QCB is Blindly Unforgeable

Revised Paper

Jannis Leuther[0009−0007−0359−3680] and Stefan Lucks[0000−0003−4906−5131]

Bauhaus-Universität Weimar, Germany
<firstname>.<lastname>@uni-weimar.de

Abstract. QCB is a proposal for a post-quantum secure, rate-one authenticated encryption with associated data scheme (AEAD) based on classical OCB3 and ΘCB, which are vulnerable against a quantum adversary in the Q2 setting. The authors of QCB prove integrity under plus-one unforgeability, whereas the proof of the stronger definition of blind unforgeability has been left as an open problem. After a short overview of QCB and the current state of security definitions for authentication, this work proves blind unforgeability of QCB. Finally, the strategy of using tweakable block ciphers in authenticated encryption is generalised to a generic blindly unforgeable AEAD model.

Keywords: Post-Quantum Cryptography · QCB · Blind Unforgeability · AEAD · Symmetric Cryptography

Note: In an earlier version of this paper [18], a claim from [2] was repeated, that *blind unforgeability* (BU) implies *plus-one unforgeability* (PO) [8]. This claim, which would have indicated that our result is strictly stronger than the PO unforgeability result from [5], has been withdrawn in an updated version of [2]. This withdrawal does not undermine the contribution of our paper. On the contrary, both BU and PO seem to be important security notions of independent importance. As it turns out, QCB satisfies both notions: QCB is both PO unforgeable (proven in [5]) and BU unforgeable (proven in our paper).

1 Introduction

Motivation. As it stands, many cryptographic algorithms currently in use are weak or outright broken when challenged by an adversary using a quantum computer [23,21]. The security of asymmetric cryptography is especially affected, as mathematical problems like integer factorization or the discrete logarithm are hardly a challenge for quantum computers. For example, integer factorization

can be solved using Shor's algorithm [27,28] with an almost exponential speed-up compared to a classical computer. While many asymmetric cryptographic algorithms are broken by design, the impact on *post-quantum security* of most symmetric cryptographic schemes is expected to be less dramatic. However, there are symmetric-key constructions that are also vulnerable to certain quantum algorithms. Simon's algorithm can render many symmetric cryptographic modes that are secure in the classical sense broken in the quantum scenario [16,5].

The authors in [16] are able to show that, additionally to the *Even-Mansour* and 3-Round Feistel construction, the *Liskov-Rivest-Wagner* (LRW) construction is also insecure against a quantum adversary. Furthermore, they describe forgery attacks against currently standardized classical authentication and authenticated encryption modes like *CBC-MAC*, *PMAC*, *GMAC*, *GCM* and *OCB* and imply that some authentication and authenticated encryption schemes are quantumly insecure even with an underlying post-quantum secure block cipher [16, pp. 2-3].

Outline. After introducing the topic and defining notations, Section 2 briefly discusses the evolution and difference of unforgeability notions *plus-one unforgeability* and *blind unforgeability*. Next, Section 3 goes into detail about *QCB*, a recent proposal for a post-quantum secure AEAD mode based on the quantumly broken *OCB*. Section 4 follows up with a proof that blind unforgeability holds for QCB. Afterwards, a generic construction for blindly unforgeable AEAD schemes is given in Section 5. Section 6 concludes.

Preliminaries. Note that there exist symmetric and asymmetric encryption algorithms for privacy, and message authentication codes (MACs) to guarantee authenticity and integrity of communication. The combination of symmetric encryption algorithms and message authentication codes form combined algorithms for authenticated encryption (AE). The term AEAD refers to an AE scheme with support for associated data that can be used to strengthen security. In this work, security schemes and algorithms employed in current digital computers will be referred to as *classical* schemes or algorithms (*e.g.,* RSA [15], AES-128 [24], OCB3 [6]). Security schemes that are designed for providing security against an adversary with a quantum computer will be called *post-quantum* (*e.g.,* QCB [5], SATURNIN [12]). Importantly, post-quantum secure security algorithms have to be designed such that they are still secure and viable when used in classical computers.

A common distinction of quantum adversary types is between a Q1 and Q2 adversary. A Q1 adversary may use a quantum computer for offline computations but is limited to classical queries to any oracle function. The stronger Q2 adversary is additionally allowed to perform superposition queries to the oracle functions. Unless mentioned otherwise, the following results assume an adversary in the Q2 model.

Notation. Addition in $\mathbb{GF}(2^n)$, XOR, is denoted as $\oplus$. The set of all possible binary strings of length n will be described as $\{0,1\}^n$.

A block cipher accepts as inputs a secret key $K \in \{0,1\}^k$ and a message block $M \in \{0,1\}^n$ to compute a ciphertext block $C \in \{0,1\}^n$:

$$E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n.$$

A block cipher E encrypting message M under key K into ciphertext C is signalled with $E_K(M) = C$ while the decryption $E_K^{-1}(C) = M$ acts as the inverse of the encryption under the same key K.

A *tweakable block cipher* (TBC) additionally accepts a *tweak* $T \in \{0,1\}^t$ as input. Tweaks can be used to define distinct families of block ciphers under the same key. It is a tool to introduce variability to many calls of a block cipher where the key does not change throughout [19,20]. Consequently, the signature of a TBC can be described as:

$$\widetilde{E} : \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n.$$

The length of a message or ciphertext X measured as the amount of bits will be described with $|X| \in \mathbb{N}_0$.

For brevity, we will abbreviate *authenticated encryption with associated data* to AEAD, *plus-one unforgeability* to PO and *blind unforgeability* to BU.

2 Evolution of Unforgeability Notions

Authentication. In classical computing, the notions of *existential unforgeability under chosen-message attacks* (EUF-CMA) and *strong existential unforgeability under chosen-message attacks* (SUF-CMA) are prevalent to describe the security of MACs. However, these notions are not applicable in the quantum setting due to the properties of a quantum system [1, pp. 1-2][9]. *E.g.,* the adversary may query in superposition and due to *no-cloning* and measurement behaviour, it is not possible to identify a correct prediction of the adversary in the superposition state [2, p. 1]. To combat the lack of a notion like UF-CMA for unforgeability in quantum computers, Boneh and Zhandry introduced *plus-one unforgeability* [9,8]. Alagic et al. followed up with the idea of *blind unforgeability* [2].

