QCB: Efficient Quantum-secure Authenticated Encryption

Ritam Bhaumik[1], Xavier Bonnetain[2], André Chailloux[1], Gaëtan Leurent[1], María Naya-Plasencia[1], André Schrottenloher[1], and Yannick Seurin[3]

[1] Inria, Paris, France
firstname.lastname@inria.fr
[2] Institute for Quantum Computing, Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada
xbonnetain@uwaterloo.ca
[3] ANSSI, Paris, France
first.last@m4x.org

Abstract. It was long thought that symmetric cryptography was only mildly affected by quantum attacks, and that doubling the key length was sufficient to restore security. However, recent works have shown that Simon's quantum period finding algorithm breaks a large number of MAC and authenticated encryption algorithms when the adversary can query the MAC/encryption oracle with a quantum superposition of messages. In particular, the OCB authenticated encryption mode is broken in this setting, and no quantum-secure mode is known with the same efficiency (rate-one and parallelizable).
In this paper we generalize the previous attacks, show that a large class of OCB-like schemes is unsafe against superposition queries, and discuss the quantum security notions for authenticated encryption modes. We propose a new rate-one parallelizable mode named QCB inspired by TAE and OCB and prove its security against quantum superposition queries.

Keywords: authenticated encryption, lightweight cryptography, QCB, post-quantum cryptography, provable security, tweakable block ciphers.

1 Introduction

The cryptographic community has launched many competitions and standardization efforts recently. The most recent ones are the CAESAR competition for authenticated encryption (AE) and the NIST standardization processes for post-quantum public-key primitives (PQC) [22] and lightweight cryptography (LWC) [23]. While these competitions have attracted a lot of attention, they have represented rather disjoint efforts: the PQC process focuses on public key cryptography, and post-quantum security has remained out of the scope of most schemes submitted to the LWC process and to the CAESAR competition. A few exceptions exist, like the LWC second-round candidate Saturnin [12] for instance, which proposes a block cipher and an AE mode aiming at post-quantum

security. This is understandable because the impact of quantum computers on symmetric cryptography is expected to be quite limited, and doubling the key length is usually considered a sufficient measure to resist quantum attacks (such as exhaustive key search with Grover's algorithm).

Security in the superposition model. However, recent work [18, 27] have shown that many MAC and authenticated encryption modes are broken in the superposition model using Simon's quantum period finding algorithm [28]. In this model, the adversary is capable of accessing a quantum encryption oracle, and of encrypting quantum states. Though the practical significance of attacks in this model is an unsettled issue in the community and opinions might differ, there is a clear consensus on the importance of having provable security in this scenario. First of all, this model is non-trivial, meaning that there exist secure schemes in this model.[d] It also offers better composability, even if we are interested only in quantum adversaries making classical queries. Finally, it captures intermediate scenarios with some level of quantum interaction between the attacker and the oracle and covers the scenarios of obfuscation or white-box encryption.

Though lightness and security against quantum adversaries are two very different topics, let us remark that they are not orthogonal. In particular, Saturnin is a submission to the LWC effort claiming security in the superposition model, based on a block cipher. But its authenticated encryption mode is not parallelizable and requires two encryption calls per message block. More precisely, it uses the encrypt-then-MAC construction and combines a quantum-secure mode of encryption (the Counter Mode) with a quantum-secure MAC similar to HMAC/NMAC.

Towards a quantum-safe rate-one AE mode. OCB [19] is one of the most influential authenticated encryption modes. OCB3 is parallelizable, and is a rate-one scheme, using just one block cipher call per block of message. It is proven secure in the classical setting provided that its underlying block cipher is a strong PRP [6]. Nevertheless, several attacks in a quantum superposition setting that use Simon's algorithm [28] were proposed in [18], with a complexity that is linear in the size of the state. These attacks, that we recall in Section 3, can efficiently recover a hidden secret period if the attacker is allowed to query messages in superposition.

Our work started with the idea to make OCB post-quantum: we wanted to identify its weaknesses, correct them and obtain a proof of quantum security. The main contribution of this paper is to fill this gap and to propose such a mode together with a proof of security.

Results and Organization of the Paper. In Section 2, we recall some standard definitions and technical material for our quantum security proofs and attacks. Note that contrary to most of the recent works on this topic, we shall not require Zhandry's random oracle recording technique [30] and we will use instead

[d] For example, indistinguishability under quantum encryption queries can be achieved by the Counter Mode from a classical PRP assumption [2].

simpler proof arguments, that we introduce here. We also introduce an extension of Hosoyamada and Sasaki's truncation technique [16] that allows to compose *any linear function* with a quantum oracle and compute it with a single query. In Section 3, we define an OCB-like mode with more complex *offsets*. The previous quantum attack on OCB used the fact that the difference between some offsets was independent of the nonce. We show how to attack this modified OCB with a *single* quantum query, yielding an attack that can be applied regardless of the nonce dependence. In Section 4, we define quantum-secure tweakable block ciphers. We are interested in adversaries making queries with classical tweaks and a superposition of messages, a setting which corresponds to the attacks on OCB. In this setting, we prove the security of two constructions and notably propose the *key-tweak insertion* TBC, which requires a related-key secure block cipher. In Section 5 we define the new rate-one parallelizable quantum safe mode, QCB, and propose two instances: one using SATURNIN with the key-tweak insertion TBC and one using the dedicated TBC TRAX-L-17 [3]. We prove in Section 6 the security of QCB if it is used with a secure TBC. We use two notions: IND-qCPA [8] and BZ-unforgeability [7]. We discuss other possible definitions in Section 7. In particular, we show that the recent "qIND-qCPA" notion of [13] leads to an attack against all practical modes of operation.

2 Preliminaries

We open this section with standard notations for permutations, block ciphers and AEAD schemes. We also define the quantum oracle access that will be given to such a scheme in our proof. We recall some standard results and definitions related to quantum provable security. Finally, we introduce our new *linear post-processing* lemma (Lemma 2) that we will use in Section 3 and Section 7.

2.1 Definitions and Notations

We let $\mathcal{P}_n$ denote the set of permutations acting on $\{0,1\}^n$. By $x \xleftarrow{\$} S$ we mean that x is taken uniformly at random from the set S. We let $\mathcal{A}^{f(\cdot)} \Rightarrow b$ (resp. $\mathcal{A}^{f(\odot)} \Rightarrow b$) denote an algorithm that performs classical queries to oracle f (resp. quantum queries to f) and outputs b. We write $\mathcal{A}^{f^{\pm}(\cdot \text{ or } \odot)}$ when $\mathcal{A}$ has access to the f and the f^{-1} oracle, which we blend into a single oracle $f^{\pm}$.

Block Ciphers. A block cipher with key space $\{0,1\}^k$ and message space $\{0,1\}^n$ is a map $E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ such that for every key $K \in \{0,1\}^k$, $M \mapsto E(K,M)$ is a permutation of $\{0,1\}^n$. We let E_K denote the map $M \mapsto E(K,M)$. If E is a block cipher then its inverse is the map $E^{-1}\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ defined by $E^{-1}(K,C) = E_K^{-1}(C)$.

AEADs. An authenticated encryption scheme with associated data (AEAD) is specified by a tuple of sets $(\mathcal{K}, \mathcal{IV}, \mathcal{A}, \mathcal{M}, \mathcal{C})$ where $\mathcal{K}$ is the key space, $\mathcal{IV}$ is the IV space, $\mathcal{A}$ is the associated data space, $\mathcal{M}$ is the message space, and $\mathcal{C}$

is the ciphertext space, and a pair of deterministic algorithms $(\mathsf{Enc}, \mathsf{Dec})$ with signatures

$$\mathsf{Enc}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{A} \times \mathcal{M} \to \mathcal{C}$$
$$\mathsf{Dec}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{A} \times \mathcal{C} \to \mathcal{M} \cup \{\perp\}.$$

We require an AEAD scheme to be correct, *i.e.*, for all $(K, IV, A, M) \in \mathcal{K} \times \mathcal{IV} \times \mathcal{A} \times \mathcal{M}$,

$$\mathsf{Dec}\left(K, IV, A, \mathsf{Enc}\left(K, IV, A, M\right)\right) = M.$$

We write $\mathsf{Enc}_K\left(IV, A, M\right)$ for $\mathsf{Enc}\left(K, IV, A, M\right)$ and similarly $\mathsf{Dec}_K\left(IV, A, C\right)$. Note that this is the most generic definition of an AEAD, but in our case, we will replace the ciphertext space $\mathcal{C}$ by $\mathcal{C} \times \mathcal{T}$, and the scheme will output a ciphertext C of variable length and an authentication tag $T \in \mathcal{T}$ of fixed size. As we consider AEADs based on block ciphers, C and M will be cut into *blocks* that we index $M_0, \ldots, M_\ell$ (resp. $C_0, \ldots, C_\ell$) where ℓ is the block length of M (resp. of C).

Quantum Computing. In this paper, an *adversary* is a quantum algorithm that accesses one or more oracles. We use the quantum circuit model, whose basics can be found in [24]. A quantum algorithm is initiated with a set of m qubits (two-level quantum systems) in a fixed state $|0\rangle$. The state of the algorithm lies in a Hilbert space of dimension 2^m, with a canonical basis $\{|i\rangle, 0 \leq i \leq 2^m - 1\}$. Basic unitary operators, coined *quantum gates* (drawn from a universal gate set), are applied on the qubits. These computations are interleaved with oracle calls and *partial measurements*, which transform a *pure state* (an element of the Hilbert space) into a *mixed state* (a probability distribution of pure states).

2.2 Quantum Oracles and Query Model

We model *quantum oracle access* to any function $f : \mathcal{X} \to \mathcal{Y}$ as a unitary operation: $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$ (this is the *standard oracle*) or as $|x\rangle|y\rangle \mapsto (-1)^{y \cdot f(x)}|x\rangle|y\rangle$ (this is the *phase oracle*). Standard and phase oracles are well-known to be equivalent; that is, a single query to one can be emulated with a single query to the other.

Choice of IVs. During the AEAD calls, IVs are classical and distinct. The only difference here with previous works (*e.g.* [8, 2]) is that the IVs are not necessarily chosen at random. In the security games for AEAD that we will define and use in Section 6, we start the game by an initialization phase in which the adversary declares the IVs that he is going to query. This includes the case where the IVs are random, provided that no collision occurs (which would add a generic term to the adversary's advantage). This also includes the case where the IVs are generated using a counter. The rationale of this definition is that it is easier to reason with a fixed set of pre-declared IVs, and we found that it included most practical use cases that we could think of.

Quantum Query Model. The input plaintext and AD will be in superposition. Furthermore, the bit-length of the message, AD and ciphertext have to be chosen classically and cannot differ within a query; that is, we encrypt a superposition of messages of a fixed length. We let the adversary choose the bit-length of the message and AD in the queries between 0 and $n\ell$ for a fixed ℓ (which determines the maximal number of blocks to be queried). Thus, ℓ will intervene as a parameter in our bounds, together with the number of queries q.

Hence, our encryption and decryption oracles are actually families of unitary operators, indexed by these lengths and by the IV choice. As the ciphertext will be longer than the plaintext, we consider that the encryption oracles for messages of m bits outputs $c(m) > m$ bits. Conversely, messages of distinct lengths may be encrypted to ciphertexts of the same length. Hence, the decryption oracle of a ciphertext of c bits writes a canonical encoding of either the message or $\perp$ on c bits. We write these oracles $O^{m,a,IV}_{\mathsf{Enc}_K}$ and $O^{c,a,IV}_{\mathsf{Dec}_K}$ respectively, with $0 \leq m, a \leq \ell n$.

The encryption $O^{m,a,IV}_{\mathsf{Enc}_K}$ is a standard oracle for Enc_K with messages of length m, AD of length a and a fixed $IV \in \mathcal{IV}$:

$$\underbrace{|A\rangle}_{a \text{ qubits}} \quad \underbrace{|M\rangle}_{m \text{ qubits}} \quad \underbrace{|X\rangle}_{\substack{c(m) \\ \text{qubits}}} \mapsto |A\rangle\,|M\rangle \quad \underbrace{|X \oplus \mathsf{Enc}_K\,(IV, A, M)\rangle}_{c(m) \text{ qubits}} \quad .$$

The decryption $O^{c,a,IV}_{\mathsf{Dec}_K}$ is a standard oracle for Dec_K with ciphertexts of length c, AD of length a and a fixed IV:

$$\underbrace{|A\rangle}_{a \text{ qubits}} \quad \underbrace{|C\rangle}_{c \text{ qubits}} \quad \underbrace{|Y\rangle}_{c \text{ qubits}} \mapsto \begin{cases} |A\rangle\,|C\rangle\left|Y \oplus \widehat{M}\right\rangle \text{ if } C = \mathsf{Enc}_K\,(IV, A, M) \\ \qquad\qquad \text{with } \widehat{M} \text{ the encoding of } M \\ |A\rangle\,|C\rangle\left|Y \oplus \widehat{\perp}\right\rangle \text{ with } \widehat{\perp} \text{ the encoding of } \perp \end{cases}$$

Counting Data, Time and Memory. While the oracles authorize messages, AD and ciphertexts to take any number of bits, the modes that we will consider are built on block ciphers with a fixed block size n. Hence, we can count the data complexity in the number of blocks queried: a query to Enc_K or to $\mathcal{O}_{\mathsf{Enc}_K}$ with r blocks costs r data. We count the time complexity either in the number of quantum gates, or in the number of block cipher calls, as a quantum standard oracle. We consider the cost of a single block cipher call to be marginal with respect to the other terms, as it is polynomial in n, making these definitions equivalent. The memory will also be counted in n-bit registers, either classical or quantum.

2.3 Distances

Usually, in game-based definitions, the adversary's advantage is a difference in probabilities to output 1 or 0. However, since our adversaries are quantum, their final state is a quantum state. It is well-known that the *Euclidean distance* between quantum states is related to the distance between the distributions that

result from measuring these states. Thus, the probabilistic interpretation of the adversary's result (measuring 0 or 1) can be replaced by an Euclidean distance.

Definition 1 (Euclidean distance). *The Euclidean distance between* $|\phi\rangle = \sum \alpha_i |i\rangle$ *and* $|\psi\rangle = \sum \beta_i |i\rangle$ *is given by:* $\| |\phi\rangle - |\psi\rangle \| = \sqrt{\sum_i |\alpha_i - \beta_i|^2}$.

