running time, the number of oracle calls, the length of each query, or the total length of responses the adversary obtains from the oracle. In this paper, we consider a problem of identifying matching attacks against Romulus-M for a class of parameters. Such attacks are optimal since obtaining a better attack complexity is impossible as they match the provable security bound, and they also show that the provable security bound is tight in the sense that obtaining a better security bound is impossible.

Provable security and attacks of Romulus-M. We focus on the provable security of Romulus-M presented in [5]. Following the standard security notions for AEAD schemes [13,12], it has two security bounds. One is for privacy and the other is for authenticity. We assume that the TBC is ideally secure, meaning that we do not consider adversaries' running time (time complexity). For privacy, we consider an adversary that can repeat a nonce up to r times. Then the privacy bound is of the form $4r\sigma_{\mathrm{priv}}/2^n$, where σ_{priv} is the effective block length which counts the number of primitive calls during the privacy game, and n is the block length of the underlying TBC. For authenticity, we consider an adversary that can repeat a nonce up to r times in encryption queries. The authenticity bound is $(4rq_e + 5rq_d)/2^n$, where q_e is the number of encryption queries and q_d is the number of decryption queries.

In this paper, for privacy, we present an attack with a success probability of at least $0.03r\sigma_{\mathrm{priv}}/2^n$, showing the tightness of the privacy bound up to a constant factor. The attack reduces to a collision finding problem, where there is a restriction that the same nonce can be repeated at most r times.

We also present an authenticity attack with a success probability of at least $0.07rq_e/2^n$ by making q_e encryption queries and one decryption query. In more detail, the authenticity provable security bound shows the infeasibility of an existential forgery, i.e., the adversary cannot find *some* non-trivial tuple of a nonce, associated data (AD), a ciphertext, and a tag that is accepted by the decryption oracle, and hence there is no need for the adversary to have control over the forged plaintext. On the other hand, our attack shows the feasibility of a universal forgery, meaning that the adversary can forge *any* tuple of a nonce, AD, and a plaintext, that could be maliciously chosen by the adversary before the start of the authenticity game. Our authenticity attack follows the idea of [6,10] that shows universal forgery attacks on various MACs with birthday-bound complexity, while in the case of Romulus-M, as in the privacy attack, there is a restriction that the same nonce can be repeated at most r times in encryption queries.

See Table 1 for the summary of provable security bounds and the success probability of our attacks in this paper.

2 Preliminaries

Authenticated encryption with associated data (AEAD). Let $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ be an AEAD scheme. For a key $K \in \mathcal{K}$, the encryption algorithm $\mathsf{Enc}_K : \mathcal{N} \times \mathcal{A} \times$

Table 1. Summary of provable security bounds [5] and the success probability of our attacks. In the table, n is the block length of the TBC, r is the number of times the adversary can repeat a nonce in encryption queries, σ_{priv} is the number of TBC calls during the privacy game, q_e is the number of encryption queries, and q_d is the number of decryption queries.

Privacy	Authenticity	Reference
$\forall \mathcal{A}, \mathbf{Adv}^{\text{priv}}_{\textsf{Romulus-M}}(\mathcal{A}) \leq \frac{4r\sigma_{\text{priv}}}{2^n}$	$\forall \mathcal{A}, \mathbf{Adv}^{\text{auth}}_{\textsf{Romulus-M}}(\mathcal{A}) \leq \frac{4rq_e + 5rq_d}{2^n}$	[5]
$\exists \mathcal{A}, \mathbf{Adv}^{\text{priv}}_{\textsf{Romulus-M}}(\mathcal{A}) \geq \frac{0.03r\sigma_{\text{priv}}}{2^n}$	$\exists \mathcal{A}, \mathbf{Adv}^{\text{auth}}_{\textsf{Romulus-M}}(\mathcal{A}) \geq \frac{0.07rq_e}{2^n}$	This paper

$\mathcal{M} \to \mathcal{C} \times \mathcal{T}$ takes a nonce $N \in \mathcal{N}$, associated data (AD) $A \in \mathcal{A}$, and a plaintext $M \in \mathcal{M}$ as input, and returns a ciphertext $C \in \mathcal{C}$ and a tag $T \in \mathcal{T}$, where $\mathcal{K}, \mathcal{N}, \mathcal{A}, \mathcal{M}, \mathcal{C}$, and $\mathcal{T}$ are the key space, nonce space, AD space, plaintext space, ciphertext space, and tag space, respectively. We write $\mathsf{Enc}_K(N, A, M) = (C, T)$. The decryption algorithm $\mathsf{Dec}_K : \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T} \to \mathcal{M} \cup \{\bot\}$ takes $(N, A, C, T) \in \mathcal{N} \times \mathcal{A} \times \mathcal{C} \times \mathcal{T}$ as input, and outputs M or $\bot$, where the symbol $\bot$ means rejection. We write $\mathsf{Dec}_K(N, A, C, T) = M$ or $\mathsf{Dec}_K(N, A, C, T) = \bot$. We require the correctness of the scheme. That is, for any $(K, N, A, M) \in \mathcal{K} \times \mathcal{N} \times \mathcal{A} \times \mathcal{M}$, we require that $\mathsf{Dec}_K(N, A, \mathsf{Enc}_K(N, A, M)) = M$ holds.

Notation. Let $\{0,1\}^n$ be the set of bit strings of n bits, and $\{0,1\}^*$ be the set of all finite bit strings, including the empty string ϵ. For $X \in \{0,1\}^*$, $|X|$ denotes its length in bits. For two bit strings X_1 and X_2, let $X_1 \| X_2$ denote their concatenation. For $X \in \{0,1\}^*$, let $(X[1], \ldots, X[\ell]) \stackrel{n}{\leftarrow} X$ denote the partition of X into n-bit strings, i.e., if X is a non-empty string, then $X[1], \ldots, X[\ell]$ are unique bit strings such that $X[1] \| \cdots \| X[\ell] = X$, $|X[i]| = n$ for $1 \leq i \leq \ell - 1$, and $1 \leq |X[\ell]| \leq n$, and if X is the empty string, then $X[1] \stackrel{n}{\leftarrow} X$, where $X[1]$ is the empty string. We write $|X|_n = \max\{1, \lceil |X|/n \rceil\}$, and it follows that $\ell = |X|_n$. For $X \in \{0,1\}^*$ with $|X| \geq \ell$, we write $\mathtt{lsb}_\ell(X)$ to denote the truncation of X to its ℓ least significant bits.