Plus-One Unforgeability. Plus-one unforgeability (PO) was proposed as a candidate to classify unforgeability on quantum computers [8, p. 598]. An adversary $\mathcal{A}$ makes $q < |\mathcal{X}|$ queries to an oracle $\mathcal{O} : \mathcal{X} \to \mathcal{Y}$. If $\mathcal{A}$ can produce $q+1$ valid input-output pairs with non-negligible probability, the plus-one unforgeability of the underlying algorithm is violated, and it is not post-quantum secure [8, p. 593]. By utilizing the *rank method*, the authors in [8, p. 602] show

that if the size of $\mathcal{Y}$ is large enough, no (quantum) algorithm can produce $k+1$ input-output pairs when given k queries. Furthermore, they prove that a post-quantum secure pseudorandom function (qPRF) [32] is plus-one unforgeable when used as a MAC [8, p. 604].

However, Alagic et al. show that PO suffers from a weakness in its definition that allows for efficient quantum attacks on MACs that are plus-one unforgeable [2, pp. 24-32]. They describe that one of the issues of the PO definition is the inability to include adversaries which need to measure states after the query phase to produce a forgery. Measuring a state would collapse the register during the security game which the PO definition does not account for. In the counterexample of Alagic et al., an adversary may perform a forgery with a single query by utilizing quantum period-finding. Importantly, due to quantum period-finding algorithms not collapsing the entire state, the adversary is able to learn the period *and* a random input-output pair of the MAC at the same time [2, p. 24].

Blind Unforgeability. Introduced in [2], blind unforgeability (BU) aims to describe an improved notion to characterize (strong) existential unforgeability of MACs when faced by a quantum adversary [2, pp. 8-10] by eliminating the weaknesses of plus-one unforgeability. Blind unforgeability of a MAC is defined through the *blind forgery game* [2, p. 3] which will also be revisited during the proof in Chapter 4.

Before the game starts, a random *blind set* is constructed. *I.e.,* for some fraction of messages in the MAC's message space, the oracle $\mathcal{O}_{\mathrm{MAC}}$ will not return a corresponding authentication tag $\tau = \mathcal{O}_{\mathrm{MAC}}(M)$ when queried but rather signal to the adversary $\mathcal{A}$ that this message M is in the blind set by returning $\perp$. $\mathcal{A}$ wins if they can create a message-tag pair (m, t) with t being a valid tag for the message m and m being a member of the blind set. In other words, if $\mathcal{A}$ succeeds, they forged a tag for a message that they were not able to get any relevant information on by querying $\mathcal{O}_{\mathrm{MAC}}$. Therefore, $\mathcal{A}$ was able to generate an *existential* forgery.

In classical computation, blind unforgeability is equal to EUF-CMA and *strong* blind unforgeability is equal to SUF-CMA [2, pp. 11-12]. Due to the incompleteness of the PO notion as described above, the conjunction of PO and BU provides a stronger security proof than only showing one or the other. BU also implies quadratic PO [2, p. 14], and at the time of writing it remains an open question if BU implies PO, which would render BU a strictly stronger security notion than PO. A (pseudo)random function $R : X \to Y$ is considered a blindly unforgeable MAC if $1/|Y|$ is negligible in n [2, p. 15]. Furthermore, as with PO, post-quantum secure pseudorandom functions (qPRFs) are blindly unforgeable MACs [2, p. 4].

Blind unforgeability is equivalent to *generalised existential unforgeability* (μ-qGEU) where $\mu = 1$ [13, p. 18].

3 QCB: Post-Quantum Secure Authenticated Encryption

QCB, as introduced in [5], is a proposal for an authenticated encryption scheme with associated data (AEAD). The authors describe it as a post-quantum secure successor to the classically secure AEAD family OCB. Apart from being parallelizable, OCB [6] is a *rate-one* authenticated encryption scheme. For each message block being encrypted, approximately one call to the secure block cipher is carried out. These properties make OCB a highly efficient classical AEAD mode and mark the motivation for the creation of QCB: Defining a post-quantum secure, rate-one, parallelizable AEAD scheme on the basis of OCB [5, p. 2]. The authors acknowledge the similarity of the scheme to ΘCB [26] and the tweakable authenticated encryption (TAE) mode [19, pp. 39-41].

As shown by Kaplan et al. [16] and being revisited in Appendix A, the OCB family of authenticated encryption algorithms is not post-quantum secure. Importantly, the underlying construction using offsets is structurally broken by applying Simon's algorithm and increasing key sizes does not act as an easy remedy to this problem. As [5, pp. 9-11] point out, this does not only affect OCB but a large family of OCB-like schemes.

Tweakable Block Ciphers in QCB. To instantiate QCB, Bhaumik et al. define a family of tweakable block ciphers (TBC) that is post-quantum secure under the condition that tweaks may not be queried in superposition by the adversary [5, pp. 11-15].

Note that there are TBCs which are considered secure even when $\mathcal{A}$ has the ability to query tweaks in superposition, like LRWQ [14]. However, these TBCs can be broken by decryption queries and have a rate of 1/3 as they use three block cipher calls for each TBC call. This renders them unattractive for QCB, as QCB tries to achieve rate-one efficiency similar to OCB [5, p. 15].

In [5, pp. 16-17], QCB is proposed to be instantiated with the key-tweak insertion TBC Saturnin [12]. The design of this block cipher was originally motivated by the *NIST Lightweight Cryptography Standardization Process* [22,31,30] and is the only candidate where its designers tried to achieve post-quantum security while remaining in the lightweight domain. Alternatives for Saturnin are discussed briefly in Appendix B. Saturnin uses 256-bit blocks and keys which renders it as a potential candidate for usage as a post-quantum secure block cipher. The authors borrow internal design ideas from AES which is heavily researched with tight security bounds already in place. There exists a variant of Saturnin denoted as Saturnin_{16} using 16 *super-rounds* increasing the resistance of the underlying compression function against related-key attacks [12, p. 7][25].

