Provably Quantum-Secure Message Authentication Code

Master's Thesis of

Jérôme Nguyen:

at the Department of Informatics
KASTEL – Institute of Information Security and Dependability

Reviewer:	Prof. Dr. Jörn Müller-Quade
Second reviewer:	Prof. Dr. Thorsten Strufe
Advisor:	M.Sc. Marcel Tiepelt
Second advisor:	M.Sc. Astrid Ottenhues

30. September 2021 – 30. March 2022

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

Abstract

The threat posed by quantum computers against public-key schemes has been known for a long time. However symmetric-key schemes were believed to be relatively safe against quantum adversaries. Recent works have shown the existence of efficient quantum attacks against a large number of *message authentication codes* (MAC) in spite of their security proof in the classical setting. In response to the new quantum cryptanalysis efforts, research has been conducted on the security proofs in the quantum setting. Some existing schemes have been proven secure in the quantum setting. In addition, new algorithms have been designed to meet the requirements of quantum security.

This master thesis examines the use of nonces in the design of quantum-secure protocols. In a previous work, a generic transformation that makes a classically-secure MAC scheme quantum-secure was introduced. We show that this transform is not secure in general. However, we then argue that the transform does its intended purpose for many specific cases. To illustrate this, we apply it to the CBC-MAC scheme and prove its security. We do this by directly proving its *existential unforgeability under quantum chosen message attack* security. This allows us to avoid technical complications, and produces a short security proof in the standard model. Additionally, we formalize some design strategies for quantum secure protocols.

Zusammenfassung

Die Gefahr von Quantencomputer gegen asymmetrische Kryptographie ist schon lange bekannt. Jedoch wurden die Auswirkungen auf die symmetrische Kryptographie als weniger einschlägig betrachtet. In den letzten Jahren sind mehrere effiziente Quantenangriffe gegen *Nachrichtenauthentifizierungscode* (message authentication code, MAC) entdeckt worden. Aus diesem Grund wurde beweisbare Sicherheit dieser Primitive im Quantenmodell erforscht. Einige existierende Algorithmen wurden als quantensicher bewiesen. Darüber hinaus wurden neuen Protokolle entworfen welche auch Quantenangriffen widerstehen können.

In dieser Masterarbeit untersuchen wir den Einsatz von Noncen in der Konstruktion von quantensicheren Protokollen. In diesem Sinne hat eine vorherige Arbeit eine allgemeine Transformation für MACs eingeführt. Wir zeigen, dass diese Transformation im Allgemeinen nicht quantensicher ist. Dennoch behaupten wir, dass die Transformation in vielen spezifischen Fällen wirksam ist. Wir behandeln denn Fall von der CBC-MAC und zeigen das die transformierte Version quantensicher ist. Zudem formalisieren wir einige Entwurfstrategien für quantensichere Protokolle.

Contents

1. Introduction

Shor's 1994 seminal paper [Sho97] proved to the cryptography world that quantum computers would break the most widely used public-key protocols, such as RSA [RSA78] and the Diffie-Hellman key exchange [DH06]. To our knowledge, no large-scale quantum computers capable of executing such an attack currently exists. Still, the development and standardisation of new protocols is a long enterprise. The quantum-secure algorithms that will be needed in the future must be designed now. This led the rise of the field of post-quantum cryptography in recent years. In 2016, the National Institute for Standards and Technology (NIST) started the process of finding a post-quantum secure standard for public-key encryption as well as digital signatures algorithms.

Noticeably, this effort did not include any work on symmetric-key protocols. Indeed, for a long time no better algorithm than the Grover search algorithm [Gro96] was known for such schemes. As this only provided a quadratic speed-up in the exhaustive key-search, it was believed that a simple re-scaling of the security parameters would be sufficient for these algorithms. In 2010 Kuwakado and Hidenori [KM10] presented a new kind of quantum attack on symmetric-key algorithms. In addition to local quantum computations, their attack assumes access to an encryption oracle that accepts queries in quantum superposition. In this stronger model, their attack breaks the 3-round Feistel cipher in polynomial time. Since then, many similar superposition attacks have been found, even on schemes that were proven secure in the classical setting.