Two quantum states $|\phi\rangle = \sum \alpha_i |i\rangle$ and $|\psi\rangle = \sum \beta_i |i\rangle$, obtained after running an adversary in two different scenarios, incur two distributions $\mathcal{D}$ and $\mathcal{D}'$ over the states in the computational basis (we could also take another basis, without any change, since composing by a unitary operator leaves the distance unchanged). These distributions are such that $\mathcal{D}(i) = |\alpha_i|^2$ and $\mathcal{D}'(i) = |\beta_i|^2$. The *total variation distance* between $\mathcal{D}$ and $\mathcal{D}'$ is defined as $\sum_i |\mathcal{D}(i) - \mathcal{D}'(i)|$ and equal to $\sum_i ||\alpha_i|^2 - |\beta_i|^2|$. Then we have:

Lemma 1 ([5], Lemma 3.6). *If* $\| |\phi\rangle - |\psi\rangle \| \leq \epsilon$, *then* $\sum_i ||\alpha_i|^2 - |\beta_i|^2| \leq 4\epsilon$.

The decision of a quantum adversary to output 0 or 1 is conditioned only on its final state. Thus, if two adversaries have similar end states, they can only win with similar probabilities.

Corollary 1. *Let* $\mathcal{A}$ *be a quantum adversary that outputs a bit* b. *Let* $\mathcal{B}$ *be another adversary that also outputs a bit* b, *and let* $|\psi\rangle$ *and* $|\phi\rangle$ *be their respective states after the last oracle query, before measuring their output in the computational basis. Then:*

$$|\Pr[\mathcal{A}(\cdot) = 1] - \Pr[\mathcal{B}(\cdot) = 1]| \leq 4\| |\psi\rangle - |\phi\rangle \| \ .$$

In practice, we will consider a game in which some parameter is selected at random (e.g. the key K), then the game runs and the final state of the adversary depends on K. We are interested in the quantity $|\Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{A}(\cdot) = 1] - \Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{B}(\cdot) = 1]|$ which determines the difference in advantage between the two adversaries. We have: $\Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{A}(\cdot) = 1] = \sum_{k \in \mathcal{K}} \Pr[K = k] \Pr[\mathcal{A}(\cdot) = 1 | K = k]$. That is, we can write:

$$\begin{aligned} |\Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{A}(\cdot) = 1] - \Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{B}(\cdot) = 1]| &\leq \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} |\Pr[\mathcal{A}(\cdot) = 1 | K = k] - \Pr[\mathcal{B}(\cdot) = 1 | K = k]| \\ &\leq \frac{4}{|\mathcal{K}|} \sum_k \| |\psi_k\rangle - |\phi_k\rangle \| \end{aligned}$$

where $|\psi_k\rangle$ and $|\phi_k\rangle$ are the final states conditioned on the fact that the selected key is k. So in practice, we will fix all the random parameters, compute the euclidean distance between the end states and take the average.

2.4 Query magnitude

We will use a "query magnitude" argument, taken from [4]. Considering an oracle O with arbitrarily defined input and output registers, we modify O on a subset

D of its inputs to make the oracle O'. If an algorithm asks queries to O, but puts only "low amplitude" on the inputs of D, then changing O into O' does not have any significant impact on the final state.

Theorem 1 (Adapted from [4], Theorem 3.3). *Let $\mathcal{A}$ be a quantum algorithm that makes q queries to an oracle O and let $|\psi_0\rangle, \ldots, |\psi_q\rangle$ be the current state before each query ($|\psi_q\rangle$ is the final state). Let O' be an oracle that is the same as O, except on some subset D of its inputs, $\mathcal{A}'$ be the same as $\mathcal{A}$, except that every query to O is replaced by a query to O', and $|\psi'_i\rangle$ the state of $\mathcal{A}'$. Let P_D be the projector on the basis states x, a, y such that $x \in D$. Then:*

$$\| \, |\psi_q\rangle - \left|\psi'_q\right\rangle \, \| \leq 2 \sum_i |P_D(|\psi_i\rangle)| \ .$$

2.5 On Random Functions and Permutations

We will use the following results from the literature. First of all, as shown by Zhandry, it is impossible to distinguish a random function with n-bit domain from a random permutation with probability bigger than $\mathcal{O}\left(\frac{q^3}{2^n}\right)$ with q queries (where the constant in the $\mathcal{O}$ is fixed by the theorem); and conversely. We refer to this statement as *PRF-PRP switching*.

Theorem 2 ([29], Theorem 3.1). *Let $h : \{0,1\}^n \to \{0,1\}^m$ be a random function. Any quantum algorithm making q quantum queries to h can only find a collision with probability at most $\mathcal{O}\left(\frac{q^3}{2^m}\right)$. If $n \leq m$, then any quantum algorithm making q queries cannot distinguish h from a random injective function except with probability $\mathcal{O}\left(\frac{q^3}{2^m}\right)$.*

Second, we use a theorem by Boneh and Zhandry that shows that a quantum algorithm making q queries to a random oracle with a domain of exponential size can only output $q+1$ valid $\{\text{input}, \text{output}\}$ pairs with negligible probability.

Theorem 3 ([7], Theorem 4.1). *Let $\mathcal{A}$ be a quantum algorithm making q queries to a random oracle $h : \{0,1\}^n \to \{0,1\}^m$, and producing $k > q$ pairs $(x_i, y_i) \in \{0,1\}^n \times \{0,1\}^m$. The probability that the x_i are distinct and $y_i = h(x_i)$ for all $1 \leq i \leq k$ is at most:*

$$\frac{1}{2^{mk}} \sum_{r=0}^{q} \binom{k}{r} (2^m - 1)^r \ .$$

If $k = q+1$ then the adversary succeeds with probability at most $\frac{q+1}{2^m}$.

We will use the terminology "$(q, q+1)$ security game" to refer to the game in which $\mathcal{A}$ accesses O_h q times and must produce $q+1$ valid pairs. An alternative proof of Theorem 3 for the $q, q+1$ case can be found in the full version of [1]. By combining this theorem with Theorem 2, we obtain a similar statement for random permutations.

Corollary 2. *There exists a constant c such that, if $\mathcal{A}$ is a quantum algorithm making q queries to a random permutation $\Pi : \{0,1\}^n \to \{0,1\}^n$ and trying to produce $q+1$ valid input-output pairs, then $\mathcal{A}$ can only suceed with probability at most: $c\frac{q^3}{2^n}$.*

The term in Corollary 2 is simply the sum of the PRP-PRF distinguishing advantage and the $(q, q+1)$ advantage. The former grows much faster with q, but we will mostly use Corollary 2 with a single query, where both terms are $\mathcal{O}\left(2^{-n}\right)$.

2.6 Computing a Linear Function of a Quantum Oracle

In [16] Hosoyamada and Sasaki show that given quantum oracle access for a function:

$$|x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$$

it is possible to compute the *truncation* of the output $f(x)$ on some bits and make a quantum query to $\mathsf{Trunc}(f(x))$ using only one quantum query to f. We now extend this result, and show that it is possible to compute *any linear function* of the output using only one quantum query. This is especially important with the oracles we will be using, since they involve nonces that are changed at each new quantum query.

The core observation in [16] is simple: the state $|0\rangle + |1\rangle$ is invariant whether we XOR a 0 or a 1 on it. Hence, before the query, in the output register, we can set the qubits we want to drop to $|0\rangle + |1\rangle$ and the qubits we want to keep to $|0\rangle$. We will now extend this result, with the following lemma:

Lemma 2 (Computing a linear function of a quantum oracle). *Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function, $O_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$. Let $g : \{0,1\}^m \to \{0,1\}^o$ be an $\mathbb{F}_2$-linear function. Then it is possible to construct the oracle $O_{g\circ f} : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus (g \circ f)(x)\rangle$ using a* single *query to O_f.*

Proof. Let O_g be a quantum oracle that implements g, assume we are given the quantum state

$$|x\rangle |y\rangle$$

We first add an ancilla register containing the uniform superposition on m bits. We then have the state

$$|x\rangle |y\rangle \sum_{z=0}^{2^m-1} |z\rangle$$

Then, we apply O_g with register z as input and y as output, and we get

$$|x\rangle \sum_{z=0}^{2^m-1} |y \oplus g(z)\rangle |z\rangle$$

Then, we apply O_f with register x as input and z as output. We get

$$|x\rangle \sum_{z=0}^{2^m-1} |y \oplus g(z)\rangle\, |z \oplus f(x)\rangle$$

Finally, we reapply O_g with register z as input and y as output. We get

$$|x\rangle \sum_{z=0}^{2^m-1} |y \oplus g(z) \oplus g(z \oplus f(x))\rangle\, |z \oplus f(x)\rangle$$

As g is linear, we have $g(z) \oplus g(z \oplus f(x)) = g(f(x))$. Hence, the state can be rewritten as

$$|x\rangle\, |y \oplus g(f(x))\rangle \sum_{z=0}^{2^m-1} |z \oplus f(x)\rangle$$

This state can then be simplified, as the z register contains the uniform superposition over m bits, independently of the value of $f(x)$, to

$$|x\rangle\, |y \oplus g(f(x))\rangle \sum_{z=0}^{2^m-1} |z\rangle$$

We can now remove the z register, as it is not entangled with the others, and obtain the quantum state we wanted. □

Remark 1. Lemma 2 can also be applied if the quantum oracle to f uses a group law different from $\oplus$ to update its output register. In that case, g shall be a linear function for the corresponding group law.

3 Offsets don't work

In this section we start by recalling the superposition attacks on OCB from [18]. We will next present a first attempt to repair it, that consisted on tweaking the value of the *offsets*, along with the new original superposition attack that shows that any offset-based variant can be broken by Simon attacks.

3.1 Simon's attack on OCB

OCB[e] [19] is one of the most influential authenticated modes. OCB3 is represented on Figure 1, with $\Delta_i = \text{gray}(i) \cdot E_K(0^n)$ (using a finite field multiplication) and $\Delta_i^{IV} = \Delta_i \oplus F_K(IV)$, with F a simple function of K and IV and $\text{gray}(i)$ the gray encoding of i.

OCB3 is classically proven secure if its underlying cipher is a strong PRP.

[e] Three versions of OCB have been proposed. We focus here on the last one, OCB3, while all three suffer from similar superposition attacks.

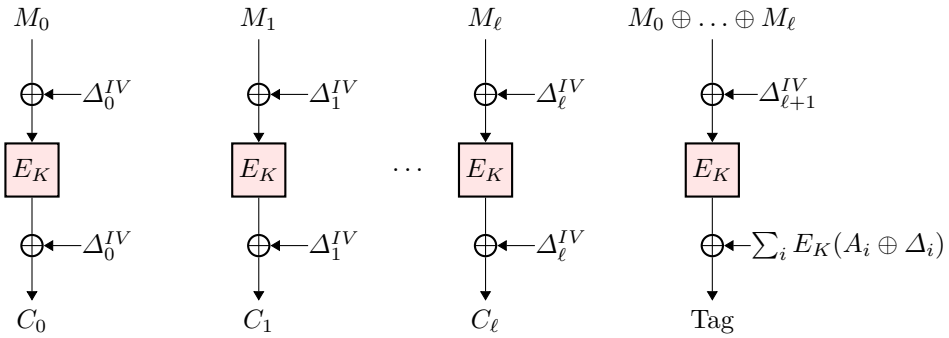

Fig. 1. OCB3. (M_i) is the message, (A_i) is the associated data.

Simon's algorithm. Simon's algorithm, proposed in [28] allows to solve efficiently, with a complexity of $\mathcal{O}(n)$, the following problem when we are allowed to ask superpositions queries to $\mathcal{F}$:

Given a Boolean function $\mathcal{F}$ on n bits and the promise that there exists s such that, for any $x \neq y$, $\mathcal{F}(x) = \mathcal{F}(y) \iff x = y \oplus s$, find s.

Simon's algorithm recover a vector orthogonal to the period with a single quantum query; with $\mathcal{O}(n)$ queries, the period is deduced with linear algebra. It still works if the promise is partially fulfilled, that is we may have $f(x) = f(y)$ and $x \neq y \oplus s$, as shown for example in [18]. For comparison, classically, the best algorithm requires $\Omega\left(\sqrt{2^n}\right)$ queries.

Quantum Superposition Attacks on OCB. Two polynomial-time attacks against OCB that require quantum superposition queries to the construction were proposed in [18]. They both use Simon's algorithm[f].

The main weakness of OCB is that the nonce only influences the construction through the value Δ, which is XORed to the internal state. The scenario of the attack considers that the attacker has access to a superposition oracle that given a superposition of messages as input, returns the superposition of their encryption. The key is a secret value and the nonce is different for each query.

The first attack considers an empty message, and two variable identical blocks of associated data. The output is then

$$E_K(IV) \oplus E_K(x \oplus \Delta_1) \oplus E_K(x \oplus \Delta_2)$$

This function is periodic, of period $\Delta_1 \oplus \Delta_2$. The function we can query each time is nonce-dependent, but the period is not. This allows to use Simon's algorithm to recover the period.

The second attack uses the same idea, but attacks the encryption part and not the authentication. Its core idea is to consider the xor of two distinct

[f] One attack on OCB presented in [18] was partial, as it assumed without any mention the use of Lemma 2.

blocks i and j that encrypt the same message block. This is equal to $f_{i,j}(x) = E_K(\Delta_i \oplus x) \oplus E_K(\Delta_j \oplus x) \oplus \Delta_i \oplus \Delta_j$.

This function is periodic, of period $\Delta_i \oplus \Delta_j = (\text{gray}(i) \oplus \text{gray}(j)) \cdot E_K(0^n)$. We can then use Simon's algorithm, and this time we need to use Lemma 2 to compute the XOR of two blocks using only one query.

Both attacks recover the difference of two offsets, which is sufficient to make some forgeries.

3.2 A First (Failed) Attempt to Fix OCB

In order to make OCB quantum-resistant, one can try to avoid those attacks by making the influence of the encrypted nonce different for each block, such that it is not possible to have a nonce-independent period. For instance, Δ_i could be changed to a multiple of $E_K(IV)$: $\Delta_i = i \cdot E_K(IV)$.

This way, the previous attack could only recover one bit of $E_K(IV)$ at a time, which is useless if the nonce changes for each query.

New superposition attack for any nonce-based solution. Actually, the previous proposal is still unsafe, but it requires a new more advanced attack that we present here. This evolved attack is inspired by the multiple-period attacks from [9]. Its core idea is to leverage the possibility to encrypt a long message to construct multiple copies of the periodic function, in such a way that one query will likely be enough to recover all the bits of the period.