The TBC used in [5] for QCB is defined as:

$$\widetilde{E}_{k,(d,IV,i)}(m) = \textsc{Saturnin}_{16}^{d}(k \oplus (IV||i), m)$$

with

$$\widetilde{E} : \mathcal{K} \times \mathcal{D} \times \mathcal{IV} \times \mathcal{I} \times \mathcal{M} \to \mathcal{C},$$
$$\widetilde{E} : \{0,1\}^{256} \times \{0,1\}^{4} \times \{0,1\}^{160} \times \{0,1\}^{96} \times \{0,1\}^{256} \to \{0,1\}^{256}.$$

Here, the tweak is denoted as the triple (d, IV, i), whereas $IV \in \mathcal{IV}$ is an initialization vector or nonce of at most 160 bits which is concatenated with $i \in \mathcal{I}$ describing the block number of the current block being encrypted (at most 2^{96} blocks are allowed). Parameter $d \in \mathcal{D}$ is the *domain separator* that can theoretically take 4 bits at maximum when Saturnin is used. For QCB, however, only 5 values in total are required [5, pp. 15-17]. $k \in \mathcal{K}$ denotes the secret key while $m \in \mathcal{M}$ denotes the data block to be encrypted with $\mathcal{C}$ representing the ciphertext space.

Structure of QCB. QCB is an AEAD mode that is instantiated with a post-quantum secure TBC. In the following, the usage of the TBC Saturnin is assumed. Figure 1 shows the encryption of ℓ plaintext blocks $M_i \in \{0,1\}^n$ into $\ell+1$ ciphertext blocks $C_i \in \{0,1\}^n$. If the last block M_* is of length $0 \leq c < n$, it will get padded with bitstring 10^{n-c-1} producing ciphertext block $C_* \in \{0,1\}^n$. Note that M_* will also be padded if it is empty ($c = 0$), which always leads to a ciphertext that will be longer than the plaintext input by at least 1 and at most n bits.

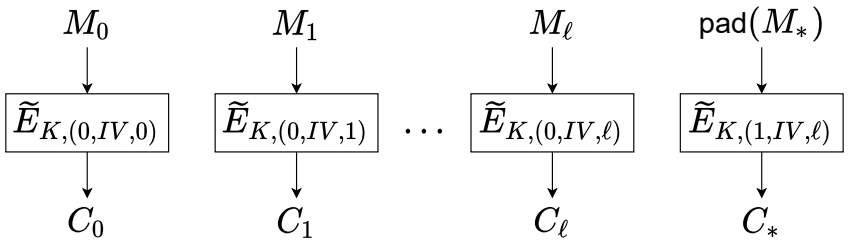

Fig. 1. Encryption of messages in QCB [5, Fig. 2].

Figure 2 illustrates how QCB uses all message blocks $M_0, \ldots, M_\ell$, all associated data blocks $A_0, \ldots, A_j$ and corresponding padding to calculate the authentication tag T. Padding of A_* behaves identically to the padding of M_* during the encryption procedure described above. Combining encryption and generation of the authentication tag leads to the full algorithm. Given a message M, associated data A, an initialization vector IV and a secret key K, QCB returns a ciphertext-tag pair C, T with $C = (C_0||C_1|| \ldots ||C_\ell||C_*)$.

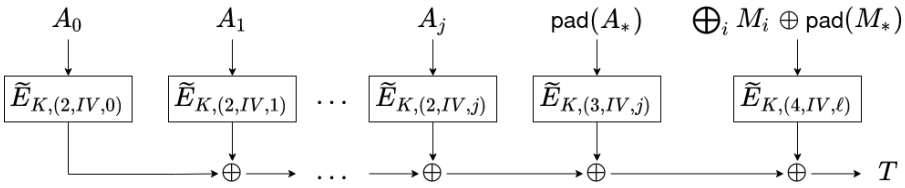

Fig. 2. Generation of the authentication tag and handling of associated data in QCB [5, Fig. 3].

QCB is Plus-One Unforgeable. Adversary $\mathcal{A}$ makes q superposition queries to QCB instantiated with the key-tweak insertion TBC $\widetilde{E}$ with ℓ blocks of messages or associated data at maximum. Furthermore, $\mathcal{A}$ makes q' encryption or decryption queries to the underlying block cipher E of the TBC $\widetilde{E}$. $\mathcal{A}$ succeeds in the plus-one unforgeability scenario if they output $q+1$ valid quadruples (A, IV, C, T) with associated data A, initialization vector IV, ciphertext C and tag T. Importantly, IVs are classical and cannot be queried in superposition. Consequently, the advantage of $\mathcal{A}$ over QCB in the definition of plus-one unforgeability (see also Section 2) is upper bounded by

$$\Pr[\mathcal{A}\ succeeds] \leq 8\sqrt{\frac{5\ell q q'^2}{2^n}} + \frac{3+c}{2^n}. \tag{1}$$

The constant c relates from the PRP-PRF distinguishing advantage and the probability $c\frac{q^3}{2^n}$ which describes an upper bound of $\mathcal{A}$ succeeding to produce $q+1$ valid input-output pairs when quantumly attacking an ideal random permutation with q queries [5, p. 8].

Evidently, in each call to the TBC, the initialization vector IV is provided as input. This fact and the requirement that no IV shall be used more than once is critical for security. QCB is secure against period-finding attacks as described in the attack on OCB in Appendix A. QCB is also secure against *quantum linearization attacks* [10] due to the continuous usage of the initialization vector in tweaks of the TBC [5, p. 15]. Consider a weakened version of QCB where the IV is not used during the processing of associated data. This algorithm can be broken with application of Deutsch's algorithm as shown in [5, Appendix B]. After n queries, an adversary is able to fully recover certain values which allow them to perform forgeries. More specifically, they are able to compute a valid tag for any message with associated data $A = 1$ if they are provided the tag of the same message with associated data $A = 0$. This attack is made possible since in the weakened version of QCB, the associated data is encrypted independently of the IV allowing an adversary to repeatedly use the encryption of blocks of associated data made in prior queries. The full specification of QCB, however, denies this vulnerability by using the IV in each block as part of the tweak for the TBC [5, Appendix B].

4 QCB is Blindly Unforgeable

On the following pages, blind unforgeability of QCB will be proven by employing a similar technique to the proof about QCB's plus-one unforgeability in [5, pp. 21-22]. The authors of QCB acknowledged the existence of blind unforgeability and left the proof for QCB as an open problem [5, pp. 27-28].

BU Game. The blind forgery experiment or blind unforgeability (BU) game as presented in [2, p. 3] can be adjusted to work with QCB (or any authenticated encryption scheme).