Padding. Let n be a multiple of 8. For $X \in \{0,1\}^*$ with $|X|$ a multiple of 8 and $|X| \leq n$, we let $\mathtt{pad}_n(X) = X$ if $|X| = n$, and $\mathtt{pad}_n(X) = X \| 0^{n-|X|-8} \| \mathtt{len}_8(X)$ if $0 \leq |X| < n$, where $\mathtt{len}_8(X)$ denotes the 8-bit binary representation of the byte length of X.

Tweakable block cipher (TBC) [8,9]. In $\mathsf{Romulus\text{-}M}$, we use a TBC $\widetilde{E} : \mathcal{K} \times \overline{\mathcal{T}} \times \{0,1\}^n \to \{0,1\}^n$, where $\mathcal{K}$ is the key space and $\overline{\mathcal{T}}$ is the tweak space, and for each $(K, \overline{T}) \in \mathcal{K} \times \overline{\mathcal{T}}$, $\widetilde{E}(K, \overline{T}, \cdot)$ is a permutation over $\{0,1\}^n$. The tweak $\overline{T}$ is of the form $\overline{T} = (T_W, B, D)$, where $T_W \in \{0,1\}^n$, $B \in \{0,1\}^8$ is used for domain separation, and $D \in \{0,1\}^{n-8}$ is used as a block counter. We write $\widetilde{E}_K^{(X,w,\bar{i})}(S)$ for the output of the TBC under the key K, tweak $(X, w, \bar{i})$, and input block S, where $\bar{i} \in \{0,1\}^{n-8}$ denotes the binary representation of $i \in \mathbb{N}$.

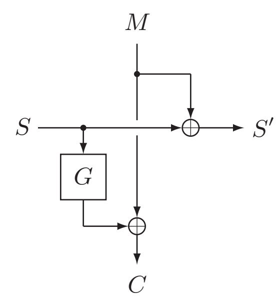

Fig. 1. The state update function ρ

A version of SKINNY [1] called Skinny-128-384+ is used in [3], in which case we have $n = 128$. The details of the TBC are irrelevant to our attacks and we treat n as a security parameter.

We write $\mathrm{Perm}(n)$ for the set of all the permutations over $\{0,1\}^n$. A random permutation is a permutation $\pi \in \mathrm{Perm}(n)$ that is chosen uniformly at random from $\mathrm{Perm}(n)$.

State update function. In $\mathsf{Romulus\text{-}M}$, we use a state update function $\rho : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$ and its inverse function $\rho^{-1} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n \times \{0,1\}^n$. They are defined as $\rho(S, M) = (S', C)$, where $C = M \oplus G(S)$ and $S' = S \oplus M$, and $\rho^{-1}(S, C) = (S', M)$, where $M = C \oplus G(S)$ and $S' = S \oplus M$. See Fig. 1. Here, $G(\cdot)$ is a linear mapping over $\{0,1\}^n$ defined by an $n \times n$ matrix. The details are irrelevant to our attack and we omit the description, which can be found in [3].

We remark that the notation ρ^{-1} is meant to be the inverse function of ρ with respect to its second argument only. We also remark that for any $(S, M) \in \{0,1\}^n \times \{0,1\}^n$, if $\rho(S, M) = (S', C)$, then $\rho^{-1}(S, C) = (S', M)$ holds.

Security notion. A privacy adversary $\mathcal{A}$ against $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ has an oracle $\mathcal{O}$, which is either the encryption algorithm Enc_K or a random oracle \$, where \$-oracle returns a uniform random bit string that has the same length as the output of Enc_K-oracle. We define the privacy advantage as

$$\mathbf{Adv}_{\Pi}^{\mathrm{priv}}(\mathcal{A}) \stackrel{\mathrm{def}}{=} \left| \Pr[\mathcal{A}^{\mathsf{Enc}_K} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1] \right|,$$

where the first probability is taken over the choice of K and the internal coin of $\mathcal{A}$, and the last one is over \$ and $\mathcal{A}$.

An authenticity adversary $\mathcal{A}$ has the encryption oracle Enc_K and the decryption oracle Dec_K. We say that $\mathcal{A}^{\mathsf{Enc}_K, \mathsf{Dec}_K}$ *forges* if it makes a decryption query (N^*, A^*, C^*, T^*) such that $\mathsf{Dec}_K(N^*, A^*, C^*, T^*) = M^*$, where (C^*, T^*) was not returned from Enc_K-oracle for an encryption query (N^*, A^*, M^*). We define the authenticity advantage as

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

where the probability is taken over the choice of K and the internal coin of $\mathcal{A}$.

This captures the authenticity in terms of *existential* forgery attacks, meaning that the adversary succeeds in forgery if it makes *some* non-trivial decryption query (N^*, A^*, C^*, T^*) that is not rejected, i.e., the forged nonce, AD, and plaintext (N^*, A^*, M^*) may not be fully controlled by the adversary. A *universal* forgery is a forgery where $\mathcal{A}$ succeeds in forgery for *any* given (N^*, A^*, M^*), that could be fully controlled by the adversary.

3 Specification and Provable Security of Romulus-M

3.1 Specification of Romulus-M

Romulus-M uses a TBC $\widetilde{E}$ as the underlying primitive. We present the algorithmic description of the encryption and decryption algorithms in Fig. 2.

First, for an input (N, A, M), where $N \in \{0,1\}^n$ and $A, M \in \{0,1\}^*$, the encryption of Romulus-M parses A and M into n-bit blocks, and processes them by applying the state update function ρ and the TBC $\widetilde{E}$ alternatively, and then a tag $T \in \{0,1\}^n$ is computed by using the nonce N as a part of the tweak for $\widetilde{E}$. Then a ciphertext C is computed starting from T, where the output of the TBC $\widetilde{E}$ is used as the randomness to encrypt the i-th plaintext block $M[i]$ into the i-th ciphertext block $C[i]$. We note that $|C| = |M|$ holds. See Fig. 3 for an illustration for the case $|A| = 2n$ and $|M| = 2n$.

