Sponge-Based Authenticated Encryption: Security Against Quantum Attackers

Christian Janson[1] and Patrick Struck[2](✉)

[1] Technische Universität Darmstadt, Darmstadt, Germany
christian.janson@cryptoplexity.de
[2] Universität Regensburg, Regensburg, Germany
patrick.struck@ur.de

Abstract. In this work, we study the security of sponge-based authenticated encryption schemes against quantum attackers. In particular, we analyse the sponge-based authenticated encryption scheme Slae as put forward by Degabriele et al. (ASIACRYPT'19) due to its modularity. We show that the scheme achieves security in the post-quantum (QS1) setting in the quantum random oracle model by using the one-way to hiding lemma. Furthermore, we analyse the scheme in a fully-quantum (QS2) setting. There we provide a set of attacks showing that Slae does not achieve ciphertext indistinguishability and hence overall does not provide the desired level of security.

1 Introduction

Authenticated encryption schemes with associated data (AEAD) [47] are the main employed cryptographic scheme when it comes to securing the communication between two parties who already share a secret key by ensuring both confidentiality and authenticity of the exchanged messages. Several works show that AEAD schemes can be constructed purely from sponges [21–24,35], which were initially introduced as a tool to construct cryptographic hash functions. Recent examples of such sponge-based AEAD schemes are Isap [22,23] and Slae [21]. Observe that these schemes are already analysed showing that they are even secure against side-channel leakage, however, their security against quantum adversaries has yet to be studied.

Unlike public key cryptography that is based on number theoretic problems, which is completely broken by Shor's algorithm [49], AEAD schemes are often assumed to be only mildly affected by Grover's algorithm [31], although this assumption turns out to be delusive in some cases [13]. To compensate this, usually one simply doubles the key length. This approach indeed works for many symmetric schemes in the standard model, namely those where their security proofs can be easily translated to one against quantum adversaries [50]. However, schemes that rely on random oracles [7] cannot be translated in a straightforward manner and hence require more attention. In particular, translating their security to hold against quantum adversaries requires a proof in the quantum random

J. H. Cheon and T. Johansson (Eds.): PQCrypto 2022, LNCS 13512, pp. 230–259, 2022.
https://doi.org/10.1007/978-3-031-17234-2_12

oracle model (QROM) [10], and it has recently been shown that proofs cannot always be translated from the ROM to the QROM [56]. In particular, this will also apply to sponge-based AEAD schemes where we typically model the block function that underlies the sponge construction as a random oracle and includes the schemes in [21–23].

The security of cryptographic primitives against quantum adversaries can nowadays be divided into two cases [27,37]. The first case corresponds to the setting of post-quantum security (usually abbreviated as QS1) where the adversary only has quantum computing power. This setting covers the scenario once the first large-scale quantum computer exists and corresponds to the setting described above which typically requires switching from the ROM to the QROM. The second case deals with the setting of quantum security (usually referred to as QS2) where protocol participants also have quantum computing power. This covers a scenario where quantum computers are ubiquitous but also earlier scenarios using more sophisticated attacks such as the *frozen smart-card* attack [28].

Observe that security in the QS2 setting is more involved since the adversary gets superposition access to the primitive, e.g., it can encrypt/sign messages in a superposition. Many schemes that are secure in the QS1 setting are however completely broken in the QS2 setting as is shown by a series of works [2,4,33,36,41,42,48]. Yet another difficulty in the QS2 setting is that there are many different security notions [1,12,14,15,25,28–30,43]. These notions use different approaches to formalise the idea of allowing the adversary to "encrypt/sign messages in a superposition" in order to obtain a security notion that translates the classical intuition of the corresponding security notion to the QS2 setting.

Our Contribution. In this work, we study the security of sponge-based authenticated encryption schemes against quantum attackers which has so far only received very little attention. In particular, we scrutinize the scheme Slae as put forward by Degabriele et al. [21] in both settings, namely in the QS1 and QS2 setting. Observe that the beauty of Slae is its simplicity in terms of their construction, i.e., Slae is a N2-composition [44] of a symmetric key encryption scheme and a message authentication code. In particular, Degabriele et al. show that Slae can be viewed in terms of smaller components (with slight improvements by [39]), i.e., the encryption scheme consists of a sponge-based pseudorandom function (PRF) and a sponge-based pseudorandom generator (PRG) while the MAC consists of the combination of a sponge-based hash function and a sponge-based PRF (a more detailed description can be found in Sect. 3). Note that our analysis does not only contribute towards the study of Slae but rather also provides a QS1 and QS2 analysis of the core primitives themselves which is of independent interest. Note that Slae is a leakage-resilient AEAD scheme. However, in this work we do not consider the leakage setting but rather use the scheme Slae due to its simplicity in order to provide a thorough security analysis of sponge-based AEAD schemes and the employed core primitives in the QS1 and QS2 setting closing this gap in the literature.

In the QS1 setting, we are able to establish security for Slae. In particular, by using the one-way to hiding lemma [3,53], we can show that the underlying building blocks, namely the sponge-based PRF and PRG are secure with respect to quantum adversaries. For the sponge-based hash function, we show that we can leverage existing results [18] to the construction specifics of Slae. Finally, being equipped with the established results, we can overall establish security of Slae in the QS1 setting.

In the QS2 setting, we analyse the ciphertext indistinguishability of Slae. Unlike the QS1 setting, there are different notions for ciphertext indistinguishability in the QS2 setting which do not form a strict hierarchy. We consider the two strongest, incomparable notions by Gagliardoni et al. [28] and Mossayebi and Schack [43]. We extend these notions to the nonce-based setting and show that Slae achieves neither of these notions by showing attacks. Finally, we argue that one may establish QS2 security in the sense of [12] of the generic construction that underlies Slae. However, the security when studying the sponge-based construction is left as an open problem.

As mentioned above, we chose to analyse Slae rather than other relevant sponge-based schemes due to its modularity. Since Slae is based on a random transformation, we can leverage techniques for the QROM, whereas other sponge-based primitives are typically based on a random permutation. Our results yield post-quantum secure pseudorandom functions, pseudorandom generators, and hash functions all constructed entirely from sponges. Since these are fundamental cryptographic building blocks our contribution is more than just a post-quantum security proof for an AEAD scheme and can be applied elsewhere. In particular, it provides a starting point for proving post-quantum security of more practical schemes.

Related Work. Sponges were introduced by Bertoni et al. [8] as a tool to construct cryptographic hash functions which resulted in the hash function SHA-3. Since then, sponges were shown to be a versatile tool allowing not only the construction of hash functions but also primitives including authenticated encryption schemes [21–24,35].

Research in the realm of QS1 security of sponges mainly targets the security of hash functions. The first result addresses sponge-based hash functions based on random transformations or non-invertible random permutations [18]. The ultimate goal is a post-quantum proof for SHA-3 which is targeted both by Unruh [55][1] and Czajkowski [16] using Zhandry's compressed oracle technique [58]. Apart from that we are not aware of other works considering the QS1 security of sponge-based constructions.

In the QS2 setting, [20] studies the quantum indifferentiability of sponges and [19] analyses the quantum indistinguishability of sponge-based pseudorandom functions. The analysis in [19] uses keyed functions for the underlying block function which allow the adversary only classical access to these block functions while it has superposition access to the resulting pseudorandom function.

[1] Observe that the current version of the paper is flawed.

Soukharev et al. [51] study the generic composition paradigms for authenticated encryption in the QS2 setting according to the security notions put forth by Boneh and Zhandry [12]. However, their proof implicitly assumes that superposition queries by the adversary can be recorded which, at this point, was unclear how to do as was pointed out Chevalier et al. [15].

Structure of the Paper. In Sect. 2 and Appendix A, we provide the necessary notation and background. The general sponge construction and the particular instantiation Slae is provided in Sect. 3. In Sect. 4, we provide a security analysis in the QS1 setting while in Sect. 5, we provide an analysis in the QS2 setting. We conclude the paper in Sect. 6 and provide proof details in Appendices B and C.

2 Preliminaries

2.1 Notation

For any positive integer $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For any two bit strings x and y of length n, $|x|$ denotes the size of x, $x \parallel y$ denotes their concatenation and by $x \cdot y = x_1 y_1 \oplus x_2 y_2 \oplus \ldots \oplus x_n y_n$ we denote their inner product. Furthermore, for a positive integer $k \leq |x|$, we use the notation $\lfloor x \rfloor_k$ to denote the string when truncated to its k least significant bits while $\lceil x \rceil^k$ denotes the string when truncated to its k most significant bits. We denote the set of bit strings of size n by $\{0,1\}^n$, and we denote by $\{0,1\}^*$ the set of all bit strings of finite length. By writing $x \leftarrow_{\$} \mathcal{X}$, we denote the process of sampling at random a value from a finite set $\mathcal{X}$ and assigning it to x. We simply denote by $\mathsf{par}(x)$ the parity of x. Furthermore, we denote by $\mathcal{Y}^{\mathcal{X}}$ the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We assume familiarity with the basics of quantum computation such as bra-ket notion for quantum states, e.g., $|x\rangle$, Hadamard operators, and measurements. For an in-depth discussion we refer to [46].

2.2 Definitions

Due to space restrictions, we provide basic definitions about authenticated encryption with associated data (AEAD) and message authentication codes (MAC) in Appendix A.

Pseudorandom Function. Next we define pseudorandom functions and their respective security.

Definition 1. *Let $\mathcal{F}\colon \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ be a deterministic function. We define the PRF advantage of an adversary $\mathcal{A}$ against $\mathcal{F}$ as*

$$\mathbf{Adv}^{\mathsf{PRF}}_{\mathcal{F}}(\mathcal{A}) = \left| \Pr_{\mathsf{K} \leftarrow_{\$} \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot)} \rightarrow 1] - \Pr_{\overline{\mathcal{F}} \leftarrow_{\$} \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot)} \rightarrow 1] \right| .$$

Pseudorandom Generator. Next we define a pseudorandom generator and its security. Observe that we specify a PRG with variable output length, where the length is specified as part of the input.

Definition 2. *Let $\mathcal{G}\colon \mathcal{S} \times \mathbb{N} \to \{0,1\}^*$ be a pseudorandom generator with associated seed space $\mathcal{S}$ and let $\ell \in \mathbb{N}$ define the PRG's output length. We define the PRG advantage of an adversary $\mathcal{A}$ against $\mathcal{G}$ as*

$$\mathbf{Adv}_{\mathcal{G}}^{\mathsf{PRG}}(\mathcal{A}) = \left| \Pr_{z \leftarrow_\$ \mathcal{S}}[\mathcal{A}(\mathcal{G}(z,\ell)) \to 1] - \Pr_{R \leftarrow_\$ \{0,1\}^\ell}[\mathcal{A}(R) \to 1] \right| .$$

Hash Function. Hash functions are a versatile cryptographic primitive that are efficiently computable functions that compress bit strings of arbitrary length to bit strings of fixed length. Hash functions do enjoy a variety of security properties and next we define collision resistance over a domain $\mathcal{X} = \{0,1\}^*$.