To generate a random blinding B_ϵ, start with the empty set $B_\epsilon = \varnothing$ and place each triple (IV, A, M) into the set with probability ϵ. Here, IV represents the initialization vector, A the associated data and M the message. The encryption oracle blinded on B_ϵ is subsequently defined as:

$$B_\epsilon \mathrm{QCB}_K(IV, A, M) \begin{cases} \perp & \text{if } (IV, A, M) \in B_\epsilon, \\ (C, T) & \text{otherwise.} \end{cases}$$

The decryption under key K will be denoted as $\mathrm{QCB}_K^{-1}(\cdot)$.

The BU game is then carried out as follows:

1. Adversary $\mathcal{A}$ selects $0 < \epsilon < 1$.
2. Key K is generated uniformly at random. A random blinding B_ϵ is generated, whereas each triple (IV, A, M) is put into the blind set with probability ϵ.
3. $\mathcal{A}$ asks q queries with messages $\{M^1, \ldots, M^q\} = \mathcal{M}_q$ to the encryption oracle blinded on B_ϵ.
4. $\mathcal{A}$ produces a candidate forgery (IV, A, C, T), where C represents the ciphertext and T the authentication tag.
5. Output **win** if $\mathrm{QCB}_K^{-1}(IV, A, C, T) \notin \{\perp\} \cup \mathcal{M}_q$ and corresponding M is such that $(IV, A, M) \in B_\epsilon$.

Definition 1 ([2], Definition 1). *QCB is blindly unforgeable if for every adversary $(\mathcal{A}, \epsilon)$, the probability of winning the blind unforgeability game is negligible.*

Theorem 1. *Let $(\mathcal{A}, \epsilon)$ be a Q2 adversary making q superposition queries to QCB with at most ℓ blocks of message and associated data combined and making q' queries to the block cipher E. Then QCB is blindly unforgeable on blind set B_ϵ with the success probability of $(\mathcal{A}, \epsilon)$ being bounded as:*

$$Pr[\mathcal{A} \text{ succeeds}] \leq \frac{\epsilon}{2^n - q} + 8\sqrt{\frac{5\ell q q'^2}{2^n}}. \tag{2}$$

Proof. Consider two BU games G_0, G_1. In G_0, adversary $\mathcal{A}$ queries QCB constructed with TBC $\widetilde{E}$ and key K selected uniformly at random. G_1 is the modification of game G_0 where the TBC $\widetilde{E}$ is replaced by a family of ideal independent random permutations for all tweaks [5, p. 21]. We make use of Lemma 3 from the PO-proof in the original QCB publication.

Lemma 1 ([5], Lemma 3).

$$Pr_{G_0}[\mathcal{A}\ succeeds] \leq Pr_{G_1}[\mathcal{A}\ succeeds] + Adv^{TPRP}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q')$$

Next, identify a bound for $Pr_{G_1}[\mathcal{A}\ succeeds]$ to get a bound on $\mathcal{A}$'s advantage against QCB as represented by $Pr_{G_0}[\mathcal{A}\ succeeds]$ on the left side of the inequation in Lemma 1.

For clarity, the BU game will be split up. For $\mathcal{A}$ to be successful in the full *BU game*, they need to be successful in both of the following events E_i:

E_1: $\mathcal{A}$ generates (IV, A, C, T) with $\mathrm{QCB}_K^{-1}(IV, A, C, T) \notin \{\perp\} \cup \mathcal{M}_q$.
E_2: The corresponding (IV, A, M) is in the blind set.

Remember that, as game G_1 is being investigated, the underlying block cipher is an ideal random permutation.

After q queries to the oracle, the probability to generate a pair (C, T) which is valid, *i.e.,* the decryption of (C, T) does not return $\perp$ or any M which was already queried in the query phase, is

$$Pr_{G_1}[\mathcal{A}\ succeeds\ in\ \mathrm{E}_1] = \frac{1}{2^{|T|} - q}, \tag{3}$$

where $|T|$ denotes the bit-length of authentication tag T. While ciphtertext C can be of variable size for different queries in an instance of QCB, the tag T is required to be of fixed size for each query [5, p. 4].

The blind set B_ϵ is of expected size $\epsilon \cdot 2^m$ with $m = |IV| + |A| + |M|$. The probability of randomly selecting any triple (IV, A, M) from the set of all possible triples is $\frac{1}{2^m}$. The expected probability of randomly hitting any item in the blind set B_ϵ is therefore

$$Pr_{G_1}[\mathcal{A}\ succeeds\ in\ \mathrm{E}_2] = \frac{|B_\epsilon|}{2^m} = \frac{\epsilon \cdot 2^m}{2^m} = \epsilon. \tag{4}$$

The probabilities for E_1 and E_2 can be treated as being independent as the blind set had been generated independently of the encryption algorithm. Thus, the expected probability that $\mathcal{A}$ successfully generates a valid forgery with the corresponding triple (IV, A, M) being in the blind set is

$$Pr_{G_1}[\mathcal{A}\ succeeds] = Pr_{G_1}[\mathcal{A}\ succeeds\ in\ \mathrm{E}_1] \cdot Pr_{G_1}[\mathcal{A}\ succeeds\ in\ \mathrm{E}_2].$$

Substituting observations from Equations 3 and 4 leads to

$$Pr_{G_1}[\mathcal{A}\ succeeds] = \frac{1}{2^{|T|} - q} \cdot \epsilon = \frac{\epsilon}{2^{|T|} - q} = \frac{\epsilon}{2^n - q}. \tag{5}$$

Length $|T|$ of the final tag T is equal to the block size n of message or associated data blocks. Subsequently, substituting Equation 5 into Lemma 1 yields

$$Pr_{G_0}[\mathcal{A}\ succeeds] \leq \frac{\epsilon}{2^n - q} + \mathrm{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q'). \tag{6}$$

The advantage $\mathrm{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q')$ of $\mathcal{A}$ against the tweakable pseudorandom permutation (TPRP) security of $\widetilde{E}$ is defined in [5, pp. 12-14, 20]. Quantum adversary $\mathcal{A}$ makes q queries with blocks of length $\leq \ell$ and q' queries to the block cipher E that is the main building block of the TBC $\widetilde{E}$. The set of tweaks that may be queried has to be pre-declared (see also [5]) and may at most be of size $5\ell q$. Furthermore, tweaks are not allowed to be queried in superposition. As described in [5, p. 20], this advantage is upper bounded by

$$\mathrm{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q') \leq 8\sqrt{\frac{5\ell q q'^2}{2^n}}. \tag{7}$$

Finally, Equations 6 and 7 produce

$$Pr_{G_0}[\mathcal{A}\ succeeds] \leq \frac{\epsilon}{2^n - q} + 8\sqrt{\frac{5\ell q q'^2}{2^n}}, \tag{8}$$

which gives an upper bound for the probability of $\mathcal{A}$ to succeed in a forgery on QCB. This bound depends on the size ϵ of the blind set, block size $n = |T|$, the amount $5\ell q$ of pre-declared tweaks and the amount q, q' of queries made by $\mathcal{A}$ to $\widetilde{E}$ or E respectively.

