Multi-User Security of the Elephant v2 Authenticated Encryption Mode

Tim Beyne[1], Yu Long Chen[1], Christoph Dobraunig[2], and Bart Mennink[3]

[1] KU Leuven and imec-COSIC, Leuven, Belgium
[2] Lamarr Security Research, Graz, Austria
[3] Radboud University, Nijmegen, The Netherlands
elephant@cs.ru.nl

Abstract. One of the finalists in the NIST Lightweight Cryptography competition is Elephant v2, a parallelizable, permutation-based authenticated encryption scheme. The original first/second-round submission Elephant v1/v1.1 was proven secure against nonce-respecting adversaries in the single-user setting. For the final round, the mode has undergone certain subtle modifications, the most important one being a change in the authentication portion of the mode. These changes require a new dedicated security proof.
In this work, we prove the security of the Elephant v2 mode. First of all, our proof shows that Elephant v2 is indeed a secure authenticated encryption scheme and that its security against nonce-respecting adversaries is on par with that of Elephant v1/v1.1. In addition, our security analysis is in the multi-user setting and demonstrates that Elephant v2 fares well if multiple devices use Elephant v2 with independent keys. Moreover, our proof shows that Elephant v2 even ensures authenticity under nonce misuse.

Keywords: authenticated encryption, lightweight, Elephant, multi-user security, nonce-misuse

1 Introduction

Authenticated encryption schemes aim to provide confidentiality and authenticity of data. If one aims to prove that a mode achieves these security properties, one typically models an adversary that has access to a single instance of the cryptographic mode (for a single secret key) and aims to break either of these properties. In practice, however, cryptographic modes are rarely used in isolation. Instead, a mode might be used by millions of people worldwide at the same time, with independent keys, and the attacker might aim to attack all instances at the same time.

This particular setting, i.e., where the adversary might attack multiple independent instances at the same time, is called the *multi-user* security setting. Biham [12] observed that recovering one key out of many is much easier than recovering only one specific key. Intuitively, in general, the adversarial success

probability increases by a factor μ if an attacker has μ instead of 1 mode at its disposal. Bellare et al. [4] proved that this μ-factor loss is tight, generically. However, for specific schemes, this μ-factor difference might induce an artificial loss in the security bound and a dedicated analysis might be more accurate. For example, Bellare and Tackmann [5] proved multi-user security of authenticated encryption in the TLS 1.3 protocol, likewise exhibiting a μ-factor loss, but Luykx et al. [22] demonstrated that a more precise analysis yields a better multi-user security bound. Likewise, for permutation-based analysis, Daemen et al. [16] derived a multi-user security bound for the keyed duplex that can, in turn, be used for sponge-based authenticated encryption. Their bound is better than the generic reduction with μ-factor loss, in the sense that their security result precisely indicates which terms in the bound are affected by the availability of multiple keys.

The issue of multi-user security is particularly important in the lightweight setting. One reason is that the internet of things leads to an increase in the number of constrained devices that often interoperate and deal with sensitive information. Another reason is that lightweight cryptographic solutions often use relatively small cryptographic primitives and operate with a smaller security margin. Therefore, an accurate estimation, rather than a rough bound, of the multi-user security of such a scheme is valuable.

In 2018, the US National Institute of Standards and Technology (NIST) initiated a competition for the design of lightweight cryptographic authenticated encryption and hashing [26]. NIST received 57 submissions, the second round covered 32 submissions, and the ongoing final round contains 10 candidates. Some candidates, namely ASCON [19], ISAP [17,18], Sparkle [2,3], and Xoodyak [14, 15], are based on the duplex construction and, at least to a certain extent, can rely on the multi-user security of the duplex [16]. Khairallah [21] analyzed multi-user security of GIFT-COFB [1] from a cryptanalytic perspective. We are not aware of other multi-user security analyses of the finalists.

1.1 Elephant

One of the finalists in the NIST Lightweight Cryptography competition is Elephant of Beyne et al. [9]. The mode of Elephant is an encrypt-then-MAC construction, with encryption performed using counter mode and authentication using a variant of the protected counter sum [6,23] MAC function. Elephant is a permutation-based construction, but, unlike sponge-based authenticated encryption, it is *parallelizable*. This way, Elephant is shown [9] to be able to operate using a cryptographic primitive that is significantly smaller than that of its competitors.

The Elephant mode itself has undergone a subtle change in its move to the final round. The original submission (of the first and second round) Elephant v1/v1.1 (henceforth simply referred to as v1) [8] performed authentication using the Wegman-Carter-Shoup [7, 28, 29] MAC function, but the final-round version Elephant v2 [11] relies on a variant of the protected counter sum [6, 23] MAC function. The transition was pre-announced in a status update [10], and was

defended by the observation that, while v1 only achieved confidentiality and authenticity in the nonce-respecting setting, v2 *additionally* achieves authenticity under nonce-reuse. We will elaborate on the differences between v1 and v2 in Section 4.3.

This change is subtle but the earlier security proof of Elephant v1 [9] is no longer applicable to Elephant v2. In addition, nonce-misuse security is yet unclear. Finally, the earlier proof of Elephant v1 does not consider the multi-user setting.

1.2 Multi-User Security of Elephant v2

In this work, we prove security of Elephant v2 [11]. The proof differs from the one of v1 [9] in that the new authenticator is considered. In addition, our result shows that Elephant v2 indeed achieves authenticity under nonce-misuse. Finally, our new proof is in the multi-user setting.

Due to the similarity between Elephant v1 and v2, the overall structure of the proof is comparable. However, the required changes to the security analysis are non-trivial. As a first step, we investigate the Simplified Masked Even-Mansour (SiM) of [9] and derive a multi-user security bound in Section 3. The derived security bound is "tight" in the number of users, in the sense that if one considers a single user, the original bound of [9] is retained. In Section 3.2, we highlight how the security analysis of SiM differs in our new proof. The second step is to prove the multi-user security of Elephant v2. The mode is first described in detail in Section 4, where we also give a treatment of the differences between v1 and v2. A difficulty in the multi-user security analysis of Elephant v2 is that one considers multiple independently keyed modes, but all these modes *employ* the same underlying cryptographic permutation. By reducing the multi-user security of the Elephant v2 mode to the multi-user security of SiM, which was proven in the first step, we isolate the usage of the same permutation in the modular building block SiM and can consider multi-user security of Elephant v2 as multiple completely independent instantiations. Then, its multi-user security can be upper bounded by the sum of μ single-user instances, where the sum is maximized over any possible distribution of the adversarial complexity over the μ instances. A final change in the security proof is that the proof properly separates Elephant v2 into a nonce-based encryption and a nonce-independent authentication part, where the proof of authenticity subsequently stands under nonce-misuse.

1.3 Outline

We describe basic notation and the multi-user security models for tweakable block ciphers and authenticated encryption schemes in Section 2. A multi-user security bound on the Simplified Masked Even-Mansour (SiM) tweakable block cipher is given in Section 3. In Section 4, we focus on Elephant: a description of Elephant v2 is given in Section 4.1, a comparison between Elephant v1 and v2 in Section 4.3, and a multi-user security bound for Elephant v2 in Section 4.2. The

multi-user security proofs of SiM and Elephant v2 are given in Appendix A and Section 5, respectively. We conclude the work in Section 6.

2 Security Model

For $n \in \mathbb{N}$, $\{0,1\}^n$ denotes the set of n-bit strings and $\{0,1\}^*$ the set of arbitrarily length strings. For $X \in \{0,1\}^*$, we define

$$X_1 \dots X_\ell \xleftarrow{n} X \tag{1}$$

as the function that partitions X into $\ell = \lceil |X|/n \rceil$ blocks of size n bits, where the last block is padded with 0s. The expression "A ? B : C" equals B if A is true, and equals C if A is false. We denote by $\lfloor x \rfloor_i$ the i leftmost bits of x.

For a finite set $\mathcal{T}$, we denote by $\mathrm{perm}(n)$ the set of all n-bit permutations and by $\mathrm{perm}(\mathcal{T}, n)$ the set of all families of permutations indexed by $T \in \mathcal{T}$. We denote by $\mathrm{func}(n)$ the set of all n-bit to n-bit functions. For a finite set $\mathcal{S}$, we denote by $s \xleftarrow{\$} \mathcal{S}$ the uniform random sampling of an element s from $\mathcal{S}$.

An adversary $\mathcal{A}$ is an algorithm that is given access to one or more oracles $\mathcal{O}$, and after interaction with $\mathcal{O}$ outputs a bit $b \in \{0,1\}$. This event is denoted as $\mathcal{A}^{\mathcal{O}} \to b$. In our work, we will be concerned with computationally unbounded adversaries $\mathcal{A}$; their complexities are only measured by the number of oracle queries. For two randomized oracles $\mathcal{O}$ and $\mathcal{P}$, we denote the advantage of an adversary $\mathcal{A}$ in distinguishing between them by

$$\Delta_{\mathcal{A}}(\mathcal{O}\ ;\ \mathcal{P}) = \left| \mathbf{Pr}\left(\mathcal{A}^{\mathcal{O}} \to 1\right) - \mathbf{Pr}\left(\mathcal{A}^{\mathcal{P}} \to 1\right) \right| . \tag{2}$$

Finally, let $k, m, n, t, \mu \in \mathbb{N}$ with $k, m, t \leq n$ throughout.

2.1 Tweakable Block Ciphers

A tweakable block cipher $\widetilde{\mathsf{E}}$ is a function that gets as input a key $K \in \{0,1\}^k$, tweak $T \in \mathcal{T}$,[4] and message $M \in \{0,1\}^n$, and outputs a ciphertext $C \in \{0,1\}^n$. The tweakable block cipher $\widetilde{\mathsf{E}}$ is required to be bijective for any fixed (K, T).

Elephant only uses its underlying primitive in the forward direction. This means that we only need to consider tweakable block ciphers that are secure against adversaries that only have access to $\widetilde{\mathsf{E}}$, and not to $\widetilde{\mathsf{E}}^{-1}$. The tweakable block cipher considered in this work is based on an n-bit permutation P, which is modeled as a random permutation $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$. We will consider the multi-user security of $\widetilde{\mathsf{E}}$, where an adversary can query up to μ instances of the scheme. Specifically, the μ-tprp security of $\widetilde{\mathsf{E}}$ against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-tprp}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\widetilde{\mathsf{E}}^{\mathsf{P}}_{K_j}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\ ;\ \left(\widetilde{\pi}_j\right)_{j=1}^{\mu}, \mathsf{P}^{\pm} \right) , \tag{3}$$

where the randomness is taken over independent $K_1, \dots, K_\mu \xleftarrow{\$} \{0,1\}^k$, $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, and $\widetilde{\pi}_1, \dots, \widetilde{\pi}_\mu \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n)$.