Definition 3. *Let $\mathcal{H}\colon \mathcal{X} \to \{0,1\}^w$ be a hash function constructed from a random transformation ρ. We define the collision-resistance advantage of an adversary $\mathcal{A}$ against $\mathcal{H}$ where the adversary has (quantum) oracle access to ρ as*

$$\mathbf{Adv}_{\mathcal{H}}^{\mathsf{CR}}(\mathcal{A}) = \Pr[(X_0, X_1) \leftarrow_\$ \mathcal{A}^\rho \colon \mathcal{H}(X_0) = \mathcal{H}(X_1) \wedge X_0 \neq X_1 \wedge X_0, X_1 \in \mathcal{X}] .$$

Since we consider hash functions in the QS1 and QS2 setting in this work, we require two additional properties when arguing about the security of a hash function, namely collapsing hash functions and zero-preimage resistance.

The collapsing property of hash functions is due to Unruh [54], who observed that collision resistance is not sufficient to construct commitment schemes secure against quantum adversaries.[2] Intuitively, a hash function is collapsing if an adversary can not distinguish between a measurement of the output (the hash value) and a measurement of the input. In [52, Lemma 25], Unruh shows that collapsing hash functions are also collision resistant. We present the formal definition of collapsing security in Appendix A.3.

Zero-preimage resistance states that it is infeasible for the adversary to output an element from the function's domain which evaluates to the zero string.

Definition 4. *Let $f^\rho\colon \{0,1\}^x \to \{0,1\}^y$ be a function. We define the* zero-preimage resistance *advantage of an adversary $\mathcal{A}$ against f^ρ where the adversary has (quantum) oracle access to ρ as*

$$\mathbf{Adv}_{f^\rho}^{\mathsf{ZP}}(\mathcal{A}) = \Pr[f^\rho(X) = 0^y \colon X \leftarrow_\$ \mathcal{A}^\rho] .$$

[2] In a nutshell, a quantum adversary can open a commitment to an arbitrary message but not to two different messages. Thus it breaks the binding property without finding a collision.

Quantum Random Oracle Model and One-way to Hiding Lemma. The quantum random oracle model (QROM) was formalised by Boneh et al. [10] extending the random oracle model (ROM) [7] to the quantum setting. The QROM has become the de-facto standard for analysing primitives which rely on random oracles. Boneh et al. [10] gave a separation between the ROM and the QROM, yet under non-standard assumptions. Recently, Yamakawa and Zhandry [56] provided a separation under standard assumptions. More precisely, let $\mathsf{H}\colon \{0,1\}^n \to \{0,1\}^n$,[3] then the QROM allows a quantum adversary access to the unitary U_{H} that does the following

$$\sum_{x,y\in\{0,1\}^n} \alpha_{x,y} |x\rangle |y\rangle \mapsto \sum_{x,y\in\{0,1\}^n} \alpha_{x,y} |x\rangle |y \oplus \mathsf{H}(x)\rangle \ .$$

We write $\mathcal{A}^{\mathsf{H}}$ to denote that $\mathcal{A}$ has oracle access to H which means having access to an oracle performing the unitary above.

The one-way to hiding (O2H) lemma is a fundamental tool for proofs in the quantum random oracle model (QROM). It provides an upper bound on the distinguishing advantage of a quantum adversary between different random oracles when having superposition access to it. The first variant was given by Unruh [53]. Subsequently, variants achieving tighter bounds were given in [3,9,40], yet at the cost of a more restricted applicability.

Below we recall the O2H lemma by Unruh [53], albeit in the formulation put forth by Ambainis et al. [3].

Lemma 5 (One-way to hiding (O2H) [3]). *Let G, $\mathsf{H}\colon \mathcal{X} \to \mathcal{Y}$ be random functions, let z be a random bitstring, and let $\mathcal{S} \subset \mathcal{X}$ be a random set such that $\forall x \notin \mathcal{S}$, $\mathsf{G}(x) = \mathsf{H}(x)$. $(\mathsf{G}, \mathsf{H}, \mathcal{S}, z)$ may have arbitrary joint distribution. Furthermore, let $\mathcal{A}^{\mathsf{H}}$ be a quantum oracle algorithm which queries H at most q times. Define an oracle algorithm $\mathcal{B}^{\mathsf{H}}$ as follows: Pick $i \leftarrow_{\$} [q]$. Run $\mathcal{A}_q^{\mathsf{H}}(z)$ until just before its i-th query to H. Measure the query in the computational basis, and output the measurement outcome. Then it holds that*

$$\left|\Pr[\mathcal{A}^{\mathsf{H}}(z) \to 1] - \Pr[\mathcal{A}^{\mathsf{G}}(z) \to 1]\right| \leq 2q\sqrt{\Pr[x \in \mathcal{S} \,|\, \mathcal{B}^{\mathsf{H}}(z) \to x]}\,.$$

3 The Sponge Construction and Slae

In this section, we provide the basic syntax about the sponge construction. Being equipped with the required syntax, we review Slae which is a N2-based authenticated encryption scheme [44] based on the sponge construction. Recall that a N2-construction follows the Encrypt-then-MAC paradigm and Slae is a refinement that builds a nonce-based AEAD scheme from a nonce-based symmetric key encryption scheme and a vector MAC.

[3] We assume that domain and co-domain are of the same size as it is the only case we are considering in this work.

3.1 Sponge Construction

The sponge construction has been introduced by Bertoni et al. [8] and has been used to build various cryptographic primitives. In Fig. 1, we provide an illustration of the plain sponge construction.

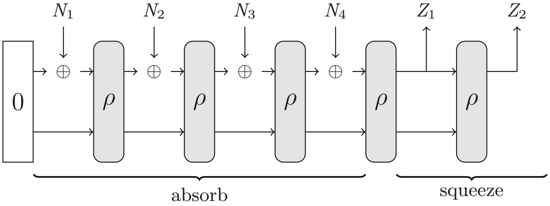

Fig. 1. Plain sponge using four rounds of absorbing and two rounds of squeezing.

The sponge construction consists of a so-called *absorbing* phase and a *squeezing* phase that is built upon a transformation ρ that is iteratively called on its input. This transformation basically maps strings of length n to strings of the same length, and in particular one can decompose n into two values $r + c$ where r is called the rate and c is called the capacity. After each iteration of the transformation we refer to its output as the state S. Furthermore, we usually refer to the leftmost r bits of the state as the outer part $\bar{S}$, which is equivalent to $\lceil S \rceil^r$, and we refer to the remaining c bits as the inner part $\hat{S}$, which is equivalent to $\lfloor S \rfloor_c$. In order to input some element N, this input is first padded to a non-zero multiple of the rate r. For this, we use an injective padding function $\texttt{pad}$ to get $l \geq 1$ input blocks $N_1 \parallel N_2 \parallel \ldots \parallel N_l = \texttt{pad}(N)$. At the ith iteration, N_i is XORed with the outer part $\bar{S}$ before being inputted to the transformation, i.e., more formally $Y_i \leftarrow (N_i \oplus \bar{S}_i) \parallel \hat{S}_i$ and evaluating $S_{i+1} \leftarrow \rho(Y_i)$. In the squeezing phase, one can produce an output in one or more iterations obtaining r bits of output per iteration, i.e., more formally at the jth iteration the output Z_j is produced by $Z_j \leftarrow \bar{S}_j$.

3.2 The FGHF' Construction and Slae

Degabriele et al. [21] provide a generic N2-construction [44] of a leakage-resilient authenticated encryption scheme with associated data called the FGHF' construction. In particular, they show that the encryption component can be constructed from a fixed-input length function family that retains pseudorandomness in the presence of leakage (F) combined with a (standard) pseudorandom generator (G) while the authentication component is built from a collision-resistant hash function (H) and a fixed-input length function family that retains both pseudorandomness and unpredictability in the presence of leakage (F'). Overall this yields a leakage-resilient AEAD scheme. Observe that Krämer and

```
Slae-Enc($\mathsf{K}, N, A, M$)
    $C \leftarrow$ SlEnc($\mathsf{K}, N, M$)
    $T \leftarrow$ SlMac($\mathsf{K}, (N, A, C)$)
    return $(C, T)$

SlEnc-Enc($\mathsf{K}, N, M$)
    $z \leftarrow$ SlFunc($\mathsf{K}, N$)
    $Z \leftarrow$ SPrg($z, |M|$)
    $C \leftarrow Z \oplus M$
    return $C$

SlMac-T($\mathsf{K}, (N, A, C)$)
    $H \leftarrow$ SvHash($N, A, C$)
    $T \leftarrow$ SlFunc($\mathsf{K}, H$)
    return $T$
```

```
SlFunc($\mathsf{K}, N$)
    $l \leftarrow \left\lceil \frac{|N|}{r} \right\rceil$
    $Y_0 \leftarrow \mathsf{K}$
    for $i = 1..l$
        $S_i \leftarrow \rho(Y_{i-1})$
        $Y_i \leftarrow (N_i \oplus \bar{S}_i) \parallel \hat{S}_i$
    $S_{l+1} \leftarrow \rho(Y_l)$
    return $S_{l+1}$

SPrg($z, y$)
    $l \leftarrow \lceil \frac{y}{r} \rceil$
    $S_0 \leftarrow z$
    for $i = 1..l$
        $S_i \leftarrow \rho(S_{i-1})$
        $Z_i \leftarrow \bar{S}_i$
    $Z \leftarrow Z_1 \parallel \ldots \parallel Z_l$
    return $\lfloor Z \rfloor_y$
```

```
SvHash($N, A, C$)
    $S_0 \leftarrow 0^n$
    $Y_0 \leftarrow (N \oplus \bar{S}_0) \parallel \hat{S}_0$
    $S_1 \leftarrow \rho(Y_0)$
    $u \leftarrow \left\lceil \frac{|A|}{r} \right\rceil$
    for $i = 1..u$
        $Y_i \leftarrow (A_i \oplus \bar{S}_i) \parallel \hat{S}_i$
        $S_{i+1} \leftarrow \rho(Y_i)$
    $\hat{S}_{u+1} \leftarrow \hat{S}_{u+1} \oplus (1 \parallel 0^{c-1})$
    $v \leftarrow \left\lceil \frac{|C|}{r} \right\rceil$
    for $i = u+1..u+v$
        $Y_i \leftarrow (C_{i-u} \oplus \bar{S}_i) \parallel \hat{S}_i$
        $S_{i+1} \leftarrow \rho(Y_i)$
    $h \leftarrow \lfloor S_{u+v+1} \rfloor_w$
    return $h$
```

Fig. 2. Pseudocode of Slae and the underlying components. We only provide the details of the encryption and tagging algorithms. Decryption and verification works in the obvious reversed way.