The decryption of Romulus-M takes (N, A, C, T) as input, and it first computes a plaintext M from N and C. Then it computes a tag T^* for (N, A, M) following the encryption algorithm, and returns M if $T^* = T$. Otherwise, it outputs $\perp$, indicating rejection.

3.2 Provable Security of Romulus-M

Romulus-M is known to be provably secure. In what follows, we assume that the underlying TBC is perfectly secure. The following provable security result regarding privacy is known.

Theorem 1 ([5]). *For any privacy adversary $\mathcal{A}$ that makes at most q_e encryption queries and can repeat a nonce at most $1 \le r \le 2^{n-1}$ times, we have*

$$\mathbf{Adv}_{\mathsf{Romulus\text{-}M}}^{\mathrm{priv}}(\mathcal{A}) \le \frac{4r\sigma_{\mathrm{priv}}}{2^n} ,$$

where σ_{priv} is the total number of effective block length of all the encryption queries.

Algorithm $\mathsf{Romulus\text{-}M.Enc}_K(N, A, M)$

$S \leftarrow 0^n$
$(X[1], \ldots, X[a]) \xleftarrow{n} A$
$(X[a+1], \ldots, X[a+m]) \xleftarrow{n} M$
$z \leftarrow |X[a+m]|$
$w \leftarrow 48$
if $|X[a]| < n$ **then** $w \leftarrow w \oplus 2$
if $|X[a+m]| < n$ **then** $w \leftarrow w \oplus 1$
if $a \bmod 2 = 0$ **then** $w \leftarrow w \oplus 8$
if $m \bmod 2 = 0$ **then** $w \leftarrow w \oplus 4$
$X[a] \leftarrow \mathtt{pad}_n(X[a])$
$X[a+m] \leftarrow \mathtt{pad}_n(X[a+m])$
$x \leftarrow 40$
for $i = 1$ **to** $\lfloor (a+m)/2 \rfloor$
  $(S, \eta) \leftarrow \rho(S, X[2i-1])$
  if $i = \lfloor a/2 \rfloor + 1$ **then** $x \leftarrow x \oplus 4$
  $S \leftarrow \widetilde{E}_K^{(X[2i], x, \overline{2i-1})}(S)$
end for
if $a \bmod 2 = m \bmod 2$ **then**
  $(S, \eta) \leftarrow \rho(S, 0^n)$
else
  $(S, \eta) \leftarrow \rho(S, X[a+m])$
$S \leftarrow \widetilde{E}_K^{(N, w, \overline{a+m})}(S)$
$(\eta, T) \leftarrow \rho(S, 0^n)$
if $M = \epsilon$ **then return** (ϵ, T)
$S \leftarrow T$
for $i = 1$ **to** m
  $S \leftarrow \widetilde{E}_K^{(N, 36, \overline{i-1})}(S)$
  $(S, C[i]) \leftarrow \rho(S, X[a+i])$
end for
$C[m] \leftarrow \mathtt{lsb}_z(C[m])$
$C \leftarrow C[1] \parallel \cdots \parallel C[m-1] \parallel C[m]$
return (C, T)

Algorithm $\mathsf{Romulus\text{-}M.Dec}_K(N, A, C, T)$

if $C = \epsilon$ **then** $M \leftarrow \epsilon$
else
  $S \leftarrow T$
  $(C[1], \ldots, C[m]) \xleftarrow{n} C$
  $z \leftarrow |C[m]|$
  $C[m] \leftarrow \mathtt{pad}_n(C[m])$
  for $i = 1$ **to** m
    $S \leftarrow \widetilde{E}_K^{(N, 36, \overline{i-1})}(S)$
    $(S, M[i]) \leftarrow \rho^{-1}(S, C[i])$
  end for
  $M[m] \leftarrow \mathtt{lsb}_z(M[m])$
  $M \leftarrow M[1] \parallel \cdots \parallel M[m-1] \parallel M[m]$
$S \leftarrow 0^n$
$(X[1], \ldots, X[a]) \xleftarrow{n} A$
$(X[a+1], \ldots, X[a+m]) \xleftarrow{n} M$
$w \leftarrow 48$
if $|X[a]| < n$ **then** $w \leftarrow w \oplus 2$
if $|X[a+m]| < n$ **then** $w \leftarrow w \oplus 1$
if $a \bmod 2 = 0$ **then** $w \leftarrow w \oplus 8$
if $m \bmod 2 = 0$ **then** $w \leftarrow w \oplus 4$
$X[a] \leftarrow \mathtt{pad}_n(X[a])$
$X[a+m] \leftarrow \mathtt{pad}_n(X[a+m])$
$x \leftarrow 40$
for $i = 1$ **to** $\lfloor (a+m)/2 \rfloor$
  $(S, \eta) \leftarrow \rho(S, X[2i-1])$
  if $i = \lfloor a/2 \rfloor + 1$ **then** $x \leftarrow x \oplus 4$
  $S \leftarrow \widetilde{E}_K^{(X[2i], x, \overline{2i-1})}(S)$
end for
if $a \bmod 2 = m \bmod 2$ **then**
  $(S, \eta) \leftarrow \rho(S, 0^n)$
else
  $(S, \eta) \leftarrow \rho(S, X[a+m])$
$S \leftarrow \widetilde{E}_K^{(N, w, \overline{a+m})}(S)$
$(\eta, T^*) \leftarrow \rho(S, 0^n)$
if $T^* = T$ **then return** M **else** $\perp$

Fig. 2. The encryption and decryption algorithms of Romulus-M [3]. The dummy variable η is always discarded.