[4] In our application, the tweak space is of a specific form and cannot be conveniently expressed as a set of binary strings.

2.2 Authenticated Encryption

An authenticated encryption scheme AE consists of two algorithms enc and dec. Encryption enc takes a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, and a message $M \in \{0,1\}^*$ as inputs, and it outputs a ciphertext $C \in \{0,1\}^{|M|}$ and a tag $T \in \{0,1\}^t$. Decryption dec takes a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^t$ as inputs, and it outputs a message $M \in \{0,1\}^{|C|}$ if the tag is correct, or a dedicated $\perp$-sign otherwise. The two functions are required to satisfy

$$\mathsf{dec}(K, N, A, \mathsf{enc}(K, N, A, M)) = M\,.$$

(Note that the output of $\mathsf{enc}(K, N, A, M)$ consists of a tuple (C, T), and hence, dec gets as input a tuple of the form (K, N, A, C, T).)

In our work, the authenticated encryption scheme AE is based on an n-bit permutation P, which is modeled as a random permutation: $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$. We will consider multi-user security of AE, where an adversary can query up to μ versions of the scheme. The multi-user security of AE against an adversary $\mathcal{A}$ is defined as

$$\mathbf{Adv}^{\mu\text{-ae}}_{\mathsf{AE}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left(\left(\mathsf{enc}^{\mathsf{P}}_{K_j}, \mathsf{dec}^{\mathsf{P}}_{K_j}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\ ;\ \left(\mathsf{rand}_j, \perp\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\right)\,, \tag{4}$$

where the randomness is taken over $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$, $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, and the functions $\mathsf{rand}_1, \ldots, \mathsf{rand}_\mu$ that for each new input (N, A, M) return a random string of size $|M| + t$ bits. The superscript $^{\pm}$ indicates two-sided access by $\mathcal{A}$. The function $\perp$ returns the $\perp$-sign for each query.

The adversary $\mathcal{A}$ is not allowed to relay the output of the encryption oracle (enc_{K_j} in the real world and rand_j in the ideal world) to the decryption oracle (dec_{K_j} in the real world and $\perp$ in the ideal world). Adversary $\mathcal{A}$ is called *nonce-respecting* if it does not make two encryption queries to the same oracle $j \in \{1, \ldots, \mu\}$ for the same nonce.

2.3 Authentication

If we restrict our focus to authentication only of an authenticated encryption scheme, we consider an adversary $\mathcal{A}$ that can query the encryption functionality enc at its discretion, and it wins if it can submit a tuple to dec that (i) succeeds and (ii) was not the result of an earlier encryption query. More formally, for an authenticated encryption scheme AE based on an n-bit permutation P, which is modeled as a random permutation $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, we will define multi-user authenticity of AE, where an adversary can query up to μ versions of the scheme, as follows:

$$\mathbf{Adv}^{\mu\text{-auth}}_{\mathsf{AE}}(\mathcal{A}) = \mathbf{Pr}\left(\mathcal{A}^{\left(\mathsf{enc}^{\mathsf{P}}_{K_j}, \mathsf{dec}^{\mathsf{P}}_{K_j}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}} \text{ forges}\right)\,, \tag{5}$$

where the randomness is taken over $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ and $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, and where a successful forgery corresponds to a query to any of the decryption oracles $\mathsf{dec}_{K_j}^{\mathsf{P}}$ that returned a valid message. The adversary $\mathcal{A}$ is not allowed to relay the output of the encryption oracle enc_{K_j} to the decryption oracle dec_{K_j}. For authenticity, we do not restrict the adversary in its choice of nonce.

3 Simplified Masked Even-Mansour

The Elephant authenticated encryption family uses its underlying permutation in a "Masked Even-Mansour" (MEM) construction [20]: the input to and output of the permutation P are masked using an LFSR (Linear Feedback Shift Register) evaluated on the secret key. To be more precise, Beyne et al. [9] described a simplification, namely "Simplified MEM" (SiM). Their arguments in favor of SiM were that (i) the tweak only consists of the exponents of the LFSRs and the nonce is not a tweak input to SiM, and (ii) Elephant only uses its primitive in the forward direction, which allows for looser conditions on the primitive.

We will describe SiM as introduced by Beyne et al. [9] in Section 3.1. In Section 3.2, we derive a multi-user security bound on the SiM mode and explain how it differs from the single-user analysis of [9].

3.1 Specification

Let $k, n, z \in \mathbb{N}$. Let $\mathsf{P} \in \mathrm{perm}(n)$ be an n-bit permutation, and let $\varphi_1, \ldots, \varphi_z : \{0,1\}^n \to \{0,1\}^n$ be z LFSRs. Let $\mathcal{T} \subseteq \mathbb{N}^z$ be a finite tweak space. Define the function $\mathsf{mask} : \{0,1\}^k \times \mathcal{T} \to \{0,1\}^n$ as follows:

$$\mathsf{mask}_K^{a_1,\ldots,a_z} = \mathsf{mask}(K, a_1, \ldots, a_z) = \varphi_z^{a_z} \circ \cdots \circ \varphi_1^{a_1} \circ \mathsf{P}(K \| 0^{n-k}) \,. \tag{6}$$

Define the tweakable block cipher $\mathsf{SiM} : \{0,1\}^k \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ as

$$\mathsf{SiM}(K, (a_1, \ldots, a_z), M) = \mathsf{P}(M \oplus \mathsf{mask}_K^{a_1,\ldots,a_z}) \oplus \mathsf{mask}_K^{a_1,\ldots,a_z} \,. \tag{7}$$

3.2 Multi-User Security of SiM

Beyne et al. [9] imposed a restriction on the tweak space $\mathcal{T}$ in order for SiM to be a secure tweakable block cipher, that was in turn taken from Granger et al. [20]. This restriction does not change for the multi-user setting, and we state it for completeness. We say that $\mathcal{T}$ is $2^{-\alpha}$-proper with respect to $(\varphi_1, \ldots, \varphi_z)$ if the function $L \mapsto \varphi_z^{a_z} \circ \cdots \circ \varphi_1^{a_1}(L)$ is $2^{-\alpha}$-uniform and $2^{-\alpha}$-XOR-uniform.

Definition 1. *Let $n, z \in \mathbb{N}$. Let $\varphi_1, \ldots, \varphi_z : \{0,1\}^n \to \{0,1\}^n$ be z LFSRs. The tweak space $\mathcal{T}$ is called* $2^{-\alpha}$-proper *with respect to $(\varphi_1, \ldots, \varphi_z)$ if the following two properties hold:*

1. *For any $Y \in \{0,1\}^n$ and $(a_1, \ldots, a_z) \in \mathcal{T} \cup \{(0, \ldots, 0)\}$,*

$$\mathbf{Pr}\left(L \xleftarrow{\$} \{0,1\}^n \;:\; \varphi_z^{a_z} \circ \cdots \circ \varphi_1^{a_1}(L) = Y\right) \leq 2^{-\alpha} \,;$$

2. *For any* $Y \in \{0,1\}^n$ *and distinct* $(a_1,\ldots,a_z),(a'_1,\ldots,a'_z) \in \mathcal{T} \cup \{(0,\ldots,0)\}$,

$$\mathbf{Pr}\left(L \xleftarrow{\$} \{0,1\}^n \;:\; \varphi_z^{a_z} \circ \cdots \circ \varphi_1^{a_1}(L) \oplus \varphi_z^{a'_z} \circ \cdots \circ \varphi_1^{a'_1}(L) = Y\right) \leq 2^{-\alpha}\,.$$

We can now prove that if the tweak space is $2^{-\alpha}$-proper for sufficiently small $2^{-\alpha}$ (note that $2^{-\alpha}$ cannot be smaller than 2^{-n}), then SiM is a *multi-user* secure tweakable block cipher.

Theorem 1. *Let* $k,n,z,\mu \in \mathbb{N}$. *Let* $\varphi_1,\ldots,\varphi_z : \{0,1\}^n \to \{0,1\}^n$ *be* z *LFSRs, and let* $\mathcal{T}$ *be a* $2^{-\alpha}$*-proper tweak space with respect to* $(\varphi_1,\ldots,\varphi_z)$. *Consider* SiM *of (7) based on a random permutation* $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$. *For any multi-user adversary* $\mathcal{A}$ *making at most* $q \leq 2^{n-1}$ *construction queries (in total to all* μ *construction oracles) and* p *primitive queries,*

$$\mathbf{Adv}^{\mu\text{-tprp}}_{\mathsf{SiM}}(\mathcal{A}) \leq \frac{q^2+2qp+(\mu-1)q}{2^\alpha} + \frac{\mu\cdot(2q+p+\frac{\mu-1}{2})}{2^n} + \frac{\mu\cdot(p+\frac{\mu-1}{2})}{2^k}\,.$$

The main difference with the single-user proof of [9] consists of bad events related to key collisions (details on the differences are given in the proof). Another change is that the analyses of the probability of certain transcripts to occur have changed slightly to facilitate the multi-user setting. When we restrict Theorem 1 to the single-user setting (i.e., $\mu = 1$), the bound of [9] is retained. As the changes compared to the proof of v1 are rather isolated, the proof is given in Appendix A.

4 Elephant Authenticated Encryption

We consider the Elephant v2 authenticated encryption mode as submitted by Beyne et al. to the final round of the NIST Lightweight Cryptography competition [11]. It is described in Section 4.1. The main Theorem 2, bounding the multiple-user security of Elephant v2 in terms of the multi-user security of SiM, is given in Section 4.2. In Section 4.3, we reflect on the differences between Elephant v2 and v1 [8], and explain how the security bound has improved.

4.1 Specification of Elephant v2

Let $k,m,n,t \in \mathbb{N}$ with $k,m,t \leq n$. Let $\mathsf{P} : \{0,1\}^n \to \{0,1\}^n$ be an n-bit permutation, and $\varphi_1 : \{0,1\}^n \to \{0,1\}^n$ be an LFSR. Define $\varphi_2 = \varphi_1 \oplus \mathsf{id}$, where id is the identity function. Define the function $\mathsf{mask} : \{0,1\}^k \times \mathbb{N}^2 \to \{0,1\}^n$ as follows:

$$\mathsf{mask}_K^{a,b} = \mathsf{mask}(K,a,b) = \varphi_2^b \circ \varphi_1^a \circ \mathsf{P}(K\|0^{n-k})\,. \tag{8}$$

We next describe the authenticated encryption mode of Elephant v2.