Struck [39] showed that leakage-resilient pseudorandom functions suffice to build the scheme of Degabriele et al. [21] dropping the unpredictability requirement.

Furthermore, Degabriele et al. [21] show that the generic construction FGHF' can be instantiated entirely from the sponge construction using a random transformation. Their particular sponge construction is called Slae which is composed of a symmetric key encryption scheme SlEnc and a MAC SlMac according to the N2-construction. In particular, viewing each of the schemes in terms of their smaller components, Degabriele et al. build SlEnc from a leakage-resilient function SlFunc and a pseudorandom generator SPrg while SlMac can be built from a collision-resistant hash function SvHash and a leakage-resilient function SlFunc, and a formal description is given in Fig. 2. Regarding the security of Slae, they prove the security via a composition theorem for the N2-construction in the leakage setting as established by Barwell et al. [5].

However, the quantum resistance of Slae has not been considered yet. In the following, we will scrutinize the Slae construction in this regard and we set the respective leakage sets to be empty. Therefore, we analyse the construction in the standard setting without leakage.

4 Post-Quantum (QS1) Security

In this section we analyse the security of Slae against quantum adversaries in the QS1 setting. The respective proofs of this Section can be found in Appendix B.

4.1 Security of SlFunc

The sponge-based pseudorandom function SlFunc is illustrated in Fig. 3 while the pseudocode can be found in Fig. 2. The function initialises the state of the sponge with the key and then absorbs the input, in case of Slae the nonce N, r bits at a time. After the nonce has been absorbed, the output is obtained by applying the transformation ρ a final time and outputting the state. Note that the function outputs the full state rather than squeezing it over several rounds. That is also the reason why ρ is required to be a random transformation rather than a random permutation. Otherwise, an adversary could simply undo the transformation from the output by applying the inverse permutation. The theorem below gives a bound on distinguishing SlFunc from a random function when having superposition access to the underlying random oracle ρ. The proof utilises the O2H lemma (cf. Lemma 5).

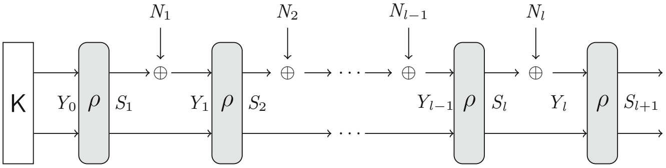

Fig. 3. Sponge-based pseudorandom function SlFunc.

Theorem 6. *Let $\mathcal{F}$ = SlFunc be the function displayed in Fig. 3. Then for any quantum adversary $\mathcal{A}$, making $q_{\mathcal{F}}$ (classical) queries to SlFunc and q_ρ (quantum) queries to ρ, it holds that*

$$\mathbf{Adv}^{\mathsf{PRF}}_{\text{SlFunc}}(\mathcal{A}) \leq \frac{q_{\mathcal{F}}^2 + q_{\mathcal{F}}}{2^{n+1}} + 2q_\rho\sqrt{\frac{2^\nu}{2^n}}\,.$$

Proof. Let $l = \left\lceil \frac{\nu}{r} \right\rceil$ be the number of absorption steps and we assume for simplicity that ν is a multiple of the rate. We further recursively define sets $\mathcal{Y}_i$ as

$$\mathcal{Y}_0 = \{\mathsf{K}\} \qquad \text{and} \qquad \mathcal{Y}_i = \{R \parallel \lfloor \rho(x) \rfloor_c \ \mid R \in \{0,1\}^r, x \in \mathcal{Y}_{i-1}\}$$

for all $i \in \{1, \ldots, l\}$, i.e., $\mathcal{Y}_i$ is the set of all possible values that can occur as input to ρ while evaluating $\mathcal{F}(\mathsf{K}, \cdot)$. It follows that $|\mathcal{Y}_i| \leq 2^{ir}$ and, in particular, $|\mathcal{Y}_l| \leq 2^{lr} = 2^\nu$. Note that every input N defines a sequence of states $Y_0, Y_1, \ldots, Y_l$

that occur while evaluating the sponge. For an input N, let $Y_i[N]$ denote the state Y_i for this particular input, e.g., $Y_1[N] = (\lceil \rho(\mathsf{K}) \rceil^r \oplus N_1) \parallel \lfloor \rho(\mathsf{K}) \rfloor_c$, where $N = N_1 \parallel \ldots \parallel N_l$. In particular, for every input N it holds that $Y_0[N] = \mathsf{K}$.

We want to bound the following difference

$$\mathbf{Adv}^{\mathsf{PRF}}_{\textsc{SlFunc}}(\mathcal{A}) = \left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot),\rho} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho} \to 1] \right| .$$

In order to do this, we define the oracle ρ_*, where $\rho_*(Y_l[N]) = \overline{\mathcal{F}}(N)$ for all $Y_l[N] \in \mathcal{Y}_l$. That is, oracle ρ_* is reprogrammed on all final input states $Y_l[N]$ to output the output of a random function $\overline{\mathcal{F}}$ on the input N. Then it holds that

$$\begin{aligned} &\left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot),\rho} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho} \to 1] \right| \\ &\leq \left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot),\rho} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho_*} \to 1] \right| \\ &\quad + \left| \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho_*} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho} \to 1] \right| . \end{aligned}$$

For the first difference on the right-hand side, the oracles are consistent in both cases. However, if the adversary finds a collision on the final input to ρ for $\textsc{SlFunc}(\mathsf{K},\cdot)$, more precisely, two inputs N and N' such that $\lceil N \rceil^{\nu-r} \neq \lceil N' \rceil^{\nu-r}$ and $Y_l[N] = Y_l[N']$, then these two inputs will result in the same output for $\mathcal{F}$ and (most likely) different outputs for $\overline{\mathcal{F}}$. Finding such a collision is a counting argument over the number of queries to the function and an application of Gaussian summation. Hence, it follows that

$$\left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot),\rho} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho_*} \to 1] \right| \leq \frac{q_{\mathcal{F}}^2 + q_{\mathcal{F}}}{2^{n+1}} .$$

For the second difference, we can apply the O2H lemma (cf. Lemma 5) which yields

$$\left| \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho_*} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho} \to 1] \right| \leq 2q_\rho \sqrt{\Pr[x \in \mathcal{Y}_l \mid \mathcal{B}^{\overline{\mathcal{F}}(\cdot),\rho} \to x]} .$$

Recall that $\mathcal{B}^{\overline{\mathcal{F}}(\cdot),\rho}$ simply runs $\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho}$ and outputs the measurement outcome of a randomly chosen query to ρ. However, $\mathcal{A}$ has no information about the set $\mathcal{Y}_l$, hence we conclude with

$$2q_\rho \sqrt{\Pr[x \in \mathcal{Y}_l \mid \mathcal{B}^{\overline{\mathcal{F}}(\cdot),\rho} \to x]} \leq 2q_\rho \sqrt{\frac{|\mathcal{Y}_l|}{2^n}} \leq 2q_\rho \sqrt{\frac{2^\nu}{2^n}} .$$

Collecting everything yields

$$\begin{aligned} \mathbf{Adv}^{\mathsf{PRF}}_{\textsc{SlFunc}}(\mathcal{A}) &= \left| \Pr_{\mathsf{K} \leftarrow\$ \mathcal{K}}[\mathcal{A}^{\mathcal{F}(\mathsf{K},\cdot),\rho} \to 1] - \Pr_{\overline{\mathcal{F}} \leftarrow\$ \mathcal{Y}^{\mathcal{X}}}[\mathcal{A}^{\overline{\mathcal{F}}(\cdot),\rho} \to 1] \right| \\ &\leq \frac{q_{\mathcal{F}}^2 + q_{\mathcal{F}}}{2^{n+1}} + 2q_\rho \sqrt{\frac{2^\nu}{2^n}} . \end{aligned}$$

□

We would like to point out the following. The length of the nonce ν is typically of fixed size, e.g., in case of the NIST lightweight cryptography standardization process [45] the nonce is assumed to be 12 bytes long. In particular, ν will be much smaller than the size of the sponge n.

4.2 Security of SPRG

In this section we show that the sponge-based pseudorandom generator SPRG is secure against adversaries having superposition access to the underlying random oracle ρ. The PRG SPRG is displayed in Fig. 4 and the respective pseudocode is given in Fig. 2. The construction deviates from more common constructions for pseudorandom generators since it initialises the state of the sponge with the seed rather than absorbing it. The output is then generated by squeezing r bits at each iteration of the sponge. Similar to the previous section, the proof relies on the O2H lemma.

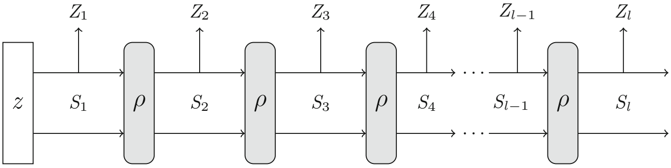

Fig. 4. Sponge-based pseudorandom generator SPRG.

Theorem 7. *Let* SPRG *be the pseudorandom generator displayed in Fig. 4. Then for any quantum adversary $\mathcal{A}$, making q (quantum) queries to ρ, and receiving an input of length μ it holds that*

$$\mathbf{Adv}^{\mathsf{PRG}}_{\mathrm{SPRG}}(\mathcal{A}) \leq \frac{2lq}{\sqrt{2^c}},$$

where $l = \left\lceil \frac{\mu}{r} \right\rceil$ is the number of squeezing steps to obtain the required output length μ.

Proof. Let $l = \left\lceil \frac{\mu}{r} \right\rceil$ be the number of squeezing steps. We assume, for sake of simplicity, that μ is a multiple of r. For a seed z, let $S_1, S_2, \ldots, S_l$ denote the sequence of states that occur during evaluation of the sponge, i.e., $S_i = \rho^{i-1}(z)$, where ρ^i corresponds to i consecutive evaluations of ρ. We want to bound the following difference

$$\mathbf{Adv}^{\mathsf{PRG}}_{\mathrm{SPRG}}(\mathcal{A}) = \left| \Pr_{z \leftarrow\$ \mathcal{S}}[\mathcal{A}^{\rho}(Z) \rightarrow 1] - \Pr_{R \leftarrow\$ \{0,1\}^{\mu}}[\mathcal{A}^{\rho}(R) \rightarrow 1] \right|,$$

where $Z = Z_1 \parallel \ldots \parallel Z_l = \mathrm{SPRG}(z, lr)$, i.e., obtaining an output of length lr using SPRG on seed z and $R = R_1 \parallel \ldots \parallel R_l$, such that $|Z_i| = |R_i| = r$.