Let g be the function that maps the sequence $(x_1, x_2, \ldots, x_{2n-1}, x_{2n})$ to $(x_1 \oplus x_2, x_3 \oplus x_4, \ldots, x_{2n-1} \oplus x_{2n})$.

We consider the function

$$f(x_1, \ldots, x_n) = g \circ \text{OCB}(x_1, x_1, x_2, x_2, \ldots, x_n, x_n)$$

Reusing the notation $f_{i,j}(x) = E_K(\Delta_i \oplus x) \oplus E_K(\Delta_j \oplus x) \oplus \Delta_i \oplus \Delta_j$, we have

$$f(x_1, \ldots, x_n) = (f_{1,2}(x_1), f_{3,4}(x_2), \ldots f_{2n-1,2n}(x_n))$$

Hence, Simon's algorithm allows us to sample one vector orthogonal to each of the periods of the involved $f_{i,j}$. As these periods are linearly dependent, this is enough to recover completely the value $E_K(IV)$, assuming n is large enough.

Conclusion. This attack shows that a solution based on offsets is unlikely to work. After this failed attempt, we decided to move one step backwards. OCB can be seen as an instantiation of the mode TAE or ΘCB, which is defined with a *Tweakable Block Cipher* (TBC). The TBC used in OCB is the LRW mode [21], which builds upon a block cipher, and is quantumly broken [18]. The attacks that we gave all seem to stem from the TBC itself, not the mode.

4 Quantum-secure Tweakable Block Ciphers

In this section, we define quantum-secure tweakable block ciphers (TBCs). We give two TBC constructions based on a block cipher. The first one requires a pseudorandom permutation (PRP) security assumption; the second one requires the *ideal cipher model*, which we will recall below. Both entail quantum security guarantees which we will explicit.

4.1 Definitions

Definition 2. *Let E be a block cipher. Let $\mathcal{A}$ be an oracle algorithm (making either classical or quantum queries depending on the case) which outputs a bit. The advantage of $\mathcal{A}$ against the PRP and Strong PRP (SPRP) security of E is defined as:*

$$\mathbf{Adv}^{\mathrm{PRP}}_{E(*)}(\mathcal{A}) := \left| \Pr_{K \xleftarrow{\$} \{0,1\}^k} [\mathcal{A}^{E_K(*)} \Rightarrow 1] - \Pr_{\Pi \xleftarrow{\$} \mathcal{P}_n} [\mathcal{A}^{\Pi(*)} \Rightarrow 1] \right|$$

$$\mathbf{Adv}^{\mathrm{SPRP}}_{E(*)}(\mathcal{A}) := \left| \Pr_{K \xleftarrow{\$} \{0,1\}^k} [\mathcal{A}^{E^{\pm}_K(*)} \Rightarrow 1] - \Pr_{\Pi \xleftarrow{\$} \mathcal{P}_n} [\mathcal{A}^{\Pi^{\pm}(*)} \Rightarrow 1] \right|$$

Depending on the access that the adversary has (classical or quantum) to the messages, we replace the $*$ symbol by $\cdot$ (classical) or $\odot$ (quantum).

Tweakable Block Ciphers. A *tweakable block cipher (TBC)* with key space $\{0,1\}^k$, tweak space $\{0,1\}^t$, and message space $\{0,1\}^n$ is a map $\widetilde{E}\colon \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ such that for every key $K \in \{0,1\}^k$ and every tweak $T \in \{0,1\}^t$, $M \mapsto \widetilde{E}(K,T,M)$ is a permutation of $\{0,1\}^n$. We let $\widetilde{E}_K$ denote the map $(T,M) \mapsto \widetilde{E}(K,T,M)$. If $\widetilde{E}$ is a TBC then its inverse is the map $\widetilde{E}^{-1}\colon \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ defined by $\widetilde{E}^{-1}(K,T,C)$ being the unique M such that $\widetilde{E}(K,T,M) = C$. A *tweakable permutation* with tweak space $\{0,1\}^t$ and message space $\{0,1\}^n$ is a map $\widetilde{\Pi}\colon \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ such that for every tweak $T \in \{0,1\}^t$, $M \mapsto \widetilde{\Pi}(T,M)$ is a permutation of $\{0,1\}^n$. We let $\widetilde{\mathcal{P}}_{t,n}$ denote the set of all tweakable permutations with tweak space $\{0,1\}^t$ and message space $\{0,1\}^n$.

Definition 3. *Let $\mathcal{A}$ be an oracle algorithm making (classical or quantum) queries and which outputs a bit. The advantage of $\mathcal{A}$ against the TPRP, resp. strong TPRP (STPRP) security of $\widetilde{E}$ is defined as*

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}}(\mathcal{A}) := \left| \Pr_{K \xleftarrow{\$} \{0,1\}^k} [\mathcal{A}^{\widetilde{E}_K(*,*)} \Rightarrow 1] - \Pr_{\widetilde{\Pi} \xleftarrow{\$} \widetilde{\mathcal{P}}_{t,n}} [\mathcal{A}^{\widetilde{\Pi}(*,*)} \Rightarrow 1] \right|$$

$$\mathbf{Adv}^{\mathrm{STPRP}}_{\widetilde{E}}(\mathcal{A}) := \left| \Pr_{K \xleftarrow{\$} \{0,1\}^k} [\mathcal{A}^{\widetilde{E}^{\pm}_K(*,*)} \Rightarrow 1] - \Pr_{\widetilde{\Pi} \xleftarrow{\$} \widetilde{\mathcal{P}}_{t,n}} [\mathcal{A}^{\widetilde{\Pi}^{\pm}(*,*)} \Rightarrow 1] \right|.$$

Depending on the access that the adversary has (classical or quantum) to the messages and to the tweaks, we replace the $*$ symbols by $\cdot$ (classical) or $\odot$ (quantum).

Pre-declaration of Tweaks. In the proofs of this section, we consider TBCs queried in superposition over the message space and *classically* over the tweaks space. We formalize this as follows: the adversary is given access to a family of standard oracles for $\widetilde{E}_K^{\pm}(T, \odot)$ indexed by the tweak space. Before each oracle call, she performs a partial measurement on her current state, extracts a classical tweak value and sets this value for the call. This is the most general setting. In our proofs, we consider a more specific case of *pre-selected tweaks*. Before the first oracle call, the adversary declares a set of tweak values $\{T_1, \ldots, T_m\}$. While running, she chooses her tweaks only in this set. Thus, the bounds that we will obtain will depend on m (the total number of available tweaks) and on the total number of queries q made by the adversary. Note that the two are independent, as tweaks may be reused and the adversary may declare more tweaks than needed. We use the notation $\mathbf{Adv}^{(\mathrm{S})\mathrm{TPRP}}_{\widetilde{E}(\cdot,\odot)}(\mathcal{A})$ for this restricted case.

TBCs from Block Ciphers. In this section, we will define and construct TBCs from block ciphers. For some of these constructions, we will prove security in the *ideal cipher model*. In the quantum setting, this model was previously considered by Hosoyamada and Yasuda [17] to analyze the Davies-Meyer and Merkle-Damgard constructions. This means that the underlying block cipher E is chosen uniformly at random from the set $\mathcal{BC}_{k,n}$ of all block ciphers with key space $\{0,1\}^k$ and message space $\{0,1\}^n$ at the beginning of the (S)TPRP distinguishing game and the adversary is allowed to make quantum queries to $E^{\pm}$ (specifying the key and the plaintext/ciphertext). The advantage is then defined as

$$\mathbf{Adv}^{(\mathrm{S})\mathrm{TPRP}}_{\widetilde{E}}(\mathcal{A}) := \left| \begin{array}{c} \Pr[\mathcal{A}^{\widetilde{E}_K^{(\pm)}(*,*),E_{\odot}^{\pm}(\odot)} \Rightarrow 1] \\ {\scriptstyle K \xleftarrow{\$} \{0,1\}^k} \\ {\scriptstyle E \xleftarrow{\$} \mathcal{BC}_{k,n}} \end{array} - \begin{array}{c} \Pr[\mathcal{A}^{\widetilde{\Pi}^{(\pm)}(*,*),E_{\odot}^{\pm}(\odot)} \Rightarrow 1] \\ {\scriptstyle \widetilde{\Pi} \xleftarrow{\$} \widetilde{\mathcal{P}}_{t,n}} \\ {\scriptstyle E \xleftarrow{\$} \mathcal{BC}_{k,n}} \end{array} \right| .$$

(Note that the adversary has access to $E^{\pm}$ even in the non-strong TPRP definition.)

4.2 Impossibility results

In order to illustrate the difficulties of building a quantum-secure TBC, even in a weak sense, let us first consider a few examples.

LRW. The LRW mode [21] uses an almost 2-xor universal hash function family $\mathcal{H}$ and adds $h \in \mathcal{H}$ to the key.

$$\widetilde{E}_{K,h}(T,x) = E_K(h(T) \oplus x) \oplus h(T)$$

An ϵ-almost 2-XOR universal hash function family $\mathcal{H}$ is such that for all x, y, z with $x \neq y$, the probability of $h(x) \oplus h(y) = z$ is small (less than ϵ) when h is chosen at random. Classically, it is a strong TBC, meaning security against adaptive chosen-ciphertext attacks.

However, the LRW mode is not a quantum-secure TBC even if we allow only classical queries to the tweaks. This was shown in [18], with an attack that is close to the OCB attacks: by querying only two classical tweaks T_0, T_1, one can build a function: $f(x) = E_k(h(T_0) \oplus x) \oplus h(T_0) \oplus E_k(h(T_1) \oplus x) \oplus h(T_1)$ which is periodic, of period $h(T_0) \oplus h(T_1)$. Using Simon's algorithm, we can recover the period of this function in $\mathcal{O}(n)$ queries. This provides a powerful distinguisher, as this property is extremely unlikely with random permutations. Note that this distinguisher still applies for any function h, even if it is an unknown qPRF.

The CMT Mode. We also consider the CMT (*CBC-MAC Tweaked*) mode from [21]: $\widetilde{E}_K(T,x) = E_K(T \oplus E_K(x))$. It is proven in [21] to be a secure tweakable block cipher (indistinguishable from a family of random permutations), but not *strong* (where we would have also access to the inverse). The issue with this construction is that it uses two block cipher calls for one TBC call, which is inefficient regarding our application in mind (a rate-one AEAD).

Proposition 1 (Theorem 1 in [21]). *Let $\mathcal{A}^{\widetilde{E}(\cdot,\cdot)}$ be an adversary making q queries, and distinguishing between the actual $\widetilde{E}_K$ for a random K and a family of random permutations Π_T. Then:*

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}(\cdot,\cdot)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{PRP}}_{E}(2q) + \frac{17q^2 - q}{2^{n+1}} \quad .$$

Thus, the CMT mode is a secure TBC construction if E is a PRP.

The classical proof of security [21] uses the classical proof for CBC-MAC, but CBC-MAC was also attacked in [18] using Simon's algorithm. In fact, with superposed tweaks, there exists a trivial distinguisher: choose two messages x_1, x_2 and call $\widetilde{E}_K(T,x_1) \oplus \widetilde{E}_K(T,x_2)$ in superposition over T, recover the hidden period $E_K(x_1) \oplus E_K(x_2)$ with Simon's algorithm.

Remark 2. CMT is not a Strong TPRP, as the following (known) attack distinguishes it from a family of PRPs within two encryption and decryption queries:

1. encrypt the same x under tweaks T_1 and T_2, obtain $C_1 = E_K(T_1 \oplus E_K(x)), C_2 = E_K(T_2 \oplus E_K(x))$;
2. decrypt C_1 under tweak T_2 and C_2 under tweak T_1.

With the CMT mode, one obtains twice the same value: $E_K^{-1}(T_1 \oplus T_2 \oplus E_K(x))$.

Key-tweak Insertion. We will consider the *key-tweak insertion TBC*, built from a block cipher E as: $\widetilde{E}_K(T, M) = E_{K \oplus T}(M)$. As the CMT mode, it admits a simple distinguisher based on Simon's algorithm if the tweaks are queried in superposition: this is the *quantum related-key attack* of [26]. It consists of emulating access to the function $f(\odot) = E_{K \oplus \odot}(0) \oplus E_{\odot}(0)$ which admits K as a period, and using Simon's algorithm again.

Despite these negative results, we will now prove the security of the CMT and key-tweak insertion TBCs if they are queried with classical, random or non-adaptive tweaks.

4.3 Proofs

Let $\widetilde{E}_K(T, x)$ denote $E_K(T \oplus E_K(x))$, the CMT mode. In Appendix A, we prove its security when $\widetilde{E}_K(T, x)$ is accessed in superposition over x, but with classical, pre-selected tweaks only, that we formalize in the following proposition:

Proposition 2. *Let $\mathcal{A}^{\widetilde{E}(\cdot,\odot)}$ be an adversary making q queries to $\widetilde{E}$ (or a random permutation family), with a set of tweaks of size m, and distinguishing between the actual $\widetilde{E}_K$ for a random K and a family of random permutations Π_T. Then:*

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}(\cdot,\odot)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{qPRP}}_{E}(2q) + \mathcal{O}\left(\left(\frac{q^3 m}{2^{n/2}}\right)^{2/3}\right) \quad .$$

Next, we consider $\widetilde{E}^{\pm}_K(t, x) = E^{\pm}_{K \oplus t}(x)$, the *key-tweak insertion TBC*. We need here the ideal cipher model: E is selected at random from all ciphers. (The probabilities are taken on average on E, but we omit this average for simplicity.) In Appendix B, we prove the Strong TPRP security of this TBC when queried under classical pre-selected tweaks with the following proposition:

Proposition 3. *Let $\mathcal{A}$ be an adversary making q queries to $\widetilde{E}^{\pm}$ and q' queries to $E^{\pm}$, with a set of tweaks of size m. Then:*

$$\left| \Pr_{K \xleftarrow{\$} \mathcal{K}}[\mathcal{A}^{\widetilde{E}^{\pm}_K(\cdot,\odot), E^{\pm}_{\odot}(\odot)} \Rightarrow 1] - \Pr_{\{\Pi_T\} \xleftarrow{\$} \mathcal{P}_n}[\mathcal{A}^{\Pi^{\pm}(\cdot,\odot), E^{\pm}_{\odot}(\odot)} \Rightarrow 1] \right| \leq 8\sqrt{\frac{mq'^2}{2^n}} \quad .$$

5 Definition of QCB

In this section, we describe the QCB mode, an AEAD based on a Tweakable Block Cipher. It is similar to the TAE mode [20, 21] and to ΘCB [25, 19]. Throughout this section, $\widetilde{E}_{k,t}$ will denote a TBC used with key k and tweak t, of block size n. We separate the tweak space in a cartesian product: $\mathcal{T} = \mathcal{D} \times \mathcal{IV} \times \mathcal{L}$. Thus, tweaks are triples (D, IV, j) where D is a *domain separator*, IV will be an IV, and j will be a block index. Only 5 values of domain separator need to be used.