4.1.1 Encryption.

Encryption enc takes as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, and a message $M \in \{0,1\}^*$, and it outputs a ciphertext $C \in \{0,1\}^{|M|}$ and a tag $T \in \{0,1\}^t$. The description of enc is given in Algorithm 1, and it is depicted in Figure 1.

Algorithm 1 Elephant v2 encryption algorithm enc

Input: $(K, N, A, M) \in \{0,1\}^k \times \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^*$
Output: $(C, T) \in \{0,1\}^{|M|} \times \{0,1\}^t$
1: $M_1 \dots M_{\ell_M} \xleftarrow{n} M$
2: **for** $i = 1, \dots, \ell_M$ **do**
3: $\quad C_i \leftarrow M_i \oplus \mathsf{P}(N \| 0^{n-m} \oplus \mathsf{mask}_K^{i-1,1}) \oplus \mathsf{mask}_K^{i-1,1}$
4: $C \leftarrow \lfloor C_1 \dots C_{\ell_M} \rfloor_{|M|}$
5: $A_1 \dots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
6: $C_1 \dots C_{\ell_C} \xleftarrow{n} C \| 1$
7: $T \leftarrow A_1$
8: **for** $i = 2, \dots, \ell_A$ **do**
9: $\quad T \leftarrow T \oplus \mathsf{P}(A_i \oplus \mathsf{mask}_K^{i-1,0}) \oplus \mathsf{mask}_K^{i-1,0}$
10: **for** $i = 1, \dots, \ell_C$ **do**
11: $\quad T \leftarrow T \oplus \mathsf{P}(C_i \oplus \mathsf{mask}_K^{i-1,2}) \oplus \mathsf{mask}_K^{i-1,2}$
12: $T \leftarrow \mathsf{P}(T \oplus \mathsf{mask}_K^{0,0}) \oplus \mathsf{mask}_K^{0,0}$
13: **return** $(C, \lfloor T \rfloor_t)$

4.1.2 Decryption. Decryption dec gets as input a key $K \in \{0,1\}^k$, a nonce $N \in \{0,1\}^m$, associated data $A \in \{0,1\}^*$, a ciphertext $C \in \{0,1\}^*$, and a tag $T \in \{0,1\}^t$, and it outputs a message $M \in \{0,1\}^{|M|}$ if the tag is correct, or a dedicated $\perp$-sign otherwise. The description of dec is given in Algorithm 2.

4.2 Multi-User Security of Elephant v2

We prove security of Elephant v2 of Section 4.1 for any $2^{-\alpha}$-proper tweak space.

Theorem 2. *Let $k, m, n, t, \mu \in \mathbb{N}$ with $k, m, t \leq n$. Let $\varphi_1, \varphi_2 : \{0,1\}^n \to \{0,1\}^n$ be LFSRs, and let $\mathcal{T}$ be a $2^{-\alpha}$-proper tweak space with respect to (φ_1, φ_2). Consider* Elephant-v2 $= (\mathsf{enc}, \mathsf{dec})$ *of Section 4.1 based on random permutation $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$. For any multi-user adversary $\mathcal{A}$ making at most $q_e \leq 2^{n-1}$ construction encryption queries (in total to all μ construction oracles) for unique nonces whenever the same oracle is queried, q_d construction decryption queries (in total to all μ construction oracles), each query at most ℓ padded nonce and associated data and message blocks, and in total at most σ padded nonce and associated data and message blocks, and p primitive queries,*

$$\mathbf{Adv}^{\mu\text{-ae}}_{\text{Elephant-v2}}(\mathcal{A}) \leq \ell \binom{q_e}{2} / 2^n + 3 \binom{q_e + q_d}{2} / 2^n + q_d / 2^t + \mathbf{Adv}^{\mu\text{-tprp}}_{\mathsf{SiM}}(\mathcal{A}') \,,$$

for some multi-user adversary $\mathcal{A}'$ that makes 2σ construction queries and p primitive queries.

The proof of Theorem 2 is given in Section 5 and is significantly different from that of Elephant v1 [9]. In a nutshell, the first change is that the new proof is in the multi-user setting, where the adversary has access to multiple instances, all

Algorithm 2 Elephant v2 decryption algorithm dec

Input: $(K, N, A, C, T) \in \{0,1\}^k \times \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^t$
Output: $M \in \{0,1\}^{|C|}$ or $\perp$

$C_1 \dots C_{\ell_M} \xleftarrow{n} C$
for $i = 1, \dots, \ell_M$ **do**
    $M_i \leftarrow C_i \oplus \mathsf{P}(N \| 0^{n-m} \oplus \mathsf{mask}_K^{i-1,1}) \oplus \mathsf{mask}_K^{i-1,1}$
$M \leftarrow \lfloor M_1 \dots M_{\ell_M} \rfloor_{|C|}$
$A_1 \dots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
$C_1 \dots C_{\ell_C} \xleftarrow{n} C \| 1$
$\bar{T} \leftarrow A_1$
for $i = 2, \dots, \ell_A$ **do**
    $\bar{T} \leftarrow \bar{T} \oplus \mathsf{P}(A_i \oplus \mathsf{mask}_K^{i-1,0}) \oplus \mathsf{mask}_K^{i-1,0}$
for $i = 1, \dots, \ell_C$ **do**
    $\bar{T} \leftarrow \bar{T} \oplus \mathsf{P}(C_i \oplus \mathsf{mask}_K^{i-1,2}) \oplus \mathsf{mask}_K^{i-1,2}$
$\bar{T} \leftarrow \mathsf{P}(\bar{T} \oplus \mathsf{mask}_K^{0,0}) \oplus \mathsf{mask}_K^{0,0}$
return $\lfloor \bar{T} \rfloor_t = T \ ? \ M \ : \ \perp$

instantiated with an independent key but with the same underlying permutation. This is resolved by performing a reduction to SiM and therewith ending up with a simplified version of Elephant v2 for which the adversarial μ-user success probability is bounded by a sum of μ single-user success probabilities, maximized over all possible distributions of adversarial complexities. Finally, the authentication of Elephant v2 is different from that of Elephant v1, and a new proof for the authentication portion is given.

A final remark on the security analysis of Theorem 2 is that, although the theorem restricts itself to nonce-respecting adversaries, this nonce-respecting behavior is *only* used for the portion of the proof related to confidentiality. The Elephant v2 mode thus even achieves *authenticity* under nonce-reuse, up to above bound minus $\ell\binom{q_e}{2}/2^n$ (coming from the confidentiality portion of the proof).

Corollary 1. *Let $k, m, n, t, \mu \in \mathbb{N}$ with $k, m, t \leq n$. Let $\varphi_1, \varphi_2 : \{0,1\}^n \to \{0,1\}^n$ be LFSRs, and let $\mathcal{T}$ be a $2^{-\alpha}$-proper tweak space with respect to (φ_1, φ_2). Consider* Elephant-v2 $= (\mathsf{enc}, \mathsf{dec})$ *of Section 4.1 based on random permutation $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$. For any multi-user adversary $\mathcal{A}$ making at most $q_e \leq 2^{n-1}$ construction encryption queries (in total to all μ construction oracles), q_d construction decryption queries (in total to all μ construction oracles), each query at most ℓ padded nonce and associated data and message blocks, and in total at most σ padded nonce and associated data and message blocks, and p primitive queries,*

$$\mathbf{Adv}^{\mu\text{-auth}}_{\text{Elephant-v2}}(\mathcal{A}) \leq 3\binom{q_e + q_d}{2}/2^n + q_d/2^t + \mathbf{Adv}^{\mu\text{-tprp}}_{\mathsf{SiM}}(\mathcal{A}'),$$

for some multi-user adversary $\mathcal{A}'$ that makes 2σ construction queries and p primitive queries.

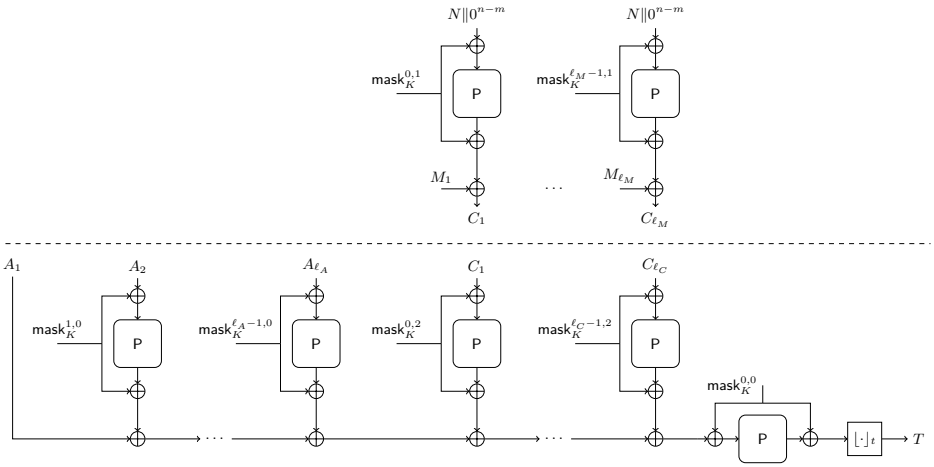

Fig. 1: Depiction of Elephant v2. For the encryption part (top): message is padded as $M_1 \dots M_{\ell_M} \xleftarrow{n} M$, and ciphertext equals $C = \lfloor C_1 \dots C_{\ell_M} \rfloor_{|M|}$. For the authentication part (bottom): nonce and associated data are padded as $A_1 \dots A_{\ell_A} \xleftarrow{n} N\|A\|1$, and ciphertext is padded as $C_1 \dots C_{\ell_C} \xleftarrow{n} C\|1$.

4.3 Comparison With Elephant v1

We will discuss how Theorem 2 improves over the result of Beyne et al. [9]. Their result was for Elephant v1, which is depicted in Figure 2.

4.3.1 Comparison Between Elephant v1 and v2. Before discussing Theorem 2 in more detail, we first consider the actual changes between Elephant v1 of Figure 2 and Elephant v2 of Figure 1.