We write $R_{[i,j]}$ for $R_i \parallel \ldots \parallel R_j$, the same for Z. In particular, $R_{[i,j]}$ for $i > j$ equals the empty string. In the following we leave out the probability spaces for readability. We obtain

$$\begin{aligned}\mathbf{Adv}^{\mathsf{PRG}}_{\mathrm{SP_{RG}}}(\mathcal{A}) &= \left|\Pr[\mathcal{A}^{\rho}(Z_{[1,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_{[1,l]}) \to 1]\right| \\ &\leq \sum_{i=1}^{l} \left|\Pr[\mathcal{A}^{\rho}(R_{[1,i-1]} \parallel Z_{[i,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_{[1,i]} \parallel Z_{[i+1,l]}) \to 1]\right| .\end{aligned}$$

We start with the first difference, that, after simple rewriting, is,

$$\begin{aligned}&\left|\Pr[\mathcal{A}^{\rho}(Z_1 \parallel Z_{[2,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_1 \parallel Z_{[2,l]}) \to 1]\right| \\ &\leq \left|\Pr[\mathcal{A}^{\rho}(Z_1 \parallel Z_{[2,l]}) \to 1] - \Pr[\mathcal{A}^{\rho_1}(R_1 \parallel Z_{[2,l]}) \to 1]\right| \\ &\quad + \left|\Pr[\mathcal{A}^{\rho_1}(R_1 \parallel Z_{[2,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_1 \parallel Z_{[2,l]}) \to 1]\right| ,\end{aligned}$$

where $\rho_1(R_1 \parallel [S_1]_c) = S_2$. Then it holds that the first difference above is 0, as the relation between R_1 and ρ_1 is the same as between Z_1 and ρ, and we merely need to bound the second difference, which only differs in the random oracle (ρ and ρ_1) at input $R_1 \parallel [S_1]_c$. Let $\mathcal{S}_1 = \{R_1 \parallel [S_1]_c\}$, then we can apply the O2H lemma (cf. Lemma 5) to obtain

$$\begin{aligned}&\left|\Pr[\mathcal{A}^{\rho_1}(R_1 \parallel Z_{[2,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_1 \parallel Z_{[2,l]}) \to 1]\right| \\ &\leq 2q\sqrt{\Pr[x \in \mathcal{S}_1 \,|\, \mathcal{B}^{\rho}(R_1 \parallel Z_{[2,l]}) \to x]} \,.\end{aligned}$$

While $\mathcal{A}$ knows R_1, it has no information about $[S_1]_c$ (note that Z_i, for $i > 1$ provides no information about $\mathcal{S}_1$ due to ρ being one-way in the random oracle model). This yields

$$\Pr[x \in \mathcal{S}_1 \,|\, \mathcal{B}^{\rho}(R_1 \parallel Z_{[2,l]}) \to x] \leq \frac{|\mathcal{S}_1|}{2^c} \leq \frac{1}{2^c} \,.$$

The same argument applies to the other differences, where more and more r bit blocks of $\mathcal{A}$'s input are replaced with R_i. More precisely, we obtain

$$\begin{aligned}&\left|\Pr[\mathcal{A}^{\rho}(R_{[1,i-1]} \parallel Z_{[i,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_{[1,i]} \parallel Z_{[i+1,l]}) \to 1]\right| \\ &\leq 2q\sqrt{\Pr[x \in \mathcal{S}_i \,|\, \mathcal{B}^{\rho}(R_{[1,i]} \parallel Z_{[i+1,l]}) \to x]} \leq \frac{2q}{\sqrt{2^c}} ,\end{aligned}$$

where $\mathcal{S}_i = \{R_i \parallel [S_i]_c\}$. Collecting everything then yields

$$\begin{aligned}\mathbf{Adv}^{\mathsf{PRG}}_{\mathrm{SP_{RG}}}(\mathcal{A}) &= \left|\Pr[\mathcal{A}^{\rho}(Z_{[1,l-1]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_{[1,l-1]}) \to 1]\right| \\ &\leq \sum_{i=1}^{l} \left|\Pr[\mathcal{A}^{\rho}(R_{[1,i-1]} \parallel Z_{[i,l]}) \to 1] - \Pr[\mathcal{A}^{\rho}(R_{[1,i]} \parallel Z_{[i+1,l]}) \to 1]\right| \\ &\leq \sum_{i=1}^{l} 2q\sqrt{\Pr[x \in \mathcal{S}_i \,|\, \mathcal{B}^{\rho}(R_{[1,i]} \parallel Z_{[i+1,l]}) \to x]} \leq \frac{2lq}{\sqrt{2^c}} \,.\end{aligned}$$

□

4.3 Security of SvHash

In this section we analyse the QS1 security of SvHash which we display in Fig. 5 and its respective pseudocode can be found in Fig. 2. Observe that in order to compute a hash digest, the internal state is initialised to an evaluation of the random transformation of a zero bit string of length n XORed with the passed nonce. Afterwards the padded associated data and padded ciphertext are absorbed blockwise. Degabriele et al. chose to employ a domain separation to separate the boundary between associated data and ciphertext consisting of XORing the string $1 \parallel 0^{c-1}$ to the inner state $\hat{S}$ as soon as the associated data has been absorbed. Observe that the domain separation can be viewed as a sponge construction with a rate increased by one bit. In this sense, an adversary $\mathcal{A}$ against SvHash with rate r and capacity c can be viewed as an adversary against the plain sponge-based hash function with rate $r+1$ and capacity $c-1$, where $\mathcal{A}$ guarantees that the $(r+1)$th bit of each input block is 0 except for the block which corresponds to absorbing the first ciphertext block. Hence a bound for the plain sponge-based hash function directly yields a bound for SvHash by accounting for the one bit loss in the capacity. The proof can be found in Appendix B.

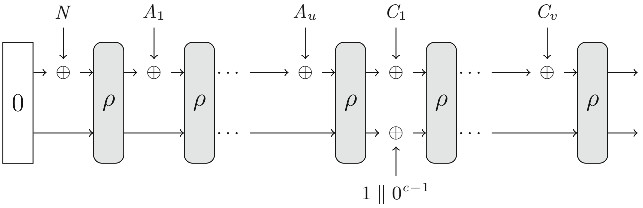

Fig. 5. Sponge-based Hash function SvHash.

Theorem 8. *Let* SvHash *be the hash function as displayed in Fig. 5. Then for any quantum adversary* $\mathcal{A}$ *making* q *(quantum) queries to* ρ*, it holds that*

$$\mathbf{Adv}^{\mathsf{CR}}_{\textsc{SvHash}}(\mathcal{A}) \leq \sqrt{\varepsilon_1} + l \cdot \varepsilon_2 + \varepsilon_3 \,,$$

where $\varepsilon_1 \leq (q+1)^2 2^{-c+4}$, $\varepsilon_2 \leq q^3 \left(\frac{\delta'+324}{2^{c-1}}\right) + 7\delta\sqrt{\frac{3(q+4)^3}{2^c}}$ *and* $\varepsilon_3 \leq q^3 \left(\frac{\delta'+324}{2^{w+1}}\right) + 7\delta\sqrt{\frac{3(q+4)^3}{2^{w+2}}}$ *with non-zero constants* δ *and* δ' *as well as* $l = \left\lceil \frac{\mu}{r} \right\rceil$ *where* μ *is the length of the (padded) message.*

4.4 Security of Slae

In this section we show that the IND-CPA and INT-CTXT security of the authenticated encryption scheme Slae in the QS1 follows from the QS1 security of the underlying primitives SlFunc, SPrg, and SvHash. The proofs can be found in Appendix B.

IND-CPA *Security of* Slae. IND-CPA security follows from SlFunc and SPrg being a secure PRF and PRG, respectively. Theorem 9 first shows that SlFunc and SPrg yield SlEnc being IND-CPA-secure while Theorem 10 then establishes the IND-CPA security of Slae.

Theorem 9. *Let* SlFunc *be a pseudorandom function and* SPrg *a pseudorandom generator. Let further* SlEnc *be the symmetric key encryption scheme constructed from* SlFunc *and* SPrg *as shown in Fig. 2. For any quantum adversary $\mathcal{A}$, making q_{Enc} queries to its encryption oracle, against the* IND-CPA *security there exist adversaries $\mathcal{A}_{prf}$ and $\mathcal{A}_{prg}$ against* SlFunc *and* SPrg*, respectively, such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\textsc{SlEnc}}(\mathcal{A}) \leq 2\,\mathbf{Adv}^{\mathsf{PRF}}_{\textsc{SlFunc}}(\mathcal{A}_{prf}) + 2q\,\mathbf{Adv}^{\mathsf{PRG}}_{\textsc{SPrg}}(\mathcal{A}_{prg})\,.$$

Theorem 10. *Let* SlEnc *be the symmetric key encryption scheme and* SlMac *be a MAC. Let further* Slae *be the authenticated encryption scheme constructed from* SlEnc *and* SlMac *as shown in Fig. 2. For any quantum adversary $\mathcal{A}$, making q_{Enc} queries to its encryption oracle, against the* IND-CPA *security there exists an adversary $\mathcal{A}_{se}$, such that*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\textsc{Slae}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\textsc{SlEnc}}(\mathcal{A}_{se})\,.$$

INT-CTXT *Security of* Slae. The INT-CTXT security follows from SlFunc being a secure PRF and SvHash being a collision-resistant hash function. In Theorem 11, we show that both yield a SUF-CMA-secure MAC SlMac. Subsequently, Theorem 12 shows that the SUF-CMA security of SlMac ensures INT-CTXT security of Slae.

Theorem 11. *Let* SlFunc *be a function and* SvHash *a hash function. Let further* SlMac *be the MAC constructed from* SlFunc *and* SvHash *as shown in Fig. 2. For any quantum adversary $\mathcal{A}$, making q_T queries to its tagging oracle and q_F to its forge oracle, against the* SUF-CMA *security there exist adversaries $\mathcal{A}_{prf}$ and $\mathcal{A}_{hash}$ against* SlFunc *and* SvHash*, respectively, such that*

$$\mathbf{Adv}^{\mathsf{SUF\text{-}CMA}}_{\textsc{SlMac}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{PRF}}_{\textsc{SlFunc}}(\mathcal{A}_{prf}) + \mathbf{Adv}^{\mathsf{CR}}_{\textsc{SvHash}}(\mathcal{A}_{hash}) + \frac{q_F}{2^{\tau}}\,.$$

Theorem 12. *Let* SlEnc *be the symmetric key encryption scheme and* SlMac *be a MAC. Let further* Slae *be the authenticated encryption scheme constructed from* SlEnc *and* SlMac *as shown in Fig. 2. For any quantum adversary $\mathcal{A}$, making q_E queries to its encryption oracle and q_F queries to its forge oracle, against the* INT-CTXT *security there exists an adversary $\mathcal{A}_{mac}$, such that*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\textsc{Slae}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{SUF\text{-}CMA}}_{\textsc{SlMac}}(\mathcal{A}_{mac})\,.$$

5 Quantum (QS2) Security

In this section we study the security of Slae in the QS2 setting, where both the adversary and the challenger are quantum. Unlike the QS1 setting, the QS2 setting comes with several security notions. We analyse Slae, or even more precisely its encryption component SlEnc, with respect to the quantum security notions put forward in [12,28,43] providing positive and negative results.