The mode is defined in Algorithm 1 and represented on Figure 2 and Figure 3. When the message and AD are cut in blocks, the last block (M_* and a_* respectively) may be empty. We define the padding scheme $\text{pad}(M_*)$ as appending $10*$ (a 1 followed by as many zeroes as necessary to fill the block). Note that due to the padding and structure of QCB, the ciphertext C is always longer than the plaintext M (by n bits at most).

Algorithm 1 QCB

Input: message M, associated data A, IV, key K
Requirements: Initialization vectors should not be reused
Output: ciphertext C, tag T

Pad the initialization vector if necessary
Split M into full blocks $M_0, M_1, \dots M_\ell$ and a final block M_* (partial or empty)
Split A into $A_0, A_1, \dots A_j, A_*$
for all $i = 0$ to ℓ **do**
    $C_i \leftarrow \widetilde{E}_{K,(0,IV,i)}(M_i)$ ▷ Encryption of block i
end for
$C_* \leftarrow \widetilde{E}_{K,(1,IV,\ell)}(\text{pad}(M_*))$ ▷ Encryption of the final block
$T \leftarrow 0$
for all $i = 0$ to j **do**
    $T \leftarrow T \oplus \widetilde{E}_{K,(2,IV,i)}(A_i)$ ▷ Absorb AD block i
end for
$T \leftarrow T \oplus \widetilde{E}_{K,(3,IV,j)}(\text{pad}(A_*))$ ▷ Absorb the final AD block
$T \leftarrow T \oplus \widetilde{E}_{K,(4,IV,\ell)}\,(M_0 \oplus \dots \oplus M_\ell \oplus \text{pad}(M_*))$
return $C = (C_0 \| C_1 \| \dots \| C_\ell \| C_*), T$

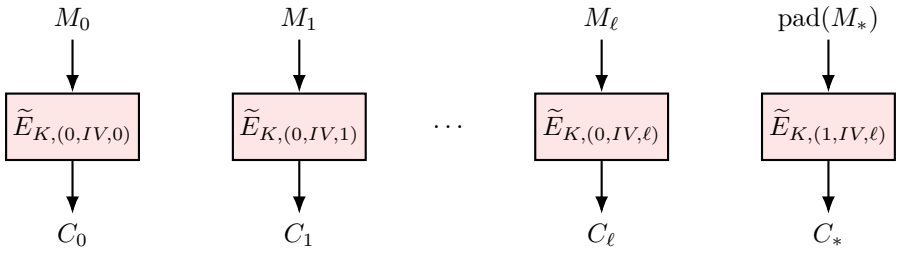

Fig. 2. QCB, encryption.

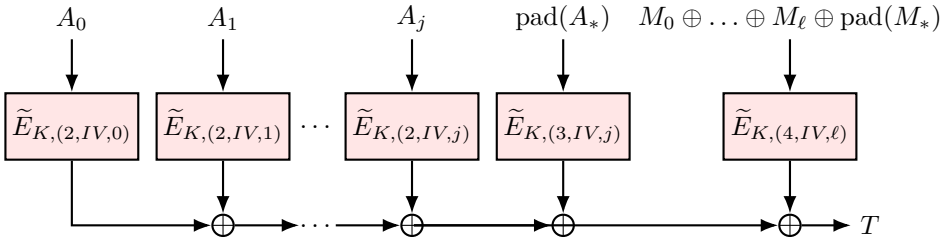

Fig. 3. QCB, processing of the associated data and computation of the tag.

Avoiding Quantum Attacks. We include the IV in the tweak when processing the AD, because otherwise there is a quantum attack based on Deutsch's algorithm (see Appendix C). In Section 6, we will prove that QCB is secure assuming a weak quantum-secure TBC. We will use the following property, which follows from its definition.

Proposition 4 (Number of tweaks (informal)). *For a given IV, there exists a set of tweaks $T(IV)$ of size $|T(IV)| = 5\ell$ such that any QCB query comprised of at most ℓn (included) bits of AD and ℓn bits of message can only reach tweaks in the set $T(IV)$.*

Proof. The tweaks are of the form (d, IV, i) where i is a block number between 0 and ℓ (included) and d a domain separator that takes 5 values. □

Instantiation with Saturnin: Saturnin-QCB. We propose to instantiate QCB with the block cipher Saturnin [12], a second-round candidate of the NIST LWC process [23]. Saturnin has 256-bit blocks and keys. In addition, the cipher admits a *domain separator* D of 4 bits. The other modes of operation of the Saturnin submission use values from 0 to 8 included, so we use $D = 9, 10, 11, 12$ and 13 in Algorithm 1. More precisely, the authors of [12] define a variant of Saturnin with 16 *Super-rounds* aiming at an increased security margin in the related-key scenario, denoted $\textsc{Saturnin}_{16}$. We define: $\widetilde{E}_{k,(D,IV,i)}(x) = \textsc{Saturnin}^D_{16}(k \oplus (IV||i), x)$, where we use the key-tweak insertion construction of Section 4 to turn $\textsc{Saturnin}^D_{16}$ into a TBC with 256-bit tweaks. The IV and the block number are simply concatenated. We use IVs of at most 160 bits and authorize up to 2^{95} blocks of data. Note that this construction motivates further inquiry of related-key attacks, as it can only be secure if $\textsc{Saturnin}_{16}$ is related-key secure.

Instantiation with a Dedicated TBC: TRAX-QCB. Block ciphers of 256 bits seem more convenient for post-quantum security. However, they are relatively rare (for example, Saturnin is the only such one in the LWC standardization process). Fortunately, it is possible to instantiate QCB with a dedicated TBC with 256-bit blocks, the TRAX-L-17 cipher of [3]. It has smaller tweaks of 128 bits, contrary to the key-tweak-insertion TBC with Saturnin, but it has the

advantage of being a dedicated design, with possibly a better security than the tight bound for the key-tweak-insertion. 128 bits allow to fit the 3 bits required for domain separation, 80 bits of IV and 45 bits of block numbering. Thus we can encrypt at most $2^{45} - 1$ blocks of plaintext and AD.

6 Security of QCB

The significant quantum attacks on modes of operation either recover the key or create forgeries using Simon's algorithm. We will show that such attacks cannot apply to QCB if the underlying TBC is weakly quantum-secure (*i.e.* secure under classical queries to the tweaks). We show that:

- QCB is IND-qCPA secure (Section 6.2): an adversary making quantum encryption queries cannot distinguish between the encryptions of two classical challenge messages;
- QCB is BZ-unforgeable (Section 6.3): an adversary making q quantum encryption queries cannot output $q + 1$ valid $\{\text{IV}, \text{AD}, \text{ciphertext}, \text{tag}\}$ quadruples.

Note that we discuss other possible (and impossible) security definitions in Section 7. Note also that in this section, we consider that an adversary makes q queries of maximal block length b each, and chooses the block length adaptively.

6.1 Definitions

In all our definitions, the adversary makes q superposition queries with distinct pre-declared IVs. The messages and ADs both have a maximal length of ℓ complete blocks, and we will bound the advantage depending on q and ℓ. We will use superscripts for separate queries, and subscripts for individual blocks within a query.

IND-qCPA. First of all, we recall the definition of the IND-qCPA security game from [8]. In [8], each call to the encryption oracle contains randomness. We extend slightly this definition by making the adversary capable of choosing his IVs. However, we request this choice to be non-adaptive. Thus, the adversary specifies at the start of the game the sequence of IVs that she is going to use. In practice these IVs can either be specified by a counter or chosen at random.

IND-qCPA game

Key generation: $K \xleftarrow{\$} \mathcal{K}, b \xleftarrow{\$} \{0,1\}$
Initialization: $\mathcal{A}$ sends to the challenger a sequence of distinct IVs: $\{IV^1, \ldots, IV^q\}$, one for each subsequent query

---

$\mathcal{A}$ can perform challenge queries and encryption queries. At the k^{th} query, the current IV is IV^k.

Challenge queries: $\mathcal{A}$ chooses two $\{\text{message}, \text{AD}\}$ pairs (M^0, A^0), (M^1, A^1) of the same length and sends them to the challenger. The challenger encrypts (IV, M^b, A^b) with the current IV and returns the result.

Encryption queries: $\mathcal{A}$ chooses a message and AD pair (M, A) length, the encryption oracle encrypts (IV, M, A) to $\mathcal{A}$ with the current IV.

Guess: $\mathcal{A}$ outputs a bit b' and wins if $b = b'$.

For each query, the message and AD length are chosen between 0 and ℓn bits for a fixed ℓ (superposed messages must have the same length).

The IND-qCPA advantage of an adversary $\mathcal{A}$ against an AEAD E is defined as:

$$\mathbf{Adv}_E^{\text{IND-qCPA}}(\mathcal{A}) = \left| \Pr\left[\mathcal{A} \text{ succeeds}\right] - \frac{1}{2} \right|$$

BZ. We define our unforgeability game, which we name "Boneh-Zhandry" (BZ) by analogy with the definition of unforgeability of [7] (which initially concerns MACs).

BZ game

Key generation: $K \xleftarrow{\$} \mathcal{K}$

Initialization: $\mathcal{A}$ sends to the challenger a sequence of distinct IVs: $\{IV^1, \ldots, IV^q\}$, one for each subsequent query

---

Encryption queries: $\mathcal{A}$ chooses a message and AD length, the encryption oracle is called on the input registers of $\mathcal{A}$, with the current IV.

Forgeries: $\mathcal{A}$ produces $q+1$ quadruples $\{A, IV, C, T\}$ and succeeds if all these quadruples are valid, that is, for all quadruples, there exists an M such that the encryption of (IV, M, A) is (C, T).

6.2 IND-qCPA Security

Theorem 4. *Let* $\mathsf{QCB}[\widetilde{E}]$ *denote the QCB function with oracle access to the tweakable blockcipher* $\widetilde{E}$*, and let* $\mathsf{QCB}[\widetilde{\Pi}]$ *be the same function with oracle access to an ideal tweakable random permutation* $\widetilde{\Pi}$*. We consider adversaries making* q *queries of block length* $\leq \ell$ *to* $\mathsf{QCB}[\widetilde{E}]$*, then we have:*

$$\mathbf{Adv}^{\text{Ind-qCPA}}_{\mathsf{QCB}[\widetilde{E}]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}(\cdot,\odot)}(5\ell q) \tag{1}$$

where we take the maximal advantage over all adversaries querying $\widetilde{E}(\cdot,\odot)$ with at most $5\ell q$ pre-declared tweaks. In the case of the key-tweak insertion TBC of Section 4, we consider adversaries making also q' queries to $E^{\pm}$ and we have:

$$\mathbf{Adv}^{\text{Ind-qCPA}}_{\mathsf{QCB}[\widetilde{E}]}(\mathcal{A}) \leq \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}(\cdot,\odot),E_{\odot}(\odot)}(5\ell q, q') \leq 8\sqrt{\frac{5\ell q q'^2}{2^n}} \ . \tag{2}$$

Proof. Suppose $\mathcal{A}$ is an adversary trying to break the IND-qCPA security of $\mathsf{QCB}[\widetilde{E}]$. $\mathcal{A}$ performs q encryption or challenge queries of maximum block length ℓ (the exact bit length of the queries can be chosen freely in the range $0,\ldots,n\ell$). If we are in the ideal cipher model, let q' be the number of queries done to E^{pm}. Consider the query number i made to QCB (encryption or challenge). From Proposition 4, in this query, the tweakable block cipher $\widetilde{E}$ is queried with tweaks in the set $T(IV^i)$ having a fixed size $|T(IV^i)| = 5\ell$.

We can therefore see $\mathcal{A}$ as an algorithm performing at most $q\ell$ queries to $\widetilde{E}$, with each tweak lying in the fixed set $T = \cup_{i=1}^{q} T(IV^i)$ with $|T| \leq 5q\ell$. If we replace $\widetilde{E}$ with $\widetilde{\Pi}$ for a random $\widetilde{\Pi}$, we get:

$$\left|\mathbf{Adv}^{\text{Ind-qCPA}}_{\mathsf{QCB}[\widetilde{E}]}(\mathcal{A}) - \mathbf{Adv}^{\text{Ind-qCPA}}_{\mathsf{QCB}[\widetilde{\Pi}]}(\mathcal{A})\right| \leq \mathbf{Adv}^{\text{TPRP}}_{\widetilde{E}(\cdot,\odot)}(5\ell q, q') \ . \tag{3}$$

Finally, consider an adversary $\mathcal{A}$ playing an IND-qCPA game with $\mathsf{QCB}[\widetilde{\Pi}]$. Recall that in the challenge phase, $\mathcal{A}$ picks two classical plaintext-AD pairs (M^0, A^0) and (M^1, A^1) of the same length, after which the challenger picks a random bit b and gives $\mathcal{A}$ (C^b, T^b), the encryption (and tag) of (M^b, A^b). Since the tweaks used for computing this encryption are all different from all the tweaks used during the query phase, and since $\widetilde{\Pi}$ is an ideal tweakable random permutation, the distribution of (C^b, T^b) is independent of the distribution of the responses received by $\mathcal{A}$ during the query phase. Since b is a random bit, if b' is the bit output by $\mathcal{A}$, the probability that $b = b'$ is always $1/2$. Furthermore, this holds irrespective of the choice of $\mathcal{A}$. Thus,

$$\mathbf{Adv}^{\text{Ind-qCPA}}_{\mathsf{QCB}[\widetilde{\Pi}]}(\mathcal{A}) = 0. \tag{4}$$

Our first result follows directly by putting this inequality into Equation 3. In the case of the key-tweak insertion TBC, we consider that the adversary also accesses $E^{\pm}$ and we combine this inequality with Proposition 3, in order to obtain Equation 2. □

6.3 Unforgeability

Now, we prove that QCB is unforgeable under our definition.