The main change is in the authentication. Elephant v1 performed authentication using the Wegman-Carter-Shoup [7, 28, 29] MAC function. To be precise, the Wegman-Carter-Shoup function is defined as a nonce-based MAC scheme $\mathsf{mac}_{K,L}(N, A, C) = \mathsf{E}_K(N) \oplus \mathsf{H}_L(A, C)$, where E is a block cipher and H a universal hash function. In Elephant v1, the block cipher is instantiated as[5]

$$\mathsf{E}_K(\cdot) = \mathsf{P}(\cdot \oplus \mathsf{mask}_K^{0,2}) \oplus \mathsf{mask}_K^{0,2} ,$$

and the universal hash function as an XOR of tweakable block cipher evaluations. In Elephant v2, one uses a similar type of universal hash function, but this time, H_L processes all of (N, A, C), where the first data block is not evaluated by a tweakable block cipher but instead added in the clear. The output of H_L is then "protected" with a block cipher call instantiated as

$$\mathsf{E}_K(\cdot) = \mathsf{P}(\cdot \oplus \mathsf{mask}_K^{0,0}) \oplus \mathsf{mask}_K^{0,0} .$$

[5] As a matter of fact, in Elephant v2, the nonce is smaller than the state size of the permutation and is appended with associated data bits. This does not change the overall argument.

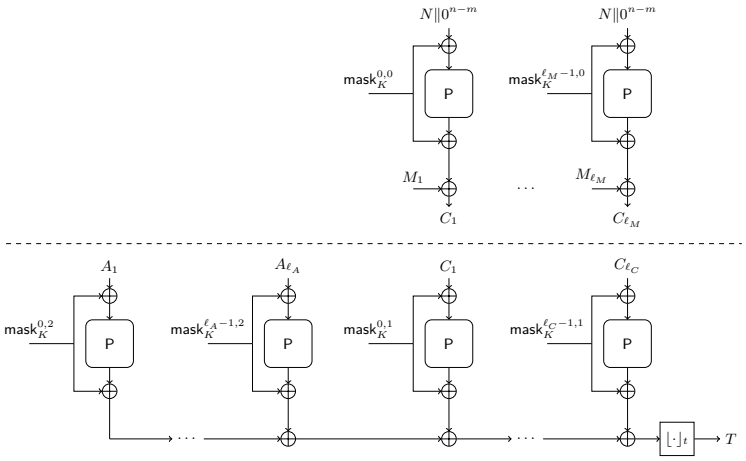

Fig. 2: Depiction of Elephant v1. For the encryption part (top): message is padded as $M_1 \dots M_{\ell_M} \xleftarrow{n} M$, and ciphertext equals $C = \lfloor C_1 \dots C_{\ell_M} \rfloor_{|M|}$. For the authentication part (bottom): nonce and associated data are padded as $A_1 \dots A_{\ell_A} \xleftarrow{n} N\|A\|1$, and ciphertext is padded as $C_1 \dots C_{\ell_C} \xleftarrow{n} C\|1$.

Concisely written, the resulting MAC scheme is of the form $\mathsf{mac}_{K,L}(N, A, C) = \mathsf{E}_K\big(\mathsf{H}_L(N, A, C)\big)$. The main advantage of the change is that the Wegman-Carter-Shoup construction is nonce-based, meaning that it *requires* the nonce to be unique for each evaluation, but the protected counter sum construction does not rely on a nonce for its security. Concretely, for Elephant v2, this implies that the authenticated encryption scheme guarantees authenticity under nonce-misuse (but no confidentiality).

A second change from Elephant v1 to v2, which is minor, is in the positioning of the masks of the tweakable block cipher calls. In Elephant v2, the roles of the masks are $(\cdot, 0)$ for associated data authentication (used to be encryption), $(\cdot, 1)$ for encryption (used to be ciphertext authentication), and $(\cdot, 2)$ for ciphertext authentication (used to be associated data authentication). This improvement gives a slight efficiency improvement, noting that the finalization, i.e., the "protection" part as mentioned above, can be performed with mask $\mathsf{mask}_K^{0,0}$. This change does not affect the security analysis.

4.3.2 Discussion of Theorem 2. The Elephant v2 submission to the NIST Lightweight Cryptography competition [26] consists of three instances: Dumbo, Jumbo, and Delirium [11]. All three variants process a key of size $k = 128$ bits and a nonce of size $m = 96$ bits. Dumbo uses an $n = 160$-bit permutation and outputs tags of size $t = 64$ bits, Jumbo uses an $n = 176$-bit permutation and outputs tags of size $t = 64$ bits, and Delirium uses an $n = 200$-bit permutation and outputs tags of size $t = 64$ bits. For each of the three variants, the authors have developed an LFSR that defines an 2^{-n}-proper mask function [11].

If we model the permutations underlying Dumbo, Jumbo, and Delirium as random permutations, we can conclude from Theorem 2 and Theorem 1 that the three functions are secure authenticated encryption schemes up to bound

$$\ell\binom{q_e}{2}/2^n + 3\binom{q_e+q_d}{2}/2^n + q_d/2^t \\ + \frac{4\sigma^2 + 4\sigma p + (\mu-1)2\sigma + \mu\cdot(4\sigma + p + \frac{\mu-1}{2})}{2^n} + \frac{\mu\cdot(p+\frac{\mu-1}{2})}{2^k},$$

against any nonce-respecting adversary that makes at most q_e encryption queries and q_d decryption queries, all of maximum length ℓ blocks and in total of length at most σ blocks, and that makes at most p evaluations of the random primitive P. Note that the dominating term in the bound is $4\sigma p/2^n$. By capping $\sigma \leq 2^{n-114}$, this term is less than 1 as long as $p \leq 2^{112}$. Likewise, by capping $\sigma \leq 2^{n-130}$, this term is less than 1 as long as $p \leq 2^{128}$. To put these numbers in perspective, Beyne et al. aimed to achieve around 112-bit security for Dumbo and close to 128-bit security for Jumbo and Delirium. However, one also needs to take the other terms of the bound into account. Most of the terms are negligible compared to $4\sigma p/2^n$, and are covered by taking a slightly stricter condition on σ. There is one exception to these negligible terms, namely the factor $p/2^k$ for Jumbo and Delirium: it equals 1 for $p = 2^{128}$. This term thus accounts for a factor 2 loss in the security strength of Jumbo and Delirium. In conclusion, we obtain that Dumbo is secure as long as $p \ll 2^{112}$ and $\sigma \ll 2^{50}/(n/8) < 2^{46}$, that Jumbo is secure as long as $p \ll 2^{127}$ and $\sigma \ll 2^{50}/(n/8) < 2^{46}$, and that Delirium is secure as long as $p \ll 2^{127}$ and $\sigma \ll 2^{74}/(n/8) < 2^{70}$. This is exactly in line with the security claims of the developers of Elephant v2 [11], and also confirms that the change from v1 to v2 has not induced an extra security loss.

In conclusion, we remark that in the above bound the term μ only appears in the minor terms and thus only plays a small role. To be precise, the complexity p is solely determined by the offline power of the adversary whereas μ (just like q_e, q_d, ℓ, σ) is determined by the actual system that the adversary aims to attack. This means that, in practice, $\mu \ll p$. Likewise, $\mu \ll \sigma$ as σ counts the *total* complexity in blocks to all oracles. From this, we can conclude that all terms involving μ are negligible compared to $4\sigma p/2^n$.

5 Proof of Theorem 2 (on Elephant)

Let $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$, $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, and $\mathsf{rand}_1, \ldots, \mathsf{rand}_\mu$ be functions that for each input $(N, A, M) \in \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^*$ return a random string of size $|M| + t$ bits. Consider a deterministic computationally unbounded multi-user adversary $\mathcal{A}$ that tries to distinguish $\mathcal{O} := ((\mathsf{enc}^{\mathsf{P}}_{K_j}, \mathsf{dec}^{\mathsf{P}}_{K_j})^{\mu}_{j=1}, \mathsf{P}^{\pm})$ from $\mathcal{P} := ((\mathsf{rand}_j, \bot)^{\mu}_{j=1}, \mathsf{P}^{\pm})$:

$$\mathbf{Adv}^{\mu\text{-ae}}_{\mathsf{Elephant\text{-}v2}}(\mathcal{A}) = \Delta_{\mathcal{A}}\left((\mathsf{enc}^{\mathsf{P}}_{K_j}, \mathsf{dec}^{\mathsf{P}}_{K_j})^{\mu}_{j=1}, \mathsf{P}^{\pm}\ ;\ (\mathsf{rand}_j, \bot)^{\mu}_{j=1}, \mathsf{P}^{\pm}\right). \quad (9)$$

5.1 First Step: Isolating SiM

As a first step, we will describe an alternative authenticated encryption scheme AE' based on tweakable permutations $\widetilde{\pi}_1, \ldots, \widetilde{\pi}_\mu \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n)$, where $\mathcal{T}$ is $2^{-\alpha}$-proper with respect to LFSRs (φ_1, φ_2). Its encryption function $\overline{\mathsf{enc}}$ and decryption function $\overline{\mathsf{dec}}$ are given in Algorithms 3 and 4, respectively, for any function $\widetilde{\pi} \in \{\widetilde{\pi}_1, \ldots, \widetilde{\pi}_\mu\}$. Unlike the original functions enc and dec of Algorithms 1 and 2, the functions $\overline{\mathsf{enc}}$ and $\overline{\mathsf{dec}}$ are not explicitly keyed by $K_1, \ldots, K_\mu$, but are instead implicitly keyed by the use of random secret tweakable permutations $\widetilde{\pi}_1, \ldots, \widetilde{\pi}_\mu$.