5.1 QS2 Security Notions for SKE

Unlike the QS1 setting, there are several notions in the QS2 setting for encryption schemes. The first notion, called IND-qCPA, was presented by Boneh and Zhandry [12]. This notion allows the adversary superposition queries in the learning (qCPA) phase, while its challenge (IND) phase is restricted to classical queries. They further showed that simply allowing a quantum indistinguishability phase results in an unachievable security notion, called fqIND-CPA. More precisely, they consider a left-or-right oracle which performs the following

$$\sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0\right\rangle \left|x_1\right\rangle \left|y\right\rangle \mapsto \sum_{x_0,x_1,y} \alpha_{x_0,x_1,y} \left|x_0\right\rangle \left|x_1\right\rangle \left|y \oplus \mathtt{Enc}(\mathsf{K}, x_b)\right\rangle .$$

This operator entangles the ciphertext register with one of the message registers. Boneh and Zhandry show how this entanglement can be exploited to determine the bit b, irrespectively of the underlying encryption scheme.

Later, Gagliardoni et al. [28] and Mossayebi and Schack [43] provided security notions which allow the challenge (IND) phase to be quantum while not suffering from the impossibility result from [12].

An exhaustive study of QS2 security notions for encryption schemes is given by Carstens et al. [14]. Their study includes the aforementioned notions, along with many variants differing in the number of queries during challenge resp. learning phase. They show, surprisingly, that the notions do not form a strict hierarchy. Instead, the notions by Gagliardoni et al. [28] and Mossayebi and Schack [43] are incomparable but, together, imply all other notions. To ensure security in the QS2 setting, schemes have to be analysed with respect to both of these notions.

Nonce-Respecting Adversaries in the QS2 Setting. Another question that arises for the security of SLAE, deals with the nonce selection. Typically, adversaries are assumed to be nonce-respecting, meaning that they never repeat a nonce. While this is well defined in both the classical as well as QS1 setting, there is no definition for such adversaries in the QS2 setting. Kaplan et al. [36] mention this problem and sidestep it by letting the game pick the nonce at random. Thus, they essentially switch to the weaker IV setting which is well-studied in the classical setting. In our adapted security notions, we let the adversary submit a nonce register along with its message(s). We observe that it is not necessary to observe nonces in superposition since all QS2 notions for encryption schemes [12,15,28,29,43] consider the randomness (in case of SLAE the nonce) to be classical.[4] To comply with this, we let the challenger measure the nonce register, thus ensuring a classical nonce, and reject a query if a nonce repeats.

5.2 Left-or-Right Security of SLENC

The notion by Gagliardoni et al. [28] follows a left-or-right approach, similar to the one by Boneh and Zhandry [12], in which the adversary submits two

[4] The same applies to QS2 notions for MACs and signatures [1,11,12,25].

messages (possibly in superposition) and receives the encryption of one of the two. The main difference is that Gagliardoni et al. use type-2 operators which operate directly on the register (instead of XORing the output to a separate output register). These operators are more powerful than the corresponding type-1 operator and they can only be realised for functions that are reversible. Type-2 operators were first studied by Kashefi et al. [38] and have further been studied by Carstens et al. [14] for symmetric key encryption and by Gagliardoni et al. [29] for public key encryption.

More formally those operators can be formalised as follows. Let $\mathcal{F}\colon \{0,1\}^n \to \{0,1\}^n$ be a function. The type-1 operator for $\mathcal{F}$ is the unitary $U_{\mathcal{F}}^{(1)}$ that does the following

$$\sum_{x,y\in\{0,1\}^n} \alpha_{x,y} \left|x\right\rangle \left|y\right\rangle \mapsto \sum_{x,y\in\{0,1\}^n} \alpha_{x,y} \left|x\right\rangle \left|y \oplus \mathcal{F}(x)\right\rangle \ .$$

Observe that the realisation of $U_{\mathcal{F}}^{(1)}$ is efficient if $\mathcal{F}$ can be realised efficiently [46]. The type-2 operator for $\mathcal{F}$ is the unitary $U_{\mathcal{F}}^{(2)}$ that does the following

$$\sum_{x\in\{0,1\}^n} \alpha_x \left|x\right\rangle \mapsto \sum_{x} \alpha_x \left|\mathcal{F}(x)\right\rangle \ .$$

A realisation of a type-2 operator is, unlike for type-1 operators, not straightforward. Kashefi et al. [38] show that they can be realised using type-1 operators for both $\mathcal{F}$ and $\mathcal{F}^{-1}$. Gagliardoni et al. [28] use this to show that type-2 operators for symmetric key encryption schemes can be realised using type-1 operators for encryption and decryption (cf. Fig. 6).

Using type-2 operators, Gagliardoni et al. [28] bypass the impossibility result by Boneh and Zhandry [12]. Since the adversary only receives a ciphertext register, it can not exploit the entanglement between registers as was the case for fqIND-CPA.

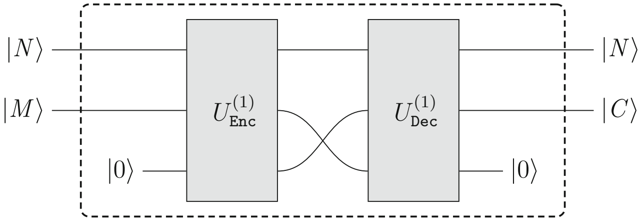

Fig. 6. Circuit for realising the type-2 operator $U^{(2)}_{\mathtt{Enc}}$ using type-1 operators $U^{(1)}_{\mathtt{Enc}}$ and $U^{(1)}_{\mathtt{Dec}}$ for `Enc` and `Dec`, respectively.

Below we define LoR-qIND security. This is the notion given in [28] restricted to a single challenge and no learning queries. The difference is that our notion allows the adversary to specify a register containing the nonce used for encryption. To ensure the usage of classical randomness, we let the challenger measure this register. We restrict ourselves to the weaker LoR-qIND notion, since we show below that SLAE does not even achieve this notion. Extension to the stronger LoR-qINDqCPA (allowing multiple challenges and learning queries) is straightforward by giving the adversary oracle access to a left-or-right oracle and a learning oracle implementing the type-2 encryption operator. The nonce-respecting property is ensured by letting the challenger reject queries for which the measurement of the nonce register yields an already measured nonce.

Definition 13. *Let* $\Sigma = (\texttt{Enc}, \texttt{Dec})$ *be symmetric key encryption scheme and the security game* LoR-qIND *be defined as in Fig. 7. For any adversary* $\mathcal{A}$ *we define its* LoR-qIND *advantage as*

$$\mathbf{Adv}_{\Sigma}^{\text{LoR-qIND}}(\mathcal{A}) = \left|2 \Pr[\text{LoR-qIND}^{\mathcal{A}} \to 1] - 1\right| .$$

```
LoR-qIND
$\mathsf{K} \leftarrow_{\$} \mathcal{K}$
implement $U_{\texttt{Enc}}^{(2)}$ using $\mathsf{K}$
$b \leftarrow_{\$} \{0,1\}$
$|N\rangle_N, |\varphi_0\rangle_M, |\varphi_1\rangle_M \leftarrow_{\$} \mathcal{A}_1()$
Measure register $|N\rangle_N$
trace out $|\varphi_{1-b}\rangle$
$|\psi\rangle \leftarrow U_{\texttt{Enc}}^{(2)}(|N\rangle_N |\varphi_b\rangle_M)$
$b' \leftarrow_{\$} \mathcal{A}_2(|\psi\rangle)$
return $(b' = b)$
```

Fig. 7. Security notion LoR-qIND following [28].

The following theorem shows that the sponge-based encryption scheme SLENC is not LoR-qIND-secure. The attack uses a Hadamard distinguisher, following the one given in [28], that exploits the quantum insecurity of the one-time pad approach. The proof details can be found in Appendix C.

Theorem 14. *Let* SLENC *be the sponge-based encryption scheme displayed in Fig. 2 with message space* $\{0,1\}^{\mu}$. *Then there exist an adversary* $\mathcal{A}$ *such that*

$$\mathbf{Adv}_{\text{SLENC}}^{\text{LoR-qIND}}(\mathcal{A}) = 1 - \frac{1}{2^{\mu}} .$$

Observe that there is no security notion for AEAD schemes using type-2 operators. Both [28] and [29] only focus on encryption schemes. The obvious question is whether the MAC can be implemented using a type-2 operator. Regardless of this, we point out that the attack does not necessarily extend to Slae. The reason is that the register containing the tag will be entangled which thwarts an attack by simply discarding the tag.

Note that the same attack applies to the encryption scheme underlying the sponge-based AEAD schemes ISAP [23] and its successor ISAP v2.0 [22].

5.3 Real-or-Random Security of SlEnc

The notion by Mossayebi and Schack [43] follows a real-or-random approach, where the adversary submits only a single message (possibly in superposition) and receives back the message along with a ciphertext. The ciphertext is either the encryption of the submitted message or of the permuted message using a permutation picked at random. Usage of the permutation ensures that the number of messages in superposition is the same for both the submitted and permuted message. Mossayebi and Schack [43] also defined the corresponding security with respect to chosen ciphertext attacks. The relevance of this notion is questionable, as it assumes non-cheating adversaries, that do not try to decrypt the challenge ciphertext with its decryption oracle.