Theorem 5. *Let $\mathcal{A}$ be an adversary making q superposition queries to QCB, of maximally ℓ blocks each (message and AD), and q' queries to E. Let $\mathcal{A}$ succeed if it outputs $q+1$ valid quadruples (A, IV, C, T). Then the success probability of $\mathcal{A}$ is upper bounded as:*

$$\Pr\left[\mathcal{A} \text{ succeeds}\right] \leq \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\cdot,\odot)}(\mathcal{B}) + \frac{3+c}{2^n}$$

where c is a the constant from Corollary 2 and $\mathcal{B}$ an adversary querying $\widetilde{E}^{\pm}$ with at most $5q\ell$ pre-declared tweaks, making at most $q\ell$ queries.

In the case of the key-tweak insertion TBC of Section 4, we consider adversaries making also q' queries to $E^{\pm}$ and we have:

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

Proof. Let G_0 be the original BZ game in which $\mathcal{A}$ interacts with QCB, instantiated with the TBC $\widetilde{E}$ and a randomly selected key k. Let G_1 be the game in which $\widetilde{E}$ is replaced by a family of independent random permutations Π_t for all tweaks t.

Lemma 3. $\Pr_{G_0}\left[\mathcal{A} \text{ succeeds}\right] \leq \Pr_{G_1}\left[\mathcal{A} \text{ succeeds}\right] + \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q')$.

Proof. The proof of this lemma comes from the argument used in Theorem 4. In G_0, $\mathcal{A}$ performs q encryption queries of block length at most ℓ. Consider the i^{th} query. From Proposition 4, in this query, the tweakable block cipher $\widetilde{E}$ is queried with tweaks in the set $T(IV^i)$ having a fixed size $|T(IV^i)| = 5\ell$.

We can therefore see $\mathcal{A}$ as an algorithm performing at most $q\ell$ queries to $\widetilde{E}$, with each tweak lying in the fixed set $T = \cup_{i=1}^{q} T(IV^i)$ with $|T| \leq 5q\ell$. If we replace $\widetilde{E}$ with $\widetilde{\Pi}$ for a random $\widetilde{\Pi}$, we go from G_0 to G_1. We therefore, have

$$\Pr_{G_0}\left[\mathcal{A} \text{ succeeds}\right] \leq \Pr_{G_1}\left[\mathcal{A} \text{ succeeds}\right] + \mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}^{\pm}(\$,\odot)}(5q\ell, q') \ . \qquad \blacksquare$$

Our goal is now to bound $\Pr_{G_1}\left[\mathcal{A} \text{ succeeds}\right]$. We run $\mathcal{A}$. Let $\mathcal{I} = \{IV'^i \mid 1 \leq q\}$ be the q declared IVs that $\mathcal{A}$ uses during its encryption queries. Let also $\mathcal{S} = \{(A^i, IV^i, C^i, T^i) \mid 1 \leq i \leq q+1\}$ denote the *forge-set*, *i.e.*, the $q+1$ quadruples in $\mathcal{A}$'s output. Finally, let $[[\cdot]]$ denote block-length. We define the following bad events:

- bad-a: For some i, $IV^i \notin \mathcal{I}$.
- bad-b: For some $i, k \neq i$, $IV^i = IV^k \in \mathcal{I}$, and $[[C^i]] \neq [[C^k]]$
- bad-c: For some $i, k \neq i$, $IV^i = IV^k \in \mathcal{I}$, $[[C^i]] = [[C^k]]$, and $[[A^i]] \neq [[A^k]]$.
- bad-d: For some $i, k \neq i$, $IV^i = IV^k \in \mathcal{I}$, $[[C^i]] = [[C^k]]$, and $[[A^i]] = [[A^k]]$.

$\mathcal{A}$ succeeds in G_1 when the $q+1$ quadruples she outputs are valid. As the $q+1$ outputs shall be distinct and $|\mathcal{I}| = q$, this implies that one of the bad events has occurred. We therefore have

$$\Pr_{G_1}\left[\mathcal{A} \text{ succeeds}\right] \leq \Pr_{G_1}\left[\mathsf{bad\text{-}a}\right] + \Pr_{G_1}\left[\mathsf{bad\text{-}b}\right] + \Pr_{G_1}\left[\mathsf{bad\text{-}c}\right] + \Pr_{G_1}\left[\mathsf{bad\text{-}d}\right]. \qquad (5)$$

We bound separately the probability of each bad event in order to conclude. For a quadruple (A, IV, C, T), with $A = (A_0, \ldots, A_j, pad(A_*))$ and $C = (C_1, \ldots, C_\ell, pad(C_*))$, we define $M_i := \Pi^{-1}_{(0,IV,i)}(C_i)$, $pad(M_*) := \Pi^{-1}_{(1,IV,\ell)}(C_*)$ and $M_{CS} := pad(M_*) \oplus \left(\bigoplus_{i=0}^{\ell} M_i\right)$. If the quadruple (A, IV, C, T) is valid in game G_1, this gives us

$$\Pi_{(4,IV,\ell)}(M_{CS}) \oplus \Pi_{(3,IV,j)}(pad(A_*)) \oplus \left(\bigoplus_{i=0}^{j} \Pi_{(2,IV,i)}(A_i)\right) = T. \tag{6}$$

From there, we have for each $i \in \{0, \ldots, \ell\}$

$$M_i = \Pi^{-1}_{(4,IV,\ell)} \left(T \oplus \Pi_{(3,IV,j)}(pad(A_*)) \oplus \left(\bigoplus_{i=0}^{j} \Pi_{(2,IV,i)}(A_i)\right)\right) \oplus pad(M_*) \oplus \left(\bigoplus_{k \neq i} M_k\right) . \tag{7}$$

This means that from a valid quadruple (A, IV, C, T), we can reconstruct each $M_i = \Pi^{-1}_{(0,IV,i)}(C_i)$ without any query to $\Pi_{0,IV,i}$ or $\Pi^{-1}_{0,IV,i}$ (but with access to other Π_t and Π_t^{-1}, in particular to compute $pad(M_*)$ and the M_k for $k \neq i$).

Similarly, for each $i \in \{0, \ldots, j\}$, we have

$$\Pi_{(2,IV,i)}(A_i) = T \oplus \Pi_{(4,IV,\ell)}(M_{CS}) \oplus \Pi_{(3,IV,j)}(pad(A_*)) \oplus \left(\bigoplus_{k \neq i} \Pi_{(2,IV,k)}(A_k)\right). \tag{8}$$

This means that for a valid quadruple (A, IV, C, T), we can reconstruct each $\Pi_{(2,IV,i)}(A_i)$ without any query to $\Pi_{(2,IV,i)}$ or $\Pi^{-1}_{(2,IV,i)}$ (but with access to other Π_t and Π_t^{-1}).

With these 2 constructions in mind, we can bound the probability of each bad event with the following lemmata.

Lemma 4.

$$\Pr_{G_1}[\mathsf{bad\text{-}a}] \leq \frac{1}{2^n}.$$

Proof. Assume $\mathcal{A}$ outputs a quadruple (A^i, IV^i, C^i, T^i) with $IV^i \notin \mathcal{I}$. Since $IV^i \notin \mathcal{I}$, the permutations $\Pi_{0,IV^i,0}$ and $\Pi^{-1}_{0,IV^i,0}$ have not been queried to compute the quadruple. From the above discussion, if the quadruple is valid, we know how to construct a valid input/output pair $(M_0^i, \Pi_{(0,IV^i,0)}(M_0^i) = C_0^i)$ without any calls to $\Pi_{0,IV^i,0}$ or $\Pi^{-1}_{0,IV^i,0}$. Because $\Pi_{0,IV^i,0}$ is a uniformly random permutation and independent from the others, this happens with probability $\frac{1}{2^n}$. ∎

Lemma 5.

$$\Pr_{G_1}[\mathsf{bad\text{-}b}] \leq \frac{1}{2^n}.$$

Proof. Assume $\mathcal{A}$ outputs two quadruples (A^i, IV^i, C^i, T^i) and (A^k, IV^k, C^k, T^k) such that $IV^i = IV^k \in \mathcal{I}$, and $[[C^i]] \neq [[C^k]]$. Without loss of generality, we assume that there exists u such that $IV^i = IV'^u$, and $\ell^i = [[C^i]]$ is different from the output block length ℓ'^u of query number u (which is a fixed value of the query). This property must be true for i or for k. If the adversary succeeds, the quadruple (A^i, IV^i, C^i, T^i) must be valid even though the function Π_{4,IV^i,ℓ^i} has never been queried. Let $j^i = [[A^i]]$. From (A^i, IV^i, C^i, T^i), we define $M^i_u := \Pi^{-1}_{(0,IV^i,u)}(C^i_u)$, $pad(M^i_*) := \Pi^{-1}_{(1,IV^i,\ell_i)}(C^i_*)$ and $M^i_{CS} := pad(M^i_*) \oplus \left(\bigoplus_{u=0}^{\ell^i} M^i_u\right)$. If the quadruple (A^i, IV^i, C^i, T^i) is valid, we have

$$\Pi_{4,IV^i,\ell^i}(M^i_{CS}) = T^i \oplus \Pi_{(3,IV^i,j^i)}(pad(A^i_*)) \oplus \left(\bigoplus_{u=0}^{j^i} \Pi_{(2,IV^i,u)}(A^i_j)\right).$$

This means we can construct a pair $(M^i_{CS}, \Pi_{4,IV^i,\ell^i}(M^i_{CS}))$ without any calls to Π_{4,IV^i,ℓ^i} or Π^{-1}_{4,IV^i,ℓ^i}. Since Π_{4,IV^i,ℓ^i} is a uniformly random permutation and independent from the others, this happens with probability $\frac{1}{2^n}$. ∎

Lemma 6.

$$\Pr_{G_1}[\mathsf{bad\text{-}c}] \leq \frac{1}{2^n}.$$

Proof. Assume $\mathcal{A}$ outputs two quadruples (A^i, IV^i, C^i, T^i) and (A^k, IV^k, C^k, T^k) such that $IV^i = IV^k \in \mathcal{I}$, $[[C^i]] = [[C^k]]$ and $[[A^i]] \neq [[A^k]]$. Without loss of generality, we assume that there exists u such that $IV^i = IV'^u$, and $j^i = [[A^i]]$ is different from the AD block length j'^u queried in query u. (This happens either for index i or index k). We focus on this quadruple (A^i, IV^i, C^i, T^i) for which Π_{3,IV^i,j^i} has never been queried. We let $\ell^i = [[C^i]]$. we define $M^i_u := \Pi^{-1}_{(0,IV^i,u)}(C^i_u)$, $pad(M^i_*) := \Pi^{-1}_{(1,IV^i,\ell^i)}(C^i_*)$ and $M^i_{CS} := pad(M^i_*) \oplus \left(\bigoplus_{u=0}^{\ell^i} M^i_u\right)$. If the quadruple is valid, we have

$$\Pi_{(3,IV,j^i)}(pad(A^i_*)) = T^i \oplus \Pi_{4,IV^i,\ell^i}(M^i_{CS}) \oplus \left(\bigoplus_{u=0}^{j^i} \Pi_{(2,IV^i,u)}(A^i_u)\right).$$

This means we can construct a pair $(pad(A^i_*), \Pi_{(3,IV^i,j^i)}(pad(A^i_*)))$ without any calls to $\Pi_{(3,IV^i,j^i)}$ or its inverse. Since it is a uniformly random permutation and independent from the others, this happens with probability $\frac{1}{2^n}$. ∎

Lemma 7. *Let c be the constant of Corollary 2, we have*

$$\Pr_{G_1}[\mathsf{bad\text{-}d}] \leq \frac{c}{2^n}.$$

Proof. Assume $\mathcal{A}$ outputs two quadruples (A^i, IV^i, C^i, T^i) and (A^k, IV^k, C^k, T^k) such that $IV^i = IV^k \in \mathcal{I}$, $[[C^i]] = [[C^k]] := \ell$ and $[[A^i]] = [[A^k]] := j$. This means we can write $C^i = (C_0^1, \ldots, C_\ell^i, C_*^i)$, $A^i = (A_0^i, \ldots, A_j^i, pad(A_*^i))$ and similarly for C^k, A^k. Assume the 2 quadruples are valid, we distinguish 2 cases:

- $\exists u, C_u^i \neq C_u^k$. From the construction following Equation 7, we can construct two different input/output pairs $(M_u^i, \Pi_{0,IV^i,u}(M_u^i) = c_u^i)$ and $(M_u^k, \Pi_{0,IV^i,u}(M_u^k) = C_u^k)$ without additional queries to $\Pi^{\pm}_{0,IV^i,u}$. However, there has been only 1 call to $\Pi_{0,IV^i,u}$ during the game (since each IV in the challenge queries is different). Therefore, we have from Corollary 2 that this can happen with probability at most $\frac{c}{2^n}$.
- $\exists u, A_u^i \neq A_u^k$. From the construction following Equation 7, we can construct two different input/output pairs $(A_u^i, \Pi_{2,IV^i,u}(A_u^i))$ and $(A_u^k, \Pi_{2,IV^i,u}(A_u^k))$ without additional queries to $\Pi^{\pm}_{2,IV^i,u}$. We conclude using a similar argument as above.

In order to conclude, notice that we have to be in one of the 2 cases above if the 2 quadruples are valid, otherwise they are equal. ■

The theorem follows from Equation 5 and Lemmas 3–7. □

7 Discussion on Security Notions

In this section, we take a broader viewpoint at suitable notions of quantum security for a combined AEAD mode. In particular, we introduce a new attack that breaks the qIND-qCPA notion given in [13] for all *online* modes (hence all practical AEAD modes). We also discuss the recent definition of *blind unforgeability* which is given in [1].

7.1 The qIND-qCPA Notion and Attacking all Online Modes

It is well-known that for any mode of encryption that XORs a keystream to the message, IND-CPA security implies IND-qCPA. In other words, a quantum adversary does not benefit from having superposition query access. This comes from the malleability of such a mode.