Algorithm 3 encryption $\overline{\mathsf{enc}}$

Input: (N, A, M)
Output: (C, T)
$M_1 \ldots M_{\ell_M} \xleftarrow{n} M$
for $i = 1, \ldots, \ell_M$ **do**
  $C_i \leftarrow M_i \oplus \widetilde{\pi}((i-1, 1), N \| 0^{n-m})$
$C \leftarrow \lfloor C_1 \ldots C_{\ell_M} \rfloor_{|M|}$
$A_1 \ldots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
$C_1 \ldots C_{\ell_C} \xleftarrow{n} C \| 1$
$T \leftarrow A_1$
for $i = 2, \ldots, \ell_A$ **do**
  $T \leftarrow T \oplus \widetilde{\pi}((i-1, 0), A_i)$
for $i = 1, \ldots, \ell_C$ **do**
  $T \leftarrow T \oplus \widetilde{\pi}((i-1, 2), C_i)$
$T \leftarrow \widetilde{\pi}((0, 0), T)$
return $(C, \lfloor T \rfloor_t)$

Algorithm 4 decryption $\overline{\mathsf{dec}}$

Input: (N, A, C, T)
Output: M or $\perp$
$C_1 \ldots C_{\ell_M} \xleftarrow{n} C$
for $i = 1, \ldots, \ell_M$ **do**
  $M_i \leftarrow C_i \oplus \widetilde{\pi}((i-1, 1), N \| 0^{n-m})$
$M \leftarrow \lfloor M_1 \ldots M_{\ell_M} \rfloor_{|C|}$
$A_1 \ldots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
$C_1 \ldots C_{\ell_C} \xleftarrow{n} C \| 1$
$\bar{T} \leftarrow A_1$
for $i = 2, \ldots, \ell_A$ **do**
  $\bar{T} \leftarrow \bar{T} \oplus \widetilde{\pi}((i-1, 0), A_i)$
for $i = 1, \ldots, \ell_C$ **do**
  $\bar{T} \leftarrow \bar{T} \oplus \widetilde{\pi}((i-1, 2), C_i)$
$\bar{T} \leftarrow \widetilde{\pi}((0, 0), \bar{T})$
return $\lfloor \bar{T} \rfloor_t = T \ ? \ M \ : \ \perp$

By a simple hybrid argument, we obtain for the distance of (9):

$$\begin{aligned}(9) \le{}& \Delta_{\mathcal{A}}\left(\left(\mathsf{enc}_{K_j}^{\mathsf{P}}, \mathsf{dec}_{K_j}^{\mathsf{P}}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm} \ ; \ \left(\overline{\mathsf{enc}}^{\mathsf{SiM}_{K_j}^{\mathsf{P}}}, \overline{\mathsf{dec}}^{\mathsf{SiM}_{K_j}^{\mathsf{P}}}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\right) \\ &+ \Delta_{\mathcal{A}}\left(\left(\overline{\mathsf{enc}}^{\mathsf{SiM}_{K_j}^{\mathsf{P}}}, \overline{\mathsf{dec}}^{\mathsf{SiM}_{K_j}^{\mathsf{P}}}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm} \ ; \ \left(\overline{\mathsf{enc}}^{\widetilde{\pi}_j}, \overline{\mathsf{dec}}^{\widetilde{\pi}_j}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\right) \\ &+ \Delta_{\mathcal{A}}\left(\left(\overline{\mathsf{enc}}^{\widetilde{\pi}_j}, \overline{\mathsf{dec}}^{\widetilde{\pi}_j}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm} \ ; \ \left(\mathsf{rand}_j, \perp\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\right) . \end{aligned} \tag{10}$$

The first distance of (10) equals 0 by design of AE'. The second distance of (10) is at most $\Delta_{\mathcal{A}'}\left(\left(\mathsf{SiM}_{K_j}^{\mathsf{P}}\right)_{j=1}^{\mu}, \mathsf{P}^{\pm} \ ; \ \left(\widetilde{\pi}_j\right)_{j=1}^{\mu}, \mathsf{P}^{\pm}\right) = \mathbf{Adv}_{\mathsf{SiM}}^{\mu\text{-tprp}}(\mathcal{A}')$, where $\mathcal{A}'$ is a multi-user adversary that makes 2σ construction queries (in total to all μ construction oracles) and p primitive queries in order to simulate $\mathcal{A}$'s oracles. For the third distance of (10), access to P does not help the adversary, and the oracle can be dropped. We obtain from (10):

$$(9) \le \mathbf{Adv}_{\mathsf{SiM}}^{\mu\text{-tprp}}(\mathcal{A}') + \Delta_{\mathcal{A}}\left(\left(\overline{\mathsf{enc}}^{\widetilde{\pi}_j}, \overline{\mathsf{dec}}^{\widetilde{\pi}_j}\right)_{j=1}^{\mu} \ ; \ \left(\mathsf{rand}_j, \perp\right)_{j=1}^{\mu}\right) . \tag{11}$$

Due to the independence of the oracles for $j = 1, \ldots, \mu$, the remaining distance in (11) can be upper bounded by the sum of the distances for the j oracles, maximized over any choice of adversaries $\mathcal{A}_1, \ldots, \mathcal{A}_\mu$ whose accumulated query complexity is at most that of $\mathcal{A}$:

$$(9) \leq \mathbf{Adv}^{\mu\text{-tprp}}_{\mathsf{SiM}}(\mathcal{A}') + \max_{\mathcal{A}_1,\ldots,\mathcal{A}_\mu} \sum_{j=1}^{\mu} \Delta_{\mathcal{A}_j}\left(\overline{\mathsf{enc}}^{\widetilde{\pi}_j}, \overline{\mathsf{dec}}^{\widetilde{\pi}_j} \; ; \; \mathsf{rand}_j, \perp\right), \tag{12}$$

where the query complexity of adversary $\mathcal{A}_j$ (for $j = 1, \ldots, \mu$) is parametrized by $q_{e,j}$, $q_{d,j}$, and σ_j, and where $\sum_{j=1}^{\mu} q_{e,j} = q_e$, $\sum_{j=1}^{\mu} q_{d,j} = q_d$, and $\sum_{j=1}^{\mu} \sigma_j = \sigma$. Here, we recall that complexity parameter ℓ denotes the maximum length of a single query, and it stays the same for each adversary.

The authenticated encryption scheme AE' can be seen as a generic encrypt-then-MAC construction, and more specifically as the N2 construction of Namprempre et al. [24], where encryption is done in counter mode and message authentication using a variant of the protected counter sum [6, 23] MAC function. We describe a dedicated security proof that is more compact and gives a better bound.

5.2 Second Step: Simplifying Authentication

The simplification of (12) allows us to focus on a single-user instance, and in this step, we drop the subscript j of the oracles for readability. Nevertheless, we keep the subscript j for $\mathcal{A}$ and for its complexities for clarity. In other words, this second step is about bounding

$$\Delta_{\mathcal{A}_j}\left(\overline{\mathsf{enc}}^{\widetilde{\pi}}, \overline{\mathsf{dec}}^{\widetilde{\pi}} \; ; \; \mathsf{rand}, \perp\right) \tag{13}$$

for $\widetilde{\pi} \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n)$ and rand a function that for each input $(N, A, M) \in \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^*$ returns a random string of length $|M| + t$ bits. The query complexity of the adversary is measured by parameters $q_{e,j}$, $q_{d,j}$, ℓ, and σ_j.

Let ρ be a random function that for each input $(N, A, C) \in \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^*$ returns a random string of length n bits. We describe an alternative authenticated encryption scheme AE'' based on tweakable permutation $\widetilde{\pi}$ and on ρ. Its encryption function $\overline{\overline{\mathsf{enc}}}$ and decryption function $\overline{\overline{\mathsf{dec}}}$ are given in Algorithms 5 and 6, respectively.

Algorithm 5 encryption $\overline{\overline{\mathsf{enc}}}$

Input: (N, A, M)
Output: (C, T)
1: $M_1 \dots M_{\ell_M} \xleftarrow{n} M$
2: **for** $i = 1, \dots, \ell_M$ **do**
3: $\quad C_i \leftarrow M_i \oplus \widetilde{\pi}((i-1, 1), N \| 0^{n-m})$
4: $C \leftarrow \lfloor C_1 \dots C_{\ell_M} \rfloor_{|M|}$
5: $A_1 \dots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
6: $C_1 \dots C_{\ell_C} \xleftarrow{n} C \| 1$
7: $T \leftarrow \rho(N, A, C)$
8: **return** $(C, \lfloor T \rfloor_t)$

Algorithm 6 decryption $\overline{\overline{\mathsf{dec}}}$

Input: (N, A, C, T)
Output: M or $\perp$
1: $C_1 \dots C_{\ell_M} \xleftarrow{n} C$
2: **for** $i = 1, \dots, \ell_M$ **do**
3: $\quad M_i \leftarrow C_i \oplus \widetilde{\pi}((i-1, 1), N \| 0^{n-m})$
4: $M \leftarrow \lfloor M_1 \dots M_{\ell_M} \rfloor_{|C|}$
5: $A_1 \dots A_{\ell_A} \xleftarrow{n} N \| A \| 1$
6: $C_1 \dots C_{\ell_C} \xleftarrow{n} C \| 1$
7: $\bar{T} \leftarrow \rho(N, A, C)$
8: **return** $\lfloor \bar{T} \rfloor_t = T \ ? \ M \ : \ \perp$

Proceeding from (13):

$$\begin{aligned}(13) \le{}& \Delta_{\mathcal{A}_j}\left(\overline{\mathsf{enc}}^{\widetilde{\pi}}, \overline{\mathsf{dec}}^{\widetilde{\pi}} \ ; \ \overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho}\right) \\ &+ \Delta_{\mathcal{A}_j}\left(\overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho} \ ; \ \overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \perp\right) \\ &+ \Delta_{\mathcal{A}_j}\left(\overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \perp \ ; \ \mathsf{rand}, \perp\right) . \end{aligned} \tag{14}$$

We will analyze the three distances of (14) separately.

First Distance of (14). Define the function $h : \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ as

$$h^{\widetilde{\pi}}(N, A, C) = A_1 \oplus \left(\bigoplus_{i=2}^{\ell_A} \widetilde{\pi}((i-1, 0), A_i)\right) \oplus \left(\bigoplus_{i=1}^{\ell_C} \widetilde{\pi}((i-1, 2), C_i)\right) ,$$

where $A_1 \dots A_{\ell_A} \xleftarrow{n} N \| A \| 1$ and $C_1 \dots C_{\ell_C} \xleftarrow{n} C \| 1$. This function is $(2^n - 1)^{-1}$-uniform: for any distinct $(N, A, C) \neq (N', A', C')$, the probability over the drawing of $\widetilde{\pi} \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n)$ that $h^{\widetilde{\pi}}(N, A, C) = h^{\widetilde{\pi}}(N', A', C')$ is at most $(2^n - 1)^{-1}$.