On the other hand, some schemes have been proven secure in this model. Towards this, new proving techniques had to be developed, since many classical theoretical tools do not translate well to the quantum setting. This opens up the question of which current protocols are already quantum secure and motivates the design of new classical quantum-secure schemes. In this line of thought, Haas proposed a general transform that takes a *message authentication code* (MAC) and makes it quantum secure using nonces [Haa20]. The idea being that with a minor design change, it would be possible to make any MAC scheme quantum secure in exchange for a small execution-time increase.

This master thesis examines the use of nonces in the design of quantum-secure protocols. More generally, it aims at identifying the design strategies that make protocols quantum secure. We focus our approach on lightweight protocols based on quantum secure block ciphers. By definition, these are schemes that have a sparing structure for efficiency reasons. They often use bitwise operation that can be efficiently implemented on low-resource devices.

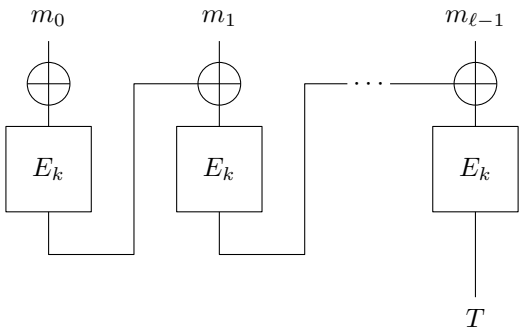


Figure 1.1.: CBC-MAC

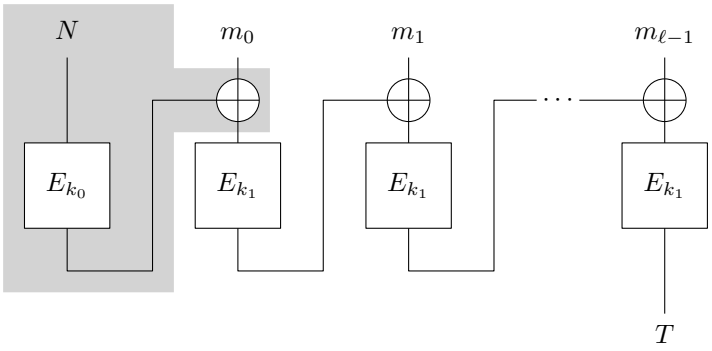


Figure 1.2.: Transformed CBC-MAC scheme. The gray part highlights the transformation.

1.1. Contributions

We start by considering Haas' transform. It essentially XORs an encrypted nonce on the first message block before passing it to the MAC algorithm. We show that the transform is not quantum-secure in general, even when the MAC is based on a quantum-secure block cipher. Indeed, we are able to build a pathological construction and give an efficient quantum attack against it.

We then argue that the transform works in specific cases. We show this by applying it to the CBC-MAC, depicted in Figure 1.1. We prove the quantum security of the transformed version, shown in Figure 1.2.

Main Theorem (Informal)**.** *If E is a quantum-secure pseudorandom permutation, then the transformed CBC-MAC is a quantum-secure MAC against nonce-respecting adversaries.*

The *existentially unforgeable under quantum chosen-message attack* (EUF-qCMA) security notion requires a successful adversary to output $q + 1$ distinct classical message-tag pairs. Hence, we chose to directly prove the EUF-qCMA security of the scheme instead of first proving its pseudorandomness as is usually the case in the classical setting proofs. This allows us to work on a set of classical values for our reduction. This leads to a compact proof that does not require highly technical techniques such as Zhandry's compressed oracle [Zha18].