The total number of effective block length refers to the total number of TBC calls during the privacy game. In more detail, if $\mathcal{A}$ makes q encryption queries $(N_1, A_1, M_1), \ldots, (N_q, A_q, M_q)$, then the number of effective block length of the i-th query is at most $\lfloor (a_i + m_i)/2 \rfloor + 1 + m_i$, and the total number of effective

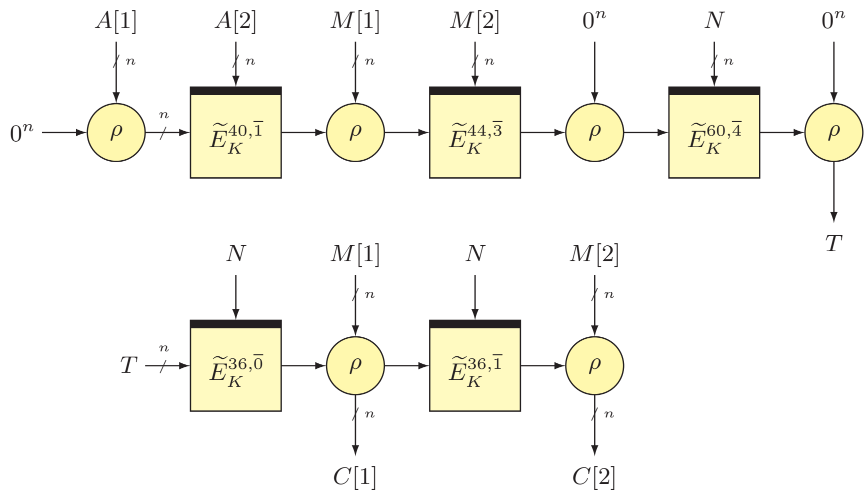

Fig. 3. The encryption of Romulus-M for the case $|A| = 2n$ and $|M| = 2n$

block length of $\mathcal{A}$ is as most $\sum_{1\le i\le q} (\lfloor (a_i + m_i)/2 \rfloor + 1 + m_i)$, where $a_i = |A_i|_n$ and $m_i = |M_i|_n$.[4] The following theorem shows the authenticity security.

Theorem 2 ([5]). *For any authenticity adversary $\mathcal{A}$ that makes q_e encryption queries and q_d decryption queries, and can repeat a nonce at most $1 \le r \le 2^{n-1}$ times in encryption queries, we have*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \le \frac{4rq_e + 5rq_d}{2^n} .$$

In Theorems 1 and 2, the case $r = 1$ corresponds to the security against nonce-respecting adversaries, and the bound becomes $\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) = 0$ for privacy, and $\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \le 5q_d/2^n$ for authenticity. See [5] for more details. We do not consider the case $r = 1$ further, since these bounds are trivially tight.

4 Distinguishing Attack on Romulus-M

In this section, we present our distinguishing attack on Romulus-M. We have the following theorem.

Theorem 3. *For* Romulus-M*, there exists a privacy adversary $\mathcal{A}$ with*

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \ge \frac{0.03r\sigma_{\mathrm{priv}}}{2^n} ,$$

where the effective block length of $\mathcal{A}$ is $\sigma_{\mathrm{priv}} = 5lr$ and $l \ge 1$ is a parameter.

[4] In [5], $\lfloor a_i/2 \rfloor + \lfloor m_i/2 \rfloor + 2 + m_i$ is used for the effective block length of the i-th query, while $\lfloor (a_i + m_i)/2 \rfloor + 1 + m_i$ is tight when the plaintext is non-empty.

Algorithm 1 Distinguishing attack on Romulus-M

```
for i = 1, ..., l do
    for j = 1, ..., r do
        (C_{i,j}, T_{i,j}) ← O(N_i, A_j, M)
    end for
end for
if T_{i,p} = T_{i,q} for some (i, p, q) then
    return 1
else
    return 0
end if
```

Proof. Let us fix $l \geq 1$, and we also fix l distinct nonces $N_1, \dots, N_l$, r distinct AD $A_1, \dots, A_r$, and an arbitrary plaintext $M = (M[1], \dots, M[m])$. For AD, their first blocks are distinct, while they share the remaining blocks. That is, for distinct $A_1[1], \dots, A_r[1] \in \{0,1\}^n$ and arbitrary $A[2], \dots, A[a] \in \{0,1\}^n$, we let

$$A_j = A_j[1] \parallel A[2] \parallel \cdots \parallel A[a]$$

for $1 \leq j \leq r$. The values of a and m can be arbitrarily for the attack to work, but we will later fix them to fit the claimed success probability.

For each $1 \leq i \leq l$ and $1 \leq j \leq r$, $\mathcal{A}$ encrypts (N_i, A_j, M) and obtains $(C_{i,j}, T_{i,j})$ from the encryption oracle. If there is a tuple (i, p, q) such that $T_{i,p} = T_{i,q}$, then $\mathcal{A}$ outputs 1, else it outputs 0. Our distinguishing attack is shown in Algorithm 1

For each encryption query, its number of effective block length is $\lfloor (a+m)/2 \rfloor + 1 + m$, and it follows that the total number of effective block length is $\sigma_{\text{priv}} = lr(\lfloor (a+m)/2 \rfloor + 1 + m)$.

First, consider the case that the oracle $\mathcal{O}$ is Romulus-M. It takes (N_i, M, A_j) as input and outputs $(C_{i,j}, T_{i,j})$. For fixed N_i, M, and $A[2], \dots, A[a]$, we observe that the mapping

$$A_j[1] \mapsto T_{i,j}$$

is a permutation over $\{0,1\}^n$. See Fig. 4 for an illustration describing this case. This property is independent of the lengths of A_j and M. For each $1 \leq i \leq r$, we see that $T_{i,1}, \dots, T_{i,r}$ are different from each other, since $A_1[1], \dots, A_r[1]$ are distinct. We therefore have $\Pr[\mathcal{A}^{\mathsf{Romulus\text{-}M}_K} \Rightarrow 1] = 0$.