In this notion, there is only a single message register that will always be entangled with the ciphertext register. This bypasses the impossibility result by Boneh and Zhandry [12].

```
RoR-qIND
$\mathsf{K} \leftarrow_{\$} \mathcal{K}$
implement $U^{(1)}_{\mathtt{Enc}}$ using $\mathsf{K}$
$b \leftarrow_{\$} \{0,1\}$
$\pi \leftarrow_{\$} \mathcal{P}(\{0,1\}^{\mu})$
$|N\rangle_N \, |M\rangle_M \, |C\rangle_C \leftarrow_{\$} \mathcal{A}_1()$
Measure register $|N\rangle_N$
if $b = 0$
    $|\psi\rangle \leftarrow U^{(1)}_{\mathtt{Enc}}(|N\rangle_N \, |M\rangle_M \, |C\rangle_C)$
if $b = 1$
    $|\psi\rangle \leftarrow ((id \otimes \pi^{-1} \otimes id) \circ U^{(1)}_{\mathtt{Enc}} \circ (id \otimes \pi \otimes id))(|N\rangle_N \, |M\rangle_M \, |C\rangle_C)$
$b' \leftarrow_{\$} \mathcal{A}_2(|\psi\rangle)$
return $(b' = b)$
```

Fig. 8. Security notion RoR-qIND following [43].

Below we define RoR-qIND security, where the adversary is restricted to a single challenge query and no learning query, again, extended by letting the adversary send a register with the nonce that is measured by the challenger.

Extension to RoR-qINDqCPA security works by providing the adversary a real-or-random challenge oracle and a learning oracle and reject queries where (measured) nonces repeat (Fig. 9).

Definition 15. *Let $\Sigma = (\mathtt{Enc}, \mathtt{Dec})$ be a symmetric key encryption scheme and the security game* RoR-qIND *be defined as in Fig. 8. For any adversary $\mathcal{A}$ we define its* RoR-qIND *advantage as*

$$\mathbf{Adv}_{\Sigma}^{\text{RoR-qIND}}(\mathcal{A}) = \left|2 \Pr[\text{RoR-qIND}^{\mathcal{A}} \to 1] - 1\right| .$$

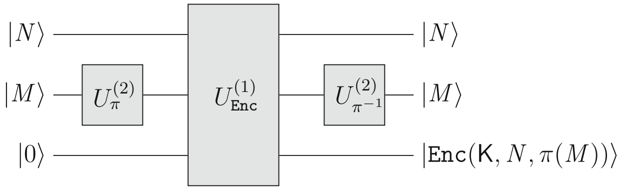

Fig. 9. Circuit for real-or-random security notion. The permutation π is applied if $b = 1$.

The following theorem shows that the sponge-based encryption scheme SlEnc is not RoR-qIND-secure. The attack follows [15] exploiting the outcome of a measurement in the Hadamard basis on two entangled registers. The full proof details can be found in the full version of the paper [34].

Theorem 16. *Let* SlEnc *be the sponge-based encryption scheme displayed in Fig. 2. Then there exist an adversary $\mathcal{A}$ such that*

$$\mathbf{Adv}_{\text{SlEnc}}^{\text{RoR-qIND}}(\mathcal{A}) = \frac{1}{2} .$$

5.4 IND-qCPA Security of Slae and FGHF'

In Sect. 5.1, we have discussed various different security notions for symmetric key encryption schemes in the QS2 setting. So far we have shown that SlEnc is neither LoR-qIND nor RoR-qIND secure. Observe that the attacks also apply to the generic construction FGHF', as the weakness lies in the one-time pad (OTP) approach exploiting an inherent insecurity of the OTP against quantum attackers.

Observe that both the generic FGHF' construction as well as Slae are stream ciphers. Following the results by Anand et al. [4], we obtain that both constructions are IND-qCPA secure which is a direct consequence from the established IND-CPA security in the QS1 sense.

6 Conclusion

In this work we have given both positive and negative results for the security of the sponge-based AEAD scheme Slae. On the one hand, we have shown that Slae as well as the underlying core primitives are post-quantum secure. On the other hand, we have shown that their quantum security is not fully clear yet. While Slae, as well as the generic FGHF' construction, are easily seen to be not quantum secure for notions that allow challenge queries by the adversary to be in superposition, its quantum security with respect to IND-qCPA is still open. More precisely, we argued that its IND-qCPA security reduces to the quantum security of the underlying function SlFunc via the generic FGHF' construction.

In the realm of quantum security, it is open to analyse the quantum security of the sponge-based function SlFunc as well as addressing the quantum unforgeability of Slae and its underlying MAC SlMac. The reason is that the landscape of quantum unforgeability notions is still unclear as the existing notions [1,12,25,30] suffer from some drawbacks that allow for intuitive forgeries that are not covered by the notions.

Regarding the post-quantum security of Slae, one can investigate whether tighter bounds can be achieved. Generally, our bounds establish for the first time post-quantum security for the AEAD scheme and the underlying primitives but they are rather conservative and there might be room for improvements. For example, for SlFunc one may be able to use the semi-classical variant of the O2H lemma developed by Ambainis et al. [3] and for SPrg one may get tighter bounds by using the doubled-sided O2H lemma by Bindel et al. [9]. One can also consider an adaptive version, where the random oracle is reprogrammed only on the points that the adversary queries to its classical oracle.

Acknowledgements. We thank anonymous reviewers for their valuable comments on an earlier draft of this work. This work was funded by the Deutsche Forschungsgemeinschaft (DFG) – SFB 1119 – 236615297.

A Additional Preliminaries

A.1 Authenticated Encryption

We begin with a definition of authenticated encryption schemes with associated data [6,47].

Definition 17. *An* authenticated encryption scheme with associated data (AEAD) $\mathtt{AEAD} = (\mathtt{Enc}, \mathtt{Dec})$ *is a pair of efficient algorithms associated with key space* $\mathcal{K}$*, nonce space* $\mathcal{N}$*, associated-data space* $\mathcal{H}$*, message space* $\mathcal{M}$*, and ciphertext space* $\mathcal{C}$ *such that:*

- *The deterministic encryption algorithm* $\mathtt{Enc}\colon \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{M} \rightarrow \mathcal{C}$ *takes as input a secret key* K*, a nonce* N*, associated data* A*, and a message* M*. It outputs a ciphertext* C*.*

- *The deterministic decryption algorithm* $\texttt{Dec}\colon \mathcal{K} \times \mathcal{N} \times \mathcal{H} \times \mathcal{C} \to \mathcal{M} \cup \{\bot\}$ *takes as input a secret key* K*, a nonce* N*, associated data* A*, and a ciphertext* C*. It outputs a message* M *or* $\bot$ *indicating an invalid ciphertext.*

We say that an AEAD scheme is correct*, if for all* $\mathsf{K} \in \mathcal{K}$*,* $N \in \mathcal{N}$*,* $A \in \mathcal{H}$ *and* $M \in \mathcal{M}$*, it holds that* $\texttt{Dec}(\mathsf{K}, N, A, \texttt{Enc}(\mathsf{K}, N, A, M)) = M$*.*

Throughout this work, we consider $\mathcal{K} = \{0,1\}^k$, $\mathcal{N} = \{0,1\}^\nu$, $\mathcal{H} = \{0,1\}^\alpha$, $\mathcal{M} = \{0,1\}^\mu$, and $\mathcal{C} = \{0,1\}^\gamma$.

Security of an AEAD scheme now demands that an adversary cannot distinguish encryptions of equal-length messages which corresponds to the usual CPA-security notion of encryption schemes. The formal description of the game can be found on the left side of Fig. 10. Additionally, security also demands that the adversary is not able to forge further valid ciphertexts which corresponds to an integrity notion on the ciphertext level. The formal description of the games can be found on the right side of Fig. 10.

```
IND-CPA
  $\mathsf{K} \leftarrow_\$ \mathcal{K}$
  $\mathcal{Q} \leftarrow \emptyset$
  $b \leftarrow_\$ \{0,1\}$
  $b' \leftarrow_\$ \mathcal{A}^{\mathsf{Enc}(\mathsf{K},\cdot,\cdot,\cdot,\cdot)}$
  return $(b' = b)$

$\mathsf{Enc}(\mathsf{K}, N, A, M_0, M_1)$
  if $|M_0| \neq |M_1|$ then
    return $\bot$
  if $N \in \mathcal{Q}$ then
    return $\bot$
  $C \leftarrow \mathsf{Enc}(\mathsf{K}, N, A, M_b)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{N\}$
  return $C$
```

```
INT-CTXT
  $\mathsf{K} \leftarrow_\$ \mathcal{K}$
  $\mathcal{Q} \leftarrow \emptyset$
  $\mathsf{win} \leftarrow 0$
  $\mathcal{A}^{\mathsf{Enc}(\mathsf{K},\cdot,\cdot,\cdot),\mathsf{Forge}(\mathsf{K},\cdot,\cdot,\cdot)}$
  return $\mathsf{win}$

$\mathsf{Enc}(\mathsf{K}, N, A, M)$
  if $(N, \cdot, \cdot) \in \mathcal{Q}$ then
    return $\bot$
  $C \leftarrow \mathsf{Enc}(\mathsf{K}, N, A, M)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(N, A, C)\}$
  return $C$

$\mathsf{Forge}(\mathsf{K}, N, A, C)$
  if $(N, A, C) \in \mathcal{Q}$ then
    return $\bot$
  $d \leftarrow \mathsf{Dec}(\mathsf{K}, N, A, C)$
  if $d \neq \bot$ then
    $\mathsf{win} \leftarrow 1$
  return $d$
```

Fig. 10. Security games for AEAD.

Definition 18. *Let* $\texttt{AEAD}$ *be an authenticated encryption scheme with associated data.*

- *For an adversary* $\mathcal{A}$*, making* q_E *queries to its encryption oracle, we define its* IND-CPA *advantage as*

$$\mathbf{Adv}^{\mathsf{IND\text{-}CPA}}_{\texttt{AEAD}}(\mathcal{A}) = 2 \Pr[\text{IND-CPA}^{\mathcal{A}} \to 1] - 1 .$$

- *For an adversary* $\mathcal{A}$*, making* q_E *and* q_F *queries to its encryption oracle and forge oracle, respectively, we define its* INT-CTXT *advantage as*

$$\mathbf{Adv}^{\mathsf{INT\text{-}CTXT}}_{\texttt{AEAD}}(\mathcal{A}) = \Pr[\text{INT-CTXT}^{\mathcal{A}} \to 1] .$$

Symmetric Key Encryption. Observe that the definition of a symmetric key encryption (SKE) scheme is very close to the given about AEAD. Note that one can obtain a SKE scheme by analogously defining an encryption scheme which does not admit associated data as a part of its input in comparison to Definition 17.

Usually one defines CPA-security of a SKE scheme. Here the formalisation is again very close to the description in Fig. 10 with the modification of not including the associated data as an input to the encryption oracle.

A.2 Message Authentication Code

Next we will provide the basic definition of a message authentication code.

