Post-Quantum Security of Keyed Sponge-Based Constructions through a Modular Approach*

Akinori Hosoyamada

NTT Social Informatics Laboratories, Tokyo, Japan
NTT Research Center for Theoretical Quantum Information, Atsugi, Japan
akinori.hosoyamada@ntt.com

Abstract. Sponge-based constructions have successfully been receiving widespread adoption, as represented by the standardization of SHA-3 and Ascon by NIST. Yet, their provable security against quantum adversaries has not been investigated much. This paper studies the post-quantum security of some keyed sponge-based constructions in the quantum ideal permutation model, focusing on the Ascon AEAD mode and KMAC as concrete instances. For the Ascon AEAD mode, we prove the post-quantum security in the single-user setting up to about $\min(2^{c/3}, 2^{\kappa/3})$ queries, where c is the capacity and κ is the key length. Unlike the recent work by Lang et al. (ePrint 2025/411), we do not need non-standard restrictions on nonce sets or the number of forgery attempts. In addition, our result guarantees even non-conventional security notions such as the nonce-misuse resilience confidentiality and authenticity under release of unverified plaintext. For KMAC, we show the security up to about $\min(2^{c/3}, 2^{r/2}, 2^{(\kappa-r)/2})$ queries, where r is the rate, ignoring some small factors. In fact, we prove the security not only for KMAC but also for general modes such as the inner-, outer-, and full-keyed sponge functions.

We take a modular proof approach, adapting the ideas by several works in the classical ideal permutation model into the quantum setting: For the Ascon AEAD mode, we observe it can be regarded as an iterative application of a Tweakable Even-Mansour (TEM) cipher with a single low-entropy key, and gives the security bound as the sum of the post-quantum TPRP advantage of TEM and the classical security advantage of Ascon when TEM is replaced with a secret random object. The proof for keyed sponges is obtained analogously by regarding them as built on an Even-Mansour (EM) cipher with a single low-entropy key.

The post-quantum security of (T)EM has been proven by Alagic et al. (Eurocrypt 2022 and Eurocrypt 2024). However, they show the security only when the keys are uniformly random. In addition, the proof techniques, so-called the resampling lemmas, are inapplicable to our case with a low-entropy key. Thus, we introduce and prove a modified resampling lemma, thereby showing the security of (T)EM with a low-entropy key.

* This article is the full version of the paper with the same title accepted to Asiacrypt 2025,

Keywords: symmetric-key cryptography · sponge · Ascon · KMAC · post-quantum cryptography · Even-Mansour · Tweakable Even Mansour

1 Introduction

Post-quantum public-key cryptography has been gaining significant attention, and the standardization process by NIST is in progress [68]. Meanwhile, ensuring the post-quantum security of *symmetric-key* cryptosystems is also crucial for protecting communications and data from adversaries equipped with quantum computers. So far, there have been no findings suggesting that an efficient quantum algorithm breaks AES like Shor's algorithm [74] breaks RSA, but the security of symmetric-key schemes will undoubtedly deteriorate. Besides the well-known Grover's algorithm [49], which provides a quadratic speed-up for the exhaustive key search, previous research has indicated that a speed-up exceeding that of Grover's is possible even in so-called the Q1 model [26], highlighting the importance of careful evaluation of practical security.

Q1 and Q2. Here, Q1 refers to an attack model in which the attacker possesses a quantum computer but can only make classical queries to the oracles of keyed constructions, such as encryption oracles, MAC oracles, and so on. There's another significant quantum attack model known as Q2 [54], which allows all queries to be in quantum superposition. However, this paper focuses on Q1, assuming that the targeted schemes are implemented on classical devices as usual.

Sponge-Based Constructions. Symmetric-key modes can be classified into some groups depending on the underlying primitives. One of the major groups is sponge-based, which builds a hash function, an Authenticated Encryption with Associated Data (AEAD) scheme, and so on, based on the sponge construction [18,19] (or its variant, duplex [21]). Sponge-based constructions have widely been adopted, including standards such as SHA-3 [69], SHA-3 derived functions [55], and Ascon [75,41]. Other notable examples include hash functions such as QUARK [9], PHOTON [52], SPONGENT [23], various keyed sponge modes [20,31,15,7,64,38], and AEADs such as SpongeWrap and its variant modes [21,38,44,63], NORX [10], Ketje [16], Keyak [17], ISAP [40], PHOTON-Beetle [11], Subterranean [37], Wage [4], SPARKLE [12], Xoodyak [36], Spook [14], to name a few.

Security Proofs in the (Quantum) Ideal Permutation Model. Sponge-based constructions are built on top of a keyless public permutation. Thus, their security proofs are usually given in the ideal permutation model, where the underlying permutation is modeled as an ideally random permutation P, and adversaries are given oracle access to P and P^{-1}.

Proofs in the classical ideal permutation model do not guarantee security against quantum adversaries, just as proofs in the classical random oracle model

do not offer such guarantees[1]. Hence, security proofs given in the classical ideal permutations model must be carefully re-examined in the quantum ideal permutation model, where quantum superposition access to P and P^{-1} is allowed.

Ascon: Our Primary Focus. We focus on keyed schemes among sponge-based constructions, especially the Ascon AEAD mode by Dobraunig et al [42], which is included in the CAESAR final portfolio [41] as well as the (draft) standard of NIST SP 800-232 [75].

Ascon's structure is quite similar to other sponge-based AEADs, especially SpongeWrap [21]. Still, the security results for other schemes do not formally guarantee Ascon's security because of additional key insertions in the initialization and finalization processes. Thus, many previous papers worked on dedicated security proofs for the Ascon AEAD mode in the classical setting. Chakraborti et al. [29] showed the single-user nonce-respecting security [71,13], and Lefevre and Mennink [58] proved multi-user security in both the nonce-respecting and nonce-misuse [72] settings. Later, Chakraborti et al. also showed security bound in the same settings [30]. Other works [50,51,58,59] also studied more non-conventional security notions/models, such as the nonce-misuse resilience [8], leakage resilience [45,70], state-recovery security, and security under Release of Unverified Plaintext (RUP) [6]. A comprehensive survey about the classical security of Ascon is given in a recent paper by Lefevre and Mennink [59].

Ascon is mainly intended for lightweight applications, such as IoT devices. However, the importance of post-quantum security remains significant. If a quantum attack were to compromise an Authenticated Encryption with Associated Data (AEAD) scheme in just a few seconds, it could lead to numerous forgeries. In that case, the scheme would no longer be reliable for lightweight applications once a large-scale, fault-tolerant quantum computer is developed. Indeed, though not included in a standard, the designers also propose a parameter setting that particularly concerns the post-quantum security (Ascon-80pq) [42].

A recent paper on ePrint by Lang et al. [57] shows the post-quantum security of Ascon up to about $\min(2^{c/3}, 2^{\kappa/2})$ queries by using the one-way to hiding (O2H) lemma [5]. While the security level guaranteed by their result is sufficiently high for practical use, their findings come with certain limitations. Specifically, they require that nonces be selected from a predefined set that is significantly smaller than 2^{c-r} (where r is the rate and c is the capacity), and they allow only a single forgery attempt, which deviates from the typical security models for authenticated encryption with associated data (AEADs). In addition, there has been no work addressing the post-quantum provable security of Ascon in nonce-misuse scenarios or under RUP, which are critically important in practice.

Keyed Sponge Functions. KMAC is another keyed sponge-based scheme standardized by NIST [55], which is (roughly) a keyed version of SHA-3. Despite

[1] Indeed, some problems are hard in the classical random oracle model while easy in the quantum random oracle model [24], as shown in [76].

its importance, there has been no work that brings a dedicated post-quantum security proof of KMAC, or for more general keyed sponge functions such as the outer-keyed sponge [20,7], inner-keyed sponge [31], and full-keyed sponge [15,64] functions, to the authors' best knowledge.

1.1 Our Contributions

Considering the above, this paper shows the PRF security of keyed sponge functions and various security notions of the Ascon AEAD mode in the quantum ideal permutation model. All the results are in the single-user setting.

First, we observe that the post-quantum security of keyed sponge function can be proven by adapting a modular proof approach in the classical setting into quantum: Some prior studies [31,7,64] in the classical setting regard keyed functions as iterative applications of a (low-entropy) single-key Even-Mansour cipher, and reduce the security to (1) the sponge hash built on top of a secret permutation that adversaries cannot access and (2) the Even-Mansour cipher. Adapting this into the quantum setting, the post-quantum security of keyed sponge functions will follow from (1) the *classical* security of the sponge hash built on top of a secret permutation and (2) the post-quantum security of a (low-entropy) single-key Even-Mansour cipher (EM) [46] in the quantum ideal permutation model. This idea enables us to delegate the most technically challenging part, handling quantum superposition queries to the ideal permutation, to the security proof of the Even-Mansour cipher.

Next, we adapt the above idea further to the Ascon AEAD mode. Ascon cannot be regarded as an iterative application of Even-Mansour because of the additional key insertions the initialization and finalization phases. Nevertheless, we observe that it can be viewed as built upon a single-key *Tweakable* Even-Mansour cipher (TEM) [32] with a low-entropy (or, non-uniform) key. Roughly, this means that the post-quantum security of the Ascon AEAD mode can be shown by proving (1) its classical security when the permutations (and keys) are replaced with an ideally random tweakable block cipher that adversaries cannot access and (2) the post-quantum security of a single-key TEM with a low-entropy key in the quantum ideal permutation model.

Now, the most technically challenging issue is how to prove the post-quantum security of such (T)EM. Two previous works have already shown the post-quantum security of the Even-Mansour cipher [1] and TEM [2], but they assume that the keys are uniform, which is not directly applicable to our situation. In addition, their core proof techniques, called the resampling lemmas, are inapplicable to TEM with a low-entropy (or, non-uniform) key: The resampling lemma in [1] is not strong enough to manage tweaks. Moreover, the security of TEM in [1] is derived as a corollary of the security result of a variant cipher (TEM-KX), and the resampling lemma in [2] is tailor-made so it will fit a special structure of TEM-KX and inapplicable to our case. Thus, we show another resampling lemma, based on which we prove the security of TEM with a low-entropy key.

Once the post-quantum security of TEM with a low-entropy key is established, we can combine it with certain classical security results to achieve the

PRF security of keyed sponge functions and various security notions of the Ascon AEAD mode in the quantum ideal permutation model. Regarding the security notions for Ascon, we show those highlighted in the recent SoK paper [59] as much as possible. As for confidentiality, we prove the standard nonce-respecting security and the nonce-misuse resilience. In terms of authenticity, we show the security in the nonce-respecting, nonce-misuse, and RUP settings. The results roughly guarantee the security until the underlying permutations are called about $\min(2^{c/3}, 2^{\kappa/3})$ times, either through processing classical encryption/decryption queries or as direct (offline) evaluations in quantum superposition. This security bound itself is not very high, but we are the first to show that Ascon does not have a structural weakness against post-quantum adversaries in the standard nonce-respecting model without restrictions on nonce sets or forgery attempts, and even in the nonce-misuse or RUP settings.

For keyed sponge functions, we show the security up to about $\min(2^{c/3}, 2^{\kappa/3})$ queries for the inner- and full-keyed sponges, and about $\min(2^{c/3}, 2^{(\kappa-r)/2}, 2^{r/2})$ for outer-keyed sponge (ignoring some small factors). In particular, if the key is sufficiently long, KMAC(XOR)128 and KMAC(XOF)256 have more than 80-bit and 160-bit security, respectively.

In the aforementioned paper by Alagic et al. [2], they also proved the security of Chaskey [65], Elephant [22], and (a variant of) Minalpher [73]. The proofs are obtained by reducing the security to TEM-KX (or its variant), and TEM with uniform keys. So, our approach can also be regarded as an adaptation of their approach to keyed sponges and Ascon.

This paper focuses on the Ascon AEAD mode and keyed sponge functions, but the same approach will work for various permutation-based constructions, as long as they can be viewed as an iterative application of (tweakable) Even-Mansour. For instance, a classical security proof of the full-state keyed duplex is given with such an approach [64], and so its post-quantum security will immediately follow from the classical result and the post-quantum security of the Even-Mansour cipher. We expect that our findings will enhance the understanding of the post-quantum security not only for existing keyed sponge-based constructions but also for future schemes. Additionally, the post-quantum security bound of TEM with a low-entropy key will also be useful for future works.

1.2 Related Works

Czajkowski et al. [34] proved that the sponge hash is collapsing (a stronger version of collision-resistance) in some settings, especially when the underlying primitive is a random function or a (non-invertible) random permutation. Another paper by Czajkowski et al. [35] showed that the sponge hash becomes quantumly-secure PRF in the Q2 model when adversaries are not given oracle access to the underlying primitive. Some other works studied the security of sponge hash when the underlying primitive is an invertible random permutation and adversaries are given quantum oracle access to it, but the number of blocks is constant [77,27,28,60,33].

A recent paper by Alagic et al. [3] shows the quantum indifferentiability of the sponge hash without restrictions on the number of blocks[2]. The security of KMAC immediately follows from this, but the derived bound is not as good as ours (security up to $O(\min(2^{r/9}, 2^{c/9}))$ queries).

Janson and Struck [53] showed the post-quantum security of a sponge-based AEAD mode called SLAE [39] with the O2H lemma [5]. Ascon's structure differs greatly from SLAE, so their technique is inapplicable to Ascon. [53] also discusses a PRF, called SLFUNC. At first glance, it may seem almost equal to the keyed sponge functions in our paper. However, the primitive underlying SLFUNC is a (non-invertible) random *function*, so their proof is inapplicable to our case, either.

1.3 Paper Organization

Section 2 introduces various notations and definitions used throughout the paper. Section 3 and Section 4 elaborate on the basic ideas on how to prove the post-quantum security of keyed sponge functions and the Ascon AEAD mode, respectively. Section 5 proves the new resampling lemma and the post-quantum security of TEM with a low-entropy (or, non-uniform) key. Section 6 and Section 7 provide formal security of Ascon and keyed sponge functions, respectively, and Section 8 concludes the paper.

2 Preliminaries

We assume that readers are familiar with the basics of quantum computation [67].

For two integers $a, b \geq 0$, let $(a \mod b)$ denote the smallest integer c in $\{0, 1, \ldots, b-1\}$ such that $(a - c)$ is a multiple of b. For any bit strings x and y, their concatenation is denoted by $x||y$, and $|x|$ denotes the bit length of x. The most and least significant n bits of x are denoted as $\mathsf{msb}_n(x)$ and $\mathsf{lsb}_n(x)$, respectively. The empty bit string is denoted by $\varnothing$, and we regard $|\varnothing| = 0$. The set of all bit strings of finite length (including the empty string) is denoted by $\{0,1\}^*$, and $(\{0,1\}^n)^+$ denotes all the bit strings whose length is a positive multiple of n. For a positive integer r and a bit string x such that $|x| = rt$ for some t, we denote the operation of splitting x into r-bit blocks as $x_1, \ldots, x_t \xleftarrow{r} x$ (i.e., $x_i \in \{0,1\}^r$ for each i and $x = x_1||\cdots||x_t$ holds).

$\mathsf{Perm}(n)$ denotes the set of all permutations over $\{0,1\}^n$. A tweakable permutation over $\{0,1\}^n$ with tweak space $\mathcal{T}$ is a function $\tilde{P} : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ such that $\tilde{P}(t, \cdot)$ is a permutation for all $t \in \mathcal{T}$. Elements of $\mathcal{T}$ are called tweaks. The inverse of $\tilde{P}(t, \cdot)$ is denoted by $\tilde{P}^{-1}(t, \cdot)$. $\mathsf{TPerm}(\mathcal{T}, n)$ denotes the set of all such tweakable permutations.

[2] When studying the quantum indifferentiability of hash functions, no secret key is involved, and thus adversaries are naturally assumed to make quantum superposition queries to both the construction and primitive oracles. This model could also be regarded as Q2.

For a probability distribution D over a (finite) set $\mathcal{X}$, the min-entropy is defined as $H_\infty(D) := -\log \max_{x\in\mathcal{X}} D(x)$. For an algorithm $\mathcal{A}$, by $x \leftarrow \mathcal{A}$ we denote that $\mathcal{A}$ runs and returns x as an output. For a finite set $\mathcal{X}$, by $x \xleftarrow{\$} \mathcal{X}$ we denote that x is chosen uniformly at random from a finite set $\mathcal{X}$.

Throughout the paper, we assume adversaries are classical or Q1, i.e., they have only classical access to the oracles of keyed constructions such as encryption and decryption oracles.

2.1 Security Notion for Keyed Functions

A keyed function with κ-bit keys is a function $F : \{0,1\}^\kappa \times \mathcal{X} \to \mathcal{Y}$ for some message space $\mathcal{X}$ and an output space $\mathcal{Y}$, where $\{0,1\}^\kappa$ is called the key space. To simplify notations, we sometimes write $F_k(x)$ instead of $F(k,x)$. The security of a keyed function is defined as the indistinguishability from a random function. For the keyed function F, we define the Pseudo-Random Function (PRF) advantage of an adversary $\mathcal{A}$ as

$$\mathsf{Adv}^{\mathtt{prf}}_F(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{F_k}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF}}\right]\right|, \tag{1}$$

where RF is a random function from $\mathcal{X}$ to $\mathcal{Y}$, and k is chosen according to a distribution which should be clear from the context.

2.2 Security Notins for (Tweakable) Block Ciphers

An n-bit block cipher with κ-bit keys is a keyed function $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ such that E_k is a permutation for each k. We denote the inverse of E by E^{-1} (i.e., $E^{-1}(k,x) := (E_k)^{-1}(x)$). The security of block ciphers is defined as the indistinguishability from an ideally random permutation. For a block cipher E, the Pseudo-Random Permutation (PRP) advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathtt{prp}}_E(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{E_k, E_k^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RP},\mathsf{RP}^{-1}}\right]\right|, \tag{2}$$

where RP is a random permutation over $\{0,1\}^n$, and k is chosen according to a distribution which should be clear from the context.

An n-bit tweakable block cipher with κ-bit keys and tweak space $\mathcal{T}$ is a keyed function $\tilde{E} : \{0,1\}^\kappa \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ such that $\tilde{E}_k(t,\cdot)$ is a permutation for all k and t. The inverse of $\tilde{E}_k(t,\cdot)$ is denoted by $\tilde{E}_k^{-1}(t,\cdot)$. The security of tweakable block ciphers is defined as the indistinguishability from an ideally random tweakable permutation. For a tweakable block cipher $\tilde{E}$, the Tweakable Pseudo-Random Permutation (TPRP) advantage is defined similarly to Eq. (2) as

$$\mathsf{Adv}^{\mathtt{tprp}}_{\tilde{E}}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{E_k, E_k^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\widetilde{\mathsf{RP}},\widetilde{\mathsf{RP}}^{-1}}\right]\right|, \tag{3}$$

where $\widetilde{\mathsf{RP}}$ is an ideally random tweakable permutation, which is sampled from $\mathsf{TPerm}(\mathcal{T}, n)$ uniformly at random.