□

In the case of using Saturnin as the TBC $\widetilde{E}$ for QCB, a message block or associated data block consists of 256 bits. As QCB generates the tag by XOR of these blocks (see Figure 2), the resulting tag T is also of size $|T| = 256$ bits. For reasonable parameters q, q', this renders the advantage of $\mathcal{A}$ to succeed in the blind unforgeability game G_0 on QCB with Saturnin negligible. For an example, consider $\mathcal{A}$ making $q = q' = 2^{32}$ queries to $\widetilde{E}, E$ respectively with $\ell = n = 256$. The probability that $\mathcal{A}$ succeeds in creating a blind forgery is therefore

$$\begin{aligned} Pr_{G_0}[\mathcal{A}\ succeeds] &\leq \frac{\epsilon}{2^{256} - 2^{32}} + 8\sqrt{\frac{5 \cdot 256 \cdot 2^{32^2}}{2^{256}}} \\ &\leq \frac{\epsilon}{2^{224}} + 8\sqrt{\frac{1}{2^{149}}}, \end{aligned}$$

which is negligible for any ϵ with $0 < \epsilon < 1$.

5 General Blindly Unforgeable Authenticated Encryption

In QCB, the authentication procedure (Figure 2) makes use of a TBC $\widetilde{E}_{K,T}$ whereas for each block cipher call, the tweak T_i is ensured to be different to the tweaks T_j with $j < i$. Similar to PMAC, this construction is parallelizable.

This idea can be generalised to describe a generic AEAD construction that is blindly unforgeable. For the security proof of blind unforgeability to hold, a set of tweaks has to be pre-declared and each tweak used throughout the BU-game needs to be inside the tweak set.

Initially, tweak space $\mathcal{T}$ and initialisation vector space $\mathcal{IV}$ are generated. Let $T \xleftarrow{gen} \mathcal{T}(X)$ denote tweak T being picked through some black box function $f(X)$ from $\mathcal{T}$. Importantly, the initialisation vector needs to be included in the (derivation of the) tweak of each TBC call to be secure against a quantum forgery attack based on Deutsch's algorithm [10]. In other words, one input to f needs to be but is not limited to an IV or *nonce* which is then used to generate a tweak. Algorithm 1 describes an authenticated encryption scheme. Algorithms 2 and 3 denote the encryption and tag generation procedures respectively. Algorithm 4 chooses and returns a tweak from $\mathcal{T}$. Importantly, the chosen tweak is removed from $\mathcal{T}$ to ensure that no tweak is used more than once. Algorithm 5 performs 10^* padding to pad a block to size n. If $|M|$ or $|A|$ respectively are a multiple of n, the blocks M_* or A_* will be of length 0. Nevertheless, they are still needed for further calculation and will be padded to length n. The Ciphertext C is therefore always at least 1 bit longer than M.

Algorithm 1: AUTHENTICATEDENCRYPTION(M, A, IV, K)

Input: Message M, associated data A, initialisation vector IV, key K
Requirements: Initialisation vectors should not be reused;
Output: Ciphertext C, tag τ
$C \leftarrow$ ENCRYPTION(M, IV, K);
$\tau \leftarrow$ GENERATETAG(M, A, IV, K);
return (C, τ)

Algorithm 2: ENCRYPTION(M, IV, K)

Input: Message M, initialisation vector IV, key K
$(M_1, \ldots, M_\ell, M_*) \leftarrow M$ with $|M_i| = n$; // $|M_*|$ can be 0.
for $i = 1$ *to* ℓ **do**
  $T_i \leftarrow$ GENTWEAK(IV);
  $C_i \leftarrow \widetilde{E}_{K,T_i}(M_i)$;
end
$T_{\ell+1} \leftarrow$ GENTWEAK(IV);
$C_* \leftarrow \widetilde{E}_{K,T_{\ell+1}}(\text{PAD}(M_*, n))$;
$C \leftarrow (C_1, \ldots, C_\ell, C_*)$;
return C

Algorithm 3: GENERATETAG(M, A, IV, K)

Input: Message M, associated data A, initialisation vector IV, key K

$(A_1, \ldots, A_j, A_*) \leftarrow A$ with $|A_i| = n$; // $|A_*|$ can be 0.
$X_0 \leftarrow 0^n$;
for $i = 1$ *to* j **do**
  $T_i \leftarrow$ GENTWEAK(IV);
  $X_i \leftarrow X_{i-1} \oplus \widetilde{E}_{K,T_i}(A_i)$;
end
$T_{j+1} \rightarrow$ GENTWEAK(IV);
$X_{j+1} \leftarrow X_j \oplus \widetilde{E}_{K,T_{j+1}}(\text{PAD}(A_*, n))$;
$M' \leftarrow \bigoplus_i M_i \oplus \text{PAD}(M_*, n)$;
$T_{j+2} \leftarrow$ GENTWEAK(IV);
$\tau \leftarrow X_{j+1} \oplus \widetilde{E}_{K,T_{j+2}}(M')$;
return τ

Algorithm 4: GENTWEAK(IV)

Input: Initialisation vector IV

$T \xleftarrow{gen} \mathcal{T}(IV)$;
$\mathcal{T} = \mathcal{T} \setminus \{T\}$;
return T

Algorithm 5: PAD(X, n)

Input: Block $X \in \{0,1\}^m$, block size n

$X' \leftarrow X||1$;
while $|X'| < n$ **do** $X' \leftarrow X'||0$;
return X'

6 Conclusion and Future Work

It is apparent that there are popular classical AEAD constructions that are structurally insecure when challenged by a quantum adversary. This means that new techniques need to be established which can substitute or repair the broken building blocks of the affected schemes and algorithms. The usage of a TBC where each tweak contains the initialization vector provides a defence strategy against quantum period finding and quantum linearization attacks. This strategy can fix vulnerabilities in known MACs and AEAD schemes like PMAC or OCB, which provide parallelizable, efficient authentication or authenticated encryption. QCB, a proposed post-quantum successor of the OCB-family, utilizes the TBC-strategy and seems to provide a post-quantum secure rate-one AEAD scheme. This recipe for security under blind unforgeability can be generalized to a more generic AEAD scheme.