Next, consider the case that $\mathcal{O}$ is a random oracle \$. For each N_i, the mapping $A_j[1] \mapsto T_{i,j}$ is an independent random function as nonces are distinct. For N_i, let p_i be the probability that there is a collision among $T_{i,1}, \dots, T_{i,r}$. Then we have $p_i = 1 - (2^n)_r/(2^n)^r$, where for integers $a \geq b \geq 1$, we let $(a)_b = a!/b! = a(a-1)\cdots(a-(b-1))$. Now we have

$$\Pr[\mathcal{A}^{\$} \Rightarrow 1] = 1 - \prod_{1 \leq i \leq l} (1 - p_i) = 1 - \left(\frac{(2^n)_r}{(2^n)^r} \right)^l \geq \left(1 - \frac{1}{e} \right) \frac{0.25 l r^2}{2^n}, \qquad (1)$$

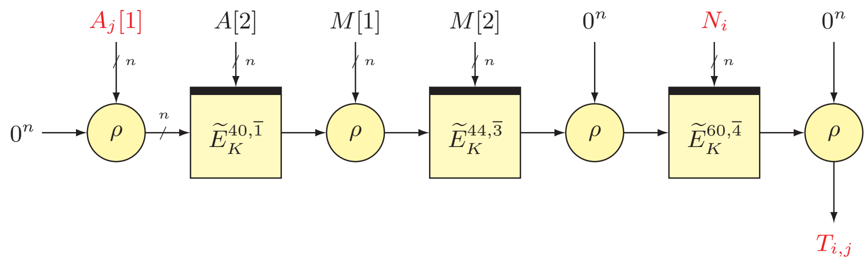

Fig. 4. Distinguishing attack on Romulus-M, showing the case $|A| = 2n$ and $|M| = 2n$. Variables that depend on i or j are highlighted in red. For each i, the mapping $A_j[1] \mapsto T_{i,j}$ is a permutation over $\{0,1\}^n$. Note that ρ is simply an XOR operation.

where e is Napier's constant, and the last inequality follows from an elementary calculation and the details are in Appendix A. The analysis until this point does not depend on the lengths of A_j nor M, and we fix their lengths $|A_j| = 2n$ and $|M| = 2n$, implying $a = 2$ and $m = 2$, in which case the total number of effective block length becomes $\sigma_{\text{priv}} = 5lr$.

Finally, the lower bound of the privacy advantage is given as

$$\begin{aligned} \mathbf{Adv}^{\text{priv}}_{\textsf{Romulus-M}}(\mathcal{A}) &= |\Pr[\mathcal{A}^{\textsf{Romulus-M}_K} \Rightarrow 1] - \Pr[\mathcal{A}^{\$} \Rightarrow 1]| \\ &\geq \left(1 - \frac{1}{e}\right) \frac{0.25lr^2}{2^n} \geq \frac{0.03r\sigma_{\text{priv}}}{2^n}, \end{aligned}$$

and we obtain the claimed success probability in Theorem 3. □

5 Universal Forgery Attack on Romulus-M

In this section, we present our universal forgery attack. For an arbitrary given challenge (N^*, A^*, M^*) that could be chosen by the adversary $\mathcal{A}$, the goal is to output (N^*, A^*, C^*, T^*) that is decrypted into M^* by the decryption algorithm of Romulus-M. In our attack, we make use of the following proposition.

Proposition 1. *Fix integers $l, r_1, r_3 \geq 1$ and let $\pi \in \mathrm{Perm}(n)$ be a random permutation. For $lr_1 + 1$ distinct bit strings $A^*[1], A_{1,i,j}[1] \in \{0,1\}^n$ for $1 \leq i \leq l$ and $1 \leq j \leq r_1$, and $lr_3 + 1$ distinct bit strings $A^*[3], A_{3,i,j}[3] \in \{0,1\}^n$ for $1 \leq i \leq l$ and $1 \leq j' \leq r_3$, it holds that*

$$\Pr\Big[\exists(i,p,q), \pi(A_{1,i,p}[1]) \oplus A^*[3] = \pi(A^*[1]) \oplus A_{3,i,q}[3]\Big] \geq \left(1 - \frac{1}{e}\right)\frac{lr_1r_3}{2^n}.$$

See Fig. 5 for the figure describing the event of Proposition 1[5]. The proof is elementary, and can be found in Appendix B.

[5] We use r_1 and r_3 instead of r_1 and r_2, since r_3 corresponds to the number of distinct blocks in the third block when we apply Proposition 1 in attacking Romulus-M.

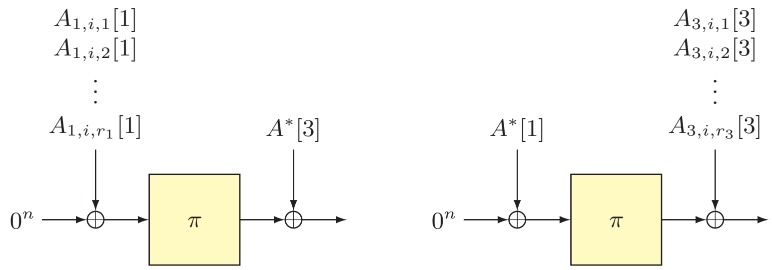

Fig. 5. The event in Proposition 1. For each $1 \leq i \leq l$, on the left, π takes r_1 distinct input values and a fixed value is XOR'ed to the output. On the left, π takes one fixed input value and r_3 distinct values are XOR'ed to the output, and we are interested in a collision between r_1 output values from the left and r_3 output values from the right.

We now have the following theorem regarding the authenticity security of Romulus-M.