Next, define $f : \{0,1\}^m \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^n$ as

$$f^{\widetilde{\pi}}(N, A, C) = \widetilde{\pi}((0, 0), h^{\widetilde{\pi}}(N, A, C)) .$$

In $\overline{\mathsf{enc}}^{\widetilde{\pi}}$ and $\overline{\mathsf{dec}}^{\widetilde{\pi}}$ the tag (before truncation) is computed as $f^{\widetilde{\pi}}(N, A, C)$, whereas in $\overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}$ and $\overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho}$ it is computed as $\rho(N, A, C)$. Therefore, the first distance of (14) is at most $\Delta_{\mathcal{A}'_j}\left(f^{\widetilde{\pi}} \ ; \ \rho\right)$, where $\mathcal{A}'_j$ is an adversary that makes $q_{e,j} + q_{d,j}$ construction queries, each query at most ℓ padded nonce and associated data and ciphertext blocks. Here, we make use of the fact that $f^{\widetilde{\pi}}$ only evaluates its tweakable permutation for tweaks $(\cdot, 0)$ and $(\cdot, 2)$, these tweaks do not occur elsewhere in the encryption and decryption function, and thus $\mathcal{A}'_j$ can properly simulate $\mathcal{A}_j$'s oracles.

Looking inside $f^{\widetilde{\pi}}$, it consists of an independent composition of $h^{\widetilde{\pi}}$, that never evaluates its primitive for tweak $(0,0)$, and $\widetilde{\pi}((0,0),\cdot)$. We replace $\widetilde{\pi}((0,0),\cdot)$ with a random function $\tau \xleftarrow{\$} \mathrm{func}(n)$, which by the PRP-PRF switching lemma comes at a cost of $\binom{q_{e,j}+q_{d,j}}{2}/2^n$:

$$\Delta_{\mathcal{A}'_j}\left(f^{\widetilde{\pi}}\ ;\ \rho\right) \leq \Delta_{\mathcal{A}'_j}\left(\tau \circ h^{\widetilde{\pi}}\ ;\ \rho\right) + \binom{q_{e,j}+q_{d,j}}{2}/2^n\,.$$

The function $\tau \circ h^{\widetilde{\pi}}$ is perfectly indistinguishable from ρ as long as no two inputs to $h^{\widetilde{\pi}}$ collide. As $h^{\widetilde{\pi}}$ is $(2^n-1)^{-1}$-uniform, we in turn have

$$\Delta_{\mathcal{A}'_j}\left(\tau \circ h^{\widetilde{\pi}}\ ;\ \rho\right) \leq \binom{q_{e,j}+q_{d,j}}{2}(2^n-1)^{-1}\,.$$

Concluding, we obtain for the first distance of (14):

$$\Delta_{\mathcal{A}_j}\left(\overline{\mathsf{enc}}^{\widetilde{\pi}}, \overline{\mathsf{dec}}^{\widetilde{\pi}}\ ;\ \overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho}\right) \leq 3\binom{q_{e,j}+q_{d,j}}{2}/2^n\,. \tag{15}$$

Second Distance of (14). The adversary $\mathcal{A}_j$ can only distinguish both worlds if it ever makes a non-trivial evaluation to $\overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho}$ that succeeds. Consider any forgery attempt (N,A,C,T). If the tuple (N,A,C) occurred in an earlier encryption query, then necessarily the tag T must differ and the forgery will not succeed. Otherwise, ρ has never been evaluated on (N,A,C) before, and $\lfloor \rho(N,A,C) \rfloor_t = T$ with probability $1/2^t$. By summing over all $q_{d,j}$ forgery attempts, we obtain for the second distance of (14):

$$\Delta_{\mathcal{A}_j}\left(\overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \overline{\overline{\mathsf{dec}}}^{\widetilde{\pi},\rho}\ ;\ \overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \bot\right) \leq q_{d,j}/2^t\,. \tag{16}$$

Third Distance of (14). We remark that every query is made for a unique nonce, and in more detail:

- The i^{th} block of ciphertext equals $\widetilde{\pi}((i-1,1),N) \oplus M_i$, where M_i is the i^{th} block of plaintext;
- The tag equals $\rho(N,A,C)$.

The tweakable permutation $\widetilde{\pi}$ is independent for different tweaks, but two different inputs for the same tweak never collide. Therefore, this third distance of (14) is upper bounded by the probability that any two out of $q_{e,j}$ evaluations of rand collide in any of the ℓ blocks, or more detailed:

$$\Delta_{\mathcal{A}_j}\left(\overline{\overline{\mathsf{enc}}}^{\widetilde{\pi},\rho}, \bot\ ;\ \mathsf{rand}, \bot\right) \leq \ell\binom{q_{e,j}}{2}/2^n\,. \tag{17}$$

Conclusion. Proceeding from (14) and the individual bounds of (15), (16), and (17), we obtain:

$$(13) \leq 3\binom{q_{e,j}+q_{d,j}}{2}/2^n + q_{d,j}/2^t + \ell\binom{q_{e,j}}{2}/2^n\,. \tag{18}$$

5.3 Third Step: Conclusion

Recall that in the second step, we dropped the subscripts and focused on a single-user case, as inspired by (12). We have to sum the bound of (18) over $j = 1, \ldots, \mu$, and maximize over the choice of $q_{e,j}$ and $q_{d,j}$ such that $\sum_{j=1}^{\mu} q_{e,j} = q_e$ and $\sum_{j=1}^{\mu} q_{d,j} = q_d$. As (18) is convex in $q_{e,j}$ and $q_{d,j}$, we obtain from (12):

$$(9) \leq \mathbf{Adv}_{\mathsf{SiM}}^{\mu\text{-tprp}}(\mathcal{A}') + 3\binom{q_e + q_d}{2}/2^n + q_d/2^t + \ell\binom{q_e}{2}/2^n \,,$$

and this completes the proof of Theorem 2.

6 Conclusion

We proved multi-user security of Elephant v2, one of the finalists in the NIST Lightweight Cryptography competition, under the assumption that the keys of all instances are mutually independent and under the assumption that the underlying permutation is random. From our observations in Section 4.3, we can conclude that the change from Elephant v1 to Elephant v2 has improved its security, with the most important change that Elephant v2 even achieves authenticity under nonce-misuse. In addition, from our analysis, we show that no unexpected things happen when moving to multiple users.

Acknowledgments. This work was supported in part by the Research Council KU Leuven: GOA TENSE (C16/15/058). Tim Beyne and Yu Long Chen are supported by a Ph.D. Fellowship from the Research Foundation - Flanders (FWO). Christoph Dobraunig is supported by the Austrian Science Fund (FWF): J 4277-N38. Bart Mennink is supported by the Netherlands Organisation for Scientific Research (NWO) under grant VI.Vidi.203.099.

References

1. Banik, S., Chakraborti, A., Iwata, T., Minematsu, K., Nandi, M., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT-COFB v1.1. Submission to NIST Lightweight Cryptography (2021)
2. Beierle, C., Biryukov, A., dos Santos, L.C., Großschädl, J., Moradi, A., Perrin, L., Shahmirzadi, A.R., Udovenko, A., Velichkov, V., Wang, Q.: Schwaemm and Esch: Lightweight Authenticated Encryption and Hashing using the Sparkle Permutation Family. Submission to NIST Lightweight Cryptography (2021)
3. Beierle, C., Biryukov, A., dos Santos, L.C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q.: Lightweight AEAD and Hashing using the Sparkle Permutation Family. IACR Trans. Symmetric Cryptol. 2020(S1), 208–261 (2020)
4. Bellare, M., Boldyreva, A., Micali, S.: Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 259–274. Springer (2000)

5. Bellare, M., Tackmann, B.: The Multi-user Security of Authenticated Encryption: AES-GCM in TLS 1.3. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 247–276. Springer (2016)
6. Bernstein, D.J.: How to Stretch Random Functions: The Security of Protected Counter Sums. J. Cryptology 12(3), 185–192 (1999)
7. Bernstein, D.J.: Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 164–180. Springer (2005)
8. Beyne, T., Chen, Y.L., Dobraunig, C., Mennink, B.: Elephant v1.1. Submission to NIST Lightweight Cryptography (2019)
9. Beyne, T., Chen, Y.L., Dobraunig, C., Mennink, B.: Dumbo, Jumbo, and Delirium: Parallel Authenticated Encryption for the Lightweight Circus. IACR Trans. Symmetric Cryptol. 2020(S1), 5–30 (2020)
10. Beyne, T., Chen, Y.L., Dobraunig, C., Mennink, B.: Status update on Elephant. Note at NIST Lightweight Cryptography (2020)
11. Beyne, T., Chen, Y.L., Dobraunig, C., Mennink, B.: Elephant v2. Submission to NIST Lightweight Cryptography (2021)
12. Biham, E.: How to decrypt or even substitute DES-encrypted messages in 2^{28} steps. Inf. Process. Lett. 84(3), 117–124 (2002)
13. Chen, S., Steinberger, J.P.: Tight Security Bounds for Key-Alternating Ciphers. In: Nguyen and Oswald [25], pp. 327–350
14. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., Van Keer, R.: Xoodyak, a lightweight cryptographic scheme. IACR Trans. Symmetric Cryptol. 2020(S1), 60–87 (2020)
15. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., Van Keer, R., Mella, S.: Xoodyak, a lightweight cryptographic scheme. Submission to NIST Lightweight Cryptography (2021)
16. Daemen, J., Mennink, B., Van Assche, G.: Full-State Keyed Duplex with Built-In Multi-user Support. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 606–637. Springer (2017)
17. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., Unterluggauer, T.: Isap v2.0. IACR Trans. Symmetric Cryptol. 2020(S1), 390–416 (2020)
18. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., Unterluggauer, T.: ISAP v2. Submission to NIST Lightweight Cryptography (2021)
19. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2. Submission to NIST Lightweight Cryptography (2021)
20. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 263–293. Springer (2016)
21. Khairallah, M.: Weak Keys in the Rekeying Paradigm: Application to COMET and mixFeed. IACR Trans. Symmetric Cryptol. 2019(4), 272–289 (2019)
22. Luykx, A., Mennink, B., Paterson, K.G.: Analyzing Multi-key Security Degradation. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 575–605. Springer (2017)
23. Luykx, A., Preneel, B., Tischhauser, E., Yasuda, K.: A MAC Mode for Lightweight Block Ciphers. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 43–59. Springer (2016)

24. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering Generic Composition. In: Nguyen and Oswald [25], pp. 257–274
25. Nguyen, P.Q., Oswald, E. (eds.): EUROCRYPT 2014, LNCS, vol. 8441. Springer (2014)
26. NIST: Lightweight Cryptography (February 2019), https://csrc.nist.gov/Projects/Lightweight-Cryptography
27. Patarin, J.: The "Coefficients H" Technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 328–345. Springer (2008)
28. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO '96. LNCS, vol. 1109, pp. 313–328. Springer (1996)
29. Wegman, M.N., Carter, L.: New Hash Functions and Their Use in Authentication and Set Equality. J. Comput. Syst. Sci. 22(3), 265–279 (1981)

A Proof of Theorem 1 (on SiM)

The proof closely follows Granger et al. [20], just like that of Beyne et al. [9] did, and is performed using the H-coefficient technique [13, 27]. The main difference is in the fact that we consider multi-user security, where the adversary can query $\mu \geq 1$ construction oracles.