Lemma 8 ([2], informal). *Define an encryption mode as $E_K(M; IV) = M \oplus f(K, IV)$ where IV is a randomly chosen IV and f is any function. If E_K is IND-CPA, then it is also IND-qCPA.*

Informal. Given a quantum adversary $\mathcal{B}$ that attacks the IND-qCPA security notion, we can construct an adversary $\mathcal{A}$ that attacks the IND-CPA security of the mode. $\mathcal{A}$ simulates $\mathcal{B}$. Whenever $\mathcal{B}$ makes a superposition query, $\mathcal{A}$ simulates this query by querying $E_K(0; IV)$ and XORing this value on the input register of $\mathcal{B}$. □

Algorithm 2 Distinguisher on the one-time pad

Input: superposition access to an n-bit function F
Output: either "F is a one-time pad" or "F is a random function"

1: Construct the state: $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |0\rangle$
2: Query F: $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |F(x)\rangle$
3: XOR x in the output: $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |F(x) \oplus x\rangle$
4: Apply Hadamard gates on the first register
5: Measure a value y_0 from the first register.
6: **if** $y_0 = 0$ **then**
7: **return** "F is a one-time pad"
8: **else**
9: **return** "F is a random function"
10: **end if**

However, such a mode also admits a well-known quantum distinguishing attack using a *single* superposition query. This attack applies regardless of the function f chosen, and in particular if f is a random oracle (this is the *one-time pad*).

Lemma 9 (Folklore, [10]). *With a single quantum query to F, Algorithm 2 returns "F is a one-time pad" with probability 1 if F is a one-time pad and "F is a random function" with probability $1 - \frac{1}{2^{n-1}}$ if F is a random function.*

Proof. Note that we can see Algorithm 2 as a call to a generalized version of the Deutsch-Jozsa algorithm [14] for distinguishing whether the function $x \mapsto F(x) \oplus x$ is constant or not using a single query.

If F is a one-time pad, then $F(x) \oplus x = f(K, IV)$, say, for some function f of the IV and key. Then the state before Step 4 is $\frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle |f(K, IV)\rangle$; after Step 4 it becomes $|0\rangle |f(K, IV)\rangle$ and we measure 0 with certainty. If F is a random function, the state before measurement is:

$$\frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} (-1)^{x \cdot y} |y\rangle |F(x) \oplus x\rangle \quad .$$

The amplitude of $|0\rangle$ in the first register depends on the number of preimages of $P(x) \oplus x$. Each value α of $P(x) \oplus x$ contributes independently to the squared amplitude of $y = 0$ by the amount: $\left(\frac{1}{2^n} \sum_{x, P(x) \oplus x = \alpha} (-1)^{x \cdot 0}\right)^2$, *i.e.* $\left(\frac{r}{2^n}\right)^2$ if $P(x) \oplus x$ has r preimages. Since we assume that F is a random function, for each constant r, the average number of images with r preimages is $\frac{1}{r!e}$ [15]. Hence, the expected probability to measure $y = 0$ in the end, over all random functions, is:

$$\sum_{r=0}^{\infty} \frac{2^n}{r!e} \left(\frac{r}{2^n}\right)^2 = \frac{1}{2^n e} \sum_{r=0}^{\infty} \frac{r^2}{r!} = \frac{1}{2^n e} \sum_{r=0}^{\infty} \left(\frac{r(r-1)}{r!} + \frac{r}{r!}\right) = \frac{2}{2^n} \quad . \qquad \square$$

Note that this also works if F is a random permutation instead of a random function (up to PRF-PRP switching). The fact that such an attack exists, although these modes are IND-qCPA secure, demonstrates a strictly stronger power of the adversary when it is only required to distinguish the function instead of breaking a more elaborate security notion. This makes *quantum challenge queries* inherently more powerful. However, they are challenging to define in a non-trivial way, as was observed in [8].

The qIND-qCPA Notion. In [13], Chevalier, Ebrahimi and Vu propose the "qIND-qCPA" security game where an adversary must distinguish between a quantum oracle for $E_K(M;IV) = M \oplus f(K,IV)$ (with IV selected uniformly at random at each new query) and a random oracle. They use Zhandry's recording technique [30] in the latter case. We shall not define the qIND-qCPA security notion in full detail and merely remark that there are no *classical* challenge queries as in IND-qCPA, and that by design, the one-time pad attack is valid.

We are now going to extend the previous distinguisher in order to attack not only keystream-based modes like CTR, but all "online" modes. By "online" mode, we mean a mode of encryption in which the plaintext blocks are read and encrypted in sequence, so that the first ciphertext block C_0 depends *only* on the first plaintext block M_0, the second ciphertext block C_1 depends only on M_0, M_1, *etc.* In fact, we can extend this definition to a much more general setting in which *one bit* of the complete ciphertext, say the last one, is independent from *one bit* of the complete plaintext, say the first one. For the sake of simplicity, we consider messages of a fixed size (since we make a single query anyway).

Lemma 10. *Let $E_K(M;IV)$ be an encryption function of messages of length m, where the first ciphertext bit is independent of the last plaintext bit. Then there exists a quantum adversary $\mathcal{A}^O$ making a single query to its oracle O and distinguishing $E_K(M;IV)$ ("real world") from a random family of permutations $\Pi_{K,IV}(M)$ ("random world") with probability of success $\frac{3}{4} \geq \frac{1}{2}$.*

Proof. Our distinguisher is based on Deutsch-Jozsa's algorithm and on the post-processing of quantum oracles of Lemma 2. The adversary fixes all the bits of M except the last one to an arbitrary value, say 0, and puts $|0\rangle + |1\rangle$ in the last bit. She queries the oracle and truncates the output to its first bit. Her state becomes:

$$|0\rangle\,|f(0)\rangle + |1\rangle\,|f(1)\rangle$$

where f is the first ciphertext bit as a function of the last plaintext bit (after the other bits have been fixed). She then uses Deutsch-Josza's algorithm to determine whether f is constant or non-constant. If f is constant, she decides that this is the real world and otherwise, the random world.

- In the random world ($O = \Pi_{K,IV}(M)$), this f should remain a random function. Thus the outputs are equal only with probability $\frac{1}{2}$: the guess is correct with probability $\frac{1}{2}$.
- In the real world, f is always constant. The guess is always correct.

Overall, the adversary is correct with probability $\frac{1}{2}\left(1+\frac{1}{2}\right)=\frac{3}{4}$. Using a full block instead of a mere bit makes the success probability exponentially close to 1 with a single query, as in the one-time pad attack. □

A consequence of this attack is that, while the qIND-qCPA definition seems nontrivial, it cannot be achieved by an online mode, including *e.g.* CBC or our proposal QCB.

Corollary 3. *No online mode of encryption is qIND-qCPA secure.*

The issue with the definition lies in the adversary's power in distinguishing random from constant functions *within a single query*. If we require the adversary to distinguish the mode from an *ideal online mode*, instead of a random permutation, our attack should not be applicable anymore. However, the definition and proofs of security may be far more involved, and we leave further exploration of this topic as an open problem.

7.2 Unforgeability for a Combined AEAD Mode

The *Blind Unforgeability* notion was introduced in [1] as a replacement for BZ-unforgeability for MACs. In [1], the authors prove that it is possible to create a MAC scheme (given by a pair Mac_K, Ver_K) such that, after having made q superposition queries to some subset of the message space, one can forge the MAC of another message outside this space. Despite that, the MAC that they give is BZ-secure.

Note that the example given in [1] is very technical, and relies heavily on the fact that the MAC treats differently different subsets of its input. This is usually not the case for practical constructions (including QCB).

Blind-unforgeability (BU) is a stronger security notion defined with the following game: the adversary is given access to a *blinded* version of Mac_K, that returns $\perp$ on some fraction ϵ of the message space. To win, the adversary has to output a valid forgery in this space. In the game, the uniform random blinding B_ϵ is created by putting every message of the message space with probability ϵ. Alternatively, the adversary could choose her own blinding, but this is equivalent for inverse-polynomial values of ϵ: in [1] (Theorem 2) the authors prove that an adversary capable of outputting a "good" forgery will still do so even if the MAC has been blinded.

BU game

Setup: the adversary selects a parameter $\epsilon < 1$. The challenger picks a random key K, a random bit b, a random blinding B_ϵ which is a fraction of the message space $\mathcal{M}$ of size ϵ.

Forgery: the adversary produces a pair (M, T) and wins if $M \in B_\epsilon$ and $\mathsf{Ver}_K(M, T) = \top$.

Encryption queries: the adversary queries the "blinded" MAC:

$$M \mapsto \begin{cases} \perp & \text{if } M \in B_\epsilon, \\ \mathsf{Mac}_K(M) & \text{otherwise .} \end{cases} \tag{9}$$

The following result, together with the example given in [1], shows that BU-unforgeability is a strictly stronger notion than BZ-unforgeability for a MAC.

Theorem 6 ([1], Theorem 1). *Any BU-unforgeable MAC is BZ-unforgeable.*

This notion is adapted for a standalone MAC. In our case, we consider a combined AEAD mode, and we would need to adapt the definition. We can propose, for example, to blind the message space. We select a subset B_ϵ of message, AD and IVs (possibly the same pairs of AD and message for all IVs, or selected differently for each one). We give the adversary access to an oracle that encrypts (IV, A, M) if it does not belong to B_ϵ and otherwise, returns $\perp$. The adversary then succeeds if she outputs a valid quadruple (A, IV, C, T) whose corresponding message M is such that $(IV, A, M) \in B_\epsilon$.

The main difference with the original BU definition is that the condition of success relies on the message M, which is not necessarily an output of the forgery (the adversary can forge on an unknown message M). Despite that, we conjecture that this definition is non-trivial and that it might be proven for QCB. This proof would likely be more technical than our original one, and we leave it as an open problem.

8 Conclusion

In this paper, we designed the first AEAD of rate one with quantum security guarantees. With a definition similar to TAE and OCB, our proposal, QCB, retains high security guarantees as soon as it is used with a quantum-secure tweakable block cipher. We explicited this security requirement and proposed a construction based on a block cipher, in the ideal cipher model: the key-tweak insertion of Section 4.

In the classical setting, the LRW construction provides a TBC of rate one (one block cipher call per TBC call) from a PRP assumption. Ours requires related-key security for the underlying block cipher. Although we do not rule out the possibility of a rate-one TBC without related-key security, the LRW approach does not seem applicable.

Thus, an interesting open question is whether it is possible to build a post-quantum AEAD of rate one from a block cipher, *with a qPRP assumption only*. It may be possible to obtain directly the security without relying explicitly on a secure TBC, though this was the subject of our first attempt, which failed due to a new attack on OCB with a *single* query.

In our security proofs, we used the IND-qCPA and BZ security notions for indistinguishability and unforgeability. Other security definitions have been proposed in the more recent literature and seem worth investigating. In this paper, we showed that the qIND-qCPA notion of [13] rules out all online encryption modes (in which some part of the output is independent on some part of the input). Nevertheless, it might be possibly to re-adapt it for the usual AEAD setting.

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, Heidelberg (May 2020)

[2] 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.) Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016. pp. 44–63. Springer, Heidelberg (2016)

[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 - (feat. CRAX and TRAX). In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 419–448. Springer, Heidelberg (Aug 2020)

[4] Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.V.: Strengths and weaknesses of quantum computing. SIAM J. Comput. 26(5), 1510–1523 (1997)

[5] Bernstein, E., Vazirani, U.V.: Quantum complexity theory. In: Kosaraju, S.R., Johnson, D.S., Aggarwal, A. (eds.) Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA. pp. 11–20. ACM (1993)

[6] Bhaumik, R., Nandi, M.: Improved security for OCB3. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 638–666. Springer, Heidelberg (Dec 2017)

[7] 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 (May 2013)

[8] 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 (Aug 2013)

[9] Bonnetain, X.: Quantum key-recovery on full AEZ. In: Adams, C., Camenisch, J. (eds.) SAC 2017. LNCS, vol. 10719, pp. 394–406. Springer, Heidelberg (Aug 2017)

[10] Bonnetain, X.: Hidden Structures and Quantum Cryptanalysis. (Structures cachées et cryptanalyse quantique). Ph.D. thesis, Sorbonne University, France (2019), https://tel.archives-ouvertes.fr/tel-02400328

[11] Bonnetain, X., Hosoyamada, A., Naya-Plasencia, M., Sasaki, Y., Schrottenloher, A.: Quantum attacks without superposition queries: The offline Simon's algorithm. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part I. LNCS, vol. 11921, pp. 552–583. Springer, Heidelberg (Dec 2019)

[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 Trans. Symm. Cryptol. 2020(S1), 160–207 (2020)

[13] Chevalier, C., Ebrahimi, E., Vu, Q.H.: On the security notions for encryption in a quantum world. QCrypt 2020 (2020), https://eprint.iacr.org/2020/237
[14] Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439(1907), 553–558 (1992)
[15] Flajolet, P., Odlyzko, A.M.: Random mapping statistics. In: Quisquater, J.J., Vandewalle, J. (eds.) EUROCRYPT'89. LNCS, vol. 434, pp. 329–354. Springer, Heidelberg (Apr 1990)
[16] Hosoyamada, A., Sasaki, Y.: Quantum Demiric-Selçuk meet-in-the-middle attacks: Applications to 6-round generic Feistel constructions. In: Catalano, D., De Prisco, R. (eds.) SCN 18. LNCS, vol. 11035, pp. 386–403. Springer, Heidelberg (Sep 2018)
[17] Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part I. LNCS, vol. 11272, pp. 275–304. Springer, Heidelberg (Dec 2018)
[18] 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 (Aug 2016)
[19] Krovetz, T., Rogaway, P.: The software performance of authenticated-encryption modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer, Heidelberg (Feb 2011)
[20] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 31–46. Springer, Heidelberg (Aug 2002)
[21] Liskov, M., Rivest, R.L., Wagner, D.: Tweakable block ciphers. Journal of Cryptology 24(3), 588–613 (Jul 2011)
[22] National Institute of Standards and Technology (NIST): Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (Dec 2016)
[23] National Institute of Standards and Technology (NIST): Submission requirements and evaluation criteria for the lightweight cryptography standardization process (Aug 2018)
[24] Nielsen, M.A., Chuang, I.L.: Quantum information and quantum computation. Cambridge: Cambridge University Press 2(8), 23 (2000)
[25] Rogaway, P.: Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer, Heidelberg (Dec 2004)
[26] Rötteler, M., Steinwandt, R.: A note on quantum related-key attacks. Inf. Process. Lett. 115(1), 40–44 (2015)
[27] Santoli, T., Schaffner, C.: Using simon's algorithm to attack symmetric-key cryptographic primitives. Quantum Inf. Comput. 17(1&2), 65–78 (2017)
[28] Simon, D.R.: On the power of quantum computation. In: 35th FOCS. pp. 116–123. IEEE Computer Society Press (Nov 1994)
[29] Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Inf. Comput. 15(7&8), 557–567 (2015), http://www.rintonpress.com/xxqic15/qic-15-78/0557-0567.pdf
[30] 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, Heidelberg (Aug 2019)

Appendix

We give below the delayed proofs of security for the CMT and key-tweak insertion TBCs.