2.3 Security Notions for AEADs

A nonce-based AEAD scheme Π consists of an encryption algorithm $\mathcal{E}$ and a decryption algorithm $\mathcal{D}$. $\mathcal{E}$ takes a key k, a nonce N, associated data A, and a message M as input, and returns a ciphertext C of length $|M|$ with a tag T of length τ. $\mathcal{D}$ takes inputs of the form (k, N, A, C, T), and returns a message of length $|C|$ or $\perp$. We require that $\mathcal{D}_k(N, A, \mathcal{E}_k(N, A, M)) = M$ is satisfied for all (k, N, A, M). We often write $\mathcal{E}_k(N, A, M)$ and $\mathcal{D}_k(N, A, C, T)$ instead of $\mathcal{E}(k, N, A, M)$ and $\mathcal{D}(k, N, A, C, T)$.

In this paper, we consider confidentiality and authenticity separately. This is because, as we will see, Ascon achieves the post-quantum authenticity even if nonces are arbitrarily repeated [72] or even unverified plaintexts are released from the decryption oracle [6], whereas the confidentiality is broken in such settings, even in the classical setting [59].

In the following definitions, k is a key sampled uniformly at random from $\{0,1\}^\kappa$ for some κ, and \$ is an oracle that always returns a random value of length $|M| + \tau$ for any input of the form (N, A, M). The following definitions basically follow [59], though we focus only on the single-user setting and change some notations.

Nonce-Respecting Security. The most standard confidentiality (resp., authenticity) notion of nonce-based AEAD schemes is defined as the indistinguishability of the encryption oracle from an ideally random functionality (resp., defined as unforgeability given oracle access to both the encryption and decryption oracles) against nonce-respecting adversaries [71,13].

The nonce-respecting confidentiality advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathtt{conf}}_{\Pi}(\mathcal{A}) := \left| \Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{E}_k}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\$}\right] \right|. \tag{4}$$

Here, $\mathcal{A}$ is assumed to be nonce-respecting, i.e., $\mathcal{A}$ can query arbitrary tuple (N, A, M) to the encryption oracle, but the nonces must be different for distinct queries.

In addition, the he nonce-respecting authenticity advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathtt{auth}}_{\Pi}(\mathcal{A}) := \Pr\left[\mathcal{A}^{\mathcal{E}_k, \mathcal{D}_k} \text{ forges}\right], \tag{5}$$

where we say that $\mathcal{A}$ forges if $\mathcal{D}_k$ returns a value that is not $\perp$. As usual, the restriction of nonce-respecting applies only to the encryption oracle; $\mathcal{A}$ is not allowed to repeat nonces in encryption queries, but allowed in decryption queries. To prevent trivial wins, $\mathcal{A}$ is prohibited from forwarding an output from the encryption oracle to the decryption oracle.

Nonce-Misuse Resistance Security. In real-world implementations, nonces are sometimes inappropriately handled and repeated in encryptions. The nonce-misuse resistance is the notion introduced to discuss security in such situations [72].

Specifically, the nonce-misuse resistance confidentiality and authenticity advantages $\mathsf{Adv}^{\mathtt{m\text{-}conf}}_{\Pi}(\mathcal{A})$ and $\mathsf{Adv}^{\mathtt{m\text{-}auth}}_{\Pi}(\mathcal{A})$ are defined in the same way as Eq. (4) and Eq. (5), respectively, except that $\mathcal{A}$ is now allowed to repeat nonces not only in decryption but also in encryption queries.

Nonce-Misuse Resillience Security. In the classical setting, the confidentiality of Ascon is broken in the nonce-misuse scenario, but it still achieves confidentiality in a weaker setting [59], which is called the nonce-misuse resilience [8]. Roughly, the nonce-misuse resilience captures that, even after some specific nonces are repeated, the confidentiality is maintained for other nonces.

For an AEAD scheme Π, the nonce-misuse resilience confidentiality advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathtt{mr\text{-}conf}}_{\Pi}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{E}_k,\mathcal{E}_k}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{E}_k,\$}\right]\right|,$$

where $\mathcal{A}$ is prohibited from querying a tuple (N, A, M) to the left (resp., right) oracle if the nonce N has been queried to the right (resp., left or right) oracle before.

Security Under Release of Unverified Plaintext. The decryption of nonce-based AEAD schemes is formally defined so it returns only a special symbol $\perp$ when the verification fails. However, in practical situations, it is sometimes inevitable to release the decrypted plaintext before verification, e.g., when the decryption is performed on a small IoT device without enough amount of memory to keep the entire plaintext until verification. To capture such situations, Andreeva et al. [6] introduced the security notions under Release of Unverified Plaintext (RUP). Since Ascon does not satisfy the confidientiality under RUP even in the classical setting [59], we explain only the authenticity notion.

Suppose that the decryption algorithm $\mathcal{D}_k$ is separated into two algorithms Dec_k and Ver_k, where the former is the decryption algorithm that always outputs a message M without verification, and Ver_k is the verification algorithm that outputs either $\top$ or $\perp$, depending on whether the verification succeeds or not. $\mathcal{D}_k(N, A, C, T) = \mathsf{Dec}_k(N, A, C, T)$ holds if $\mathsf{Ver}_k(N, A, C, T) = \top$, and $\mathcal{D}_k(N, A, C, T) = \perp$ if $\mathsf{Ver}_k(N, A, C, T) = \perp$. When oracle access to the triple $(\mathcal{E}_k, \mathsf{Dec}_k, \mathsf{Ver}_k)$ is given, the RUP authenticity advantage of an adversary $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\Pi}(\mathcal{A}) := \Pr\left[\mathcal{A}^{\mathcal{E}_k,\mathsf{Dec}_k,\mathsf{Ver}_k} \text{ forges}\right],$$

where we say that $\mathcal{A}$ forges if Ver_k returns $\perp$. To prevent trivial wins, $\mathcal{A}$ is prohibited from forwarding an output of the encryption oracle $\mathcal{E}_k$ to the verification oracle Ver_k. When discussing RUP authenticity, we always allow $\mathcal{A}$ to repeat nonces for all the oracles.

2.4 Security in the (Quantum) Ideal Permutation Model

When a scheme, e.g., a block cipher or an AEAD, is built upon a public permutation P, we discuss the security in the (quantum) ideal permutation model

where P is modeled as an ideally random permutation, and an adversary is given oracle access to P and P^{-1}, as well as keyed oracles. When an adversary is quantum, we assume quantum superposition access to P and P^{-1} is allowed. Queries can be fully adaptive; adversaries can choose a query value to a classical oracle (that depends on P) after seeing results of quantum queries to P or P^{-1}, and vice versa. For example, the definition of the PRF advantage of a keyed function $F_k[P]$ built on top of P becomes like $\mathsf{Adv}_F^{\mathtt{prf}}(\mathcal{A}) = \left|\Pr\left[1 \leftarrow \mathcal{A}^{F_k[P],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF},P,P^{-1}}\right]\right|$.

2.5 On the Types of Queries and Count of the Lengths

Suppose a scheme (e.g., AEAD) is built upon a smaller primitive, e.g., an ideally random permutation P. In that case, we refer to the scheme's oracle (resp., the primitive's oracle) as the construction oracle (resp., primitive oracle) and queries to the construction oracle (resp., primitive oracle) as construction queries (resp., primitive queries). In addition, we refer to the total number of primitive calls required to process a query to the construction oracle as *the number of effective blocks of the query*.

2.6 (Tweakable) Even-Mansour Ciphers

The Original Construction. Given a key $(k_1, k_2) \in (\{0,1\}^n)^2$ and a public permutation $P \in \mathsf{Perm}(n)$, the Even-Mansour cipher [46] is defined as

$$E_{k_1,k_2}[P](x) := P(x \oplus k_1) \oplus k_2.$$

The classical and quantum security of the construction is summarized as follows.

- In the classical ideal permutation model, the construction is proven to be a secure PRP up to $O(2^{n/2})$ queries [46] when the keys are random.
- In the Q2 model (i.e., when quantum queries to the encryption oracle are allowed, unlike our setting), Kuwakado and Morii showed a polynomial time quantum attack breaking the construction with Simon's algorithm [56].
- In the Q1 model (i.e., when queries to $E_k[P], E_k^{-1}[P]$ are classical while those to P, P^{-1} are quantum, like this paper's setting), attacks of complexity around $O(2^{n/3})$ are known [56,25]. However, no polynomial-time attack has been found, unlike in the Q2 model. At Eurocrypt 2022, Alagic et al. [1] showed that the construction is indeed post-quantumly secure (i.e., secure in the Q1 model) up to $O(2^{n/3})$ queries if k_1 and k_2 are chosen uniformly at random.

In later sections, we will also consider variant constructions such that $k_1 = k_2$ and the key is sampled from a low-entropy (or, non-uniform) distribution (e.g., $k_1 = k_2 = 0^{n-\kappa}||k'$ for some uniformly random $k' \in \{0,1\}^\kappa$). We also refer to them as the Even-Mansour cipher.

Tweakable Even-Mansour Cipher. Let $P \in \mathsf{Perm}(n)$ be a public permutation, $\mathcal{T}$ be a (tweak) space, and $f_1, f_2 : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ some two functions. The Tweakable Even-Mansour cipher [32] (TEM) is an n-bit tweakable block cipher with tweak space $\mathcal{T}$ and n-bit key, whose encryption is defined as

$$\mathsf{TEM}_k[P](t,x) := P(f_1(t,k) \oplus x) \oplus f_2(t,k).$$

Here, $t \in \mathcal{T}$ denotes a tweak, and k is an n-bit key (sampled from a possibly low-entropy, or non-uniform distribution).

In the classical ideal permutation model, Cogliati et al. [32] proved that TEM is a secure TPRP up to $O(2^{n/2})$ queries if $f_1 = f_2$ and it is an almost XOR universal function. At Eurocrypt 2024, Alagic et al. [2] showed that the construction is also a post-quantumly secure TPRP up to $O(2^{n/3})$ queries in the quantum ideal permutation model if f_1 and f_2 are *proper*, and k is chosen uniformly at random. Here, proper functions are defined as follows.

Definition 1 (Proper functions). *A function $f : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is proper if $f(t,\cdot)$ is a permutation over $\{0,1\}^n$ for all $t \in \mathcal{T}$, and it holds that $\Pr_{x \xleftarrow{\$} \{0,1\}^n}[f(t,x) \oplus f(t',x) = y] \leq 2^{-n}$ for all y and $t \neq t'$.*

2.7 On Quantum Adversaries Making Only Classical Queries

If a classical security result concerns only the number of queries as an adversary's computational resources (and does not care about running time, space complexity, etc), then the result is valid even for quantum adversaries that make only classical queries: In general, the behavior of quantum algorithms without oracles can be simulated by classical algorithms with arbitrary precision if there is no restriction on time and space complexity. Thus, quantum algorithms with *classical* oracles can also be simulated by classical algorithms. We will use this fact without any mention.

3 Observations on Keyed Sponge Functions

3.1 Sponge Hash

The sponge construction is a hash function of variable output lengths, or an eXtendable Output Function (XOF), built upon a public permutation [18,19]. Given an n-bit permutation P, two parameters r and c such that $r+c = n$ (which are called rate and capacity), and an injective function $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^r)^+$ (which is called the padding function), the sponge construction computes the output as follows (see also Fig. 1).

Absorbing. Each input message $M \in \{0,1\}^*$ is first computed padded and splitter into r-bit blocks, i.e., $M_1, \ldots, M_\ell \xleftarrow{r} \mathsf{pad}(M)$ is computed[3]. The sponge function sets the initial internal state as $S \leftarrow 0^n$, and then iteratively updates it as $S \leftarrow P(S \oplus (M_i || 0^c))$ for $i = 1, \ldots, \ell$ in sequential order.

[3] How ℓ depends on the length of M varies depending on the concrete definition of pad.

Squeezing. Let Z_1 be the most significant r bits of the internal state S just after the absorbing phase. If an m-bit output is needed for some $m \leq r$, then $\mathsf{msb}_m(Z_1)$ is returned as the output. If a longer output (length m for some $m \geq r$) is needed, then the state is repeatedly updated as $S \leftarrow P(S)$ and $Z_i \leftarrow \mathsf{msb}_r(S)$ is computed, for $i = 2, \ldots, \lceil m/r \rceil$, and finally $\mathsf{msb}_m(Z_1 || \cdots || Z_{\lceil m/r \rceil})$ is returned as the output.

We denote the hash function by $\mathsf{Sponge}[P]$. The construction is proven to be secure up to $O(2^{c/2})$ queries [19] in the sense of indifferentiability [61].

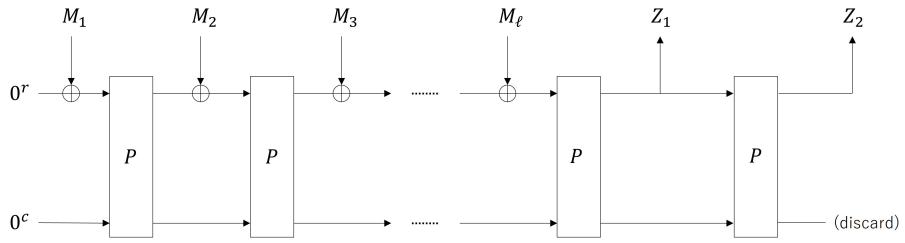

Fig. 1. The sponge hash function ($\mathsf{Sponge}[P]$). Inner keyed sponge ($\mathsf{IKS}_k[P]$) is obtained by replacing the 0^c in the lower left with $0^{c-\kappa}||k$.

Variants. For later use, we define some variants as follows.

- $\mathsf{Sponge}^{\mathsf{w/o\text{-}pad}}[P]$ is the sponge function without the padding function such that the length of input messages are always a multiple of r. (The domain is $(\{0,1\}^r)^+$ instead of $\{0,1\}^*$, and $\mathsf{Sponge}[P](M) = \mathsf{Sponge}^{\mathsf{w/o\text{-}pad}}[P](\mathsf{pad}(M))$ holds.)
- $\mathsf{Sponge}_{IV}[P]$ denotes the sponge function such that the initial state is changed from 0^n to some value $IV \in \{0,1\}^n$ ($\mathsf{Sponge}_{0^n}[P] = \mathsf{Sponge}[P]$ holds). We also define $\mathsf{Sponge}_{IV}^{\mathsf{w/o\text{-}pad}}[P]$ analogously.
- $\mathsf{FSSponge}[P]$ (abbreviation of Full-State Sponge) is the construction that absorbs message blocks into full states. That is, now the range of the padding function is $(\{0,1\}^n)^+$ instead of $(\{0,1\}^r)^+$, message blocks M_i are in $\{0,1\}^n$ instead of $\{0,1\}^r$ (computed as $M_1, \ldots, M_\ell \xleftarrow{n} \mathsf{pad}(M)$), and the state update of the absorbing part is done as $S \leftarrow P(S \oplus M_i)$ for $i = 1, \ldots, \ell$. The squeezing part remains unchanged. We also define $\mathsf{FSSponge}_{IV}$, $\mathsf{FSSponge}^{\mathsf{w/o\text{-}pad}}$, and $\mathsf{FSSponge}_{IV}^{\mathsf{w/o\text{-}pad}}$ analogously.

3.2 Keyed Sponge Functions

There are three major types of keyed functions built upon sponge hash, which are called the Outer-Keyed Sponge (OKS) [20,7], Inner-Keyed Sponge (IKS) [31], and Full-Keyed Sponge (FKS) [15,64].

Outer-Keyed Sponge (OKS*).* OKS just appends a key k as a prefix to input messages of the original sponge hash. More specifically, given the key k and input message M, OKS first pads k into k^* with some constant strings $\mathsf{s}_1, \mathsf{s}_2$ as the prefix and suffix so the length becomes a multiple of r. That is,

$$k^* := \mathsf{s}_1 || k || \mathsf{s}_2 \tag{6}$$

and $|k^*|$ is a multiple of r (here, the length of $\mathsf{s}_1, \mathsf{s}_2$ may be greater than equal to r). Then, OKS computes the output value as $\mathsf{OKS}_k[P](M) := \mathsf{Sponge}[P](k^*||M)$. The KMAC and KMACXOF functions specified by NIST [55] are instantiations of OKS with $n = 1600$ and $c = 256$ or $c = 512$. (See Section A in the appendix for the specification details of KMAC(XOF).)

Inner-Keyed Sponge (IKS*).* IKS sets a key k of length κ ($\leq c$) in the capacity part of the initial state (see also Fig. 1). That is, the function is defined as $\mathsf{IKS}_k[P](M) := \mathsf{Sponge}_{0^{n-\kappa}||k}[P](M)$.

Full-Keyed Sponge (FKS). FKS is defined in the same was as IKS, but using the $\mathsf{FSSponge}$ instead of Sponge. Specifically, given a key k of length κ ($\leq c$), the function FKS is defined as $\mathsf{FKS}_k[P] := \mathsf{FSSponge}_{0^{n-\kappa}||k}[P]$.

The security of all the three schemes is well-studied in the classical ideal permutation model [20,15,7,47,64,66,62]. For later use, we also define $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}$, $\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}$, and $\mathsf{FKS}^{\mathsf{w/o\text{-}pad}}$ as the padding-less version of OKS, IKS, and FKS, respectively (the domain of $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}$ and $\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}$ is $(\{0,1\}^r)^+$ while $(\{0,1\}^n)^+$ for $\mathsf{FKS}^{\mathsf{w/o\text{-}pad}}$).

3.3 Idea for Post-Quantum Security Proof of Keyed Sponges

Inner-Keyed Sponge. Inner-keyed sponge was first introduced by Chang et al. [31]. Their most important observation was that IKS can be viewed as iterative applications of a (single-key) Even-Mansour cipher as in Fig. 2. Namely, denoting the Even-Mansour cipher corresponding to the dotted box in the figure as $E_k[P]$, we have $\mathsf{IKS}_k[P] = \mathsf{Sponge}[E_k[P]]$.

Chang et al.'s original motivation for the alternative view was to prove the security of some keyed sponge constructions in the classical standard model[4], but similar approaches are later also used to prove the security of not only IKS but also other constructions in the classical ideal permutation model by, e.g., Andreeva et al. [7] and Mennink et al. [64].

Our observation here is that it also helps provide a post-quantum security proof in the quantum ideal permutation model. The idea is quite simple, but it will be the basis of a post-quantum proof approach for the Ascon AEAD mode, so we explain in detail.

[4] By "standard model", we ambiguously mean a model that do not introduce any idealized primitive (e.g., a random oracle or an ideal permutation) and assume that a primitive with a secret key (e.g., a block cipher) satisfies a falsifiable hypothesis (e.g., it is a PRP, PRF, and so on).