Other classical schemes suffer from the same vulnerabilities against a quantum adversary like OCB. Enhancing those schemes with the aforementioned structure may prove to be a viable method to eliminate attack vectors in the quantum scenario.

A Quantum Attacks against Symmetric Cryptography

Simon's Algorithm for Period-Finding. Given a black-box function $f : \{0,1\}^n \to \{0,1\}^n$ with some unknown period $s \in \{0,1\}^n$ and $f(x) = f(y) \Leftrightarrow ((x = y) \vee (x = y \oplus s))$ for all $x, y \in \{0,1\}^n$. *I.e.,* there exist two distinct values x, y for which f produces the same result. The difference $x \oplus y$ between these values is s. In the context of this chapter, a function that satisfies this property is also described as satisfying *Simon's promise*. Finding s on a classical computer takes $\Theta(2^{n/2})$ queries to f. Simon's algorithm [29] can find s with $\mathcal{O}(n)$ queries to the black-box function on a quantum computer. The following paragraphs highlight some of the impactful attacks presented from Kaplan et al. against CBC-MAC, LRW, PMAC and OCB [16].

CBC-MAC. Consider some adversary $\mathcal{A}$ who has access to the encryption oracle $E_k : \{0,1\}^n \to \{0,1\}^n$ and a function f satisfying Simon's promise. Furthermore, $\mathcal{A}$ can query f in superposition if they have quantum oracle access to E_k. If $\mathcal{A}$ can find the hidden difference s, it is sufficient to break the cryptographic scheme. In this attack, $s = E(M_1) \oplus E(M_2)$ for two distinct messages M_1, M_2.

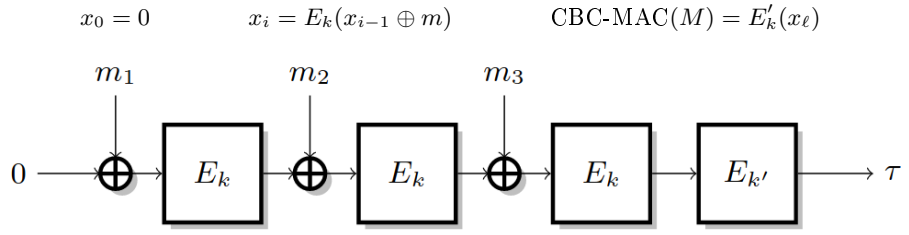

Fig. 3. Encrypted CBC-MAC [16, Fig. 9]. Here, k, k' denote two independent keys, $M = m_1 || \ldots || m_\ell$ is the message divided into ℓ blocks and τ the resulting authentication tag.

Figure 3 shows the standardized encrypted CBC-MAC. Classically, it is considered secure (up to the birthday bound) [4]. According to the attack strategy

described above, f is defined as

$$\begin{aligned} f : \{0,1\} \times \{0,1\}^n &\to \{0,1\}^n \\ b, x &\mapsto \text{CBC-MAC}(\alpha_b || x) = E'_k(E_k(x \oplus E_k(\alpha_b))) \end{aligned}$$

with α_0, α_1 representing two distinct message blocks [16, p. 15]. This function f satisfies Simon's promise with $s = 1||E_k(\alpha_0) \oplus E_k(\alpha_1)$. Consequently, applying Simon's algorithm will return $E_k(\alpha_0) \oplus E_k(\alpha_1)$ which allows for the forgery of messages. Query the oracle to receive tag $\tau_0 = \text{CBC-MAC}(\alpha_0||m_1)$ for an arbitrary m_1. Next, query the oracle for tag $\tau_1 = \text{CBC-MAC}(\alpha_1||m_1 \oplus E_k(\alpha_0) \oplus E_k(\alpha_1))$. It holds that $\tau_1 = \tau_2$ and a valid tag has been forged successfully [16, pp. 15-16]. This attack directly violates *plus-one unforgeability* as well as *blind unforgeability*, which are examined in Chapter 2. If adversary $\mathcal{A}$ repeats the forgery step $q + 1$ times making $2q + 1$ classical and quantum queries to the oracle, they can produce $2(q + 1)$ messages with valid tags.

Liskov-Rivest-Wagner (LRW) Construction. By employing the *LRW* construction, a block cipher E is transformed into a tweakable block cipher E^*, whereas E^* is a family of unrelated block ciphers. The construction is defined as

$$E^*_{t,k}(x) = E_k(x \oplus h(t)) \oplus h(t)$$

with h being an (almost) universal hash function [19,20]. Here, h and k are both part of the joint key. Furthermore, for two arbitrary tweaks $t_0 \neq t_1$, the function f is defined as [16, p. 13]

$$\begin{aligned} f : \{0,1\}^n &\to \{0,1\}^n \\ x &\mapsto E^*_{t_0,k}(x) \oplus E^*_{t_1,k}(x) \\ f(x) &= E_k(x \oplus h(t_0)) \oplus h(t_0) \oplus E_k(x \oplus h(t_1)) \oplus h(t_1). \end{aligned}$$

This function satisfies Simon's promise with $f(x) = f(x \oplus s) = f(x \oplus h(t_0) \oplus h(t_1))$. Therefore, by running Simon's algorithm $\mathcal{O}(n)$ times, an attacker can recover $s = h(t_0 \oplus h(t_1))$. The difference s is orthogonal to all the values measured in Simon's algorithm and therefore appears $\mathcal{O}(n)$ times during the computation. As this structure would not occur when f is a random function, it allows for an efficient distinguisher between an ideal random tweakable permutation and the *LRW* construction for defining tweakable block ciphers [16, pp. 13-14].