A Proof of Security of the CMT Mode

In this section, we let $\widetilde{E}_K$ denote $E_K(T \oplus E_K(x))$, the CMT mode. We recall Proposition 2:

Proposition 2. *Let $\mathcal{A}^{\widetilde{E}(\cdot,\odot)}$ be an adversary making q queries to $\widetilde{E}$ (or a random permutation family), with a set of tweaks of size m, and distinguishing between the actual $\widetilde{E}_K$ for a random K and a family of random permutations Π_T. Then:*

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}(\cdot,\odot)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{qPRP}}_{E}(2q) + \mathcal{O}\left(\left(\frac{q^3 m}{2^{n/2}}\right)^{2/3}\right) .$$

We start the proof with a technical lemma.

Lemma 11. *Fix m distinct tweaks $t_1, \ldots, t_m \in \{0,1\}^n$. Then, choose S random distinct values $x_1, \ldots, x_S \in \{0,1\}^n$. We have*

$$P_{q,S} := \Pr[\exists i, j \in [m], \exists k, l \in [S] : t_i \oplus x_k = t_j \oplus x_l] \leq \frac{S(S+1)m^2}{2(2^n - S)}.$$

Proof. Let $T = \{t_i \oplus t_j\}_{i,j \in [m]}$. We have $|T| \leq m^2$. First notice that because the t_i are distinct, we have $S_{m,1} = 0$. We now write

$$\begin{aligned}
P_{m,S+1} &= \Pr[\exists i, j \in [m], \exists k, l \in [S+1] : t_i \oplus x_k = t_j \oplus x_l] \\
&= \Pr[\exists i, j \in [m], \exists k, l \in [S] : t_i \oplus x_k = t_j \oplus x_l] + \Pr[\exists i, j \in [m], \exists k \in [S] : t_i \oplus x_k = t_j \oplus x_{S+1}] \\
&= P_{m,S} + \Pr[\exists k \in [S] : x_{S+1} \in \{t \oplus x_k\}_{t \in T}] \\
&\leq P_{m,S} + \frac{Sm^2}{2^n - S} \qquad \text{since } x_{S+1} \text{ is chosen uniformly from } \{0,1\}^n \backslash \{x_1, \ldots, x_S\}.
\end{aligned}$$

This gives

$$P_{m,S+1} = \sum_{k=1}^{S} \frac{km^2}{2^n - k} \leq \sum_{k=1}^{S} \frac{km^2}{2^n - S} = \frac{S(S+1)m^2}{2(2^n - S)}. \qquad \square$$

Notice that if we take $S = \sqrt{\frac{\varepsilon 2^n}{m^2}} \ll 2^n$, we have $P_{m,S+1} = O(\varepsilon)$.

Proof of Proposition 2. We consider a sequence of hybrid games, starting from the real setting and going to Π_T. Let G_1 be a game in which $\mathcal{A}$ interacts with $\widetilde{E}^{\pm}(\cdot, \odot)$, with a key K chosen at random.

Game G_1	Game G_2	Game G_3
$K \xleftarrow{\$} \mathcal{K}$.	$\Pi \xleftarrow{\$} \mathcal{P}_n$.	$F \xleftarrow{\$} \mathcal{F}_n$.
$\widetilde{E}_K(t,x) := E_K(t \oplus E_K(x))$.	$\widetilde{\Pi} := \Pi(t \oplus \Pi(x))$.	$\widetilde{\Pi} := F(t \oplus F(x))$.
Run $\mathcal{A}^{\widetilde{E}_K(\cdot,\odot)}$.	Run $\mathcal{A}^{\widetilde{\Pi}(\cdot,\odot)}$.	Run $\mathcal{A}^{\widetilde{\Pi}(\cdot,\odot)}$.

Let G_2 be a game in which $\mathcal{A}$ interacts with $\widetilde{\Pi} = \Pi(t \oplus \Pi(x))$, where Π is a permutation selected at random. Then by definition of qPRP security:

$$\left| \Pr_{G_1}[\mathcal{A} \Rightarrow 1] - \Pr_{G_2}[\mathcal{A} \Rightarrow 1] \right| \leq \mathbf{Adv}_E^{\text{q-PRP}}(2q)$$

since we have to replace E by Π a total of $2q$ times. Next, we introduce another game G_3 in which Π is replaced by a random oracle F. By the PRP-PRF switching lemma (Theorem 2):

$$\left| \Pr_{G_2}[\mathcal{A} \Rightarrow 1] - \Pr_{G_3}[\mathcal{A} \Rightarrow 1] \right| \leq \mathcal{O}\left(\frac{q^3}{2^n}\right)$$

where n is the block and tweak size; because the queries of the adversary contain $2q$ queries to Π in total.

Let $t_1, \ldots, t_m$ be the tweaks declared by the adversary. Note that we could have $m \geq q$. The adversary is only allowed to choose his tweaks from this predefinite set. We can consider this set to be constant and reason on average on it; note that the tweaks could be random, but their randomness is independent from the choice of the permutations. We introduce a game G_4 in which F is replaced by a random oracle G with a codomain $X = G(\{0,1\}^n)$ reduced to a size S, being such that no collision of the form:

$$G(x) \oplus t_i = G(x') \oplus t_j \implies G(x) \oplus G(x') = t_i \oplus t_j$$

occurs for any pair x, x' of inputs and t_i, t_j of tweaks. The set X is chosen at random, depending on the tweaks, and its definition implies that the sets $t_i \oplus X$ are all pairwise disjoint. This sets a rather restrictive upper bound on S. More precisely, we use Lemma 11: by picking elements for X uniformly at random, the bound $S = \sqrt{\frac{\epsilon 2^n}{m^2}}$ ensures that the probability that one of the pairwise sums collides with one of the $t_i \oplus t_j$ is $\mathcal{O}(\epsilon)$. We shall then take ϵ small enough to ensure that this case does not occur.

Game G_4	Game G_5
Choose X at random of size S	Choose X at random of size S
$G \xleftarrow{\$} \{\{0,1\}^n \to X\}$	$G \xleftarrow{\$} \{\{0,1\}^n \to X\}$
$G_t(t,x) := G(t \oplus G(x))$.	$H(t_i,x) := \begin{cases} 0 \text{ if } x \in \bigcup_i (t_i \oplus X) \\ G(t_i \oplus G(x)) \text{ otherwise} \end{cases}$
Run $\mathcal{A}^{G_t(\cdot,\odot)}$.	Run $\mathcal{A}^{H(\cdot,\odot)}$.

The adversary cannot distinguish between G_3 and G_4 unless she finds a collision of G within $2q$ queries, or if the choice of X was a bad one. By the

collision bound with small range:

$$\left|\Pr_{G_3}[\mathcal{A} \Rightarrow 1] - \Pr_{G_4}[\mathcal{A} \Rightarrow 1]\right| \leq \mathcal{O}\left(\frac{q^3}{S}\right) + \mathcal{O}(\epsilon) \quad .$$

Our goal is to use this oracle G with a reduced image to show independence between the queries $G(t_i \oplus G(x))$ in G_4.

In G_4, the results of queries to $G(t_i \oplus G(x))$ for different tweaks are not necessarily independent, although the inputs to the outer G are disjoints subsets of its domain. We consider the set $Y = \bigcup_i (t_i \oplus X)$. This is a set of size $\mathcal{O}(mS)$ which, depending on the values of the t_i and our choice of X (both of which are fixed within the game and uncontrolled by the adversary), can be any subset of $\{0,1\}^n$ of size $\mathcal{O}(mS)$. We introduce a new game G_5 (still depending on X) in which we replace each query to $G(t_i \oplus G(x))$ by a query to an oracle $H(t_i, \odot)$ that: on input $x \in Y$, replies 0, and on input $x \notin Y$, replies by $G(t_i \oplus G(x))$.

As we perform the same projector on all oracles (the inputs that are modified are the same), we can replace all of them at once. Consider the state $\left|\phi_j^Y\right\rangle$ of $\mathcal{A}$ during G_5 before the j-th query to any of the oracles H_i and $|\psi_j\rangle$ the state during G_4 at the same moment (note that only G_5 depends on Y). We show that both games will look equivalent to the adversary, because she cannot know Y. At the end of the games:

$$\| \left|\phi_{q+1}^Y\right\rangle - |\psi_{q+1}\rangle \| \leq 2 \sum_{1 \leq j \leq q} |P_Y(|\psi_j\rangle)|,$$

where P_Y is the projector on the part of the state which corresponds to inputs $x \in Y$. We take the average over all choices of Y.

$$\mathop{\mathbb{E}}_Y \left(\| \left|\phi_{q+1}^Y\right\rangle - |\psi_{q+1}\rangle \| \right) \leq 2 \mathop{\mathbb{E}}_Y \left(\sum_{1 \leq j \leq q} |P_Y(|\psi_j\rangle)| \right) \quad .$$

On the left, we swap the sums over j and Y. A given x is in a given Y with probability $\frac{mS}{2^n}$, and our choices are symmetric. Let N be the total number of choices for Y that we make. For a given j we compute:

$$\sum_Y |P_Y |\psi_j\rangle|^2 = \sum_Y \sum_{x \in Y} |P_x |\psi_j\rangle|^2 = \frac{NmS}{2^n} \sum_{x \in \{0,1\}^n} |P_x |\psi_j\rangle|^2 = \frac{NmS}{2^n}$$

and we use Jenssen's inequality to show that:

$$\left(\sum_Y |P_Y(|\psi_j\rangle)| \right)^2 \leq N \sum_Y |P_Y |\psi_j\rangle|^2 \leq \frac{N^2 mS}{2^n} \implies \mathop{\mathbb{E}}_Y \left(\sum_Y |P_Y(|\psi_j\rangle)| \right) \leq \sqrt{\frac{mS}{2^n}} \quad .$$

Thus we can bound:

$$\mathop{\mathbb{E}}_Y \left(\| \left|\phi_{q+1}^Y\right\rangle - |\psi_{q+1}\rangle \| \right) \leq 2q \sqrt{\frac{mS}{2^n}} = 2\sqrt{\frac{q^2 mS}{2^n}}$$

and by Corollary 1:

$$\left|\Pr_{G_4}[\mathcal{A} \Rightarrow 1] - \Pr_{G_5}[\mathcal{A} \Rightarrow 1]\right| \leq 8\sqrt{\frac{q^2 m S}{2^n}} \ .$$

Having done that, we remark that the queries in G_5 are now (finally) totally independent: they are of the form $F_i(t_i \oplus G(x))$ where G is a random function with codomain X and F_i is an independent random function with domain and codomain X. We can replace them by $F_i(x)$ directly. The F_i are (at most) q different small-range random functions, to which the adversary makes a total of q queries. We can replace them by full-range random functions and the difference in distinguishing advantage will be $\mathcal{O}\left(\frac{q^3}{S}\right)$.

Finally, we replace them by independent permutations of $\{0,1\}^n$ and the difference in distinguishing advantage is: $\mathcal{O}\left(\frac{q^3}{2^n}\right)$.

Thus the total bound that we get is:

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}(\cdot,\odot)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{qPRP}}_{E}(2q) + \mathcal{O}\left(\frac{q^3}{2^n}\right) + \mathcal{O}\left(\frac{q^3}{S}\right) + \mathcal{O}\left(\epsilon\right) + 8\sqrt{\frac{q^2 m S}{2^n}} \ .$$

Meanwhile, we know that $S = \sqrt{\frac{\epsilon 2^n}{m^2}}$, which gives a bound:

$$\mathbf{Adv}^{\mathrm{TPRP}}_{\widetilde{E}(\cdot,\odot)}(\mathcal{A}) \leq \mathbf{Adv}^{\mathrm{qPRP}}_{E}(2q) + \mathcal{O}\left(\frac{q^3 m}{\sqrt{\epsilon} 2^{n/2}} + \epsilon + \frac{\epsilon^{1/4}}{2^{n/4}}\right) \ .$$

which makes the first two terms dominant, with a bound $\mathcal{O}\left(\left(\frac{q^3 m}{2^{n/2}}\right)^{2/3}\right)$. □

Remark 3. Proposition 2 shows that qPRP security is enough to obtain a weak quantum-secure TBC in our sense, albeit with two block cipher calls. However, the proof is definitely not tight. This tightness is mainly lost when we modify the codomain of the random function G in order to make the q queries independent.

The best attack that we can give is the following.

Lemma 12. *There exists a quantum algorithm that for any E, makes q queries to a TBC $\widetilde{E}(\cdot,\odot)$ and outputs "CMT" with probability $\mathcal{O}\left(q^3/2^n\right)$ if $\widetilde{E}$ is a CMT mode, and probability $\mathcal{O}\left(q^3/2^{2n}\right)$ if $\widetilde{E}$ is a family of random permutations.*

Proof. The difference between a CMT construction and a family of random permutations can be detected when an internal collision $t_1 \oplus E_K(x_1) = t_2 \oplus E_K(x_2)$ occurs. Since the adversary can only make queries to the full construction, she observes that $E_K(t_1 \oplus E_K(x_1)) = E_K(t_2 \oplus E_K(x_2))$. Collisions between $\Pi_{t_1}(x_1)$ and $\Pi_{t_2}(x_2)$ for a permutation family Π can occur at random. But in the CMT case, we also have $E_K(t_1 \oplus 1 \oplus E_K(x_1)) = E_K(t_2 \oplus 1 \oplus E_K(x_2))$ (and the same for any constant instead of 1).

Thus, we define a function: $f(t,x) = (\widetilde{E}(t,x), \widetilde{E}(t \oplus 1, x))$ and we look for a collision of f. In the CMT case, such a collision indicates, with almost certainty, that we have found an internal collision $t_1 \oplus E_K(x_1) = t_2 \oplus E_K(x_2)$. In the random case, it will happen only with smaller probability. The quantum attack does:

1. Query $f(t_i, 0)$ for $q/2$ tweaks t_i (this means $q/2$ internal values $t_i \oplus E_k(0)$)
2. With Grover's algorithm, search a $(0,x)$ such that $\exists i, f(0,x) = f(t_i,0), (0,x) \neq (t_i,0)$
3. Measure after $q/2$ iterations and output "CMT" if a collision is found. In the CMT case, this happens with probability $\mathcal{O}\left(q^3/2^n\right)$ and $\mathcal{O}\left(q^3/2^{2n}\right)$ in the random case (we can decrease it arbitrarily by increasing the output size of f).