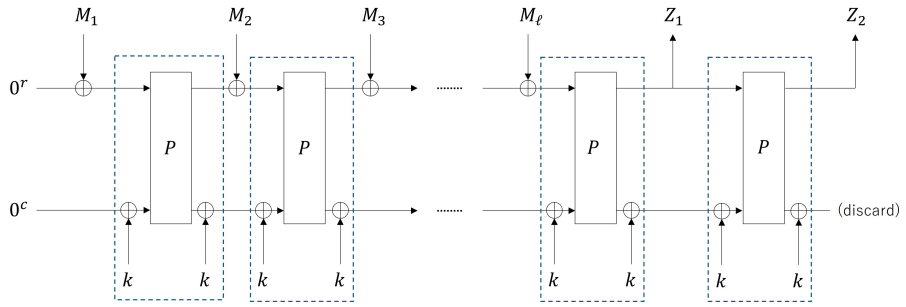

Fig. 2. An alternative view of the inner keyed sponge (when $|k| = c$). Each dotted box can be regarded as a single call to the Even-Mansour construction with c-bit key k.

Specifically, $\mathsf{IKS}_k[P]$ is a post-quantumly secure PRF in the quantum ideal permutation model if the following two claims hold.

1. The (single-key) Even-Mansour cipher $E_k[P]$ is a post-quantumly secure PRP in the quantum ideal permutation model. Namely, the classical oracles of $E_k[P]$ and a random permutation π (independent of P) are indistinguishable, even if quantum oracle access to P and P^{-1} is additionally given. (This implies that the classical oracles of $\mathsf{IKS}_k[P] = \mathsf{Sponge}[E_k[P]]$ and $\mathsf{Sponge}[\pi]$ are indistinguishable, even if quantum oracle access to P and P^{-1} is additionally given. Note that π is secret and the oracle is not given to adversaries.)
2. The classical oracles of $\mathsf{Sponge}[\pi]$ and a random function RF are indistinguishable if π is a secret permutation to which adversaries do not have oracle access. (This implies that the classical oracles of $\mathsf{Sponge}[\pi]$ and RF are indistinguishable, even if *quantum* oracle access to P and P^{-1} is additionally given, as π and RF are independent of P.)

The second claim immediately follows from the classical result on the indifferentiability of sponge [19] (see also Section 2.7). In addition, the post-quantum security of the Even-Mansour construction in the ideal permutation model has already been shown [1] in a specific setting. The cipher $E_k[P]$ in Fig. 2 is slightly different from the setting of [1] because the two keys before and after P are the same, and the entropy of the key k is at most c ($< n$). Yet, it is natural to expect that the security of $E_k[P]$ may also be proven in a similar way.

We will later show that the post-quantum security of the inner-keyed sponge can indeed be shown by this approach.

Full-Keyed Sponge. The proof idea for FKS is similar. The construction is decomposed into the iterative application of the Even-Mansour, so the post-quantum security of FKS follows from the classical PRF security of $\mathsf{FSSponge}[\pi]$ (for a secret permutation π) and the post-qunatum PRP security of the Even-Mansour cipher with a low-entropy key in the quantum ideal permutation model.

(This can be regarded as an adaptation of the classical proof idea [64] into the quantum setting.)

Outer-Keyed Sponge. To prove the security of OKS, we split the construction into the key processing part and the remaining part, and show the PRNG security of the former and the PRF security of the latter separately. This is also an adaptation of the previous works in the classical setting [7,47,62] into the quantum setting[5]. For the sake of brevity, below we ignore the padding and consider the security of $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}$ instead of OKS.

Let $\mathsf{KProc}[P]$ be the key processing part of the outer-keyed sponge. That is, $\mathsf{KProc}[P]$ is a function that computes an n-bit string from a key k as follows: First, $\mathsf{KProc}[P]$ computes k^* as in Eq. (6). Assume $k^* = k_1^*||\cdots||k_{|k^*|/r}^*$ holds for some $k_i^* \in \{0,1\}^r$ $(i = 1, \ldots, |k^*|/r)$. Then, $\mathsf{KProc}[P]$ sets the initial state as $S_0 \leftarrow 0^n$, and updates it $S_i \leftarrow P(S_{i-1} \oplus (k_i^*||0^c))$ for $i = 1, \ldots, |k^*|/r$. Finally, $\mathsf{KProc}[P]$ returns the n-bit state $S_{|k^*|/r}$ as the output.

If the key length is sufficiently long, the PRNG security of $\mathsf{KProc}[P]$ can again be reduced to the security of the Even-Mansour cipher: For example, suppose $\mathsf{s}_1 = \mathsf{s}_2 = \varnothing$ and $|k| = 2r$ (and thus $k^* = k$), and let $E_k[P]$ be a single-key Even-Mansour cipher such that $E_k[P](x) := P(x \oplus (k_1^*||0^c)) \oplus (k_2^*||0^c)$ Then, $\mathsf{KProc}[P](k) = P(E_k[P](0^n))$ holds. If $E_k[P]$ is a post-quantumly secure PRP, then the value $E_k[P](0^n)$ is indistinguishable from a random string, and so is $\mathsf{KProc}[P](k) = P(E_k[P](0^n))$.

In addition, $\mathsf{OKS}_k^{\mathsf{w/o\text{-}pad}}[P] = \mathsf{Sponge}_{k'}^{\mathsf{w/o\text{-}pad}}[P]$ holds with $k' = \mathsf{KProc}[P](k)$. As discussed above, $k' = \mathsf{KProc}[P](k)$ is indistinguishable from random if k is sufficiently long. The PRF security of $\mathsf{Sponge}_{k'}^{\mathsf{w/o\text{-}pad}}[P]$ can be shown in the same way as FKS and IKS, given k' is uniformly random[6]. Hence, the PRF security of $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}$ follows.

More precise analysis is given in later sections.

4 Observations on Ascon

In light of the observation on keyed sponge functions in the previous section, here we investigate the possibility of proving the post-quantum security of the AEAD mode of Ascon by some alternative view like the one in Fig. 2.

4.1 Speficitation of Ascon

First, we quickly review the specification, following [42]. The AEAD mode of Ascon is a sponge-based scheme parametrized by four variables κ, r, a, and b. It

[5] The previous works essentially considered a slightly different notion [7,47,62] for KProc, which is so called the key prediction security [62]. This paper considers the PRNG security instead because the key prediction security is defined on the history of queries to P or P^{-1}, but all the values can be queried at once (in superposition) in the quantum setting.

[6] To apply the view of Fig. 2 again, we reveal the the rate part of k' to the adversary.

is built upon two 320-bit permutations, denoted by p^a and p^b. κ is the key length (≤ 160), and r is the rate. a and b are the number of rounds of the underlying permutations.

We sometimes denote the length of the permutation by n (i.e., $n = 320$), and the capacity $(n-r)$ by c. Let $\mathsf{pad} : \{0,1\}^* \to (\{0,1\}^r)^+$ be the padding function defined as $\mathsf{pad}(x) = x||10^{r-1-(x \mod r)}$.

Taking a κ-bit key k, a 128-bit nonce N, an associated data A, and a message M, the encryption algorithm $\mathcal{E}_{\kappa,r,a,b}(k, N, A, M)$ works as follows (see also Fig. 3).

Initialization. Set the initial state as $S \leftarrow p^a(IV_{\kappa,r,a,b}||k||N) \oplus (0^{n-\kappa}||k)$, where $IV_{\kappa,r,a,b}$ is a prespecified constant.

Processing Associated Data. If A is not empty, pad and split it into r-bit blocks as $A_1, \ldots, A_s \xleftarrow{r} \mathsf{pad}(A)$, and update the internal state as $S \leftarrow p^b(S \oplus (A_i||0^c))$ for $i = 1, \ldots, s$. Finally, (regardless of whether A is empty or not), update the state S as $S \leftarrow S \oplus (0^{n-1}||1)$.

Processing Plaintexts. Pad and split M into r-bit blocks as $M_1, \ldots, M_t \xleftarrow{r} \mathsf{pad}(M)$. For $i = 1, \ldots, t$, compute the ciphertext block $C_i \in \{0,1\}^r$ as $C_i \leftarrow M_i \oplus \mathsf{msb}_r(S)$, and then update the state S as $S \leftarrow p^b(S \oplus (M_i||0^c))$ if $i < t$ and $S \leftarrow S \oplus (M_i||0^c)$ if $i = t$. Let the ciphertext C be the most significant $|M|$ bits of $C_1||\cdots||C_t$.

Finalization. Update the state as $S \leftarrow p^a(S \oplus (0^r||k||0^{c-\kappa})) \oplus (0^{n-\kappa}||k)$, and compute the tag T as $T \leftarrow \mathsf{lsb}_{128}(S)$. Finally, output the ciphertext C and the tag T.

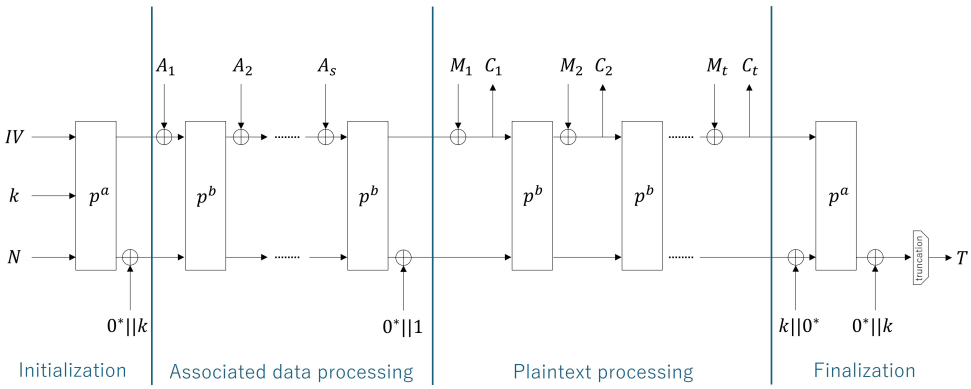

Fig. 3. Encryption by Ascon.

The decryption algorithm $\mathcal{D}_{\kappa,r,a,b}(k, N, A, M)$ is naturally defined from $\mathcal{E}_{\kappa,r,a,b}(k, N, A, M)$: The first two phases for initialization and processing associated data

are the same, while the latter two phases are replaced with the following procedure[7].

> **Processing Ciphertexts.** Pad and split C into r-bit blocks as $C_1, \ldots, C_t \xleftarrow{r} \mathsf{pad}(C)$. For $i = 1, \ldots, t$, compute the message block $M_i \in \{0,1\}^r$ as $M_i \leftarrow C_i \oplus \mathsf{msb}_r(S)$, and then update the state S as $S \leftarrow p^b(C_i||\mathsf{lsb}_c(S))$ if $i < t$ and $S \leftarrow (C_i||\mathsf{lsb}_c(S))$ if $i = t$. Let M be the most significant $|C|$ bits of $M_1||\cdots||M_t$.
> **Finalization.** Update the state as $S \leftarrow p^a(S \oplus (0^r||k||0^{c-\kappa})) \oplus (0^{n-\kappa}||k)$, and compute $\tilde{T}$ as $\tilde{T} \leftarrow \mathsf{lsb}_{128}(S)$. If $T = \tilde{T}$, then output the message M. If $T \neq \tilde{T}$, then output $\perp$.

In [42], the designers recommend two primary parameter settings: $(\kappa, r, a, b) = (128, 64, 12, 6)$ and $(\kappa, r, a, b) = (128, 128, 12, 8)$. The AEAD with the former setting is named Ascon-128 and the latter Ascon-128a, respectively. The designers also defines Ascon-80pq with the parameter setting $(\kappa, r, a, b) = (160, 64, 12, 6)$, expecting it to achieve 80-bit post-quantum security. The NIST SP 800-232 initial public draft [75] specifies Ascon-128a as a draft standard, modifying the name to Ascon-AEAD128. As additional options, the draft allows to truncate tag up to 64 bits, and allow to mask the nonce with an additional 128-bit key, but we do not consider these options in this section.

In the previous works showing the security of the Ascon AEAD mode in the classical ideal permutation model, there are two patterns on how to model the permutations p^a and p^b. The first is to model p^a and p^b as different public random permutations, and the second is to model them as a single public random permutation P. This paper adopts the latter and set $p^a = p^b = P$, following [29,30,59].

The Ascon family includes not only the AEAD mode but also a hash mode, but this paper studies only the former. In what follows, we refer to the AEAD mode just as Ascon.

4.2 An Alternative View of Ascon

We observe that Ascon can be regarded as built upon a TEM with low-entropy (or, non-uniform) keys if p^a and p^b are modeled as a single random permutation P.

AsconAlt: Ascon Viewed as an AEAD Built upon a TBC. Let n, r, κ be parameters such that $\max\{\kappa + r, \kappa + 128\} \leq n$. Let P be an n-bit ranodm permutation, and $\tilde{E}_k[P]$ be the n-bit tweakable block cipher with κ-bit key and

[7] When considering RUP authenticity, the message M is always returned, regardless of $T = \tilde{T}$ or not.

tweak space $\mathcal{T} = \{1, 2, 3\}$ such that

$$\tilde{E}_k[P](1, x) = P\left(x \oplus k_1\right) \oplus k_3, \tag{7}$$

$$\tilde{E}_k[P](2, x) = P\left(x \oplus k_2 \oplus k_3\right) \oplus k_2 \oplus k_3, \tag{8}$$

$$\tilde{E}_k[P](3, x) = P\left(x \oplus k_2\right) \oplus k_2, \tag{9}$$

where $k_1 := 0^{n-128-\kappa}||k||0^{128}$, $k_2 := 0^{n-\kappa}||k$, $k_3 := 0^r||k||0^{n-r-\kappa}$. Now, let $n = 320$, and suppose the encryption algorithm of Ascon is modified so the initialization is done as $S \leftarrow \tilde{E}_k[P](1, IV_{\kappa,r,a,b}||0^\kappa||N)$, associated data and plaintexts are processed with $\tilde{E}_k[P](2, \cdot)$ instead of p^b, and the state update in the finalization is done as $S \leftarrow \tilde{E}_k[P](3, S)$ (see Fig. 4). Then, the resulting algorithm, which we denote by AsconAlt, is equivalent to the original Ascon (with both p^a and p^b being replaced by P).

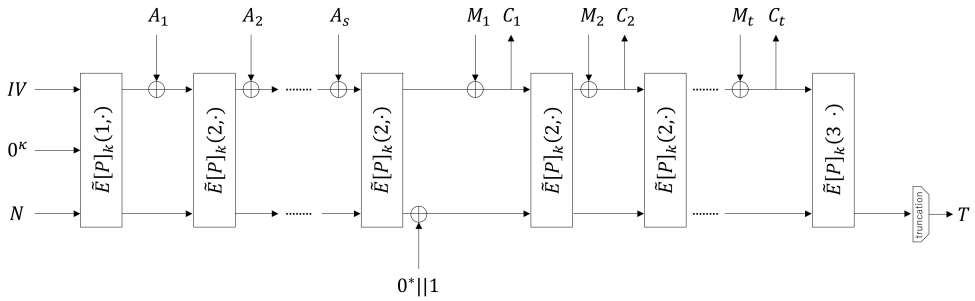

Fig. 4. Encryption by AsconAlt.

To see this, substitute the right sides of Eq. (7)-(9) into Fig. 4. Then, many of k_2 and k_3 are canceled out: For example, $\tilde{E}_k[P](1, \cdot)$ (the leftmost one in 4) adds k_3 after applying P, but this k_3 is canceled by the next $\tilde{E}_k[P](2, \cdot)$ that adds $k_2 \oplus k_3$. After the cancellation, only k_2 is left between the first (leftmost) and second applications of P. Cancelling out other (sub-)keys analogously, we obtain exactly Fig. 3 (with p^a and p^b replaced by P), just as IKS is obtained by canceling out intermediate keys of Fig. 2.

$\tilde{E}_k[P]$ is a TEM with Proper Functions and a low-entropy Key. Let $f(X) \in \mathbb{F}_2[X]$ be an irreducible polynomial of degree n, and identify $\{0,1\}^n$ with $\mathbb{F}_2[X]/(f(X))$ ($\cong \mathbb{F}_{2^n}$). Then, we have $k_1 = X^{128} \cdot (0^{n-\kappa}||k)$, $k_2 = 0^{n-\kappa}||k$, and $k_3 = X^{n-r-\kappa} \cdot (0^{n-\kappa}||k)$. In addition, let $\mathcal{T}$ be again $\{1, 2, 3\}$, and define $f_1, f_2 : \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ by

$$f_1(1, z) := X^{128} \cdot z, \quad f_1(2, z) := \left(X^{n-r-\kappa} + 1\right) \cdot z, \quad f_1(3, z) := z,$$
$$f_2(1, z) := X^{n-r-\kappa} \cdot z, \quad f_2(2, z) := \left(X^{n-r-\kappa} + 1\right) \cdot z. \quad f_2(3, z) := z.$$

Then $\mathsf{TEM}_{0^{n-\kappa}||k}[P] = \tilde{E}_k[P]$ holds, and both f_1 and f_2 are obviously proper. Thus, the TBC underlying AsconAlt is a TEM built with proper functions. In addition, the key of this TEM is not uniformly distributed over $\{0,1\}^n$ because it is of the form $0^{n-\kappa}||k$ for some $k \in \{0,1\}^\kappa$.

4.3 Idea for a Post-Quantum Security Proof

Analogously to Section 3.3, we obtain a post-quantum security proof of Ascon in the quantum ideal permutation model if the following two claims hold.

1. $\tilde{E}_k[P]$ defined above (or equivalently, TEM with low-entropy keys) is a post-quantumly secure TPRP in the quantum ideal permutation model. This means the following: Let AsconAlt$[\tilde{\pi}]$ denote the AsconAlt construction with $\tilde{E}_k[P]$ in Fig. 4 being replaced by an ideally random tweakable permutation $\tilde{\pi}$ with tweak space $\mathcal{T} = \{1,2,3\}$ ($\tilde{\pi}$ is independent of P). Then, the classical oracles of Ascon (which is equal to AsconAlt$[\tilde{E}_k[P]]$) and AsconAlt$[\tilde{\pi}]$ are indistinguishable, even if quantum oracle access to P and P^{-1} is given.
2. AsconAlt$[\tilde{\pi}]$ is a secure AEAD if only classical queries are allowed to adversaries (P no longer appears here).

The second claim could be shown by classical security proof techniques since all the queries are classical (see also Section 2.7). The issue is how to prove the first claim: The previous work [2] showed the security of TEM when the n-bit key is chosen uniformly at random. However, their proof technique is inapplicable to our TEM with a low-entropy key. Thus, we introduce a modified version of the technique and use it to prove the post-quantum security of our TEM. The next section elaborates on the details of this.

Remark 1. In the classical setting, the security proof of Chaskey [65] is given by showing a security notion, which is named the 3PRP security, of an Even-Mansour cipher. In the paper by Alagic et al. on TEM-KX and TEM (with uniform keys) [2], they observed that the 3PRP security is essentially the TPRP security of a TEM with uniform keys, and proved the post-quantum security of Chaskey by reducing it to TEM. Thus, our approach can also be regarded as an adaptation of their approach to Ascon and TEM with a low-entropy (or, non-uniform) key.

Remark 2. Our idea fully relies on the model in which both p^a and p^b in Ascon are replaced with a single random permutation P. If they are modeled as two independent random permutations, the reductions to AsconAlt and TEM do not work, and another proof technique will be necessary.