Theorem 4. *For* Romulus-M*, there exists an authenticity adversary* $\mathcal{A}$ *with*

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \geq \frac{0.07 r q_e}{2^n},$$

where $\mathcal{A}$ *makes* $q_e = lr + 1$ *encryption queries and* $q_d = 1$ *decryption query, and* $l \geq 1$ *is a parameter.*

Proof. We fix $l \geq 1$, and let $N_1, \ldots, N_l$ be l distinct nonces that are different from N^*, the nonce in the challenge. We divide r as $r = r_1 + r_3$, where $r_1 = \lfloor r/2 \rfloor$ and $r_3 = \lceil r/2 \rceil$. We then prepare $lr_1 + lr_3$ AD

$$A_{1,1,1}, \ldots, A_{1,1,r_1}, \ldots, A_{1,l,1}, \ldots, A_{1,l,r_1}, \tag{2}$$

$$A_{3,1,1}, \ldots, A_{3,1,r_3}, \ldots, A_{3,l,1}, \ldots, A_{3,l,r_3}, \tag{3}$$

where lr_1 AD in Eq. (2) are distinct and are different from $A^*[1]$ in the first block, and lr_3 AD in Eq. (3) are distinct and are different from $A^*[3]$ in the third block. Specifically, for the challenge AD $A^* = (A^*[1], \ldots, A^*[a])$, let

$$A_{1,1,1}[1], \ldots, A_{1,1,r_1}[1], \ldots, A_{1,l,1}[1], \ldots, A_{1,l,r_1}[1] \in \{0,1\}^n \setminus \{A^*[1]\}$$

be lr_1 distinct n-bit strings, and we define $A_{1,i,j}$ as

$$A_{1,i,j} = A_{1,i,j}[1] \parallel A^*[2] \parallel \cdots \parallel A^*[a].$$

Similarly, we let

$$A_{3,1,1}[3], \ldots, A_{3,1,r_3}[3], \ldots, A_{3,l,1}[3], \ldots, A_{3,l,r_3}[3] \in \{0,1\}^n \setminus \{A^*[3]\}$$

be lr_3 distinct n-bit strings, and let

$$A_{3,i,j'} = A^*[1] \parallel A^*[2] \parallel A_{3,i,j'}[3] \parallel A^*[4] \parallel \cdots \parallel A^*[a].$$

Algorithm 2 Universal forgery attack on Romulus-M

for $i = 1, \ldots, l$ **do**
    for $j = 1, \ldots, r_1$ **do**
        $(C_{1,i,j}, T_{1,i,j}) \leftarrow \mathsf{Enc}_K(N_i, A_{1,i,j}, M^*)$
    end for
    for $j' = 1, \ldots, r_3$ **do**
        $(C_{3,i,j'}, T_{3,i,j'}) \leftarrow \mathsf{Enc}_K(N_i, A_{3,i,j'}, M^*)$
    end for
end for
if $T_{1,i,p} = T_{3,i,q}$ for some (i, p, q) **then**
    $A \leftarrow A_{1,i,p}[1] \parallel A^*[2] \parallel A_{3,i,q}[3] \parallel A^*[4] \parallel \cdots \parallel A^*[a]$
    $(C, T) \leftarrow \mathsf{Enc}_K(N^*, A, M^*)$
    return (N^*, A^*, C, T)
else
    return failure
end if

Having prepared all these bit strings, our universal forgery attack is presented in Algorithm 2. For each i, j, j', we encrypt $(N_i, A_{1,i,j}, M^*)$ and $(N_i, A_{3,i,j'}, M^*)$, and obtain $(C_{1,i,j}, T_{1,i,j})$ and $(C_{3,i,j'}, T_{3,i,j'})$. For each i, $T_{1,i,1}, \ldots, T_{1,i,r_1}$ are distinct random values with the same reasoning as in the proof of Theorem 3. We also observe that $T_{3,i,1}, \ldots, T_{3,i,r_3}$ are distinct random values.

Next, we search for a tuple of indices (i, p, q) such that $T_{1,i,p} = T_{3,i,q}$, which holds if and only if $\widetilde{E}_K^{40,\bar{1}}(A_{1,i,p}[1]) \oplus A^*[3] = \widetilde{E}_K^{40,\bar{1}}(A^*[1]) \oplus A_{3,i,q}[3]$ holds. See Fig. 6 illustrating this case. Proposition 1 gives the probability of this event, and we see

$$\Pr[1 \le \exists i \le l, 1 \le \exists p \le r_1, 1 \le \exists q \le r_3, T_{1,i,p} = T_{3,i,q}] \ge \left(1 - \frac{1}{e}\right) \frac{l r_1 r_3}{2^n}\,. \quad (4)$$

If we find a tuple of indices (i, p, q) that satisfies $T_{1,i,p} = T_{3,i,q}$, then we make an encryption query (N^*, A, M^*), where

$$A = A_{1,i,p}[1] \parallel A^*[2] \parallel A_{3,i,q}[3] \parallel A^*[4] \parallel \cdots \parallel A^*[a]$$

and obtain (C, T). Then we make a decryption query (N^*, A^*, C, T). The oracle returns (N^*, A^*, M^*), and the adversary succeeds in the universal forgery since $\widetilde{E}_K^{40,\bar{1}}(A_{1,i,p}[1]) \oplus A_{3,i,q}[3] = \widetilde{E}_K^{40,\bar{1}}(A^*[1]) \oplus A^*[3]$ holds. See Fig. 7 describing the encryption and decryption queries.

The adversary makes $l(r_1 + r_3) + 1$ encryption queries, followed by one decryption query. Since $r = r_1 + r_3$, it follows that the adversary makes $q_e = lr + 1$ encryption queries. The success probability of our attack is given by Eq. (4), and we thus have

$$\mathbf{Adv}^{\mathrm{auth}}_{\mathsf{Romulus\text{-}M}}(\mathcal{A}) \ge \left(1 - \frac{1}{e}\right) \frac{l r_1 r_3}{2^n} \ge \frac{1}{3}\left(1 - \frac{1}{e}\right) \frac{q_e r_1}{2^n} \ge \frac{0.07 r q_e}{2^n}$$

from $q_e = l(r_1 + r_3) + 1 \le 3 l r_3$ and $r = r_1 + r_3 \le 3 r_1$, and this completes the proof of Theorem 4. $\square$