PMAC. The attack on CBC-MAC can be used to attack other message authentication codes as well. PMAC [26], for example, works as follows:

$$c_i = E_k(m_i \oplus \Delta_i) \qquad \text{PMAC}(M) = E^*_k(m_\ell \oplus \sum_i c_i)$$

with E^* being a tweakable block cipher derived from E. PMAC has the same internal structure as CBC-MAC when only messages consisting of two blocks

are considered: $\text{PMAC}(m_1||m_2) = E_k^*(m_2 \oplus E_k(m_1 \oplus \Delta_0))$. $\mathcal{A}$ can therefore execute the identical attack as used to break CBC-MAC. Query the tag $\tau_0 = \text{PMAC}(\alpha_0||m_1||m_2)$ for arbitrary message blocks m_1, m_2. Consequently, $\tau_1 = \text{PMAC}(\alpha_1||m_1||m_2 \oplus E_k(\alpha_0) \oplus E_k(\alpha_1)) = \tau_0$ and a valid forgery has been achieved [16, pp. 16-17].

A different attack can be carried out by utilizing the vulnerabilities of *LRW* to gain knowledge of the differences Δ_i. First, the function fulfilling Simon's promise is defined as

$$\begin{aligned} f : \{0,1\}^n &\to \{0,1\}^n \\ m &\mapsto \text{PMAC}(m||m||0^n) = E_k^*(E_k(m \oplus \Delta_0) \oplus E_k(m \oplus \Delta_1)). \end{aligned}$$

The hidden difference s is given with $f(m) = f(m \oplus s) = f(m \oplus \Delta_0 \oplus \Delta_1)$. Therefore, $s = \Delta_0 \oplus \Delta_1$ can be recovered by an adversary efficiently using Simon's Algorithm in $\mathcal{O}(n)$ iterations. The adversary queries tag $\tau_1 = \text{PMAC}(m_1||m_1)$ for an arbitrary message block m_1. It holds that τ_1 is equal to $\tau_2 = \text{PMAC}(m_1 \oplus \Delta_0 \oplus \Delta_1||m_1 \oplus \Delta_0 \oplus \Delta_1)$ and therefore a valid forgery was generated.

PMAC is based on the *XE* construction, which is an instantiation of *LRW*. In PMAC, the offsets are calculated with $\Delta_i = \gamma(i) \cdot L$ with $\gamma(i)$ being the *Gray encoding* of i and $L = E_k(0)$ [26, p. 21]. This leads to an adversary being able to learn L from the hidden period $s = \Delta_0 \oplus \Delta_1$ with $L = (\Delta_0 \oplus \Delta_1) \cdot (\gamma(0) \oplus \gamma(1))^{-1}$. With this knowledge, the adversary can compute each Δ_i and forge any arbitrary message.

OCB. Finally, to attack the authenticated encryption mode OCB, it can be observed that OCB reduces to a randomized variant of PMAC when the message is empty [16, p. 20]. Encrypted ciphertexts c_i and authentication tag τ are generated by OCB as

$$\begin{aligned} c_i &= E_k(m_i \oplus \Delta_i^N) \oplus \Delta_i^N, \\ \tau &= E_k\left(\Delta_\ell'^N \oplus \sum_i m_i\right) \oplus \sum_i E_k(a_i \oplus \Delta_i) \end{aligned}$$

with nonce N, message $M = m_1|| \ldots ||m_\ell$ and associated data $A = a_i|| \ldots ||a_\ell$. Using an empty message ϵ, OCB generates the tag τ with

$$\text{PMAC}_k(N, \varepsilon, A) = \phi_k(N) \oplus \sum_i E_k(a_i \oplus \Delta_i).$$

Note that $\phi_k(N)$ denotes a permutation under key k whose specific description is of no interest to us. This construction can be attacked as described by the second attack on PMAC based on the *LRW* vulnerabilities. Consider a family

of functions f_N with

$$\begin{aligned} f_N : \{0,1\}^n &\to \{0,1\}^n \\ x &\mapsto \mathrm{PMAC}_k(N, \epsilon, x||x) \\ f_N(x) &= E_k(x \oplus \Delta_0) \oplus E_k(x \oplus \Delta_1) \oplus \phi_k(N). \end{aligned}$$

Each function f_N for any N satisfies Simon's promise: $f_N(a) = f_N(a \oplus s) = f_N(a \oplus \Delta_0 \oplus \Delta_1)$. This allows for the recovery of the hidden period $s = \Delta_0 \oplus \Delta_1$. An adversary can now query the authenticated encryption with ciphertext and tag pair $C_1, \tau_1 = \mathrm{OCB}(N, M, a||a)$ for arbitrary message M, an arbitrary block a and random Nonce N. C_1, τ_1 is also a valid authenticated encryption of $\mathrm{OCB}(N, M, a \oplus \Delta_0 \oplus \Delta_1 || a \oplus \Delta_0 \oplus \Delta_1)$ with the same nonce N [16, p. 20].

B Instantiation of QCB with TRAX and Pholkos.

When using Saturnin as the TBC for QCB, due to the *key-tweak-insertion* construction, each message or associated data block is encrypted with a separate block-key based on the key k which is modified by a distinct tweak for each block cipher call. For an adversary $\mathcal{A}$, it is therefore sufficient to find only one of these block-keys to break the TBC and thus QCB. Consequently, there are more chances of $\mathcal{A}$ breaking one of the TBC iterations than there would be for a block cipher that uses the same key for each block. Keep in mind that the latter construction would then be structurally vulnerable to quantum attacks like quantum linearization.