5 On the Post-Quantum Security of TEM with Low-Entropy Keys

In this section, we review the previous results and proof techniques about EM and TEM in [1] and [2], see why the core technique (resampling lemma) in [2]

used to prove TEM with uniform keys is inapplicable to our case, and explain how to prove the security of TEM with low-entropy (or, non-uniform) keys.

In what follows, for arbitrary $s_0, s_1 \in \{0,1\}^n$, by swap_{s_0,s_1} we denote the permutation over $\{0,1\}^n$ defined by $\mathsf{swap}_{s_0,s_1}(s_b) := s_{b\oplus 1}$ and $\mathsf{swap}_{s_0,s_1}(x) := x$ for $x \neq s_0, s_1$

5.1 Proof for Even-Mansour with Uniform Keys in [1]

As mentioned before, Alagic et al. [1] showed that Even-Mansour construction is a post-quantumly secure PRP up to $O(2^{n/3})$ queries, when the two n-bit keys are uniformly random. Their core technical tools are two lemmas about *arbitrary reprogramming* and *resampling*. The latter will also be important in our technical discussions, so we restate it here.

Lemma 1 (First resampling lemma [1]). *Consider the following two-stage experiment with a (quantum) distinguisher $\mathcal{D}$.*

1. *A random permutation $P_0 \in \mathsf{Perm}(n)$ is chosen, and $\mathcal{D}$ runs relative to the quantum oracles of P_0 and P_0^{-1}.*
2. *Choose $s_0, s_1 \in \{0,1\}^n$ uniformly at random, define $P_1 := P_0 \circ \mathsf{swap}_{s_0,s_1}$, and let b be a uniform random bit. $\mathcal{D}$ is given s_0, s_1, performs some additional computation with the quantum oracle access to P_b and P_b^{-1}, and finally outputs a bit b'.*

If the number of queries by $\mathcal{D}$ in Step 1 is at most q, then

$$|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]| \leq 4\sqrt{q/2^n}$$

holds.

We refer to Lemma 1 as the first resampling lemma to differentiate it from variants introduced later. It is a permutation version of a technique for random *functions* in a previous work [48, Theorem 1].

Very roughly, in the post-quantum security proof of the Even-Mansour, the resampling lemma is used to show the following thing: For each classical query x to the encryption oracle (resp., y to decryption oracle) and a random value s_1, an adversary does not notice even if the output values of P on $s_0 := x \oplus k_1$ and s_1 (resp., the output values of P^{-1} on $s_0 := y \oplus k_2$ and s_1) are swapped. (Together with another thing proven with the arbitrary reprogramming lemma, this eventually leads to proving that the adversary does not notice even if all the outputs of the encryption/decryption oracles are replaced with random values, and so the Even-Mansour is indistinguishable from random.)

5.2 Proof for TEM with Uniform Keys in [2]

Alagic et al. [2] showed the post-quantum security of TEM when the key is chosen from $\{0,1\}^n$ uniformly at random. The result is derived as a corollary of the post-quantum TPRP security of a variant construction TEM-KX defined by

$$\mathsf{TEM\text{-}KX}_k[P](t,x) := P(f_1(t, P(k||0^{n-\kappa})) \oplus x) \oplus f_2(t, P(k||0^{n-\kappa})),$$

where k is now a uniform key of length κ ($\leq n$) and f_1, f_2 are proper functions. Indeed, the post-quantum TPRP security of TEM with uniform keys (when f_1 and f_2 are proper) immediately follows from the post-quantum TPRP security of $\mathsf{TEM\text{-}KX}$ with $\kappa = n$.

To prove the security of $\mathsf{TEM\text{-}KX}$, Alagic et al. utilized the arbitrary reprogramming lemma again and a modified version of the resampling lemma (Lemma 1), which is adjusted so it can handle tweaks and it will fit the unique characteristic of $\mathsf{TEM\text{-}KX}$ that the key is processed with P before input to f_1 and f_2. The modified version, which we call the second resampling lemma, is as follows.

Lemma 2 (Second resampling lemma [2]). *Let $S \subset \mathsf{Perm}(n)$. Consider the following two-stage experiment with a (quantum) distinguisher $\mathcal{D}$.*

1. *A random permutation $P_0 \in \mathsf{Perm}(n)$ is chosen, and $\mathcal{D}$ runs relative to the quantum oracle of P_0 and P_0^{-1}. $\mathcal{D}$ outputs a distribution D over $\{0,1\}^n$ and a permutation $\tau \in S$.*
2. *Sample $\hat{s} \leftarrow D$, set $s_0 \leftarrow \tau(P_0(\hat{s}))$, and choose $s_1 \in \{0,1\}^n$ uniformly at random. Define $P_1 := P_0 \circ \mathsf{swap}_{s_0,s_1}$, and let b be a uniform random bit. $\mathcal{D}$ is given $\hat{s}$, performs some additional computation with the quantum oracle access to P_b and P_b^{-1}, and finally outputs a bit b'.*

If the number of queries by $\mathcal{D}$ in Step 1 is at most q, then

$$\left|\Pr\left[b' = 1 \mid b = 1\right] - \Pr\left[b' = 1 \mid b = 0\right]\right| \leq \sqrt{\varepsilon} \cdot \left(1 + \sqrt{q + \log(11|S|/\sqrt{\varepsilon})}\right)$$

holds, where $\varepsilon := 2 \cdot \mathbf{E}_{(D,\tau)\leftarrow\mathcal{D}^{P_0,P_0^{-1}}} \left[\max_{x\in\{0,1\}^n} \Pr_{x'\leftarrow D}\left[x' = x\right]\right]$.

5.3 Proof for TEM with Low-Entropy Key

To obtain a security proof for Ascon based on the idea we discussed in Section 4.3, we need a security proof for TEM when the key is low-entropy (or, non-uniform). Specifically, we need to prove the following proposition.

Proposition 1. *Suppose that f_1 and f_2 are proper, and the key $k \in \{0,1\}^n$ is sampled from a distribution D_{key} with min-entropy at least κ. For a (quantum) adversary $\mathcal{A}$ making at most q_C classical queries to TEM and TEM^{-1} and q_Q quantum queries to P and P^{-1},*

$$\mathsf{Adv}^{\mathtt{tprp}}_{\mathsf{TEM}}(\mathcal{A}) \leq 4\sqrt{\frac{(q_C)^2 q_Q}{2^\kappa}} + \frac{2(q_C)^2}{2^\kappa} + 2\sqrt{\frac{2q_C(q_Q)^2}{2^\kappa}}.$$

holds.

In fact, to prove the post-quantum security of Ascon, it suffices to show the claim of the proposition when only the encryption oracle of TEM is given while the decryption oracle TEM^{-1} is not. Still, we prove the strong claim above, hoping it

will help understand the post-quantum security of other constructions in future works.

To prove the proposition, we need another resampling lemma for the following reasons.

1. In the proof for TEM-KX, the tailor-made second resampling lemma (Lemma 2) is applied to manage the special structure that the key is processed with P before input to f_1 or f_2 (very roughly, $\hat{s}$ in the lemma corresponds to the key in the security proof of TEM-KX). However, it is not applicable to TEM, as the key is directly input to f_1 and f_2.
2. The first resampling lemma is not applicable, either: To handle tweaks, we have to let the distinguisher $\mathcal{D}$ in the resampling lemmas choose a distribution D from which s_0 is sampled (like the end of the first stage and the beginning of the first stage in Lemma 2).

With this in mind, we show a new variant of the resampling lemma below.

Lemma 3 (Our resampling lemma). *Consider the following two-stage experiment with a (quantum) distinguisher $\mathcal{D}$.*

1. *A random permutation $P_0 \in \mathsf{Perm}(n)$ is chosen, and $\mathcal{D}$ runs relative to the quantum oracles of P_0 and P_0^{-1}. Then, $\mathcal{D}$ chooses a distribution D over $\{0,1\}^n$ such that $H_\infty(D) \geq \kappa$ and outputs it.*
2. *Sample s_0 from D and choose $s_1 \in \{0,1\}^n$ uniformly at random. Define $P_1 := P_0 \circ \mathsf{swap}_{s_0,s_1}$, and let b be a uniform random bit. $\mathcal{D}$ is given s_0, s_1, performs some additional computation with the quantum oracle access to P_b and P_b^{-1}, and finally outputs a bit b'.*

If the number of queries by $\mathcal{D}$ in Step 1 is at most q, then

$$|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]| \leq 4\sqrt{q/2^\kappa} \tag{10}$$

holds.

Note that this lemma exactly matches the first resampling lemma if $|S| = 1$ and $\kappa = n$ (which implies that D is always the uniform distribution).

Once Lemma 3 is shown, the proof of Proposition 1 proceeds similarly to the proof for TEM-KX in [2], except that the new resampling lemma (Lemma 3) is used instead of Lemma 2. See Section C in the appendix for a complete proof.

On Proof of Lemma 3. The proof of Lemma 3 proceeds analogously to that of the first resampling lemma (Lemma 1 = [1, Lemma 5]) to some extent. Still, we need some dedicated analysis. Below, we provide a comparison with the proof of previous techniques, especially [48, Theorem 1], the first resampling lemma, and the second resampling lemma (Lemma 2 = [2, Lemma 3]).

[48, Theorem 1] concerns Random Oracles (ROs). The value s_0 (the input for which the value of an RO is reprogrammed) is sampled from a distribution D. Adversaries can choose D arbitrarily. Since the outputs of ROs are independent,

the quantum registers of the oracle's (purified) state storing $\mathrm{RO}(x)$ and $\mathrm{RO}(y)$ are unentangled for $x \neq y$. The proof of [48, Theorem 1] heavily relies on such properties specific to ROs (e.g., "Lemma 1 clearly also holds...applied" of [48, pp.662–663]).

The first resampling lemma [1, Lemma 5] reprograms a random permutation P_0 on a uniformly sampled s_0. Random permutations generally require higher-level proof techniques than ROs: Since outputs of P_0 are not independent, the oracle's quantum registers storing them are entangled. Moreover, superposition queries are made to not only P_0 but also P_0^{-1}. To deal with these difficulties, [1] uses sophisticated proof techniques specific to random permutations.

The second resampling lemma [2, Lemma 3] generalizes the first resampling lemma, allowing adversaries to choose a distribution D and a permutation τ. s_0 is generated as $s_0 := \tau(P_0(s))$, where s is sampled from D. In [2], the technical details of the proof are deeply interwoven with the assumption that s_0 is generated using P_0.

Our resampling lemma (Lemma 3) also generalizes the first resampling lemma, assuming that adversaries are allowed to choose D and that s_0 is *directly* sampled from D. This assumption is similar to [48, Theorem 1] and the second resampling lemma. However, in extending the first resampling lemma to ours, it seems difficult to leverage the proof techniques in [48,2] because of their characteristics mentioned above. Thus, we need a dedicated analysis. (Still, we do not argue that our proof itself is as fundamentally innovative as previous works. Our focus rather lies in proving the post-quantum security of important schemes, particularly Ascon, for the first time without non-standard restrictions.)

See Section B in the appendix for a complete proof of Lemma 3.

6 Post-Quantum Security of Ascon

This section shows the post-quantum security of Ascon in the quantum ideal permutation model, based on the idea from Section 4.3, applying Proposition 1. In what follows, τ denotes the tag length ($64 \leq \tau \leq 128$).

6.1 Security of AsconAlt

Again, let AsconAlt$[\tilde{\pi}]$ denote the AsconAlt construction of Fig. 4 with the underlying tweakable block cipher replaced with an ideally random tweakable permutation $\tilde{\pi}$. Then, the following two propositions hold.

Proposition 2 (RUP Authenticity of AsconAlt$[\tilde{\pi}]$). *Let $\mathcal{A}$ be a (possibly quantum) adversary against the RUP authenticity of AsconAlt$[\tilde{\pi}]$ making at most q_e (resp., q_d) classical queries to the encryption oracle $\mathcal{E}_k$ (resp., the decryption oracle Dec_k and the verification oracle Ver_k), the number of effective blocks being at most σ_e (resp., σ_d) in total. Then, it holds that*

$$\mathsf{Adv}^{\texttt{rup-auth}}_{\mathrm{AsconAlt}[\tilde{\pi}]}(\mathcal{A}) \leq \frac{2(\sigma_e+\sigma_d)}{2^\kappa} + \frac{18(\sigma_e+\sigma_d)^2}{2^c} + \frac{2q_d}{2^\tau} + \frac{2(\sigma_e+\sigma_d)^2}{2^\kappa}$$

Proposition 3 (Nonce-Misuse Resilience Confidentiality of AsconAlt$[\tilde{\pi}]$). *Let $\mathcal{A}$ be a (possibly quantum) adversary against nonce-misuse resilience confidentiality of AsconAlt$[\tilde{\pi}]$ making q_e classical encryption queries, the number of effective blocks being at most σ_e in total. Then, it holds that*

$$\mathsf{Adv}^{\texttt{mr-conf}}_{\mathrm{AsconAlt}[\tilde{\pi}]}(\mathcal{A}) \leq \frac{\sigma_e}{2^\kappa} + \frac{18(\sigma_e)^2}{2^c} + \frac{2(\sigma_e)^2}{2^\kappa}$$

Since all the queries are classical and there is no quantum oracle allowing $\mathcal{A}$ to make superposition queries, the propositions can be shown only with classical discussions (see also Section 2.7), as below.

Proof of Proposition 2. We prove Proposition 2 by using the following theorem in the classical setting shown in [59][8].

Theorem 1 (Theorem 6 of [59]). *Let $\mathcal{A}$ be the classical adversary against the RUP authenticity of Ascon in the classical ideal permutation model. Assume that $\mathcal{A}$ makes at most q_e (resp., q_d) classical queries to the encryption oracle $\mathcal{E}_k$ (resp., the decryption oracle Dec_k and the verification oracle Ver_k), the number of effective blocks being at most σ_e (resp., σ_d) in total, and at most q_C (classical) queries to P and P^{-1}. Then, it holds that*

$$\mathsf{Adv}^{\texttt{rup-auth}}_{\mathrm{Ascon}}(\mathcal{A}) \leq \frac{2(q_C+\sigma_e+\sigma_d)}{2^\kappa} + \frac{18(q_C+\sigma_e+\sigma_d)(\sigma_e+\sigma_d)}{2^c} + \frac{2q_d}{2^\tau}.$$

Proof (of Proposition 2). As all the queries are classical, it suffices to show the claim when $\mathcal{A}$ is a classical algorithm.

Let $\mathcal{E}_k$, Dec_k, and Ver_k be the encryption, decryption, and verification oracles of Ascon, respectively. In addition, let $\mathcal{E}'[\tilde{E}_k[P]]$, $\mathsf{Dec}'[\tilde{E}_k[P]]$, and $\mathsf{Ver}'[\tilde{E}_k[P]]$ (resp., $\mathcal{E}'[\tilde{\pi}]$, $\mathsf{Dec}'[\tilde{\pi}]$, and $\mathsf{Ver}'[\tilde{\pi}]$) denote the encryption, decryption, and verification oracles of AsconAlt built on top of the tweakable block cipher $\tilde{E}_k[P]$ defined as in Fig. 4 (resp., instantiated with a random tweakable block cipher $\tilde{\pi}$). Recall that Fig. 4 is just an alternative view of Ascon, and so

$$(\mathcal{E}_k, \mathsf{Dec}_k, \mathsf{Ver}_k) = (\mathcal{E}'[\tilde{E}_k[P]], \mathsf{Dec}'[\tilde{E}_k[P]], \mathsf{Ver}'[\tilde{E}_k[P]]) \tag{11}$$

holds.

Define probability p_1 and p_2 by $p_1 := \Pr\left[\mathcal{A}^{\mathcal{E}'[\tilde{E}_k[P]],\mathsf{Dec}'[\tilde{E}_k[P]],\mathsf{Ver}'_k[\tilde{E}_k[P]]} \text{ forges}\right]$ and $p_2 := \Pr\left[\mathcal{A}^{\mathcal{E}'[\tilde{\pi}],\mathsf{Dec}'[\tilde{\pi}],\mathsf{Ver}'[\tilde{\pi}]} \text{ forges}\right]$.

Let $\mathcal{B}$ be a (classical) adversary that tries to distinguish $(\tilde{E}_k[P], P, P^{-1})$ from $(\tilde{\pi}, P, P^{-1})$ as follows: $\mathcal{B}$ runs $\mathcal{A}$. When $\mathcal{A}$ makes a query to the encryption, decryption, or verification oracle, $\mathcal{B}$ answers by computing $\mathcal{E}'[\mathcal{O}]$, $\mathsf{Dec}'[\mathcal{O}]$, or $\mathsf{Ver}'[\mathcal{O}]$, respectively, where $\mathcal{O}$ is the construction oracle ($\tilde{E}_k[P]$ or $\tilde{\pi}$) given to $\mathcal{B}$. ($\mathcal{B}$ does not make queries to P nor P^{-1}.) $\mathcal{B}$ outputs 1 if $\mathcal{A}$ forges, and outputs 0 otherwise.

[8] The original theorem is proven in the multi-user setting. We use it here by setting the number of users to 1.

Then, $\mathcal{B}$ perfectly simulates $(\mathcal{E}'[\tilde{E}_k[P]], \mathsf{Dec}'[\tilde{E}_k[P]], \mathsf{Ver}'[\tilde{E}[P]])$ and $(\mathcal{E}'[\tilde{\pi}], \mathsf{Dec}'[\tilde{\pi}], \mathsf{Ver}'[\tilde{\pi}])$ when the construction oracle $\mathcal{O}$ given to $\mathcal{B}$ is $\tilde{E}_k[P]$ and $\tilde{\pi}$, respectively. In addition, $\mathcal{B}$ makes at most $(\sigma_e + \sigma_d)$ queries to either $\tilde{E}_k[P]$ and $\tilde{\pi}$, while no queries to P or P^{-1}. Hence,

$$|p_1 - p_2| = \mathsf{Adv}^{\mathtt{tprp}}_{\tilde{E}[P]}(\mathcal{B}) \leq \frac{2(\sigma_e + \sigma_d)^2}{2^\kappa} \tag{12}$$

follows from Proposition 1 (because post-quantum security bounds are also valid as classical security bounds).

Moreover, since Ascon is equivalent to AsconAlt built on the cipher $\tilde{E}_k[P]$ (and so Eq. (11) holds), $p_1 = \mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\mathrm{Ascon}}(\mathcal{A}) \leq \frac{2(\sigma_e+\sigma_d)}{2^\kappa} + \frac{18(\sigma_e+\sigma_d)^2}{2^c} + \frac{2q_d}{2^\tau}$ follows from Theorem 1. Therefore, $\mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\mathrm{AsconAlt}[\tilde{\pi}]}(\mathcal{A}) = p_2 \leq |p_1 - p_2| + p_1 \leq \frac{2(\sigma_e+\sigma_d)}{2^\kappa} + \frac{18(\sigma_e+\sigma_d)^2}{2^c} + \frac{2q_d}{2^\tau} + \frac{2(\sigma_e+\sigma_d)^2}{2^\kappa}$ holds, which completes the proof.

Proof of Proposition 3. In the classical setting, the following theorem holds [59].