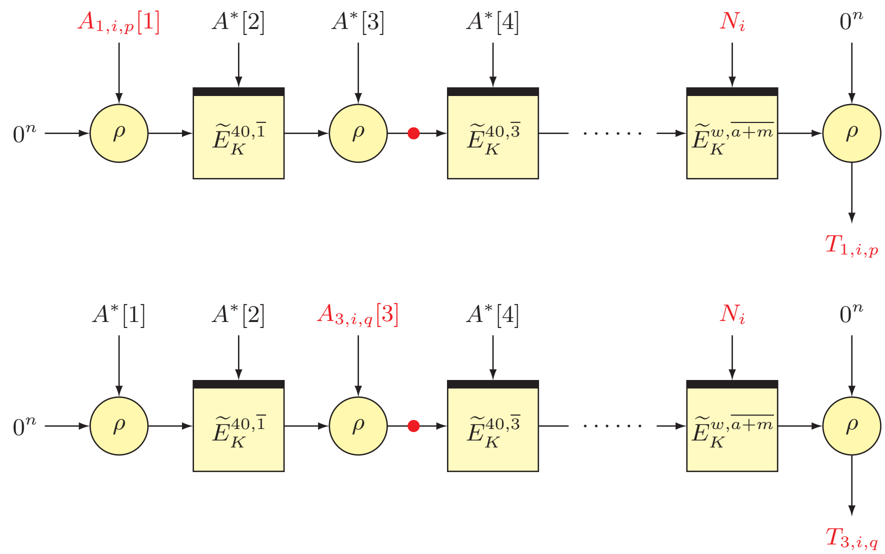

Fig. 6. Variables that depend on i, j, or j' are highlighted in red, and we are interested in the collision between the two red points. We see that $T_{1,i,p} = T_{3,i,q}$ holds iff $\widetilde{E}_K^{40,\bar{1}}(A_{1,i,p}[1]) \oplus A^*[3] = \widetilde{E}_K^{40,\bar{1}}(A^*[1]) \oplus A_{3,i,q}[3]$ holds.

6 Conclusions

In this paper, we have presented matching attacks on Romulus-M. Concretely, our distinguishing attack has an advantage of $0.03r\sigma_{\text{priv}}/2^n$, while the provable security bound is $4r\sigma_{\text{priv}}/2^n$, and our authenticity attack has an advantage of $0.07rq_e/2^n$, while the provable security bound is $(4rq_e + 5rq_d)/2^n$. Our authenticity attack is a universal forgery, which is the strongest attack scenario of authenticity. The results show that the provable security bounds of Romulus-M are tight for a large class of parameters.

The authenticity bound has two terms, where the last term is $O(rq_d/2^n)$. There is a trivial attack that gives a success probability of $O(q_d/2^n)$, while we do not know if there is a matching attack whose success probability scales with respect to r, and filling the gap is an open problem.

References

1. 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.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5, https://doi.org/10.1007/978-3-662-53008-5_5

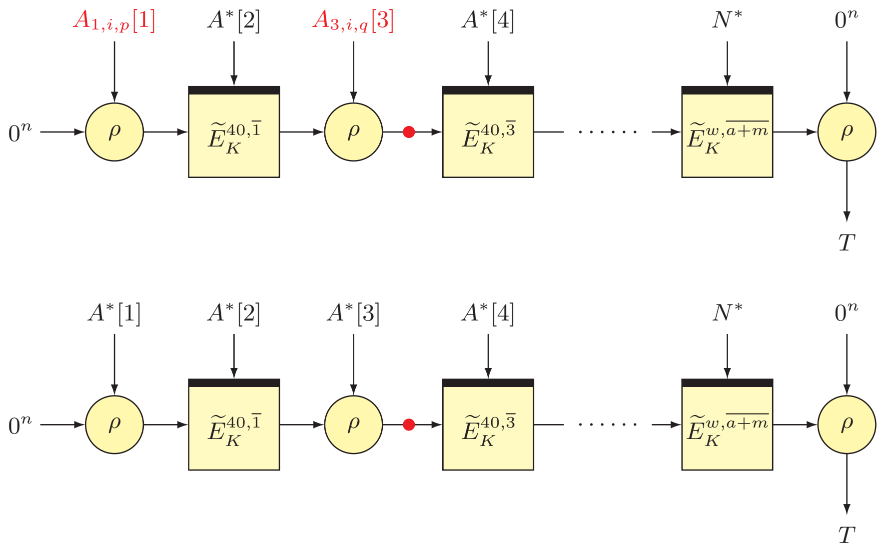

Fig. 7. The top figure is the encryption query, and the bottom one is the decryption query. The collision between two red points in Fig. 6 implies a collision between two red points in this figure, and the decryption query will be accepted.

2. Berti, F., Guo, C., Pereira, O., Peters, T., Standaert, F.: Tedt, a leakage-resist AEAD mode for high physical security applications. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2020**(1), 256–320 (2020). https://doi.org/10.13154/tches.v2020.i1.256-320, https://doi.org/10.13154/tches.v2020.i1.256-320
3. Guo, C., Iwata, T., Khairallah, M., Minematsu, K., Peyrin, T.: Romulus v1.3. Submission to the NIST lightweight cryptography standardization process (2021), https://csrc.nist.gov/Projects/lightweight-cryptography/
4. Guo, C., Iwata, T., Minematsu, K.: New indifferentiability security proof of MDPH hash function. IET Inf. Secur. (2022), to appear.
5. Iwata, T., Khairallah, M., Minematsu, K., Peyrin, T.: Duel of the titans: The Romulus and Remus families of lightweight AEAD algorithms. IACR Trans. Symmetric Cryptol. **2020**(1), 43–120 (2020). https://doi.org/10.13154/tosc.v2020.i1.43-120, https://doi.org/10.13154/tosc.v2020.i1.43-120
6. Jia, K., Wang, X., Yuan, Z., Xu, G.: Distinguishing and second-preimage attacks on cbc-like macs. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) Cryptology and Network Security, 8th International Conference, CANS 2009, Kanazawa, Japan, December 12-14, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5888, pp. 349–361. Springer (2009). https://doi.org/10.1007/978-3-642-10433-6_23, https://doi.org/10.1007/978-3-642-10433-6_23
7. Lee, J.: Security evaluation of Romulus. A third party security proof (2021), https://romulusae.github.io/romulus/security
8. Liskov, M.D., Rivest, R.L., Wagner, D.A.: Tweakable block ciphers. In: Yung, M. (ed.) Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryp-

tology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2442, pp. 31–46. Springer (2002). https://doi.org/10.1007/3-540-45708-9_3, https://doi.org/10.1007/3-540-45708-9_3