Let $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$, $\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n)$, and $\widetilde{\pi}_1, \ldots, \widetilde{\pi}_\mu \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n)$, where $\mathcal{T}$ is $2^{-\alpha}$-proper with respect to LFSRs $(\varphi_1, \ldots, \varphi_z)$. Consider a computationally unbounded adversary $\mathcal{A}$ that tries to distinguish $\mathcal{O} := ((\widetilde{\mathsf{E}}^{\mathsf{P}}_{K_j})_{j=1}^{\mu}, \mathsf{P}^{\pm})$ from $\mathcal{P} := ((\widetilde{\pi}_j)_{j=1}^{\mu}, \mathsf{P}^{\pm})$. Without loss of generality, we can consider it to be deterministic: for any probabilistic adversary there exists a deterministic one that has at least the same success probability. The interaction of $\mathcal{A}$ with its oracle ($\mathcal{O}$ or $\mathcal{P}$) is gathered in a view ν. Denote by $D_{\mathcal{O}}$ (resp., $D_{\mathcal{P}}$) the probability distribution of views in interaction with $\mathcal{O}$ (resp., $\mathcal{P}$). Denote by $\mathcal{V}$ the set of "attainable views", i.e., views ν such that $\mathbf{Pr}\left(D_{\mathcal{P}} = \nu\right) > 0$.

Lemma 1 (H-coefficient technique). *Consider a partition $\mathcal{V} = \mathcal{V}_{\mathrm{good}} \cup \mathcal{V}_{\mathrm{bad}}$ of the set of views into "good" and "bad" views. Let $\varepsilon \in [0,1]$ be such that $\frac{\mathbf{Pr}(D_{\mathcal{O}}=\nu)}{\mathbf{Pr}(D_{\mathcal{P}}=\nu)} \geq 1 - \varepsilon$ for all $\nu \in \mathcal{V}_{\mathrm{good}}$. Then,*

$$\Delta_{\mathcal{A}}\left(\mathcal{O}\,;\,\mathcal{P}\right) \leq \varepsilon + \mathbf{Pr}\left(D_{\mathcal{P}} \in \mathcal{V}_{\mathrm{bad}}\right). \tag{19}$$

For view $\nu = \{(x_1, y_1), \ldots, (x_q, y_q)\}$ consisting of q input/output tuples, we denote by $\mathcal{O} \vdash \nu$ the event that oracle $\mathcal{O}$ satisfies that $\mathcal{O}(x_i) = y_i$ for all $i = \{1, \ldots, q\}$.

The remainder of the proof is structured as follows. We specify the views of an adversary in Section A.1 and define the bad views in Section A.2. The probability of bad views is analyzed in Section A.3 and the probability ratio for good views is considered in Section A.4. Section A.5 concludes the proof.

A.1 Views

The adversary can make q construction queries to $(\widetilde{\mathsf{E}}^{\mathsf{P}}_{K_j})_{j=1}^{\mu}$ or $(\widetilde{\pi})_{j=1}^{\mu}$, all *in forward direction only*. Each such query is made for user index $j_i \in \{1, \ldots, \mu\}$,

some tweak $\bar{a}_i = (a_1, \ldots, a_z)_i$, and message input M_i, and results in an output C_i. The q queries are summarized in a view

$$\nu_c = \{(j_1, \bar{a}_1, M_1, C_1), \ldots, (j_q, \bar{a}_q, M_q, C_q)\}\,.$$

The adversary can make p primitive queries to $\mathsf{P}^{\pm}$, and these are likewise summarized in a view

$$\nu_p = \{(X_1, Y_1), \ldots, (X_p, Y_p)\}\,.$$

After the conversation of $\mathcal{A}$ with its oracle, but before it makes its final decision, we reveal the key material used in the interaction. This can be done without loss of generality; it only improves the adversarial success probability. The first values that are revealed are values $K_1, \ldots, K_\mu$. In the real world, these are the keys $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ that are actually used by the construction oracle; in the ideal world, these are dummy keys $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. The second values that are revealed are values $L_1, \ldots, L_\mu \in \{0,1\}^n$. In the real world, these are the values $L_j = \mathsf{P}(K_j \| 0^{n-k})$ for $j = 1, \ldots, \mu$; in the ideal world, these are dummy keys $L_1, \ldots, L_\mu \xleftarrow{\$} \{0,1\}^n$.[6] The revealed data is summarized in a view

$$\nu_k = \{(K_1, L_1), \ldots, (K_\mu, L_\mu)\}\,.$$

(Note that in the single-user setting, where $\mu = 1$, ν_k is a singleton.) The complete view is defined as $\nu = (\nu_c, \nu_p, \nu_k)$. We assume that the adversary never makes any duplicate query, hence ν_c and ν_p contain no duplicate elements.

A.2 Definition of Good and Bad Views

In the real world, all tuples in ν_p define exactly one input-output pair for P. Likewise, the tuples in ν_k are input-output pairs for P. Using these tuples, one can observe that any tuple $(j_i, \bar{a}_i, M_i, C_i) \in \nu_c$ also defines an input-output pair for P, namely

$$\left(M_i \oplus \bar{\varphi}^{\bar{a}_i}(L_{j_i}), C_i \oplus \bar{\varphi}^{\bar{a}_i}(L_{j_i})\right)\,,$$

see (7), where we define $\bar{\varphi}^{\bar{a}_i} := \varphi_z^{a_{z\,i}} \circ \cdots \circ \varphi_1^{a_{1\,i}}$ for brevity. If among all these $q+p+\mu$ input-output pairs defined by ν, there are two that have colliding input or output values, we consider ν to be a bad view. Formally, ν is called "bad" if one of the following conditions is satisfied, where we recall that the user index j in a tuple in ν_c determines which key tuple from ν_k has to be used:

$$\mathrm{bad}_{c,c}:\ \text{for some distinct } (j, \bar{a}, M, C), (j', \bar{a}', M', C') \in \nu_c\text{:}$$

[6] In the original analysis of MEM [20] (that was about single-user security only), the mask involves a computation $\mathsf{P}(K\|N)$ for nonce N. This not only complicates the values that have to be revealed; it also results in a larger view and hence a higher collision probability among tuples in the view.

$$\bar{\varphi}^{\bar{a}}(L_j) \oplus \bar{\varphi}^{\bar{a}'}(L_{j'}) \in \{M \oplus M', C \oplus C'\}\,,$$

$\mathrm{bad}_{c,p}$: for some $(j, \bar{a}, M, C) \in \nu_c$ and $(X, Y) \in \nu_p$:
$\bar{\varphi}^{\bar{a}}(L_j) \in \{M \oplus X, C \oplus Y\}\,,$

$\mathrm{bad}_{c,k}$: for some $(j, \bar{a}, M, C) \in \nu_c$ and $(K, L) \in \nu_k$:
$\bar{\varphi}^{\bar{a}}(L_j) \in \{M \oplus K \| 0^{n-k}, C \oplus L\}\,,$

$\mathrm{bad}_{p,k}$: for some $(X, Y) \in \nu_p$ and $(K, L) \in \nu_k$:
$X = K \| 0^{n-k}$ or $Y = L\,,$

$\mathrm{bad}_{k,k}$: for some distinct $(K, L), (K', L') \in \nu_k$:
$K = K'$ or $L = L'\,.$

We write $\mathrm{bad} = \mathrm{bad}_{c,c} \vee \mathrm{bad}_{c,p} \vee \mathrm{bad}_{c,k} \vee \mathrm{bad}_{p,k} \vee \mathrm{bad}_{k,k}$.

The definition of bad events differs from the single-user analysis of Beyne et al. [9] in the adjustment of bad events $\mathrm{bad}_{c,k}$ and $\mathrm{bad}_{p,k}$ and the addition of the bad event $\mathrm{bad}_{k,k}$. The events $\mathrm{bad}_{c,k}$ and $\mathrm{bad}_{p,k}$ have been adjusted as construction or permutation queries may now collide with μ different key tuples. The addition of the new bad event $\mathrm{bad}_{k,k}$ come from the fact that different key tuples might collide.

A.3 Probability of Bad View in Ideal World

Our goal is to bound $\mathbf{Pr}\left(D_{\mathcal{P}} \in \mathcal{V}_{\mathrm{bad}}\right)$, the probability of a bad view in the ideal world $\mathcal{P} = ((\widetilde{\pi}_j)_{j=1}^{\mu}, \mathsf{P}^{\pm})$. For brevity, denote by $D_{\mathcal{P}} \propto \mathrm{bad}$ the event that $D_{\mathcal{P}}$ satisfies bad. By the union bound,

$$\begin{aligned}\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}\right) &= \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,c} \vee \mathrm{bad}_{c,p} \vee \mathrm{bad}_{c,k} \vee \mathrm{bad}_{p,k} \vee \mathrm{bad}_{k,k}\right) \\ &\leq \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,c}\right) + \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,p}\right) + \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,k}\right) \\ &+ \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{p,k}\right) + \mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{k,k}\right)\,. \qquad (20)\end{aligned}$$

We will analyze the five probabilities separately, thereby noticing that (i) K_1, $\ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$ and $L_1, \ldots, L_\mu \xleftarrow{\$} \{0,1\}^n$ are random variables in the ideal world, and (ii) as the adversary only makes forward construction queries, each tuple $(j, \bar{a}, M, C) \in \nu_c$ satisfies that C is randomly drawn from a set of size at least $2^n - q$.