Theorem 2 (Theorem 3 of [59]). *Let $\mathcal{A}$ be a (classical) adversary against the nonce-misuse resilience confidentiality of Ascon in the classical ideal permutation model. Suppsoe that $\mathcal{A}$ makes at most q_e encryption queries, the number of effective blocks being at most σ_e in total, and at most q_C queries to P and P^{-1}. Then, it holds that*

$$\mathsf{Adv}^{\mathtt{mr\text{-}conf}}_{\mathrm{Ascon}}(\mathcal{A}) \leq \frac{\sigma_e + q_C}{2^\kappa} + \frac{18\sigma_e(\sigma_e + q_C)}{2^c}.$$

Proposition 3 follows from this theorem and Proposition 1 in exactly the same way as Proposition 2 follows from Theorem 1 and Proposition 1 as above.

6.2 Security of Ascon

Combining Proposition 1 with Proposition 2, we can show the following theorem.

Theorem 3 (Post-Quantum Authenticity of Ascon). *Let $\mathcal{A}$ be a quantum adversary against the post-quantum nonce-respecting authenticity (resp., nonce-misuse resistance authenticity, or RUP authenticity) of Ascon in the quantum ideal permutation model. Suppose $\mathcal{A}$ makes at most q_d classical decryption queries (or, q_d classical decryption and verification queries when considering RUP authnticity) and at most q_e classical encryption queries, the number of effective blocks being at most σ_d and σ_e in total, and at most q_Q quantum queries to P and P^{-1}. Then, the advantage $\mathsf{Adv}^{\mathtt{auth}}_{\mathrm{Ascon}}(\mathcal{A})$ (resp., $\mathsf{Adv}^{\mathtt{m\text{-}auth}}_{\mathrm{Ascon}}(\mathcal{A})$, or $\mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\mathrm{Ascon}}(\mathcal{A})$) is upper bounded by*

$$\begin{aligned}&\frac{2(\sigma_e + \sigma_d)}{2^\kappa} + \frac{18(\sigma_e + \sigma_d)^2}{2^c} + \frac{2q_d}{2^\tau} + \frac{2(\sigma_e + \sigma_d)^2}{2^\kappa} \\ &\quad + 4\sqrt{\frac{(\sigma_e + \sigma_d)^2 q_Q}{2^\kappa}} + \frac{2(\sigma_e + \sigma_d)^2}{2^\kappa} + 2\sqrt{\frac{2(\sigma_e + \sigma_d)(q_Q)^2}{2^\kappa}}.\end{aligned} \tag{13}$$

Proof. The claims for the nonce-respecting and nonce-misuse resistance authenticity advantages immediately follow from the claim for the RUP authenticity advantage. Thus, in what follows, we show that $\mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\mathrm{Ascon}}(\mathcal{A})$ is upper bounded by the value of Eq. (13).

Let $\mathcal{E}_k$, Dec_k, and Ver_k be the encryption, decryption, and verification oracles of Ascon, respectively. In addition, let $\mathcal{E}'[\tilde{E}_k[P]]$, $\mathsf{Dec}'[\tilde{E}_k[P]]$, and $\mathsf{Ver}'[\tilde{E}_k[P]]$ (resp., $\mathcal{E}'[\tilde{\pi}]$, $\mathsf{Dec}'[\tilde{\pi}]$, and $\mathsf{Ver}'[\tilde{\pi}]$) denote the encryption, decryption, and verification oracles of AsconAlt built on top of the tweakable block cipher $\tilde{E}_k[P]$ defined as in Fig. 4 (resp., instantiated with a random tweakable block cipher $\tilde{\pi}$). Recall that Fig. 4 is just an alternative view of Ascon, and so

$$(\mathcal{E}_k, \mathsf{Dec}_k, \mathsf{Ver}_k) = (\mathcal{E}'[\tilde{E}_k[P]], \mathsf{Dec}'[\tilde{E}_k[P]], \mathsf{Ver}'[\tilde{E}_k[P]]) \tag{14}$$

holds.

Let

$$p_1 := \Pr\left[\mathcal{A}^{\mathcal{E}'[\tilde{E}_k[P]],\mathsf{Dec}'[\tilde{E}_k[P]],\mathsf{Ver}'_k[\tilde{E}_k[P]],P,P^{-1}} \text{ forges}\right]$$

and

$$p_2 := \Pr\left[\mathcal{A}^{\mathcal{E}'[\tilde{\pi}],\mathsf{Dec}'[\tilde{\pi}],\mathsf{Ver}'[\tilde{\pi}],P,P^{-1}} \text{ forges}\right].$$

Then, due to Eq. (14), we have

$$\mathsf{Adv}^{\mathtt{rup\text{-}auth}}_{\mathrm{Ascon}}(\mathcal{A}) = p_1 \leq |p_1 - p_2| + p_2. \tag{15}$$

Let $\mathcal{B}$ be an adversary that tries to distinguish $(\tilde{E}_k[P], P, P^{-1})$ from $(\tilde{\pi}, P, P^{-1})$ (the oracles of $\tilde{E}_k[P]$ and $\tilde{\pi}$ are classical while P and P^{-1} are quantum) as follows: $\mathcal{B}$ runs $\mathcal{A}$. When $\mathcal{A}$ makes a classical query to the encryption, decryption, or verification oracle, $\mathcal{B}$ answers by computing $\mathcal{E}'[\mathcal{O}]$, $\mathsf{Dec}'[\mathcal{O}]$, or $\mathsf{Ver}'[\mathcal{O}]$, respectively, where $\mathcal{O}$ is the classical oracle ($\tilde{E}_k[P]$ or $\tilde{\pi}$) given to $\mathcal{B}$. When $\mathcal{A}$ makes a quantum query to P or P^{-1}, $\mathcal{B}$ just forwards them to the oracles given to $\mathcal{B}$ and the responses to $\mathcal{A}$. $\mathcal{B}$ outputs 1 if $\mathcal{A}$ forges, and outputs 0 otherwise.

Then, $\mathcal{B}$ perfectly simulates $(\mathcal{E}'[\tilde{E}_k[P]], \mathsf{Dec}'[\tilde{E}_k[P]], \mathsf{Ver}'[\tilde{E}[P]], P, P^{-1})$ and $(\mathcal{E}'[\tilde{\pi}], \mathsf{Dec}'[\tilde{\pi}], \mathsf{Ver}'[\tilde{\pi}]P, P^{-1})$ when the classical oracle $\mathcal{O}$ given to $\mathcal{B}$ are $\tilde{E}_k[P]$ and $\tilde{\pi}$, respectively. In addition, $\mathcal{B}$ makes at most $(\sigma_e + \sigma_d)$ classical queries to either $\tilde{E}_k[P]$ and $\tilde{\pi}$. Hence, by applying Proposition 1, we have

$$\begin{aligned} |p_1 - p_2| &= \mathsf{Adv}^{\mathtt{tprp}}_{\tilde{E}[P]}(\mathcal{B}) \\ &\leq 4\sqrt{\frac{(\sigma_e+\sigma_d)^2 q_Q}{2^\kappa}} + \frac{2(\sigma_e+\sigma_d)^2}{2^\kappa} + 2\sqrt{\frac{2(\sigma_e+\sigma_d)(q_Q)^2}{2^\kappa}}. \end{aligned} \tag{16}$$

Next, let $\mathcal{C}$ be an adversary that tries to break the RUP authenticity of $(\mathcal{E}'[\tilde{\pi}], \mathsf{Dec}'[\tilde{\pi}], \mathsf{Ver}'[\tilde{\pi}])$ as follows: First, $\mathcal{C}$ samples a random permutation P by itself, and then runs $\mathcal{A}$. When $\mathcal{A}$ makes quantum queries, $\mathcal{C}$ responds by using the sampled permutation P and its inverse. When $\mathcal{A}$ makes a classical query, $\mathcal{C}$ responds by forwarding them and responses to/from the classical oracles given to itself.

Then,

$$p_2 = \Pr\left[\mathcal{C}^{\mathcal{E}'[\tilde{\pi}],\mathsf{Dec}'[\tilde{\pi}],\mathsf{Ver}'[\tilde{\pi}]} \text{ forges}\right]$$

holds. In addition, the probability that $\mathcal{C}$ succeeds to forge is upper bounded as in Proposition 2. Therefore, we have

$$p_2 \leq \frac{2(\sigma_e+\sigma_d)}{2^\kappa} + \frac{18(\sigma_e+\sigma_d)^2}{2^c} + \frac{2q_d}{2^\tau} + \frac{2(\sigma_e+\sigma_d)^2}{2^\kappa}. \tag{17}$$

The claim of the theorem follows from Eq. (15)-(17). □

For confidentiality, the following theorem holds.

Theorem 4 (Post-Quantum Confidentiality of Ascon). *Let $\mathcal{A}$ be a quantum adversary against the post-quantum nonce-respecting confidentiality (resp., nonce-misuse resilience confidentiality) of Ascon in the quantum ideal permutation model. Suppose $\mathcal{A}$ makes at most q_e classical encryption queries, the number of effective blocks being at most σ_e in total, and at most q_Q quantum queries to P and P^{-1}. Then, the advantage $\mathsf{Adv}^{\mathtt{conf}}_{\mathrm{Ascon}}(\mathcal{A})$ (resp., $\mathsf{Adv}^{\mathtt{mr\text{-}conf}}_{\mathrm{Ascon}}(\mathcal{A})$) is upper bounded by*

$$\frac{\sigma_e}{2^\kappa} + \frac{18(\sigma_e)^2}{2^c} + \frac{2(\sigma_e)^2}{2^\kappa} + 4\sqrt{\frac{(\sigma_e)^2 q_Q}{2^\kappa}} + \frac{2(\sigma_e)^2}{2^\kappa} + 2\sqrt{\frac{2\sigma_e(q_Q)^2}{2^\kappa}}.$$

The proof for nonce-misuse resilience confidentiality is analogous to Theorem 3 (Proposition 3 is used instead of Proposition 2). The claim for nonce-respecting confidentiality immediately follows from that for nonce-misuse confidentiality.

Interpretation of the Security Bounds. Roughly, Theorem 3 and Theorem 4 tell that Ascon is secure as long as the number of permutation calls (either as quantum primitive queries or through classical construction queries) and the number of decryption queries are much smaller than $\min(2^{c/3}, 2^{\kappa/3})$ and 2^τ, respectively. Specifically, around 40-bit post-quantum security is guaranteed for the NIST version, and 50-bit for Ascon-pq80.

The guaranteed security levels are not very high. Yet, we can be confident that the Ascon AEAD mode does not have a structural weakness against quantum adversaries, even if nonce sets are not restricted, many forgery attempts are made, and in some nonce-misuse scenarios or even under RUP setting (for authenticity). This paper is the first to show the post-quantum security of Ascon in such settings.

Remark 3. NIST's initial draft of SP 800-32 allows to mask nonces with an additional 128-bit key. Of course, using the additional key does not decrease the post-quantum security bounds. At the same time, we have not achieved better bounds than Theorem 3 and Theorem 4 even with the additional key, either, because we did not find another view of Ascon in a way that the additional key contributes to increasing the security of TEM.

7 Post-Quantum Security Proof for Keyed Sponge Functions

This section proves the post-quantum security proof for IKS, FKS, and OKS in the quantum ideal permutation model, based on the idea from Section 3.3 and applying Proposition 1.

7.1 Security of Inner-Keyed Sponge

Theorem 5. *Let $\mathcal{A}$ be a quantum adversary against the post-quantum PRF security of IKS in the quantum ideal permutation model. Assume that the key of IKS is sampled uniformly at random from $\{0,1\}^\kappa$. Suppose $\mathcal{A}$ makes classical construction queries with the number of effective blocks at most σ in total, and at most q_Q quantum queries to P and P^{-1}. Then,*

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}}(\mathcal{A}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^\kappa}} + \frac{2\sigma^2}{2^\kappa} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^\kappa}} + \frac{\sigma^2}{2^c} \tag{18}$$

holds.

Proof. Let π and P' be another random permutation independent of P, and $E_k[P]$ be the Even-Mansour cipher defined as $E_k[P](x) = P(x \oplus (0^{n-\kappa}||k)) \oplus (0^{n-\kappa}||k)$. Then, since $\mathsf{IKS}_k[P] = \mathsf{Sponge}[E_k[P]]$ holds, we have that

$$\begin{aligned}
\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}}(\mathcal{A}) &= \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{IKS}_k[P],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF},P,P^{-1}}\right]\right| \\
&\leq \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{Sponge}[E_k[P]],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{Sponge}[\pi],P,P^{-1}}\right]\right| \quad (19) \\
&\quad + \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{Sponge}[\pi],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF},P,P^{-1}}\right]\right| \quad (20)
\end{aligned}$$

holds.

Since P is called at most σ times in the construction queries by $\mathcal{A}$ in the real world,

$$(19) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^\kappa}} + \frac{2\sigma^2}{2^\kappa} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^\kappa}}. \tag{21}$$

follows from Proposition 1 (by setting the tweak space size to be 1).

In addition, regarding $\mathcal{A}^{(\cdot),P,P^{-1}}$ as an adversary making only classical queries,

$$(20) \leq \frac{\sigma^2}{2^c} \tag{22}$$

follows from the classical indifferentiability bound [19]).

The claim of the theorem follows from Eq. (19)-(22). □

7.2 Security of Full-State Keyed Sponge

For FKS, the following theorem holds.

Theorem 6. *Let $\mathcal{A}$ be a quantm adversary against the post-quantum PRF security of FKS in the ideal permutation model. Assume that the key of FKS is sampled uniformly at random from $\{0,1\}^\kappa$. Suppose $\mathcal{A}$ makes at most q classical construction queries with the number of effective blocks at most ℓ each, and at most q_Q quantum primitive queries to P and P^{-1}. Then,*

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{FKS}}(\mathcal{A}) \leq \frac{2(q\ell)^2}{2^n} + \frac{2q^2\ell}{2^c} + 4\sqrt{\frac{(q\ell)^2 q_Q}{2^\kappa}} + \frac{2(q\ell)^2}{2^\kappa} + 2\sqrt{\frac{2q\ell(q_Q)^2}{2^\kappa}}.$$

holds.

The proof is analogous to Theorem 5, except that the bound of Eq. (22) is replaced with another bound shown in [64] for $\mathsf{FSSponge}$ built on a secret permutation (see Lemma 5 in Section D in the appendix for the concrete statement).

7.3 Security of Outer-Keyed Sponge

Recall that the key after the padding in OKS is $k^* = \mathsf{s}_1||k||\mathsf{s}_2$ for some constant strings s_1 and s_2. Based on the idea in Section 3.3, we can also show the following theorem. For a complete proof, see Section E in the appendix.

Theorem 7. *Assume that $\kappa > r$ and the key $k \in \{0,1\}^\kappa$ is sampled uniformly at random. Let $\kappa' := \min\left(\kappa - r + (|\mathsf{s}_1| \mod r), r - (|\mathsf{s}_1| \mod r)\right)$. Let $\mathcal{A}$ be a quantum adversary against the post-quantum PRF security of OKS in the quantum ideal permutation model. Suppose that $\mathcal{A}$ makes classical construction queries with the number of effective blocks at most σ in total, and at most q_Q quantum primitive queries to P and P^{-1}. Then,*

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{OKS}}(\mathcal{A}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^c}} + \frac{3\sigma^2}{2^c} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^c}} + 8\sqrt{\frac{2(q_Q + \sigma + (|k^*|/r))^2}{2^{\kappa'}}}$$

holds.

For the case of KMAC and KMACXOF [55], $(|\mathsf{s}_1| \mod r)$ is a constant of length $O(\log \kappa)$ (see Section A in the appendix for the specification details of KMAC(XOF)). Thus, the theorem assures that they are post-quantumly secure up to $O(\min(2^{c/3}, 2^{(\kappa - r + \log\kappa)/2}, 2^{(r - \log\kappa)/2}))$ queries. More specifically, the KMAC(XOF)128 (resp., KMAC(XOF)256) function has about 84-bit (resp., 169-bit) post-quantum security for a sufficiently long key (e.g., $\kappa = 2^{11}$) because $(n, r, c) = (1600, 1344, 256)$ (resp., $(n, r, c) = (1600, 1048, 512)$).

8 Concluding Remarks

This paper proved the post-quantum security of the Ascon AEAD mode and the keyed sponge functions, especially KMAC, through a modular proof approach.

For the Ascon AEAD mode, we observed that it can be regarded as an iterative application of TEM with a low-entropy key, and gave an upper bound on the various post-quantum security advantages of Ascon as the sum of (i) the *classical* security advantage of Ascon when TEM is replaced with an ideally random tweakable permutation, and (ii) the post-quantum TPRP advantage of TEM with a low-entropy key. To show that (ii) is indeed small, we introduced and proved a variant of the resampling lemmas of [1] and [2]. Our results guarantee the security of the Ascon AEAD mode up to about $\min(2^{c/3}, 2^{\kappa/3})$ queries, without restrictions on nonce sets or the number of forgery attempts, and even in some nonce-misuse and RUP settings.

The proofs of the keyed sponge functions were also obtained analogously, by viewing them as iterative applications of a single-key Even-Mansour cipher. The results guarantee the security up to about $2^{\kappa/3}$ queries for the inner- and full-keyed sponges, and about $\min(2^{\kappa/3}, 2^{(\kappa-r)/2}, 2^{r/2})$ queries for the outer-keyed sponge (and KMAC), when some small factors are ignored.

Proving Post-Quantum Security of Other Constructions. We have mainly focused on keyed sponge functions and Ascon AEAD mode. However, the proof idea of (1) regarding a construction as built upon a (tweakable) Even-Mansour cipher, (2) proving the *classical* security of the construction when the Even-Mansour cipher is replaced with a secret ideal primitive, and then (3) concluding the post-quantum security of the construction from (2) and Proposition 1, can also be applied to various other sponge-based constructions.

For example, in [64], the security of the Full-Keyed Duplex (FKD) construction in the classical ideal permutation model is exactly based on the above (1) and (2), with the classical security of Even-Mansour. Thus, we will easily achieve the post-quantum security bound of FKD in the quantum ideal permutation model (by replacing the terms derived from the classical security bound of the Even-Mansour with those from Proposition 1). In the same paper [64], the security of the Full-State SpongeWrap (FSW) AEAD mode is reduced to the security of FKD, and so we can also achieve the post-quantum security bound of FSW in the same way. The post-quantum (single-user) security of other constructions such as SpongeWrap [21], the duplex constructions introduced in [38] and [44], MonkeySpongeWrap [63], and Ascon-PRF [43], could also be proven analogously. Meanwhile, for some other construction such as NORX [10], we have not observed a way to reduce the security to the (Tweakable) Even-Mansour, and so a different approach will be needed.