9. Liskov, M.D., Rivest, R.L., Wagner, D.A.: Tweakable block ciphers. J. Cryptol. **24**(3), 588–613 (2011). https://doi.org/10.1007/s00145-010-9073-y, https://doi.org/10.1007/s00145-010-9073-y
10. Liu, F., Liu, F.: Universal forgery with birthday paradox: Application to blockcipher-based message authentication codes and authenticated encryptions. IACR Cryptol. ePrint Arch. p. 653 (2017), http://eprint.iacr.org/2017/653
11. Naito, Y.: Optimally indifferentiable double-block-length hashing without post-processing and with support for longer key than single block. In: Schwabe, P., Thériault, N. (eds.) Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11774, pp. 65–85. Springer (2019). https://doi.org/10.1007/978-3-030-30530-7_4, https://doi.org/10.1007/978-3-030-30530-7_4
12. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002. pp. 98–107. ACM (2002). https://doi.org/10.1145/586110.586125, https://doi.org/10.1145/586110.586125
13. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4004, pp. 373–390. Springer (2006). https://doi.org/10.1007/11761679_23, https://doi.org/10.1007/11761679_23

A Proof of Eq. (1)

Here, we show Eq. (1), namely,

$$1-\left(\frac{(2^n)_r}{(2^n)^r}\right)^l \geq \left(1-\frac{1}{e}\right)\frac{0.25lr^2}{2^n}\,.$$

Let $p = 1-(2^n)_r/(2^n)^r$. Then we have

$$\begin{aligned} p &= 1-\prod_{1\leq i\leq r-1}\left(1-\frac{i}{2^n}\right) \geq 1-\prod_{1\leq i\leq r-1}\exp\left(-\frac{i}{2^n}\right) \\ &= 1-\exp\left(\frac{-0.5r(r-1)}{2^n}\right)\,, \end{aligned}$$

where the inequality uses the fact that $1 - x \leq \exp(-x)$ holds for any x and is obtained by setting $x = i/2^n$. We have

$$\begin{aligned}1 - (1-p)^l \geq 1 - \exp\left(\frac{-0.5lr(r-1)}{2^n}\right) &\geq \left(1 - \frac{1}{e}\right)\frac{0.5lr(r-1)}{2^n} \\ &\geq \left(1 - \frac{1}{e}\right)\frac{0.25lr^2}{2^n},\end{aligned}$$

where the second inequality follows from the fact that

$$\left(1 - \frac{1}{e}\right) x \leq 1 - \exp(-x) \tag{5}$$

holds for $0 \leq x \leq 1$, and the last one uses $r(r-1) \geq r^2/2$ for $r \geq 2$. $\square$

B Proof of Proposition 1

We consider the complementary event of Proposition 1, which is

$$\Pr\left[\forall(i,p,q), \pi(A_{1,i,p}[1]) \oplus A^*[3] \neq \pi(A^*[1]) \oplus A_{3,i,q}[3]\right], \tag{6}$$

and derive its upper bound. Now, we claim that Eq. (6) is upper bounded by

$$\frac{2^n \cdot [(2^n - (r_3+1))_{r_1}]^l}{(2^n)_{lr_1+1}}. \tag{7}$$

To see this, the denominator is $(2^n)_{lr_1+1}$, since we are dealing with lr_1+1 input-output pairs of π. For the numerator, we first arbitrary fix $\pi(A^*[1])$. Then we count the number of possible choices of $(\pi(A_{1,1,1}[1]), \ldots, \pi(A_{1,1,r_1}[1]))$ such that the set of r_1 elements $\{\pi(A_{1,1,1}[1]) \oplus A^*[3], \ldots, \pi(A_{1,1,r_1}[1]) \oplus A^*[3]\}$ and the set of r_3 elements $\{\pi(A^*[1]) \oplus A_{3,1,1}[3], \ldots, \pi(A^*[1]) \oplus A_{3,1,r_3}[3]\}$ are disjoint. That is, we require that the following two sets are disjoint.

$$\begin{cases}\{\pi(A_{1,1,1}[1]), \ldots, \pi(A_{1,1,r_1}[1])\}, \\ \{\pi(A^*[1]) \oplus A_{3,1,1}[3] \oplus A^*[3], \ldots, \pi(A^*[1]) \oplus A_{3,1,r_3}[3] \oplus A^*[3]\}.\end{cases}$$

Observe that the last set is a fixed set of r_3 elements, and that $\pi(A^*[1])$ is not included in the last set, since $A_{3,1,1}[3] \oplus A^*[3], \ldots, A_{3,1,r_3}[3] \oplus A^*[3]$ are all non-zero.

Therefore, the number of possible choices of $(\pi(A_{1,1,1}[1]), \ldots, \pi(A_{1,1,r_1}[1]))$ is at most $(2^n - (r_3+1))_{r_1}$. We obtain Eq. (7) by applying the same upper bound for $(\pi(A_{1,i,1}[1]), \ldots, \pi(A_{1,i,r_1}[1]))$ for $2 \leq i \leq l$.

Next, we have

$$
\begin{aligned}
\text{Eq. (7)} &\le \prod_{0\le j\le l-1}\prod_{1\le i\le r_1}\frac{2^n-(r_3+i)}{2^n-(jr_1+i)} \\
&\le \left(1-\frac{r_3}{2^n-r_1}\right)^{r_1}\left(1-\frac{r_3}{2^n-2r_1}\right)^{r_1}\cdots\left(1-\frac{r_3}{2^n-lr_1}\right)^{r_1} \\
&\le \left(1-\frac{r_3}{2^n}\right)^{lr_1}.
\end{aligned}
$$

The claimed bound in Proposition 1 is obtained as

$$
1-\left(1-\frac{r_3}{2^n}\right)^{lr_1} \ge 1-\exp\left(-\frac{lr_1r_3}{2^n}\right) \ge \left(1-\frac{1}{e}\right)\frac{lr_1r_3}{2^n},
$$

where we used $1-x\le\exp(-x)$ for the first inequality, and Eq. (5) for the last one. $\square$