Event $\mathrm{bad}_{c,c}$. For $\mathrm{bad}_{c,c}$, let $(j, \bar{a}, M, C), (j', \bar{a}', M', C') \in \nu_c$ be any two distinct tuples. If $j = j'$ and $\bar{a} = \bar{a}'$, then necessarily $M \neq M'$ and $C \neq C'$, and $\mathrm{bad}_{c,c}$ holds with probability 0. Otherwise, if $j = j'$ but $\bar{a} \neq \bar{a}'$, we can deduce from $2^{-\alpha}$-properness of $\mathcal{T}$, namely property 2 of Definition 1, that event $\mathrm{bad}_{c,c}$ holds with probability at most $2/2^{\alpha}$. Finally, if $j \neq j'$, the subkeys $L_j, L_{j'}$ are independent and we can likewise deduce from $2^{-\alpha}$-properness of $\mathcal{T}$, namely property 1 of Definition 1, that event $\mathrm{bad}_{c,c}$ holds with probability at most $2/2^{\alpha}$. Thus, summing over all $\binom{q}{2}$ possible choices of queries,

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,c}\right) \leq \frac{q(q-1)}{2^{\alpha}}\,.$$

Event $\mathrm{bad}_{c,p}$. For $\mathrm{bad}_{c,p}$, let $(j, \bar{a}, M, C) \in \nu_c$ and $(X, Y) \in \nu_p$ be any two tuples. We can deduce from $2^{-\alpha}$-properness of $\mathcal{T}$, namely property 1 of Definition 1, that event $\mathrm{bad}_{c,p}$ holds with probability at most $2/2^{\alpha}$. Thus, summing over all qp possible choices of queries,

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,p}\right) \leq \frac{2qp}{2^{\alpha}}\,.$$

Event $\mathrm{bad}_{c,k}$. For $\mathrm{bad}_{c,k}$, let $(j, \bar{a}, M, C) \in \nu_c$ and $(K, L) \in \nu_k$ be any two tuples. We consider the two equations of $\mathrm{bad}_{c,k}$ separately. For the first equation,

$$\bar{\varphi}^{\bar{a}}(L_j) = M \oplus K \| 0^{n-k}\,,$$

we will use that $L_j \xleftarrow{\$} \{0,1\}^n$ is a randomly generated value independent of K. We can deduce from $2^{-\alpha}$-properness of $\mathcal{T}$, namely property 1 of Definition 1, that this equation holds with probability at most $1/2^{\alpha}$.

For the second equation,

$$\bar{\varphi}^{\bar{a}}(L_j) = C \oplus L\,,$$

it might be that $L = L_j$, and we cannot rely on Definition 1. Instead, we will use that all construction queries are made in the forward direction, and that C is randomly drawn from a set of size at least $2^n - q$ elements. The above equation thus holds with probability at most $1/(2^n - q)$.

Thus, summing over all μq possible choices of queries,

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{c,k}\right) \leq \frac{\mu q}{2^{\alpha}} + \frac{\mu q}{2^n - q}\,.$$

Event $\mathrm{bad}_{p,k}$. For $\mathrm{bad}_{p,k}$, let $(X, Y) \in \nu_p$ and $(K, L) \in \nu_k$ be any two tuples. As $K \xleftarrow{\$} \{0,1\}^k$ and $L \xleftarrow{\$} \{0,1\}^n$, the tuples set $\mathrm{bad}_{p,k}$ with probability at most $1/2^k + 1/2^n$. Thus, summing over all μp possible choices of queries,

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{p,k}\right) \leq \frac{\mu p}{2^k} + \frac{\mu p}{2^n}\,.$$

Event $\mathrm{bad}_{k,k}$. For $\mathrm{bad}_{k,k}$, let $(K, L), (K', L') \in \nu_k$ be any two distinct tuples. As $K, K' \xleftarrow{\$} \{0,1\}^k$ and $L, L' \xleftarrow{\$} \{0,1\}^n$, the tuples set $\mathrm{bad}_{k,k}$ with probability at most $1/2^k + 1/2^n$. Thus, summing over all $\binom{\mu}{2}$ possible choices of queries,

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}_{k,k}\right) \leq \frac{\mu(\mu-1)}{2^{k+1}} + \frac{\mu(\mu-1)}{2^{n+1}}\,.$$

Conclusion. Concluding, we obtain for (20):

$$\mathbf{Pr}\left(D_{\mathcal{P}} \propto \mathrm{bad}\right) \leq \frac{q^2 + 2qp + (\mu-1)q}{2^{\alpha}} + \frac{\mu \cdot (2q + p + \frac{\mu-1}{2})}{2^n} + \frac{\mu \cdot (p + \frac{\mu-1}{2})}{2^k}\,, \tag{21}$$

using that $2^n - q \geq 2^{n-1}$.

A.4 Probability Ratio for Good Views

Consider any good view $\nu \in \mathcal{V}_{\text{good}}$. We will prove the inequality $\mathbf{Pr}\left(D_{\mathcal{O}} = \nu\right) \geq \mathbf{Pr}\left(D_{\mathcal{P}} = \nu\right)$. The proof is a direct generalization of that of Granger et al. [20], noting that in our case, we consider multi-user security. The proof is included for completeness.

Real World. In the real world $\mathcal{O} = ((\widetilde{\mathsf{E}}^{\mathsf{P}}_{K_j})_{j=1}^{\mu}, \mathsf{P}^{\pm})$, goodness of the view means that $\nu = (\nu_c, \nu_p, \nu_k)$ defines exactly $q + p + \mu$ input-output pairs for P, and no two of them collide on the input or output, and ν_k consists of random values $K_1, \ldots, K_\mu \xleftarrow{\$} \{0,1\}^k$. Therefore, we obtain:

$$\begin{aligned}\mathbf{Pr}\left(D_{\mathcal{O}} = \nu\right) &= \mathbf{Pr}\left(K'_1, \ldots, K'_\mu \xleftarrow{\$} \{0,1\}^k \;:\; K'_1 = K_1 \wedge \cdots \wedge K'_\mu = K_\mu\right) \cdot \\ &\quad \mathbf{Pr}\left(\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n) \;:\; (\widetilde{\mathsf{E}}^{\mathsf{P}}_{K_j})_{j=1}^{\mu} \vdash \nu_c \wedge \mathsf{P} \vdash \nu_p \wedge \mathsf{P} \vdash \nu_k\right) \\ &= \frac{1}{2^{k\mu}} \cdot \frac{(2^n - (q+p+\mu))!}{2^n!}\,. \end{aligned} \tag{22}$$

Ideal World. In the ideal world $\mathcal{P} = ((\widetilde{\pi}_j)_{j=1}^{\mu}, \mathsf{P}^{\pm})$, the view $\nu = (\nu_c, \nu_p, \nu_k)$ consists of three lists of independent tuples: ν_c defines exactly q input-output pairs for $\widetilde{\pi}_j$, ν_p defines exactly p input-output pairs for P, and ν_k consists of μ random tuples $(K_1, L_1), \ldots, (K_\mu, L_\mu) \xleftarrow{\$} \{0,1\}^k \times \{0,1\}^n$. For counting, it is convenient to group the tuples in ν_c depending on the user index j and tweak value $\bar{a}$. For $J \in \{1, \ldots, \mu\}$ and $T \in \mathcal{T}$, define

$$q_{J,T} = \left|\{(j, \bar{a}, M, C) \in \nu_c \mid j = J \wedge \bar{a} = T\}\right|\,,$$

where $\sum_{(J,T) \in \{1,\ldots,\mu\} \times \mathcal{T}} q_{J,T} = q$. We obtain:

$$\begin{aligned}\mathbf{Pr}\left(D_{\mathcal{P}} = \nu\right) &= \mathbf{Pr}\left(K'_1, \ldots, K'_\mu \xleftarrow{\$} \{0,1\}^k \;:\; K'_1 = K_1 \wedge \cdots \wedge K'_\mu = K_\mu\right) \cdot \\ &\quad \mathbf{Pr}\left(L'_1, \ldots, L'_\mu \xleftarrow{\$} \{0,1\}^n \;:\; L'_1 = L_1 \wedge \cdots \wedge L'_\mu = L_\mu\right) \cdot \\ &\quad \mathbf{Pr}\left(\widetilde{\pi}_1, \ldots, \widetilde{\pi}_\mu \xleftarrow{\$} \mathrm{perm}(\mathcal{T}, n) \;:\; (\widetilde{\pi}_j)_{j=1}^{\mu} \vdash \nu_c\right) \cdot \\ &\quad \mathbf{Pr}\left(\mathsf{P} \xleftarrow{\$} \mathrm{perm}(n) \;:\; \mathsf{P} \vdash \nu_p\right) \\ &= \frac{1}{2^{(k+n)\mu}} \cdot \prod_{\substack{J \in \{1,\ldots,\mu\} \\ T \in \mathcal{T}}} \frac{(2^n - q_{J,T})!}{2^n!} \cdot \frac{(2^n - p)!}{2^n!} \\ &= \frac{1}{2^{k\mu}} \cdot \left(\frac{(2^n - 1)!}{2^n!}\right)^{\mu} \cdot \prod_{\substack{J \in \{1,\ldots,\mu\} \\ T \in \mathcal{T}}} \frac{(2^n - q_{J,T})!}{2^n!} \cdot \frac{(2^n - p)!}{2^n!} \\ &\leq \frac{1}{2^{k\mu}} \cdot \frac{(2^n - (q+p+\mu))!}{2^n!}\,, \end{aligned} \tag{23}$$

using that for any $\sigma + \tau \leq 2^n$ we have $\frac{(2^n-\sigma)!}{2^n!} \cdot \frac{(2^n-\tau)!}{2^n!} \leq \frac{(2^n-(\sigma+\tau))!}{2^n!}$.

Conclusion. Combining (22) and (23), we obtain that for any good view $\nu \in \mathcal{V}_{\text{good}}$:

$$\frac{\mathbf{Pr}\left(D_{\mathcal{O}} = \nu\right)}{\mathbf{Pr}\left(D_{\mathcal{P}} = \nu\right)} \geq 1\,. \tag{24}$$

A.5 Conclusion

By the H-coefficient technique (Lemma 1), we directly obtain from (21) and (24):

$$\mathbf{Adv}^{\mu\text{-tprp}}_{\widetilde{\mathsf{E}}}(\mathcal{A}) \leq 0 + \frac{q^2 + 2qp + (\mu-1)q}{2^{\alpha}} + \frac{\mu \cdot (2q + p + \frac{\mu-1}{2})}{2^{n}} + \frac{\mu \cdot (p + \frac{\mu-1}{2})}{2^{k}}\,.$$