Limitations and Future Works. Security bounds by our approach are capped by those for (T)EM, which are about $2^{\kappa/3}$. The aforementioned bounds of $\min(2^{c/3}, 2^{\kappa/3})$ for Ascon, $2^{\kappa/3}$ for IKS and FKS, and $\min(2^{\kappa/3}, 2^{(\kappa-r)/2}, 2^{r/2})$ for OKS (and KMAC) are all derived from this limitation. We are unaware of matching attacks and expect that the bounds are not tight. Another technique is necessary for a better bound (indeed, the work by Lang et al. [57] uses the O2H

lemma [5] to achieve a better bound, though assuming some additional constraints on construction queries). Our approach fully relies on the assumption that the construction oracle is keyed and queries are classical, and so it is not helpful for proving indifferentiability or Q2 security. In addition, the reduction of Ascon to TEM relies on the assumption that both the two permutations p^a and p^b of Ascon can be modeled as a single random permutation. Our idea does not work if they are modeled as independent permutations.

Showing a better security bound, especially a tight one, is definitely important future work. Proving security of Ascon in the multi-user setting, under leakage, or state recovery security, is also an intriguing research topic. Studying how to show the security of Ascon when p^a and p^b are modeled as independent permutations may also be of interest.

References

1. Alagic, G., Bai, C., Katz, J., Majenz, C.: Post-quantum security of the Even-Mansour cipher. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 458–487. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_17
2. Alagic, G., Bai, C., Katz, J., Majenz, C., Struck, P.: Post-quantum security of tweakable Even-Mansour, and applications. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part I. LNCS, vol. 14651, pp. 310–338. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58716-0_11
3. Alagic, G., Carolan, J., Majenz, C., Tokat, S.: The sponge is quantum indifferentiable. IACR Cryptology ePrint Archive 2025/731 (2025)
4. AlTawy, R., Gong, G., Mandal, K., Rohit, R.: WAGE: An authenticated encryption with a twist. IACR Trans. Symm. Cryptol. **2020**(S1), 132–159 (2020). https://doi.org/10.13154/tosc.v2020.iS1.132-159
5. Ambainis, A., Hamburg, M., Unruh, D.: Quantum security proofs using semi-classical oracles. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 269–295. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26951-7_10
6. Andreeva, E., Bogdanov, A., Luykx, A., Mennink, B., Mouha, N., Yasuda, K.: How to securely release unverified plaintext in authenticated encryption. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 105–125. Springer, Berlin, Heidelberg (Dec 2014). https://doi.org/10.1007/978-3-662-45611-8_6
7. Andreeva, E., Daemen, J., Mennink, B., Van Assche, G.: Security of keyed sponge constructions using a modular proof approach. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 364–384. Springer, Berlin, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-48116-5_18
8. Ashur, T., Dunkelman, O., Luykx, A.: Boosting authenticated encryption robustness with minimal modifications. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10403, pp. 3–33. Springer, Cham (Aug 2017). https://doi.org/10.1007/978-3-319-63697-9_1
9. Aumasson, J.P., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A lightweight hash. In: Mangard, S., Standaert, F.X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 1–15. Springer, Berlin, Heidelberg (Aug 2010). https://doi.org/10.1007/978-3-642-15031-9_1

10. Aumasson, J.P., Jovanovic, P., Neves, S.: NORX v3.0. Submisison to the CAESAR competition (2016)
11. Bao, Z., Chakraborti, A., Datta, N., Guo, J., Nandi, M., Peyrin, T., Yasuda, K.: PHOTON-Beetle Authenticated Encryption and Hash Family. Submission to NIST LWC Standardization Process (2019)
12. Beierle, C., Biryukov, A., Cardoso dos Santos, L., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q.: Lightweight AEAD and hashing using the Sparkle permutation family. IACR Trans. Symm. Cryptol. **2020**(S1), 208–261 (2020). https://doi.org/10.13154/tosc.v2020.iS1.208-261
13. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. Journal of Cryptology **21**(4), 469–491 (Oct 2008). https://doi.org/10.1007/s00145-008-9026-x
14. Bellizia, D., Berti, F., Bronchain, O., Cassiers, G., Duval, S., Guo, C., Leander, G., Leurent, G., Levi, I., Momin, C., Pereira, O., Peters, T., Standaert, F.X., Udvarhelyi, B., Wiemer, F.: Spook: Sponge-based leakage-resistant authenticated encryption with a masked tweakable block cipher. IACR Trans. Symm. Cryptol. **2020**(S1), 295–349 (2020). https://doi.org/10.13154/tosc.v2020.iS1.295-349
15. Bertoni, G., Daeman, J., Peters, M., Van Assche, G.: Permutation-based encryption, authentication and authenticated encryption. Directions in Authenticated Ciphers (2012)
16. Bertoni, G., Daeman, J., Peters, M., Van Assche, G., Van Keer, R.: CAESAR submission: KETJE v2. Submisison to the CAESAR competition (2016)
17. Bertoni, G., Daeman, J., Peters, M., Van Assche, G., Van Keer, R.: CAESAR submission: KEYAK v2. Submisison to the CAESAR competition (2016)
18. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. Ecrypt Hash Workshop (2007)
19. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Berlin, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3_11
20. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the security of the keyed sponge construction. Symmetric Key Encryption Workshop 2011 (2011)
21. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Duplexing the sponge: Single-pass authenticated encryption and other applications. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 320–337. Springer, Berlin, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-28496-0_19
22. Beyne, T., Cheng, Y., Dobraunig, C., Mennink, B.: Elephant v2 (2021)
23. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: A lightweight hash function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer, Berlin, Heidelberg (Sep / Oct 2011). https://doi.org/10.1007/978-3-642-23951-9_21
24. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer, Berlin, Heidelberg (Dec 2011). https://doi.org/10.1007/978-3-642-25385-0_3
25. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline Simon's algorithm. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 552–583. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34578-5_20

26. Bonnetain, X., Schrottenloher, A., Sibleyras, F.: Beyond quadratic speedups in quantum attacks on symmetric schemes. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 315–344. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07082-2_12
27. Carolan, J., Poremba, A.: Quantum one-wayness of the single-round sponge with invertible permutations. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VI. LNCS, vol. 14925, pp. 218–252. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68391-6_7
28. Carolan, J., Poremba, A., Zhandry, M.: (quantum) indifferentiability and precomputation. IACR Cryptology ePrint Archive 2024/1727 (2024)
29. Chakraborty, B., Dhar, C., Nandi, M.: Exact security analysis of ASCON. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part III. LNCS, vol. 14440, pp. 346–369. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8727-6_12
30. Chakraborty, B., Dhar, C., Nandi, M.: Tight multi-user security of ascon and its large key extension. In: Tianqing Zhu, Y.L. (ed.) ACISP 24, Part I. LNCS, vol. 14895, pp. 57–76. Springer, Singapore (Jul 2024). https://doi.org/10.1007/978-981-97-5025-2_4
31. Chang, D., Dworkin, M., Hong, S., Kelsey, J., Nandi, M.: A keyed sponge construction with pseudorandomness in the standard model. Third SHA-3 Candidate Conference (2012)
32. Cogliati, B., Lampe, R., Seurin, Y.: Tweaking Even-Mansour ciphers. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 189–208. Springer, Berlin, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-47989-6_9
33. Cojocaru, A., Hhan, M., Liu, Q., Yamakawa, T., Yun, A.: Quantum lifting for invertible permutations and ideal ciphers. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025. LNCS, vol. 16001, pp. 481–512. Springer, Cham (2025), https://doi.org/10.1007/978-3-032-01878-6
34. Czajkowski, J., Bruinderink, L.G., Hülsing, A., Schaffner, C., Unruh, D.: Post-quantum security of the sponge construction. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018. pp. 185–204. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_9
35. Czajkowski, J., Hülsing, A., Schaffner, C.: Quantum indistinguishability of random sponges. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 296–325. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26951-7_11
36. Daemen, J., Hoffert, S., Peeters, M., Van Assche, G., Van Keer, R.: Xoodyak, a lightweight cryptographic scheme. IACR Trans. Symm. Cryptol. **2020**(S1), 60–87 (2020). https://doi.org/10.13154/tosc.v2020.iS1.60-87
37. Daemen, J., Massolino, P.M.C., Mehrdad, A., Rotella, Y.: The subterranean 2.0 cipher suite. IACR Trans. Symm. Cryptol. **2020**(S1), 262–294 (2020). https://doi.org/10.13154/tosc.v2020.iS1.262-294
38. 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, Cham (Dec 2017). https://doi.org/10.1007/978-3-319-70697-9_21
39. Degabriele, J.P., Janson, C., Struck, P.: Sponges resist leakage: The case of authenticated encryption. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019,

Part II. LNCS, vol. 11922, pp. 209–240. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34621-8_8
40. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Mennink, B., Primas, R., Unterluggauer, T.: ISAP v2.0. IACR Trans. Symm. Cryptol. **2020**(S1), 390–416 (2020). https://doi.org/10.13154/tosc.v2020.iS1.390-416
41. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2 Submission to the caesar competition (2016)
42. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1.2: Lightweight authenticated encryption and hashing. Journal of Cryptology **34**(3), 33 (Jul 2021). https://doi.org/10.1007/s00145-021-09398-9
43. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon MAC, PRF, and short-input PRF - lightweight, fast, and efficient pseudorandom functions. In: Oswald, E. (ed.) CT-RSA 2024. LNCS, vol. 14643, pp. 381–403. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58868-6_15
44. Dobraunig, C., Mennink, B.: Leakage resilience of the duplex construction. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 225–255. Springer, Cham (Dec 2019). https://doi.org/10.1007/978-3-030-34618-8_8
45. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: 49th FOCS. pp. 293–302. IEEE Computer Society Press (Oct 2008). https://doi.org/10.1109/FOCS.2008.56
46. Even, S., Mansour, Y.: A construction of a cipher from a single pseudorandom permutation. Journal of Cryptology **10**(3), 151–162 (Jun 1997). https://doi.org/10.1007/s001459900025
47. Gazi, P., Pietrzak, K., Tessaro, S.: The exact PRF security of truncation: Tight bounds for keyed sponges and truncated CBC. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 368–387. Springer, Berlin, Heidelberg (Aug 2015). https://doi.org/10.1007/978-3-662-47989-6_18
48. Grilo, A.B., Hövelmanns, K., Hülsing, A., Majenz, C.: Tight adaptive reprogramming in the QROM. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 637–667. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92062-3_22
49. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
50. Guo, C., Pereira, O., Peters, T., Standaert, F.X.: Towards low-energy leakage-resistant authenticated encryption from the duplex sponge construction. Cryptology ePrint Archive, Report 2019/193 (2019), https://eprint.iacr.org/2019/193
51. Guo, C., Pereira, O., Peters, T., Standaert, F.X.: Towards low-energy leakage-resistant AE from the duplex sponge. IACR Trans. Symm. Cryptol. **2020**(1), 6–42 (2020). https://doi.org/10.13154/tosc.v2020.i1.6-42
52. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON family of lightweight hash functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer, Berlin, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9_13
53. Janson, C., Struck, P.: Sponge-based authenticated encryption: Security against quantum attackers. In: Cheon, J.H., Johansson, T. (eds.) Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022. pp. 230–259. Springer, Cham (Sep 2022). https://doi.org/10.1007/978-3-031-17234-2_12

54. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symm. Cryptol. **2016**(1), 71–94 (2016). https://doi.org/10.13154/tosc.v2016.i1.71-94, https://tosc.iacr.org/index.php/ToSC/article/view/536
55. Kelsey, J., Chang, S., Perlner, R.: SHA-3 derived functions: cSHAKE, KMAC, TupleHash and ParallelHash. NIST Special Publication 800-185 (2016)
56. Kuwakado, H., Morii, M.: Security on the quantum-type even-mansour cipher. In: Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2012, Honolulu, HI, USA, October 28-31, 2012. pp. 312–316. IEEE (2012), https://ieeexplore.ieee.org/document/6400943/
57. Lang, N., Lucks, S., Mennink, B., Talnikar, S.: Security of the ascon authenticated encryption mode in the presence of quantum adversaries. IACR Cryptology ePrint Archive 2025/411 (2025)
58. Lefevre, C., Mennink, B.: Generic security of the ascon mode: On the power of key blinding. In: Eichlseder, M., Gambs, S. (eds.) Selected Areas in Cryptography - SAC 2024 - 31st International Conference, Montreal, QC, Canada, August 28-30, 2024, Revised Selected Papers, Part II. Lecture Notes in Computer Science, vol. 15517, pp. 3–32. Springer (2024). https://doi.org/10.1007/978-3-031-82841-6_1, https://doi.org/10.1007/978-3-031-82841-6_1
59. Lefevre, C., Mennink, B.: Sok: Security of the ascon modes. IACR Trans. Symmetric Cryptol. **2025**(1), 138–210 (2025). https://doi.org/10.46586/TOSC.V2025.I1.138-210, https://doi.org/10.46586/tosc.v2025.i1.138-210
60. Majenz, C., Malavolta, G., Walter, M.: Permutation superposition oracles for quantum query lower bounds. In: Koucký, M., Bansal, N. (eds.) STOC 2025. pp. 1508–1519. ACM (2025). https://doi.org/10.1145/3717823.3718266, https://doi.org/10.1145/3717823.3718266
61. Maurer, U.M., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Berlin, Heidelberg (Feb 2004). https://doi.org/10.1007/978-3-540-24638-1_2
62. Mennink, B.: Key prediction security of keyed sponges. IACR Trans. Symm. Cryptol. **2018**(4), 128–149 (2018). https://doi.org/10.13154/tosc.v2018.i4.128-149
63. Mennink, B.: Understanding the duplex and its security. IACR Trans. Symm. Cryptol. **2023**(2), 1–46 (2023). https://doi.org/10.46586/tosc.v2023.i2.1-46
64. Mennink, B., Reyhanitabar, R., Vizár, D.: Security of full-state keyed sponge and duplex: Applications to authenticated encryption. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 465–489. Springer, Berlin, Heidelberg (Nov / Dec 2015). https://doi.org/10.1007/978-3-662-48800-3_19
65. Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., Verbauwhede, I.: Chaskey: An efficient MAC algorithm for 32-bit microcontrollers. In: Joux, A., Youssef, A.M. (eds.) SAC 2014. LNCS, vol. 8781, pp. 306–323. Springer, Cham (Aug 2014). https://doi.org/10.1007/978-3-319-13051-4_19
66. Naito, Y., Yasuda, K.: New bounds for keyed sponges with extendable output: Independence between capacity and message length. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 3–22. Springer, Berlin, Heidelberg (Mar 2016). https://doi.org/10.1007/978-3-662-52993-5_1
67. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010)
68. NIST: Post-quantum cryptography, https://csrc.nist.gov/projects/post-quantum-cryptography. Accessed on May 16, 2025.

69. NIST: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. FIPS PUB 202 (2015)
70. Pereira, O., Standaert, F.X., Vivek, S.: Leakage-resilient authentication and encryption from symmetric cryptographic primitives. In: Ray, I., Li, N., Kruegel, C. (eds.) ACM CCS 2015. pp. 96–108. ACM Press (Oct 2015). https://doi.org/10.1145/2810103.2813626
71. Rogaway, P.: Nonce-based symmetric encryption. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 348–359. Springer, Berlin, Heidelberg (Feb 2004). https://doi.org/10.1007/978-3-540-25937-4_22
72. Rogaway, P., Shrimpton, T.: A provable-security treatment of the key-wrap problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Berlin, Heidelberg (May / Jun 2006). https://doi.org/10.1007/11761679_23
73. Sasaki, Y., Todo, Y., Aoki, K., Naito, Y., Sugawara, T., Murakami, Y., Matsui, M., Hirose, S.: Minalpher v1.1. Submission to CAESAR commpetition (2015)
74. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th FOCS. pp. 124–134. IEEE Computer Society Press (Nov 1994). https://doi.org/10.1109/SFCS.1994.365700
75. Turan, M.S., McKay, K.A., Chang, D., Kang, J., Kelsey, J.: Ascon-based lightweight cryptography standards for constrained devices: Authenticated encryption, hash, and extendable output functions. NIST SP 800-32 ipd (2024)
76. Yamakawa, T., Zhandry, M.: Verifiable quantum advantage without structure. In: 63rd FOCS. pp. 69–74. IEEE Computer Society Press (Oct / Nov 2022). https://doi.org/10.1109/FOCS54457.2022.00014
77. Zhandry, M.: Redeeming reset indifferentiability and applications to post-quantum security. In: Tibouchi, M., Wang, H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 518–548. Springer, Cham (Dec 2021). https://doi.org/10.1007/978-3-030-92062-3_18

A Specification of KMAC128, KMAC256, KMACXOF128, and KMACXOF256

This section explains the specifications of KMAC128, KMAC256, KMACXOF128, and KMACXOF256, following SP 800-185 [55].

A.1 Component Functions

First, we review the component functions required to describe the specifications. The first two functions $\mathsf{pad10^*1}(r, m)$ and Keccak$[c]$ are from FIPS 202 [69], while the remaining ones from SP 800-185 [55].

$\mathsf{pad10^*1}(r, m)$. Given a positive integer r and non-negative integer m, the function $\mathsf{pad10^*1}$ returns a bit string $P := 10^z1$, where z is the minimum non-negative integer such that $(m + z + 2)$ is a positive multiple of r.

Keccak[c](X, L). Keccak[c](X, L) is the sponge hash function built on top of a 1600-bit permutation. Here, X denotes the input string, and L is the output length. The padding function is given by $X \mapsto X||\mathsf{pad}10^*1(1600-c, |X|)$. The capacity is c and the rate is $(1600-c)$.

enc$_8$(x). Given an integer $x \in \{0, \ldots, 255\}$, the function enc$_8$ returns an 8-bit string representing x.

right_encode(x) and left_encode(x). Let x be an integer such that $0 \leq x < 2^{2040}$ and n be the smallest positive integer such that $2^{8n} > x$. right_encode(x) and left_encode(x) return an $8(n+1)$-bit string representing x. The difference between the two functions is in endian, but the details are irrelevant to our results, so we omit them.

encode_string(S). Given a bit string S such that $|S| < 2^{2040}$, the function encode_string returns left_encode($|S|$)$||S$.

bytepad(X, w). Given a bit string $X \in \{0,1\}^*$ and a positive integer w, the function bytepad returns $X' :=$ left_encode(w)$||X||0^m$, where m is the minimum non-negative integer such that ($|$left_encode(w)$| + |X| + m$) is a multiple of $8w$.

cSHAKE128(X, L, N, S) and cSHAKE256(X, L, N, S). cSHAKE128 and cSHAKE256 are hash functions derived from Keccak[c]. N and S are parameters for customization such that $|N|, |S| < 2^{2040}$. The case of $|N| = 0$ is irrelevant for KMAC(XOF), so we assume $|N| > 0$. Given an input message X,

- cSHAKE128 returns

  Keccak[256](bytepad(encode_string(N)$||$encode_string(S), 168)$||X||$00, L).

- cSHAKE256 returns

  Keccac[512](bytepad(encode_string(N)$||$encode_string(S), 136)$||X||$00, L).

A.2 Specifications of KMAC(XOF)

KMAC(XOF) is parameterized by an optional customization parameter $S \in \{0,1\}^*$. For KMAC128 and KMAC256, the output length is denoted by L. The padding function depends on S, L (for KMAC128 and KMAC256), and the key length κ. It is required that $\kappa, |S| < 2^{2040}$ ($L < 2^{2040}$ is additionally required for KMAC128 and KMAC256). Let N be a 32-bit string representing the character string "KMAC".