However, the authors of QCB mention the scarcity of usable 256-bit block ciphers. They do suggest to alternatively use the dedicated TBC TRAX-L-17 [3] which is based on 256-bit message blocks and keys but a smaller tweak than Saturnin with 128 bits. This would allow for IVs of 80 bits and at most $2^{45} - 1$ blocks of plaintext and associated data [5, p. 17]. An alternative that may provide a better trade-off between security and efficiency is the TBC Pholkos [11]. Pholkos is a recent proposal for a post-quantum-secure TBC with a tweak size of 128 bits, block sizes of 256 or 512 bits and keys of size 256 bit. It is a substitution-permutation network (SPN) inspired by AESQ [7] and Haraka [17]. Any input plaintext block is encrypted in 8-14 steps depending on the configuration of block and key size. Initially, the n-bit plaintext block is split into $\frac{n}{128}$ 128-bit blocks which are then split into four 32-bit words each. Subsequently, each step performs the similar rounds as found in the classical block cipher AES [24]. A *tweakey* is used for the AddRoundKey step of AES, whereas a round tweakey is generated by a schedule from the secret key and the tweak. An advantage of Pholkos is that the block cipher AES is well researched in terms of cryptanalysis and security. Furthermore, efficient implementations in soft- and hardware already exist. Pholkos-QCB provides a larger security margin than Saturnin-QCB due to the larger tweak space.

References

1. Alagic, G., Gagliardoni, T., Majenz, C.: Unforgeable quantum encryption. In: Advances in Cryptology – EUROCRYPT 2018. pp. 489–519. Springer International Publishing, Cham (2018)
2. Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. Lecture Notes in Computer Science pp. 788–817 (2020)
3. Beierle, C., Biryukov, A., dos Santos, L.C., Großschädl, J., Perrin, L., Udovenko, A., Velichkov, V., Wang, Q.: Alzette: A 64-bit arx-box. In: Annual International Cryptology Conference. pp. 419–448. Springer (2020)
4. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. Journal of Computer and System Sciences **61**(3), 362–399 (2000)
5. Bhaumik, R., Bonnetain, X., Chailloux, A., Leurent, G., Naya-Plasencia, M., Schrottenloher, A., Seurin, Y.: Qcb: Efficient quantum-secure authenticated encryption. IACR Cryptol. ePrint Arch. **2020**, 1304 (2020)
6. Bhaumik, R., Nandi, M.: Improved security for ocb3 (Nov 2017). https://doi.org/10.1007/978-3-319-70697-9_22, https://eprint.iacr.org/2017/845.pdf
7. Biryukov, A., Khovratovich, D.: Paeq: parallelizable permutation-based authenticated encryption. In: International Conference on Information Security. pp. 72–89. Springer (2014)
8. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: Advances in Cryptology – EUROCRYPT 2013. pp. 592–608. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
9. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Advances in Cryptology - CRYPTO 2013. pp. 361–379. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
10. Bonnetain, X., Leurent, G., Naya-Plasencia, M., Schrottenloher, A.: Quantum linearization attacks. Cryptology ePrint Archive, Report 2021/1239 (2021)
11. Bossert, J., List, E., Lucks, S., Schmitz, S.: Pholkos–efficient large-state tweakable block ciphers from the aes round function. In: Cryptographers' Track at the RSA Conference. pp. 511–536. Springer (2022)
12. Canteaut, A., Duval, S., Leurent, G., Naya-Plasencia, M., Perrin, L., Pornin, T., Schrottenloher, A.: Saturnin: a suite of lightweight symmetric algorithms for post-quantum security. IACR Transactions on Symmetric Cryptology **2020**(S1), 160–207 (Jun 2020). https://doi.org/10.13154/tosc.v2020.iS1.160-207, https://tosc.iacr.org/index.php/ToSC/article/view/8621
13. Doosti, M., Delavar, M., Kashefi, E., Arapinis, M.: A unified framework for quantum unforgeability (2021). https://doi.org/10.48550/ARXIV.2103.13994, https://arxiv.org/abs/2103.13994
14. Hosoyamada, A., Iwata, T.: Provably quantum-secure tweakable block ciphers. IACR Transactions on Symmetric Cryptology pp. 337–377 (2021)
15. IEEE: Ieee standard specifications for public-key cryptography. IEEE Std 1363-2000 pp. 1–228 (2000). https://doi.org/10.1109/IEEESTD.2000.92292
16. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. https://arxiv.org/pdf/1602.05973.pdf (2016)
17. Kölbl, S., Lauridsen, M.M., Mendel, F., Rechberger, C.: Haraka v2–efficient short-input hashing for post-quantum applications. IACR Transactions on Symmetric Cryptology pp. 1–29 (2016)

18. Leuther, J., Lucks, S.: Qcb is blindly unforgeable. In: International Conference on Codes, Cryptology, and Information Security. pp. 91–108. Springer (2023)
19. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Annual International Cryptology Conference. pp. 31–46. Springer (2002)
20. Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. Journal of cryptology **24**(3), 588–613 (2011)
21. Mavroeidis, V., Vishi, K., Zych, M.D., Jøsang, A.: The impact of quantum computing on present cryptography. International Journal of Advanced Computer Science and Applications **9**(3) (2018). https://doi.org/10.14569/ijacsa.2018.090354, http://dx.doi.org/10.14569/IJACSA.2018.090354
22. McKay, K., Bassham, L., Sönmez Turan, M., Mouha, N.: Report on lightweight cryptography. Tech. rep., National Institute of Standards and Technology (2016)
23. Moody, D., Chen, L., Jordan, S., Liu, Y.K., Smith, D., Perlner, R., Peralta, R.: Nist report on post-quantum cryptography (Apr 2016). https://doi.org/10.6028/NIST.IR.8105
24. National Institute of Standards and Technology (NIST): Announcing the advanced encryption standard (aes) (Nov 2001)
25. Roetteler, M., Steinwandt, R.: A note on quantum related-key attacks. Information Processing Letters **115**(1), 40–44 (2015)
26. Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 16–31. Springer (2004)
27. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
28. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing **26**(5), 1484–1509 (Oct 1997). https://doi.org/10.1137/s0097539795293172, http://dx.doi.org/10.1137/S0097539795293172
29. Simon, D.: On the power of quantum computation. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 116–123 (1994). https://doi.org/10.1109/SFCS.1994.365701
30. Sönmez Turan, M., McKay, K., Chang, D., Çalık, Ç., Bassham, L., Kang, J., Kelsey, J.: Status report on the second round of the nist lightweight cryptography standardization process. Tech. rep., National Institute of Standards and Technology (2021)
31. Turan, M.S., McKay, K.A., Çalik, Ç., Chang, D., Bassham, L., et al.: Status report on the first round of the nist lightweight cryptography standardization process. National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Rep.(NISTIR) (2019)
32. Zhandry, M.: How to construct quantum random functions. Cryptology ePrint Archive, Report 2012/182 (2012)