Definition 19. *A* message authentication code (MAC) $\mathtt{MAC} = (\mathtt{Tag}, \mathtt{Vfy})$ *is a pair of efficient algorithms associated with key space $\mathcal{K}$ and domain space $\mathcal{X}$ such that:*

- *The deterministic tagging algorithm* $\mathtt{Tag}\colon \mathcal{K} \times \mathcal{X} \to \{0,1\}^\tau$ *takes as input a key* K *and an element X. It returns a tag T of size $\{0,1\}^\tau$.*
- *The deterministic verification algorithm* $\mathtt{Vfy}\colon \mathcal{K} \times \mathcal{X} \times \{0,1\}^\tau \to \{0,1\}$ *takes as input a key* K*, an element X, and a tag T and outputs* 1 *indicating that the input is valid, or otherwise* 0*.*

We say that a MAC scheme is correct*, if for all* $\mathsf{K} \in \mathcal{K}$ *and any admissible input $X \in \mathcal{X}$, it holds that* $\mathtt{Vfy}(\mathsf{K}, X, \mathtt{Tag}(\mathsf{K}, X)) = 1$*.*

```
SUF-CMA
  $\mathsf{K} \leftarrow_{\$} \mathcal{K}$
  $\mathcal{Q} \leftarrow \emptyset$
  $\mathsf{win} \leftarrow 0$
  $\mathcal{A}^{\mathsf{Tag}(\mathsf{K},\cdot),\mathsf{Forge}(\mathsf{K},\cdot,\cdot)}$
  return $\mathsf{win}$
```

```
$\mathtt{Tag}(\mathsf{K}, M)$
  $T \leftarrow \mathtt{Tag}(\mathsf{K}, M)$
  $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(M, T)\}$
  return $T$
```

```
$\mathsf{Forge}\ (\mathsf{K},\ M,\ T)$
  if $(M, T) \in \mathcal{Q}$ then
    return $\perp$
  $d \leftarrow \mathtt{Vfy}(\mathsf{K}, M, T)$
  if $d = 1$ then
    $\mathsf{win} \leftarrow 1$
  return $d$
```

Fig. 11. Security game for MAC.

Definition 20. *Let* $\mathtt{MAC}$ *be a message authentication code. We define the* SUF-CMA *advantage of an adversary $\mathcal{A}$ making at most q_T queries to its tag oracle and q_F many queries to its forge oracle as*

$$\mathbf{Adv}^{\mathsf{SUF\text{-}CMA}}_{\mathtt{MAC}}(\mathcal{A}) = \Pr[\text{SUF-CMA}^{\mathcal{A}} \to 1]\,,$$

where the respective game is depicted in Fig. 11.

A.3 Hash Function

In this section, we simply review the collapsing property of hash functions [54] in the formalisation of [18].

Definition 21. *For algorithms $\mathcal{A}$ and $\mathcal{B}$, consider the following games given in Fig. 12. There are quantum registers S and M, and $\mathcal{M}(M)$ is a measurement of M in the computational basis.*

For a set $\mathbf{m}$, we call an adversary $(\mathcal{A}, \mathcal{B})$ valid on $\mathbf{m}$ for H^{O} if and only if $\Pr[H^{\mathsf{O}}(m) = h \wedge m \in \mathbf{m}]$ when we run $(S, M, h) \leftarrow \mathcal{A}^{\mathsf{O}}()$ and measure M in the computational basis as m.

A function H is collapsing on $\mathbf{m}$ if and only if for any quantum-polynomial-time adversary $(\mathcal{A}, \mathcal{B})$ that is valid for H^{O} on $\mathbf{m}$ and $|\Pr[b = 1\colon \mathsf{Game}_1] - \Pr[b = 1\colon \mathsf{Game}_2]|$ is negligible.

Game_1	Game_2
$(S, M, h) \leftarrow \mathcal{A}^{\mathsf{O}}()$	$(S, M, h) \leftarrow \mathcal{A}^{\mathsf{O}}()$
$m \leftarrow \mathcal{M}(M)$	
$b \leftarrow \mathcal{B}^{\mathsf{O}}(S, M)$	$b \leftarrow \mathcal{B}^{\mathsf{O}}(S, M)$

Fig. 12. Collapsing games.

B QS1 Proofs

B.1 Proof of Theorem 8

Proof. The above collision resistance bound can be obtained from a combination of results from Czajkowski et al. [17] and Unruh [52] with a slight modification that stems from the way SvHash is constructed. Observe that the small modification is due to the interpretation that we consider a sponge-based hash function with the capacity being reduced by one bit and hence the rate being increased by one bit. We take care of this one bit loss when applying the following results.

A crucial property in the realm of hash functions in the post-quantum setting is called the collapsing property which is a strengthening of collision resistance and Unruh has showed in [52,54] that if a hash function is collapsing then this also implies that it is quantum collision resistant. Additionally, Czajkowski et al. [17,18] showed that if the underlying function of the sponge construction is a random transformation then the sponge construction is collapsing. Being equipped with their result, we can derive the required bound for our setting.

We will follow the proof strategy put forward by Czajkowski et al. [17, Theorem 33] to show that the sponge construction is collapsing. This requires to show that the inner state $\hat{S}$ is collapsing in the absorbing phase while the outer state

$\bar{S}$ is collapsing in the squeezing phase and that there are no zero-preimages in the inner state $\hat{S}$. Then using [52, Lemma 25] provides us with the implication that the sponge construction is then also collision resistant. It now remains to apply the above strategy appropriately to derive the bound.

We have that $l = \left\lceil \frac{\mu}{r} \right\rceil$ and by [17, Theorem 33], we know that the collapsing advantage is bounded by $\sqrt{\varepsilon_1} + l \cdot \varepsilon_2 + \varepsilon_3$, where ε_1 corresponds to the probability of finding zero-preimages, ε_2 corresponds to the collapsing advantage of the inner state and ε_3 corresponds to the collapsing advantage of the outer state, respectively. By applying [17, Lemma 19], we obtain that $\varepsilon_1 \leq (q+1)^2 2^{-c+4}$. By a simple combination of [17, Lemma 32] and [52, Theorem 38], we can derive $\varepsilon_2 \leq q^3 \left(\frac{\delta' + 324}{2^{c-1}}\right) + 7\delta\sqrt{\frac{3(q+4)^3}{2^c}}$ and $\varepsilon_3 \leq q^3 \left(\frac{\delta' + 324}{2^{w+1}}\right) + 7\delta\sqrt{\frac{3(q+4)^3}{2^{w+2}}}$ where both δ and δ' are non-zero constants. Then by [52, Lemma 25], we have a tight reduction from collapsing to collision resistance and hence the same bound holds for the collision resistance of the sponge construction. □

B.2 Proof of Theorem 9

Proof. The proof can be obtained from [21] by dropping everything related to the leakage setting. It proceeds in two game hops. The first game hop replaces the function SlFunc by a random function which can be straightforwardly bound by the PRF advantage of SlFunc. More precisely, $\mathcal{A}_{prf}$ uses its own oracle for everything related to SlFunc while simulating SPrg using (classical) queries to the random oracle ρ. All (quantum) queries by $\mathcal{A}$ to ρ are simply forwarded by $\mathcal{A}_{prf}$, as are the responses back to $\mathcal{A}$.

The second game hop replaces the output of SPrg by a random output. A standard hybrid argument [26] shows that this can be bound by the security of SPrg. The reduction $\mathcal{A}_{prg}$ picks a random query of $\mathcal{A}$ to its encryption oracle, where it uses its own input (either the output of SPrg or a random bit string) to encrypt the message. Prior queries are answered by XORing random bit string to the message while subsequent queries are answered by simulating SPrg using (classical) queries to ρ. All (quantum) queries by $\mathcal{A}$ to ρ are simply forwarded by $\mathcal{A}_{prg}$, as are the responses back to $\mathcal{A}$.

The resulting game yields identically distributed ciphertexts, irrespectively of the message. The factor 2 accounts for doing the game hops for both cases $b = 0$ and $b = 1$. □

B.3 Proof of Theorem 10

Proof. The proof proceeds by a simple reduction. In more detail, the reduction $\mathcal{A}_{se}$ picks a key for the MAC SlMac. For every query to the encryption oracle by $\mathcal{A}$, $\mathcal{A}_{se}$ invokes its own encryption oracle and locally computes the tag of the ciphertext using (classical) queries to ρ before sending the ciphertext and the tag back to $\mathcal{A}$. Every (quantum) query by $\mathcal{A}$ to ρ is simply forwarded by $\mathcal{A}_{se}$. □

B.4 Proof of Theorem 11

Proof. We assume that all messages queried by $\mathcal{A}$ result in different hash values, otherwise, we obtain a simple reduction $\mathcal{A}_{hash}$ from the collision resistance of SvHash.

Then the proof proceeds by a game hop in which SlFunc is replaced by a random function. The reduction $\mathcal{A}_{prf}$ will invoke its own function to simulate the tagging and verification of SlMac and (classical) queries to ρ to evaluate SvHash. Every (quantum) query to ρ by $\mathcal{A}$ is simply forwarded by $\mathcal{A}_{prf}$.

The resulting game is bound by a simple counting argument that $\mathcal{A}$ predicts the output of a random function. □

B.5 Proof of Theorem 12

Proof. The reduction $\mathcal{A}_{mac}$ picks a key for the symmetric key encryption scheme SlEnc. For every query to the encryption oracle by $\mathcal{A}$, $\mathcal{A}_{mac}$ locally computes the ciphertext using (classical) queries to ρ and obtains the tag using its own tagging oracle. It then sends the ciphertext and the tag back to $\mathcal{A}$. For every forgery attempt by $\mathcal{A}$, $\mathcal{A}_{mac}$ queries the ciphertext and the tag as its own forgery attempt. If the tag verifies correctly, $\mathcal{A}_{mac}$ locally decrypts the ciphertext using the sampled key and (classical) queries to ρ and sends the message back to $\mathcal{A}$, otherwise, i.e., if the tag was invalid, $\mathcal{A}_{mac}$ simply returns $\perp$ to $\mathcal{A}$. Every (quantum) query by $\mathcal{A}$ to ρ is simply forwarded by $\mathcal{A}_{se}$. □

C QS2 Proofs

C.1 Proof of Theorem 14

Proof. We construct the following adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. It picks a nonce $N \leftarrow_\$ \{0,1\}^\nu$ and prepares the states $|\varphi_0\rangle = H\,|0^\mu\rangle = |+\rangle^{\otimes\mu}$ and $|\varphi_1\rangle = |0^\mu\rangle$. It outputs the state

$$|\varphi\rangle = |N\rangle \otimes |+\rangle^{\otimes\mu} \otimes |\varphi_1\rangle \ .$$

Upon receiving the state $|\psi\rangle$, $\mathcal{A}_2$ applies the Hadamard operator to it and measures the register. If the measurement output is 0^μ, $\mathcal{A}_0$ outputs 0, otherwise, it outputs 1.

Before analysing the different cases, note that measuring the nonce register as well as tracing out one of the message registers does not affect the other registers as they are all unentangled. Let us now start with the case distinctions.