Given an input message X and a key k,

- KMAC128 returns

  $\mathrm{cSHAKE128}(\mathrm{bytepad}(\mathrm{encode\_string}(k), 168)||X||\mathrm{right\_encode}(L), L, N, S)$.

- KMAC256 returns

  $\mathrm{cSHAKE256}(\mathrm{bytepad}(\mathrm{encode\_string}(k), 136)||X||\mathrm{right\_encode}(L), L, N, S)$.

- KMACXOF128 returns

  $\mathrm{cSHAKE128}(\mathrm{bytepad}(\mathrm{encode\_string}(k), 168)||X||\mathrm{right\_encode}(0), L, N, S)$.

- KMACXOF256 returns

  $\mathrm{cSHAKE256}(\mathrm{bytepad}(\mathrm{encode\_string}(k), 136)||X||\mathrm{right\_encode}(0), L, N, S)$.

B Proof of Lemma 3

Proof (of Lemma 3). We follow the notations of [1, Section 4.2] so it will be clear which part is different from the proof of the first resampling lemma [1, Section 4.2]. (In the notational setup below, the main difference from [1] is that projectors P_D used to choose the distribution D are introduced.)

Notational Setup. Let $\mathcal{D}_0$ and $\mathcal{D}_1$ be the algorithms that correspond to the first and second steps of $\mathcal{D}$, respectively. Without loss of generality, we assume that $\mathcal{D}_0$ chooses a distribution D from a finite set S (because the number of qubits available for $\mathcal{D}_0$ is finite).

We first introduce four quantum registers, E, F, X, and Y. Here, E is basically the register of $\mathcal{D}$ for offline computations. F is the $(n2^n)$-qubit register for the oracle to store the truth table of functions (or permutations), which consists of a collection of smaller registers $\{F_x\}_{x\in\{0,1\}^n}$. For an arbitrary function $f : \{0,1\}^n \to \{0,1\}^n$, its values are stored as $|f\rangle_F = \bigotimes_{x\in\{0,1\}^n} |f(x)\rangle_{F_x}$. X and Y are the n-qubit registers used to send a query and receive the response. If an algorithm is given oracle access to a function f, then quantum queries are processed by the unitary operator O defined by

$$O_{XYF}\,|x\rangle_X\,|y\rangle_Y\,|f\rangle_F = |x\rangle_X\,|y \oplus f(x)\rangle_Y\,|f\rangle_F\,.$$

In addition, the unitary operator O^{inv} is defined by

$$O^{\mathrm{inv}}_{XYF}\,|x\rangle_X\,|y\rangle_Y\,|f\rangle_F = |x \oplus (\oplus_{x':f(x')=y} x')\rangle_X\,|y\rangle_Y\,|f\rangle_F\,.$$

If f is a permutation, then the action of O^{inv}_{XYF} corresponds to querying y to f^{-1} and receive the response $f^{-1}(y)$ in register X.

At the beginning of the experiment (before $\mathcal{D}_0$ starts running), register F is initialized to be

$$|\phi_0\rangle_F := \bigotimes_{\pi\in\mathsf{Perm}(n)} \frac{1}{\sqrt{2^n!}}\,|\pi\rangle_F\,,$$

and queries by $\mathcal{D}$ to P (resp., P^{-1}) are processed by applying the unitary operator O (resp., O^{inv}).

For arbitrary $s_0, s_1 \in \{0,1\}^n$, let $\mathsf{Swap}_{F_{s_0},F_{s_1}}$ be the unitary operator that swaps the two registers F_{s_0} and F_{s_1}. That is, $\mathsf{Swap}_{F_{s_0},F_{s_1}} |w\rangle_{F_{s_0}} |z\rangle_{F_{s_1}} = |z\rangle_{F_{s_0}} |w\rangle_{F_{s_1}}$ holds. In addition, the four projectors $P_{s_0s_1}$, $P^{\mathrm{inv}}_{s_0s_1}$, $\bar{P}_{s_0s_1}$, and $\bar{P}^{\mathrm{inv}}_{s_0s_1}$ are defined by

$$
\begin{aligned}
(P_{s_0s_1})_X &:= \begin{cases} \mathbf{1} & \text{if } s_0 = s_1, \\ \mathbf{1} - |s_0\rangle\langle s_0|_X - |s_1\rangle\langle s_1|_X & \text{if } s_0 \neq s_1, \end{cases} \\
\left(P^{\mathrm{inv}}_{s_0s_1}\right)_{FY} &:= \begin{cases} \mathbf{1} & \text{if } s_0 = s_1, \\ \sum_{y\in\{0,1\}^n} |y\rangle\langle y|_Y \otimes (\mathbf{1} - |y\rangle\langle y|)^{\otimes 2}_{F_{s_0}F_{s_1}} & \text{if } s_0 \neq s_1, \end{cases} \\
\bar{P}_{s_0s_1} &:= \mathbf{1} - P_{s_0s_1}, \\
\bar{P}^{\mathrm{inv}}_{s_0s_1} &:= \mathbf{1} - P^{\mathrm{inv}}_{s_0s_1}.
\end{aligned}
$$

Without loss of generality, we assume that $\mathcal{D}_0$ makes exactly q quantum queries, $\mathcal{D}_0$ does not perform intermediate measurements, and the distribution D is chosen by $\mathcal{D}_0$ as follows:

1. At the beginning, $\mathcal{D}_0$ sets a specific part of the offline register E, which we refer to as register Z, to be $|0\rangle_Z$. Register Z is untouched until $\mathcal{D}_0$ chooses a distribution.
2. After making q quantum queries, $\mathcal{D}_0$ writes (in quantum superposition) the information about which distribution D to choose into register Z.
3. Then, measure register Z to obtain a distribution D.

We denote the projector $|D\rangle\langle D|_Z$ as P_D.

Let $|\psi\rangle$ be the quantum state right before the measurement to choose the distribution D. Then,

$$
|\psi\rangle = U_q O_q \cdots U_1 O_1 \left(|\psi'_0\rangle_{XYE} |\phi_0\rangle_F\right) \tag{23}
$$

holds, where each O_i (which is either O_{XYF} or O^{inv}_{XYF}) is the unitary operator corresponding to the i-th quantum query, and U_i is a unitary operator acting on registers XYE that corresponds to some offline computation by $\mathcal{D}_0$, and $|\psi'_0\rangle$ is the initial state of $\mathcal{D}_0$. In addition, for arbitrary $s_0, s_1 \in \{0,1\}^n$, let

$$
\begin{aligned}
|\psi_{\mathrm{good}}(s_0,s_1)\rangle &:= z \cdot U_q O_q P^q_{s_0s_1} \cdots U_1 O_1 P^1_{s_0s_1} \left(|\psi'_0\rangle_{XYE} |\phi_0\rangle_F\right), \\
|\psi_{\mathrm{bad}}(s_0,s_1)\rangle &:= |\psi\rangle - |\psi_{\mathrm{bad}}(s_0,s_1)\rangle,
\end{aligned}
$$

where

$$
P^i_{s_0s_1} := \begin{cases} (P_{s_0s_1})_X & \text{if } O_i = O_{XYF}, \\ \left(P^{\mathrm{inv}}_{s_0s_1}\right)_{YF} & \text{if } O_i = O^{\mathrm{inv}}_{XYF}, \end{cases}
$$

and z is a complex number such that $|z| = 1$ and $\langle\psi|\psi_{\mathrm{good}}(s_0,s_1)\rangle \in \mathbb{R}_{\geq 0}$.

Just as in [1, Section 4.2], the vector $|\psi_{\mathrm{good}}(s_0, s_1)\rangle$ is invariant under the action of $\mathsf{Swap}_{F_{s_0},F_{s_1}}$, that is,

$$\mathsf{Swap}_{F_{s_0},F_{s_1}} |\psi_{\mathrm{good}}(s_0, s_1)\rangle = |\psi_{\mathrm{good}}(s_0, s_1)\rangle \tag{24}$$

holds.

Moreover, let

$$\epsilon(s_0, s_1) := \left\| \bar{P}^i_{s_0 s_1} U_{i-1} O_{i-1} \cdots U_1 O_1 |\psi'_0\rangle_{XYE} |\phi_0\rangle_F \right\|_2^2,$$

where $\bar{P}^i_{s_0 s_1} := \mathbf{1} - P^i_{s_0 s_1}$.

Proving Eq. (10). Recall that, at the end of $\mathcal{D}_0$, the register Z (which is a part of register E for offline computation) is measured to choose a distribution D. The (pure) state just before the measurement is $|\psi\rangle$ of Eq. (23). By the measurement, $|\psi\rangle$ changes to the mixed state

$$\sum_{D \in S} P_D |\psi\rangle \langle\psi| P_D.$$

After that, s_0 and s_1 are sampled, and the state changes to

$$\rho_0 := \sum_{\substack{D \in S, \\ s_0, s_1 \in \{0,1\}^n}} \frac{D(s_0)}{2^n} P_D |\psi\rangle \langle\psi|_{XYEF} P_D \otimes |s_0, s_1\rangle \langle s_0, s_1|_C,$$

where C is an additional register.

If $b = 1$ and the values of the permutation are swapped on s_0 and s_1, the state further changes to

$$\rho_1 := \sum_{\substack{D \in S, \\ s_0, s_1 \in \{0,1\}^n}} \frac{D(s_0)}{2^n} \mathsf{Swap}_{F_{s_0},F_{s_1}} P_D |\psi\rangle \langle\psi| P_D \mathsf{Swap}_{F_{s_0},F_{s_1}} \otimes |s_0, s_1\rangle \langle s_0, s_1|.$$

From the standard argument for distinguishing advantages,

$$|\Pr[b' = 1 \mid b = 1] - \Pr[b' = 1 \mid b = 0]| \leq \mathsf{TD}(\rho_0, \rho_1) \tag{25}$$

follows, where TD is the trace distance.

We observe that the state ρ_0 can also be produced from $|\psi\rangle$ as follows.

1. Without measuring register Z (on which the information of D is written), create the superposition[9] of s_0 and s_1 of the form $\sum_{s_0,s_1} \sqrt{\frac{D(s_0)}{2^n}} |s_0, s_1\rangle_C$ for each D, obtaining the (pure) state

$$|\xi_0\rangle := \sum_{s_0,s_1,D} \sqrt{\frac{D(s_0)}{2^n}} P_D |\psi\rangle_{XYEF} |s_0, s_1\rangle_C.$$

[9] In other words, apply a unitary operator V satisfying $V |D\rangle_Z |0^n, 0^n\rangle_C = \sum_{s_0,s_1} \sqrt{D(s_0)/2^n} |D\rangle_Z |s_0, s_1\rangle_C$.

2. Measure the registers Z and C to obtain the classical information of D, s_0, s_1.

Similarly, the state ρ_1 can be produced by measuring the registers Z and C of the pure state

$$|\xi_1\rangle := \sum_{s_0,s_1,D} \sqrt{\frac{D(s_0)}{2^n}} \mathsf{Swap}_{F_{s_0},F_{s_1}} P_D \, |\psi\rangle_{XYEF} \, |s_0,s_1\rangle_C \, .$$

Since measurements decrease the trace distance between two states, and the trace distance between the two pure states $|\xi_0\rangle\langle\xi_0|$ and $|\xi_1\rangle\langle\xi_1|$ is upper bounded by the norm of the difference $(|\xi_0\rangle - |\xi_1\rangle)$, it holds that

$$\mathsf{TD}\left(\rho_0,\rho_1\right) \leq \mathsf{TD}\big(\, |\xi_0\rangle\langle\xi_0| \, , |\xi_1\rangle\langle\xi_1| \,\big) \leq \big\| \, |\xi_0\rangle - |\xi_1\rangle \, \big\|_2. \tag{26}$$

In addition, we have

$$\begin{aligned}
\big\| \, |\xi_0\rangle - |\xi_1\rangle \, \big\|_2^2 &\overset{\text{(i)}}{=} \sum_{D,s_0,s_1} \frac{D(s_0)}{2^n} \Big\| P_D(\mathbf{1} - \mathsf{Swap}_{F_{s_0},F_{s_1}}) \, |\psi\rangle \Big\|_2^2 \\
&\overset{\text{(ii)}}{=} \sum_{D,s_0,s_1} \frac{D(s_0)}{2^n} \Big\| P_D \left(|\psi_{\mathrm{bad}}(s_0,s_1)\rangle - \mathsf{Swap}_{F_{s_0},F_{s_1}} \, |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right) \Big\|_2^2 \\
&\overset{\text{(iii)}}{\leq} 4 \sum_{D,s_0,s_1} \frac{D(s_0)}{2^n} \left\| P_D \, |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right\|_2^2 \\
&\overset{\text{(iv)}}{\leq} 4 \sum_{D,s_0,s_1} \frac{1}{2^{n+\kappa}} \left\| P_D \, |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right\|_2^2 = 4 \sum_{s_0,s_1} \frac{1}{2^{n+\kappa}} \left\| |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right\|_2^2 \\
&= 4 \cdot 2^{n-\kappa} \cdot \mathbb{E}_{s_0,s_1 \xleftarrow{\$} \{0,1\}^n} \left[\left\| |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right\|_2^2 \right],
\end{aligned} \tag{27}$$

where we used the fact[10] that $\Big[P_D, \mathsf{Swap}_{F_{s_0},F_{s_1}}\Big] = 0$ for (i) and (iii), Eq. (24) for (ii), and the assumption that the min-entropy of D is at least κ for (iv).

In addition,

$$\left\| |\psi_{\mathrm{bad}}(s_0,s_1)\rangle \right\|_2^2 \leq 2 \sum_{i=1}^{q} \epsilon_i(s_0,s_1), \tag{28}$$

$$\mathbb{E}_{s_0,s_1 \xleftarrow{\$} \{0,1\}^n} \left[\epsilon_i(s_0,s_1) \right] \leq \frac{2}{2^n} \tag{29}$$

are proven in exactly the same way as [1, Section 4.2][11].

Putting equations (25)-(29) together, we obtain the desired claim. □

[10] P_D acts only on (a part of) register E while $\mathsf{Swap}_{F_{s_0},F_{s_1}}$ acts only on register F.

[11] In [1, Section 4.2], the upper bound of the expectation value of $\epsilon_i(s_0,s_1)$ is implicitly shown as Eq. (13) and Eq. (15).

C Proof of Proposition 1

The proof of the proposition is analogous to the proof for TEM-KX in [2], except that the new resampling lemma (Lemma 3) is used instead of Lemma 2. Below, we mainly explain the difference from the proof for TEM-KX in [2], following the notations of [2] as much as possible.

In what follows, P is the public random n-bit permutation underlying TEM ($\mathcal{A}$ has quantum oracle access to both P and P^{-1}), and k denotes an n-bit key sampled from D_{key}. The construction oracle in the ideal world (an ideally random tweakable permutation over $\{0,1\}^n$ with tweak space $\mathcal{T}$) is denoted as $\tilde{E}$. Without loss of generality, $\mathcal{A}$ is assumed not to make redundant classical queries. That is, if $\mathcal{A}$ queries a value (t,x) (resp., (t,y)) to the encryption (resp., decryption) oracle and receives y (resp., x) as the response, then $\mathcal{A}$ will never query (t,x) to the encryption oracle nor (t,y) to the decryption oracle. For $s_0, s_1 \in \{0,1\}^n$, let swap_{s_0,s_1} denote the permutation over $\{0,1\}^n$ defined by $\mathsf{swap}_{s_0,s_1}(s_b) := s_{b\oplus 1}$ and $\mathsf{swap}_{s_0,s_1}(x) := x$ for $x \neq s_0, s_1$.

For each i, the i-th classical query to the construction oracle (TEM or $\tilde{E}$) and the response are denoted as (t_i, x_i, y_i, b_i). Here, t_i is the tweak. x_i and y_i correspond to the plaintext and ciphertext. $b_i = 0$ (resp., $b_i = 1$) means that the query is made to the forward (resp., inverse) oracle, i.e., $\mathsf{TEM}_k[P]$ or $\tilde{E}$ (resp., $\mathsf{TEM}_k^{-1}[P]$ or $\tilde{E}^{-1}$). The list of the first j classical queries and responses are denoted as $T_j = ((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j))$.

Given a permutation P, a key k, and the list T_j, define permutations $\overrightarrow{S}_{T_j,P,k}$, $\overleftarrow{S}_{T_j,P,k}$, and $P^{T_j,k}$ as

$$\begin{aligned}
\overrightarrow{S}_{T_j,P,k} &:= \mathsf{swap}^{1-b_1}_{P(x_1\oplus f_1(t_1,k)),y_1\oplus f_2(t_1,k)} \circ \cdots \circ \mathsf{swap}^{1-b_j}_{P(x_j\oplus f_1(t_j,k)),y_j\oplus f_2(t_j,k)} \\
\overleftarrow{S}_{T_j,P,k} &:= \mathsf{swap}^{b_j}_{P(x_j\oplus f_1(t_j,k)),y_j\oplus f_2(t_j,k)} \circ \cdots \circ \mathsf{swap}^{b_1}_{P(x_1\oplus f_1(t_1,k)),y_1\oplus f_2(t_1,k)} \\
P^{T_j,k} &:= \overleftarrow{S}_{T_j,P,k} \circ \overrightarrow{S}_{T_j,P,k} \circ P.
\end{aligned}$$

Very roughly, $P^{T_j,k}$ is defined so it will be consistent with the classical query/response history $T_j = ((t_1, x_1, y_1, b_1), \ldots, (t_j, x_j, y_j, b_j))$ (i.e., $\mathsf{TEM}_k[P^{T_j,k}](t_i, x_i) = y_i$ will hold) as much as possible, while the difference from P will be minimal.

For $j = 0, \ldots, q_C$, the experiments $\mathbf{H}_j$, $\mathbf{H}_j^*$, $\mathbf{H}_j^{**}$, and $\mathbf{H}_j'$ are defined as follows.

Experiment $\mathbf{H}_j$.

1. Run the adversary $\mathcal{A}$ relative to the classical oracle of $\tilde{E}$ and the quantum oracle of P (and their inverse oracles). Stop when $\mathcal{A}$ submits the $(j+1)$-th classical query.
2. Change the oracles from $\tilde{E}$ and P to $\mathsf{TEM}_k[P^{T_j,k}]$ and $P^{T_j,k}$ (and their inverses), and resume running $\mathcal{A}$.

Experiment $\mathbf{H}_j^*$.

1. Run the adversary $\mathcal{A}$ relative to the classical oracle of $\tilde{E}$ and the quantum oracle of P (and their inverse oracles). Stop when $\mathcal{A}$ submits the $(j+1)$-th classical query. Suppose the $(j+1)$-th classical query is to the forward oracle, which we denote by (t_{j+1}, x_{j+1}) (if the query is to the inverse oracle, the oracles are analogously defined.).
2. Let $s_0 := f_1(t_{j+1}, k) \oplus x_{j+1}$, choose $s_1 \in \{0,1\}^n$ uniformly at random, and let $P^{(1)} := P \circ \mathsf{swap}_{s_0,s_1}$. Change the oracles from $\tilde{E}$ and P to $\mathsf{TEM}_k[(P^{(1)})^{T_j,k}]$ and $(P^{(1)})^{T_j,k}$ (and their inverses), and resume running $\mathcal{A}$.