This concludes the proof. □

Remark 4. The attack of Lemma 12 does not work if the tweaks are used only once, or if they are random, as a standard Grover search in Step 2 is inapplicable. We can resort to the classical collision attack, which succeeds with probability $\mathcal{O}\left(q^2/2^n\right)$.

B Proof of Security of the Key-tweak Insertion TBC

In this section, we let $\widetilde{E}_K(T,x)$ denote $E_{K \oplus T}(x)$, the *key-tweak insertion TBC*. We need here the ideal cipher model: E is selected at random from all ciphers. We recall Proposition 3:

Proposition 3. *Let $\mathcal{A}$ be an adversary making q queries to $\widetilde{E}^{\pm}$ and q' queries to $E^{\pm}$, with a set of tweaks of size m. Then:*

$$\left| \Pr_{K \xleftarrow{\$} \mathcal{K}} [\mathcal{A}^{\widetilde{E}_K^{\pm}(\cdot,\odot), E^{\pm}_{\odot}(\odot)} \Rightarrow 1] - \Pr_{\{\Pi_T\} \xleftarrow{\$} \mathcal{P}_n} [\mathcal{A}^{\Pi^{\pm}(\cdot,\odot), E^{\pm}_{\odot}(\odot)} \Rightarrow 1] \right| \leq 8\sqrt{\frac{mq'^2}{2^n}} \ .$$

Proof of Proposition 3. We will consider hybrid games, where we change the oracles that $\mathcal{A}$ accesses and bound the difference between her probabilities of success.

Let $t_1, \ldots, t_m$ be the tweaks of the declared set. This list is not deterministic, but it is given by the game, and does not depend on the adversary's state (in particular, it is non-adaptive). Thus, it suffices to reason with an arbitrary list and to take the average over all possibilities (the bound obtained will be the same in all cases). Note that the definition of our hybrid games will be dependent on this list.

Let G_0 be the "real world" in wich $\mathcal{A}$ interacts with $\widetilde{E}^{\pm}$ and $E^{\pm}$, for $k \xleftarrow{\$} \mathcal{K}$. We also define the game $G_0[K]$ where a key K is fixed.

Game G_0
$\forall k \in \mathcal{K}, E_k \xleftarrow{\$} \mathcal{P}_n.$
$K \xleftarrow{\$} \mathcal{K}.$
$\widetilde{E}_K(t,x) := E_{t\oplus K}(x).$
Run $\mathcal{A}^{\widetilde{E}^{\pm}_K(\cdot,\odot),E^{\pm}_{\odot}(\odot)}$.

Game $G_0[K]$
$\forall k \in \mathcal{K}, E_k \xleftarrow{\$} \mathcal{P}_n.$
$\widetilde{E}_K(t,x) := E_{t\oplus K}(x).$
Run $\mathcal{A}^{\widetilde{E}^{\pm}_K(\cdot,\odot),E^{\pm}_{\odot}(\odot)}$.

We have by definition

$$\Pr\Big[\forall k\in\mathcal{K}, E_k \xleftarrow{\$} \mathcal{P}_n : \mathcal{A}^{\widetilde{E}^{\pm}_K(\cdot,\odot),E^{\pm}_{\odot}(\odot)} \Rightarrow 1\Big] = \Pr[G_0 \Rightarrow 1]$$
$$= \mathop{\mathbb{E}}_{K\xleftarrow{\$}\mathcal{K}} \left(\Pr[G_0[K] \Rightarrow 1\right).$$

Let G_1 be a hybrid game in which $\widetilde{E}^{\pm}$ is replaced by a family of permutations $\Pi_{t_1},\ldots,\Pi_{t_m}$, and $E^{\pm}$ is replaced by $E'^{\pm}$, which is equal to $E^{\pm}$ for all keys, except $K\oplus t_1,\ldots,K\oplus t_m$, where we constrain: $E'^{\pm}_{K\oplus t_i} = \Pi_{t_i}$.

Game $G_1[K]$
$\forall i \in [m], \Pi_{t_i} \xleftarrow{\$} \mathcal{P}_n\,.$
$\forall k \in \mathcal{K}, E'_k \xleftarrow{\$} \mathcal{P}_n.$
$\forall i \in [m], E_{K\oplus t_i} := \Pi_{t_i},\ \ \forall k \notin \{K\oplus t_i\}_{i\in[m]}, E_k := E'_k\,.$
Run $\mathcal{A}^{\Pi^{\pm}(\cdot,\odot),E^{\pm}_{\odot}(\odot)}$.

Notice that if we define $\widetilde{E}_K(t,x) := E_{t\oplus K}(x)$, we have in this game that $\forall i \in [m], \forall x, \widetilde{E}_K(t_i,x) = \Pi_{t_i}(x)$, which implies that $\mathcal{A}^{\widetilde{E}^{\pm}_K(\cdot,\odot),E^{\pm}_{\odot}(\odot)} = \mathcal{A}^{\Pi^{\pm}(\cdot,\odot),E^{\pm}_{\odot}(\odot)}$ when we only query $\Pi_t(x)$ for tweaks $t = t_1,\ldots,t_m$.

Lemma 13. *For any key $K \in \mathcal{K}$,*

$$\Pr[G_0[K] \Rightarrow 1] = \Pr[G_1[K] \Rightarrow 1]. \tag{10}$$

Proof. The two games are syntactically equivalent. The only change is in the order in which we select the new permutations at random. In G_0, we first pick E_k for each $k \in \mathcal{K}$ and we define $\widetilde{E}$ accordingly. In $G_1[K]$, we select first randomly the permutations for $\widetilde{E}$ and then the other permutations E_k for $k \notin \{K\oplus t_i\}_{i\in[m]}$. ■

Next, we create another hybrid G_2 in which $\mathcal{A}$ interacts with the family Π, and the unmodified $E^{\pm}$, which is then independent of Π.

Game G_2
$\forall i \in [m], \Pi_{t_i} \xleftarrow{\$} \mathcal{P}_n.$
$\forall k \in \mathcal{K}, E'_k \xleftarrow{\$} \mathcal{P}_n.$
$\forall k \in \mathcal{K}, E_k := E'_k\,.$
Run $\mathcal{A}^{\Pi^{\pm}(\cdot,\odot),E^{\pm}_{\odot}(\odot)}$.

Notice that it is equivalent to write directly $E_k \xleftarrow{\$} \mathcal{P}_n$ in this game but writing it the way we did will make notations easier in the proof.

We will show that the difference between these two games is small, on average on K. To show this, notice that going from $G_1[K]$ to G_2, we only change E_k for $k \in \{K \oplus t_i\}_{i\in[m]}$ and using query magnitude arguments, we show that this leads to a small change in the game value, on average on K.

Lemma 14.

$$\mathop{\mathbb{E}}_{K\xleftarrow{\$}\mathcal{K}} \left(\Pr[G_1[K] \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]\right) \leq 8\sqrt{\frac{mq'^2}{2^n}} \ . \tag{11}$$

Proof. Let $G_1[K, E', \Pi]$ and $G_2[E', \Pi]$ be the games $G_1[K]$ and G_2 where we additionally fix all the choices of E'_k and Π_{t_i}. Let us also fix such a choice E' and Π. Let $|\psi_i\rangle$ the state of $\mathcal{A}$ in $G_2[E', \Pi]$ before the i^{th} query to $E^{\pm}_{\odot}(\odot)$ and $\left|\phi_i^K\right\rangle$ the state of $\mathcal{A}$ in $G_1[K, E', \Pi]$ at the same point.

Between the two games $G_1[K, E', \Pi]$ and $G_2[E', \Pi]$, we change the choice of $E^{\pm}_k$ only for $k \in \{K \oplus t_i\}_{i\in[m]}$. After q' queries, we therefore have by Theorem 1:

$$\| \left|\phi^K_{q'+1}\right\rangle - |\psi_{q'+1}\rangle \| \leq 2 \sum_{1\leq i\leq q'} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,|$$

where $P_{K,t_1,\ldots,t_m}$ is the projector on the part of the input that corresponds to a key $k \in \{K \oplus t_i\}_{i\in[m]}$. When K cycles over all possible keys, $\mathcal{K} = \{0,1\}^n$, the set $\{K \oplus t_i\}_{i\in[m]}$ describes $\{0,1\}^n$ exactly m times. Thus, we have:

$$\sum_{K\in\mathcal{K}} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,|^2 = m \sum_{x\in\{0,1\}^n} |P_x |\phi_i\rangle \,|^2 = m$$

by normalization, and by Jensen's inequality:

$$\left(\sum_{K\in\mathcal{K}} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,|\right)^2 \leq |\mathcal{K}| \sum_{K\in\mathcal{K}} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,|^2 = 2^n m \ .$$

Afterwards, we use Corollary 1:

$$\begin{aligned}\left|\Pr[G_1[K, E', \Pi] \Rightarrow 1] - \Pr[G_2[E', \Pi] \Rightarrow 1]\right| &\leq 4\| \left|\phi^K_{q'+1}\right\rangle - |\psi_{q'+1}\rangle \| \\ &\leq 8 \sum_{1\leq i\leq q'} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,|\end{aligned}$$

and we take the average over K:

$$\begin{aligned}\mathop{\mathbb{E}}_K \left(\left|\Pr[G_1[K, E', \Pi] \Rightarrow 1] - \Pr[G_2[E', \Pi] \Rightarrow 1]\right|\right) &\leq \frac{8}{|\mathcal{K}|} \sum_{1\leq i\leq q'} \sum_{K\in\mathcal{K}} |P_{K,t_1,\ldots,t_m} |\phi_i\rangle \,| \\ &\leq 8\sqrt{\frac{q'^2 m}{2^n}}.\end{aligned}$$

This holds for all E', Π hence by taking the average over these, we have

$$\mathop{\mathbb{E}}_{K}\left(\left|\Pr[G_1[K] \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]\right|\right) \leq 8\sqrt{\frac{q'^2 m}{2^n}}.$$

which concludes the proof of the lemma. ■

We can now finish the proof of our theorem. Game G_2 is the ideal world. Combining our two lemmata, we can conclude:

$$\left|\Pr[G_0 \Rightarrow 1] - \Pr[G_2 \Rightarrow 1]\right| \leq 8\sqrt{\frac{mq'^2}{2^n}} \ . \qquad \square$$

Remark 5. Making the proof work for general *adaptative* tweaks, which are chosen by the adversary depending on her current state, turned out to be much more difficult than we initially anticipated. In particular, our query magnitude argument cannot be used as is, since we do not know in advance the positions at which we would like to change the outputs of $E^{\pm}$. Despite that, we conjecture that the same bound can be achieved for adaptive tweaks, as there does not seem to be any better attack. We leave this as an open question.

The bound given by Proposition 3 is not tight because of the $\sqrt{\cdot}$. However, for a constant success probability, the bound is matched by the following attack.

Lemma 15. *There exists a quantum algorithm that for any E, makes q queries to the TBC, q' queries to E and succeeds in recovering the key of a key-tweak insertion TBC with probability $\mathcal{O}\left(\frac{qq'^2}{2^n}\right)$ (thus distinguishing the instance from a random family of permutations).*

Proof. The attack runs in three phases:

1. The adversary makes q queries of the form $E_{K\oplus i}(0)$ for $i = 0, \ldots, q-1$ and stores the couples $E_{K\oplus i}(0), E_{K\oplus i\oplus 1}(0)$ in a database $\mathcal{D}$.
2. Using Grover's algorithm, the adversary searches for an element z such that $(E_z(0), E_{z\oplus 1}(0)) \in \mathcal{D}$. As $\mathcal{D}$ is of size q, Grover search would require $\mathcal{O}\left(\sqrt{\frac{2^n}{q}}\right)$ queries to succeed with constant probability. After q' queries, the probability of success is $\mathcal{O}\left(\frac{qq'^2}{2^n}\right)$.
3. Let t, z be the obtained pair such that $E_z(0) = E_{K\oplus t}(0)$ and $E_{z\oplus 1}(0) = E_{K\oplus t\oplus 1}(0)$. The use of two elements makes the probability of a false positive (a random collision) exponentially low. The adversary then concludes that $z = K \oplus t$ *i.e.* $K = z \oplus t$ and checks that the key was correctly guessed.

If the TBC queried is a random family of permutations, then no solution exists at Step 2. After running q' iterations of Grover search, the attacker measures a random element that does not pass the check. $\square$

Remark 6. The attack of Lemma 15 works even if the adversary does not control the tweaks queried. It requires $\mathcal{O}(q)$ quantum-accessible classical memory. If the tweaks are controlled (but still non adaptatively), the *offline Simon's algorithm* of [11] reduces the memory down to $\mathcal{O}(n^2)$ qubits. The attack then is exactly the related-key attack of [11, Section 6.1]

C Attack on a Weakened QCB

In this section, we describe a forgery attack on a variant of QCB in which the IV is not used in the AD processing. It is also applicable to ΘCB3 [25, 19], and it is the first quantum forgery attack on this mode when used with an ideal TBC.

In this case, the tag of the empty message and one block of associated data is

$$T = f(IV) \oplus \widetilde{E}_{K,(2,0)}(a_0)$$

Hence, we can query the one-bit input function

$$T(x) = f(IV) \oplus \widetilde{E}_{K,(2,0)}(0||x)$$

Using Lemma 2, we can compute the i-th output bit of this function, that we note $T_i(x)$.

We propose to use Deutsch's algorithm, which allows to tell in one query wether a one-bit input, one-bit output function is constant or not, on the function $T_i(x)$. It will be constant if and only if the i-th bit of $\widetilde{E}_{K,(2,0)}(0||0) \oplus \widetilde{E}_{K,(2,0)}(0||1)$ is 0. Hence, in n queries (one to each of the $T_i(x)$), we can fully recover the value of $\widetilde{E}_{K,(2,0)}(0||0) \oplus \widetilde{E}_{K,(2,0)}(0||1)$. This is enough to make some forgeries, as it allows to compute a valid tag for any message with the associated data 1 given the tag for the same message with the associated data 0. We could also proceed similarly for any values of the type $\widetilde{E}_{K,(2,i)}(a) \oplus \widetilde{E}_{K,(2,i)}(b)$.

Note that this attack cannot be applied to QCB, as it requires the AD to be encrypted independently of the IV. Since the IV is used in all blocks in QCB, it is impossible for the adversary to mount such an attack, which relies on re-using the encryptions of some AD blocks of previous queries.