If $b = 0$, the game encrypts the left message, i.e., the state

$$|\varphi\rangle = H\,|0^\mu\rangle = |+\rangle^{\otimes\mu} = \frac{1}{\sqrt{2^\mu}}\left(\sum_{x\in\{0,1\}^\mu} |x\rangle\right) .$$

$\mathcal{A}_2$ receives the state

$$|\psi\rangle = \frac{1}{\sqrt{2^\mu}} \left(\sum_{x\in\{0,1\}^\mu} |N\rangle \, |x \oplus \textsc{SPrg}(\textsc{SlFunc}(\mathsf{K}, N))\rangle \right) = |\varphi\rangle \ ,$$

i.e., the state $|\varphi\rangle$ is left unchanged. Application of the Hadamard operator therefore yields the state $|0^\mu\rangle$, for which the measurement outcome is 0^μ with probability 1. Thus we get

$$\Pr[\mathcal{A}^{\text{LoR-qIND}} \to 0 \,|\, b = 0] = 1 \,.$$

If $b = 1$, $\mathcal{A}_2$ receives the state

$$|\psi\rangle = |\mathtt{Enc}(\mathsf{K}, N, 0^\mu)\rangle = |0^\mu \oplus \textsc{SPrg}(\textsc{SlFunc}(\mathsf{K}, N))\rangle \ .$$

Application of the Hadamard operator yields

$$H\,|\psi\rangle = \frac{1}{\sqrt{2^\mu}} \left(\sum_{x\in\{0,1\}^\mu} (-1)^{x\cdot\textsc{SPrg}(\textsc{SlFunc}(\mathsf{K},N))} \, |x\rangle \right) \ .$$

Measurement yields a random $x \in \{0,1\}^\mu$. Since $\mathcal{A}_2$ outputs 0 if and only if the measurement yields 0^μ, we obtain

$$\Pr[\mathcal{A}^{\text{LoR-qIND}} \to 0 \,|\, b = 1] = \frac{1}{2^\mu} \,.$$

Collecting everything yields

$$\begin{aligned}\mathbf{Adv}^{\text{LoR-qIND}}(\mathcal{A}) &= \left|\Pr[\mathcal{A}^{\text{LoR-qIND}} \to 0 \,|\, b = 0] - \Pr[\mathcal{A}^{\text{LoR-qIND}} \to 0 \,|\, b = 1]\right| \\ &= 1 - \frac{1}{2^\mu} \,.\end{aligned}$$

□

References

1. Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 788–817. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_27
2. Alagic, G., Russell, A.: Quantum-secure symmetric-key cryptography based on hidden shifts. In: Coron, J.-S., Nielsen, J.B. (eds.) EUROCRYPT 2017, Part III. LNCS, vol. 10212, pp. 65–93. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-56617-7_3
3. 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 (2019). https://doi.org/10.1007/978-3-030-26951-7_10

4. Anand, M.V., Targhi, E.E., Tabia, G.N., Unruh, D.: Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 44–63. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_4
5. Barwell, G., Martin, D.P., Oswald, E., Stam, M.: Authenticated encryption in the face of protocol and side channel leakage. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part I. LNCS, vol. 10624, pp. 693–723. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70694-8_24
6. Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_41
7. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) ACM CCS 1993, pp. 62–73. ACM Press, November 1993
8. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: Sponge functions. In: ECRYPT Hash Workshop (2007)
9. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: Hofheinz, D., Rosen, A. (eds.) TCC 2019, Part II. LNCS, vol. 11892, pp. 61–90. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-36033-7_3
10. 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, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_3
11. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 592–608. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_35
12. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_21
13. Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Yu., 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 (2019). https://doi.org/10.1007/978-3-030-34578-5_20
14. Carstens, T.V., Ebrahimi, E., Tabia, G.N., Unruh, D.: On quantum indistinguishability under chosen plaintext attack. Cryptology ePrint Archive, Report 2020/596 (2020). https://eprint.iacr.org/2020/596
15. Chevalier, C., Ebrahimi, E., Vu, Q.-H.: On security notions for encryption in a quantum world. Cryptology ePrint Archive, Report 2020/237 (2020). https://eprint.iacr.org/2020/237
16. Czajkowski, J.: Quantum indifferentiability of SHA-3. IACR Cryptology ePrint Archive 2021/192 (2021)
17. Czajkowski, J., Groot Bruinderink, L., Hülsing, A., Schaffner, C., Unruh, D.: Post-quantum security of the sponge construction. Cryptology ePrint Archive, Report 2017/771 (2017). https://eprint.iacr.org/2017/771

18. Czajkowski, J., Groot Bruinderink, L., Hülsing, A., Schaffner, C., Unruh, D.: Post-quantum security of the sponge construction. In: Lange, T., Steinwandt, R. (eds.) PQCrypto 2018. LNCS, vol. 10786, pp. 185–204. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-79063-3_9
19. 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 (2019). https://doi.org/10.1007/978-3-030-26951-7_11
20. Czajkowski, J., Majenz, C., Schaffner, C., Zur, S.: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428 (2019). https://eprint.iacr.org/2019/428
21. 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 (2019). https://doi.org/10.1007/978-3-030-34621-8_8
22. Dobraunig, C., et al.: ISAP v2.0. IACR Trans. Symm. Cryptol. **2020**(S1), 390–416 (2020)
23. Dobraunig, C., Eichlseder, M., Mangard, S., Mendel, F., Unterluggauer, T.: ISAP - towards side-channel secure authenticated encryption. IACR Trans. Symm. Cryptol. **2017**(1), 80–105 (2017)
24. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1. 2. Submission to the CAESAR Competition (2016)
25. Doosti, M., Delavar, M., Kashefi, E., Arapinis, M.: A unified framework for quantum unforgeability. CoRR, abs/2103.13994 (2021)
26. Fischlin, M., Mittelbach, A.: An overview of the hybrid argument. Cryptology ePrint Archive, Report 2021/088 (2021). https://eprint.iacr.org/2021/088
27. Gagliardoni, T.: Quantum security of cryptographic primitives. Ph.D. thesis, Darmstadt University of Technology, Germany (2017)
28. Gagliardoni, T., Hülsing, A., Schaffner, C.: Semantic security and indistinguishability in the quantum world. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part III. LNCS, vol. 9816, pp. 60–89. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_3
29. Gagliardoni, T., Krämer, J., Struck, P.: Quantum indistinguishability for public key encryption. In: Cheon, J.H., Tillich, J.-P. (eds.) PQCrypto 2021 2021. LNCS, vol. 12841, pp. 463–482. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81293-5_24
30. Garg, S., Yuen, H., Zhandry, M.: New security notions and feasibility results for authentication of quantum data. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part III. LNCS, vol. 10402, pp. 342–371. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63715-0_12
31. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th ACM STOC, pp. 212–219. ACM Press, May 1996
32. Hosoyamada, A., Sasaki, Yu.: Quantum Demiric-Selçuk meet-in-the-middle attacks: applications to 6-round generic feistel constructions. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 386–403. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_21
33. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Yu., Iwata, T.: Quantum chosen-ciphertext attacks against feistel ciphers. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 391–411. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12612-4_20

34. Janson, C., Struck, P.: Sponge-based authenticated encryption: security against quantum attackers. Cryptology ePrint Archive, Report 2022/139 (2022). https://eprint.iacr.org/2022/139
35. Jovanovic, P., Luykx, A., Mennink, B.: Beyond $2^{c/2}$ security in sponge-based authenticated encryption modes. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part I. LNCS, vol. 8873, pp. 85–104. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45611-8_5
36. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 207–237. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_8
37. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symm. Cryptol. **2016**(1), 71–94 (2016). https://tosc.iacr.org/index.php/ToSC/article/view/536
38. Kashefi, E., Kent, A., Vedral, V., Banaszek, K.: Comparison of quantum oracles. Phys. Rev. A **65**(5), 050304 (2002)
39. Krämer, J., Struck, P.: Leakage-resilient authenticated encryption from leakage-resilient pseudorandom functions. In: Bertoni, G.M., Regazzoni, F. (eds.) COSADE 2020. LNCS, vol. 12244, pp. 315–337. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68773-1_15
40. Kuchta, V., Sakzad, A., Stehlé, D., Steinfeld, R., Sun, S.-F.: Measure-rewind-measure: tighter quantum random oracle model proofs for one-way to hiding and CCA security. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part III. LNCS, vol. 12107, pp. 703–728. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_24
41. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round feistel cipher and the random permutation. In: ISIT 2010 (2010)
42. Kuwakado, H., Morii, M.: Security on the quantum-type even-mansour cipher. In: ISITA 2012 (2012)
43. Mossayebi, S., Schack, R.: Concrete security against adversaries with quantum superposition access to encryption and decryption oracles. CoRR, abs/1609.03780 (2016)
44. Namprempre, C., Rogaway, P., Shrimpton, T.: Reconsidering generic composition. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 257–274. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55220-5_15
45. National Institute of Standards and Technology. Lightweight cryptography standardization process (2015)
46. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th edn. Cambridge University Press, New York (2011)
47. Rogaway, P.: Authenticated-encryption with associated-data. In: Atluri, V. (ed.) ACM CCS 2002, pp. 98–107. ACM Press, November 2002
48. Rötteler, M., Steinwandt, R.: A note on quantum related-key attacks. Inf. Process. Lett. **115**(1), 40–44 (2015)
49. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: 35th FOCS, pp. 124–134. IEEE Computer Society Press, November 1994
50. Song, F.: A note on quantum security for post-quantum cryptography. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 246–265. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_15

51. Soukharev, V., Jao, D., Seshadri, S.: Post-quantum security models for authenticated encryption. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 64–78. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_5
52. Unruh, D.: Computationally binding quantum commitments. Cryptology ePrint Archive, Report 2015/361 (2015). https://eprint.iacr.org/2015/361
53. Unruh, D.: Revocable quantum timed-release encryption. J. ACM **62**(6), 1–76 (2015)
54. Unruh, D.: Computationally binding quantum commitments. In: Fischlin, M., Coron, J.-S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 497–527. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_18
55. Unruh, D.: Compressed permutation oracles (and the collision-resistance of sponge/SHA3). Cryptology ePrint Archive, Report 2021/062 (2021). https://eprint.iacr.org/2021/062
56. Yamakawa, T., Zhandry, M.: Classical vs quantum random oracles. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021, Part II. LNCS, vol. 12697, pp. 568–597. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77886-6_20
57. Zhandry, M.: How to construct quantum random functions. In: 53rd FOCS, pp. 679–687. IEEE Computer Society Press, October 2012
58. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part II. LNCS, vol. 11693, pp. 239–268. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26951-7_9