Experiment $\mathbf{H}_j^{**}$. This is almost the same as $\mathbf{H}_j^*$, but the difference is that $\tilde{E}$ is used to answer the $(j+1)$-th classical query, and s_1 is defined as $s_1 := (P^{T_j,k})^{-1}(y_{j+1} \oplus f_2(t_{j+1}, k))$.

Experiment $\mathbf{H}_j'$.

1. Run the adversary $\mathcal{A}$ relative to the classical oracle of $\tilde{E}$ and the quantum oracle of P (and their inverse oracles). Stop when $\mathcal{A}$ receives the response to the $(j+1)$-th classical query.
2. Change the oracles from $\tilde{E}$ and P to $\mathsf{TEM}_k[P^{T_{j+1},k}]$ and $P^{T_{j+1},k}$ (and their inverses), and resume running $\mathcal{A}$.

The oracles of the four games look roughly as follows. For ease of notations, $P^{T_j,k}$ is abbreviated as P^j, $(P^{(1)})^{T_j,k}$ as $(P^{(1)})^j$, and so on. Each number i in the top row shows that the column corresponds to the i-th classical query.

	1	$j+1$	$j+2$
$\vdots$			
$\mathbf{H}_j$	$P, \tilde{E}, P, \cdots P,$	$\mathsf{TEM}[P^j],$	$P^j,$ $\mathsf{TEM}[P^j],$ $P^j,$ $\cdots$
$\mathbf{H}_j^*$	$P, \tilde{E}, P, \cdots P,$	$\mathsf{TEM}[(P^{(1)})^j],$	$(P^{(1)})^j,$ $\mathsf{TEM}[(P^{(1)})^j],$ $(P^{(1)})^j,$ $\cdots$
$\mathbf{H}_j^{**}$	$P, \tilde{E}, P, \cdots P,$	$\tilde{E},$	$(P^{(1)})^j,$ $\mathsf{TEM}[(P^{(1)})^j],$ $(P^{(1)})^j,$ $\cdots$
$\mathbf{H}_j'$	$P, \tilde{E}, P, \cdots P,$	$\tilde{E},$	$P^{j+1},$ $\mathsf{TEM}[P^{j+1}],$ $P^{j+1},$ $\cdots$
$\mathbf{H}_{j+1}$	$P, \tilde{E}, P, \cdots P,$	$\tilde{E},$	$P,$ $\mathsf{TEM}[P^{j+1}],$ $P^{j+1},$ $\cdots$
$\vdots$			

Lemma 4. *It holds that*

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^*) = 1]\right| \leq 4\sqrt{q_Q/2^\kappa}, \tag{30}$$

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j^*) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1]\right| \leq j/2^\kappa, \tag{31}$$

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j^{**}) = 1] - \Pr[\mathcal{A}(\mathbf{H}_j') = 1]\right| \leq 3j/2^\kappa, \tag{32}$$

$$\left|\Pr[\mathcal{A}(\mathbf{H}_j') = 1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1}) = 1]\right| \leq 2q_{Q,j+1}\sqrt{2(j+1)/2^\kappa}, \tag{33}$$

where $q_{Q,j+1}$ is the expected number of the quantum queries $\mathcal{A}$ makes between the $(j+1)$-th and $(j+2)$-th classical queries.

Proof. To prove Eq. (30), we reduce the problem of distinguishing $\mathbf{H}_j$ and $\mathbf{H}_j^*$ to the problem of distinguishing P_0 and P_1 in our resampling lemma (Lemma 3): Construct a distinguisher $\mathcal{D}$ for our resampling lemma running as follows.

1. $\mathcal{D}$ is given quantum oracle access to a random permutation P_0 and its inverse. $\mathcal{D}$ samples an ideally random tweakable permutation $\tilde{E}$, and runs $\mathcal{A}$. When $\mathcal{A}$ makes classical and quantum queries, $\mathcal{D}$ answers by using $\tilde{E}$ and P_0 (and their inverses) until $\mathcal{A}$ submits the $(j+1)$-th classical query. Assume it is to the forward oracle, denoted as (t_{j+1}, x_{j+1}) (the discussions when the query is to the inverse oracle are analogous). As before, let $T_j = ((t_i, x_i, y_i, b_i))_{i=1,\ldots,j}$ be the history of the classical queries by $\mathcal{A}$ and the responses. $\mathcal{D}$ chooses a distribution D so that sampling s_0 from D will be equivalent to sampling k from D_{key} and set $s_0 := f_1(t_{j+1}, k) \oplus x_{j+1}$, and finally $\mathcal{D}$ outputs D.
2. s_0 is sampled according to D, s_1 is chosen uniformly at random from $\{0,1\}^n$, and P_1 is defined as $P_1 := P_0 \circ \mathsf{swap}_{s_0,s_1}$. After a random bit b is chosen, the bit strings s_0, s_1 and quantum oracle access to P_b are given to $\mathcal{D}$. Now, $\mathcal{D}$ computes $k := f_1^{-1}(t_{j+1}, s_0 \oplus x_{j+1})$, and resumes running $\mathcal{A}$. When $\mathcal{A}$ makes queries, $\mathcal{D}$ answers by using $\mathsf{TEM}_k\left[(P_b)^{T_j,k}\right]$ and $(P_b)^{T_j,k}$ instead of $\tilde{E}$ and P_0. $\mathcal{D}$ returns $\mathcal{A}$'s final output as its own output.

Then, this $\mathcal{D}$ perfectly simulates the experiments $\mathbf{H}_j$ and $\mathbf{H}_j^*$ for $\mathcal{A}$ when $b=0$ and $b=1$, respectively. Hence, Eq. (30) follows from Lemma 3.

The proofs for Eq. (31)-Eq. (33) are almost exactly the same as the corresponding proofs in [2, Section 4.1]. More specifically, those in [2, Section 4.1] are for TEM-KX, but they also become valid proofs for TEM with low-entropy keys if the terms $f_i(t, P(k||0^{n-\kappa}))$ are replaced with $f_i(t,k)$. Thus, we omit the details. □

Finishing the Proof. $\mathbf{H}_0$ and $\mathbf{H}_{q_C}$ correspond to the real and ideal worlds. From Lemma 4, it follows that

$$\begin{aligned}
&|\Pr[\mathcal{A}(\mathbf{H}_0)=1] - \Pr[\mathcal{A}(\mathbf{H}_{q_C})=1]| \\
&\le \sum_{j=0}^{q_C-1} \left|\Pr[\mathcal{A}(\mathbf{H}_j)=1] - \Pr[\mathcal{A}(\mathbf{H}_j^*)=1]\right| + \left|\Pr[\mathcal{A}(\mathbf{H}_j^*)=1] - \Pr[\mathcal{A}(\mathbf{H}_j^{**})=1]\right| \\
&\qquad + \left|\Pr[\mathcal{A}(\mathbf{H}_j^{**})=1] - \Pr[\mathcal{A}(\mathbf{H}_j')=1]\right| + \left|\Pr[\mathcal{A}(\mathbf{H}_j')=1] - \Pr[\mathcal{A}(\mathbf{H}_{j+1})=1]\right| \\
&\le \sum_{j=0}^{q_C-1} \left(4\sqrt{q_Q/2^\kappa} + j/2^\kappa + 3j/2^\kappa + 2q_{Q,j+1}\sqrt{2(j+1)/2^\kappa}\right) \\
&\le 4\sqrt{\frac{(q_C)^2 q_Q}{2^\kappa}} + \frac{2(q_C)^2}{2^\kappa} + 2\sqrt{\frac{2q_C(q_Q)^2}{2^\kappa}}.
\end{aligned}$$

Hence, the claim of Proposition 1 holds.

D Security of FSSponge with a Secret Permutation

On the security of FSSponge built on a secret permutation π, Mennink et al. [64] showed the following lemma. (The original lemma was shown only for classical adversaries, but it also holds for quantum adversaries because all the queries are classical.)

Lemma 5 (Lemma 2 in [64]). *Let $\mathcal{A}$ be a (possibly quantum) adversary against PRF security of* $\mathsf{FSSponge}[\pi]$*, where π is a* secret *n-bit permutation to which $\mathcal{A}$ does not have access. Suppose $\mathcal{A}$ makes at most q classical construction queries, with the number of effective blocks at most ℓ each. Then,*

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{FSSponge}[\pi]}(\mathcal{A}) \leq \frac{2(q\ell)^2}{2^n} + \frac{2q^2\ell}{2^c}$$

holds.

E Proof for Theorem 7

Let $\mathsf{KProc}[P] : \{0,1\}^\kappa \to \{0,1\}^n$ be key processing part of OKS defined as in Section 3.3.

In addition, for $k_r \in \{0,1\}^r$ and $k_c \in \{0,1\}^c$, let $\mathsf{IKS}'_{k_r,k_c}[P]$ be the variant of IKS such that (1) both the rate and capacity of the initial value of the sponge construction are replaced from 0^r and 0^c to k_r and k_c, respectively, and (2) the padding function is removed from the underlying sponge function, and the length of each input is always a multiple of r. That is, $\mathsf{IKS}'_{k_r,k_c}[P] := \mathsf{Sponge}^{\mathsf{w/o\text{-}pad}}_{k_r,k_c}[P]$. We assume that k_r and k_c are chosen uniformly at random.

In what follows, we first prove the PRF security of IKS' (Section E.1), secondly the PRNG security of KProc (Section E.2), and finally complete the security proof of OKS (Section E.3).

E.1 Security Proof for IKS'

Proposition 4. *Let $\mathcal{A}$ be a quantum adversary against post-quantum PRF security of* IKS' *in the quantum ideal permutation model. Assume that k_r and k_c are chosen uniformly at random. Suppose $\mathcal{A}$ makes classical construction queries with the number of effective blocks at most σ in total, and at most q_Q quantum primitive queries to P and P^{-1}. Then,*

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}'}(\mathcal{A}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^c}} + \frac{3\sigma^2}{2^c} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^c}} \tag{34}$$

holds.

Proof. Let $\mathcal{B}$ be an adversary against the post-quantum PRF security of $\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}_{k_c}[P]$ running as follows (k_c is chosen uniformly at random from $\{0,1\}^c$).

1. $\mathcal{B}$ chooses $k_r \in \{0,1\}^r$ uniformly at random.

2. $\mathcal{B}$ runs $\mathcal{A}$. When $\mathcal{A}$ makes quantum queries to P and P^{-1}, $\mathcal{B}$ just forwards queries and responses. When $\mathcal{A}$ makes a classical construction query M, then $\mathcal{B}$ queries $M \oplus (k_r||0^{|M|-r})$ to the construction oracle, and return the response to $\mathcal{A}$.
3. Finally, $\mathcal{B}$ returns what $\mathcal{A}$ outputs.

Then, $\mathcal{B}$ perfectly simulates the oracles $(\mathsf{IKS}'_{k_r,k_c}[P], P, P^{-1})$ and (RF, P, P^{-1}) when the construction oracle given to $\mathcal{B}$ is $(\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}_{k_c}[P], P, P^{-1})$ and (RF, P, P^{-1}), respectively. Thus, we have

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}'}(\mathcal{A}) = \mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}}(\mathcal{B}). \tag{35}$$

In addition, we can show

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}^{\mathsf{w/o\text{-}pad}}}(\mathcal{B}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^c}} + \frac{3\sigma^2}{2^c} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^c}}. \tag{36}$$

in the same way as we showed Theorem 5. (Note that κ in Theorem 5 corresponds to c here.)

Therefore, the claim of the proposition follows from Eq. (35) and Eq. (36). □

E.2 Security Proof for KProc

For an adversary $\mathcal{A}$, we define the Pseudo-Random Number Generator (PRNG) advantage of KProc in the quantum ideal permutation model as

$$\mathsf{Adv}^{\mathtt{prng}}_{\mathsf{KProc}}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{P,P^{-1}}(\mathsf{KProc}[P](k))\right] - \Pr\left[1 \leftarrow \mathcal{A}^{P,P^{-1}}(R)\right]\right|,$$

where k and R are chosen uniformly at random from $\{0,1\}^\kappa$ and $\{0,1\}^n$, respectively.

Then, the following proposition holds.

Proposition 5. *As in Theorem 7, assume that $\kappa > r$, and let $\kappa' := \min(\kappa - r + (|\mathsf{s}_1| \mod r), r - (|\mathsf{s}_1| \mod r))$. Let $\mathcal{A}$ be a quantum adversary against the post-quantum PRNG security of KProc that makes at most q_Q quantum queries to P and P^{-1}. Then,*

$$\mathsf{Adv}^{\mathtt{prng}}_{\mathsf{KProc}}(\mathcal{A}) \leq 8\sqrt{\frac{2(q_Q + \lceil |k^*|/r \rceil)^2}{2^{\kappa'}}}$$

holds.

Proof. We prove the claim when $\mathsf{s}_1 = \varnothing$ and s_2 is the minimum-length string of the form $0\cdots0$ such that the length of $k^* = k||\mathsf{s}_2 = k||0\cdots0$ becomes a multiple of r. The proof for other cases is analogous.

Let k'_1 be a κ'-bit key chosen uniformly at random, and $E_{k'_1}[P]$ be the n-bit Even-Mansour cipher defined as $E_{k'_1}[P](x) = P(x \oplus (k'_1||0^{n-\kappa'})) \oplus (k'_1||0^{n-\kappa'})$.

Let $\mathcal{B}$ be a quantum adversary $\mathcal{B}$ against the post-quantum PRP security of the above $E_{k'_1}[P]$ running as follows.

1. Sample r-bit strings k'_0 and $k'_2, k'_3, \ldots, k'_{\lceil \kappa/r \rceil - 1}$ uniformly at random. Query $k'_0||0^{n-r}$ to the classical encryption oracle ($E_{k'_1}[P]$ or a random permutation) to obtain a response S_1.
2. Compute the value $S_i \leftarrow P(S) \oplus (k_i||0^{n-r})$ for $i = 2, \ldots, \lceil \kappa/r \rceil - 1$ (if $\lceil \kappa/r \rceil \geq 3$).
3. Compute $S_* \leftarrow P(S_{\lceil \kappa/r \rceil - 1})$.
4. Run $\mathcal{A}$, giving S_* as the input. When $\mathcal{A}$ makes quantum queries to P and P^{-1}, just forward queries to/responses from the quantum oracles given to $\mathcal{B}$. When $\mathcal{A}$ returns an output, return it as $\mathcal{B}$'s own output.

The input S_* for $\mathcal{A}$ is perfectly indistinguishable from $\mathsf{KProc}[P](k)$ for random $k \in \{0,1\}^\kappa$ (resp., a uniformly random n-bit value) if the classical encryption oracle given to $\mathcal{B}$ is $E_{k'_1}[P]$ (resp., a random permutation). Hence,

$$\mathsf{Adv}^{\mathtt{prng}}_{\mathsf{KProc}}(\mathcal{A}) = \mathsf{Adv}^{\mathtt{prp}}_{E}(\mathcal{B}) \tag{37}$$

holds. In addition, $\mathcal{B}$ makes 1 construction query and at most $(q_Q + (|k^*|/r))$ primitive queries to P and P^{-1}. Therefore, by Proposition 1, we have

$$\begin{aligned}\mathsf{Adv}^{\mathtt{prp}}_{E}(\mathcal{B}) &\leq 4\sqrt{\frac{(q_Q + (|k^*|/r))}{2^{\kappa'}}} + \frac{2}{2^{\kappa'}} + 2\sqrt{\frac{2(q_Q + (|k^*|/r))^2}{2^{\kappa'}}} \\ &\leq 8\sqrt{\frac{2(q_Q + (|k^*|/r))^2}{2^{\kappa'}}}.\end{aligned} \tag{38}$$

The claim of the proposition follows from Eq. (37) and Eq. (38). □

E.3 Finishing Proof for Theorem 7

Proof (of Theorem 7). Let $\mathcal{B}$ be a quantum adversary against the post-quantum PRNG security of KProc running as follows.

1. First, $\mathcal{B}$ is given an input $R \in \{0,1\}^n$, which is $\mathsf{KProc}[P](k)$ for a random $k \in \{0,1\}^\kappa$ in the real world, and a uniform random bit string in the ideal world. Let R_r (resp., R_c) be the most significant r bits (resp., the remaining c bits) of R.
2. $\mathcal{B}$ runs the adversary $\mathcal{A}$. When $\mathcal{A}$ makes quantum primitive queries, $\mathcal{B}$ just forward them to its own oracles and return the responses to $\mathcal{A}$. When $\mathcal{A}$ queries a value M to the classical construction oracle, then $\mathcal{B}$ computes $\mathsf{IKS}'_{R_r,R_c}[P](M)$ and returns it to $\mathcal{A}$.

Then, $\mathcal{B}$ perfectly simulates the classical construction oracle of $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}_k[P]$ in the real world (i.e., when $R = \mathsf{KProc}[P](k)$), and $\mathsf{IKS}'_{k_r,k_c}[P]$ with random k_r, k_c

in the ideal world (i.e., when R is a random string). Therefore,

$$\begin{aligned}
&\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}}(\mathcal{A}) \\
&= \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{OKS}_k^{\mathsf{w/o\text{-}pad}}[P],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF},P,P^{-1}}\right]\right| \\
&\leq \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{OKS}_k^{\mathsf{w/o\text{-}pad}}[P],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{IKS}'_{k_r,k_c}[P],P,P^{-1}}\right]\right| \\
&\quad + \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{IKS}'_{k_r,k_c}[P],P,P^{-1}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathsf{RF},P,P^{-1}}\right]\right| \\
&= \mathsf{Adv}^{\mathtt{prng}}_{\mathsf{KProc}}(\mathcal{B}) + \mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}'}(\mathcal{A}) \qquad (39)
\end{aligned}$$

holds.

Since $\mathcal{B}$ makes at most $(q_Q + \sigma)$ quantum primitive queries to P and P^{-1},

$$\mathsf{Adv}^{\mathtt{prng}}_{\mathsf{KProc}}(\mathcal{B}) \leq 8\sqrt{\frac{2(q_Q + \sigma + (|k^*|/r))^2}{2^{\kappa'}}} \qquad (40)$$

follows from Proposition 5. In addition, by Proposition 4,

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{IKS}'}(\mathcal{A}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^c}} + \frac{3\sigma^2}{2^c} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^c}} \qquad (41)$$

holds. Therefore,

$$\mathsf{Adv}^{\mathtt{prf}}_{\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}}(\mathcal{A}) \leq 4\sqrt{\frac{\sigma^2 q_Q}{2^c}} + \frac{3\sigma^2}{2^c} + 2\sqrt{\frac{2\sigma(q_Q)^2}{2^c}} + 8\sqrt{\frac{2(q_Q + \sigma + (|k^*|/r))^2}{2^{\kappa'}}}$$

follows from Eq. (39)-Eq. (41). This shows that the claim of the theorem holds (because OKS is obtained from $\mathsf{OKS}^{\mathsf{w/o\text{-}pad}}$ by restricting the domain). □