1.2. Related Works

After their first paper on the quantum attack against the 3-round Feistel [KM10], Kuwakado and Hidenori published in 2012 a superposition attack on the Evan-Mansour block cipher [KM12]. In 2016, Kaplan et al. [Kap+16] give attacks on the LRW tweakable block cipher, block cipher based authentication protocols such as the OCB, GCM, or the CBC-MAC, as well as multiple CAESAR candidates. Furthermore, they show that slide attacks obtain an exponential speed-up in the quantum-setting.

In order to start proving the security of schemes, the appropriate attacker models had to be formalized first. Towards this, Boneh and Zhandry define the *indistinguishable under quantum chosen-message attacks* (IND-qCPA) model in [BZ13b]. This is meant to be a quantum equivalent to the *indistinguishable under chosen-message attacks* (IND-CPA) security definition for encryption schemes. When defining security models, a fine balance has to be met between giving a strong model that will indeed guarantee good security in practice, and a model so strong that no scheme could possibly achieve it. Bhaumik et al. further discuss this model in [Bha+20], and give an impossibility result for an alternative, stronger definition. In 2013, Boneh and Zhandry define the EUF-qCMA security for MACs in [BZ13a]. This security notion has been widely used in the subsequent papers, such as the one by Kaplan et al. we mentioned above. However, in 2017 Garg et al. propose a different model in [GYZ17]. They show that certain MACs that are secure in Boneh and Zhandry's model suffer from unwanted attacks. For the same reasons, a third model has been published in 2018 by Alagic et al. [Ala+20]. However, Boneh and Zhandry's original EUF-qCMA definition remains the most widely used in spite of its flaws. Indeed, it is technically less demanding to prove security in this model.

Many of the reduction technique used in the classical setting cannot be used directly in the quantum setting. In particular, we cannot easily record quantum queries, as this would mean measuring them which perturbs the adversary's state. Zhandry's groundbreaking *compressed oracle* technique [Zha18] is meant to bridge this gap. It allows to record the queries to a random oracle under certain conditions. When using this tool, we can use lazy-sampling ideas for the quantum random oracle. This means that some classical intuition can be applied to quantum security proofs. This technique has given rise to many subsequent works. Hosoyamada and Iwata have given an alternative formalization of the technique called *recording standard oracle with error* (RstOE). The compressed oracle is a highly technical tool, the RstOE is designed to be simpler to use. The "Bad-flag" analysis is another common classical technique that cannot be used direclty in the quantum setting. In [Unr15], Unruh presents the *one-way to hiding* lemma. This is the quantum equivalent to the classical "Bad-flag" analysis. Czajkowski et al. show how to apply the compressed oracle technique in conjunction with the one-way to hiding lemma [Cza+19]. They use this to define a game-playing proof framework.

Alagic and Russel have presented a generic transformation against superposition attacks [AR17]. Their idea is to replace additions in $(\mathbb{Z}/2)^n$, i.e the XOR, with an operation over alternative finite groups such as $(\mathbb{Z}/2^n)$. Indeed they observe that the group $(\mathbb{Z}/2)^n$ has too much structure, which is exploited in the attacks through Simon's algorithm. They applied their transformation on the Evan-Mansour construction and the CBC-MAC. They

are then able to give a reduction from the security of these schemes to the general hidden shift problem, which is considered a good quantum hardness assumption. A follow-up to this work has been published by Bonnetain and Naya-Plasencia in 2018 [BN18]. They claim that while Alagic and Russel's construction does indeed increase the security of the schemes, it comes at the price of efficiency. They show that the construction requires a significant increase in the size of the internal states of the schemes. This makes the transformation ill suited in resource-constrained settings.

1.3. Structure of the Work

Chapter 2 describes notation and gives the basic definitions used in the rest of the paper. Chapter 3 contains a discussion on the issues that arise when conducting security proofs in the Q2 setting. We then address the quantum security of the generic transform by Haas in Chapter 4 and prove the quantum-security of the transformed CBC-MAC. Finally, in Chapter 5 we relate the security of the Haas transformation with earlier security proofs in the Q2 setting to give some heuristics on designing quantum-secure protocols.

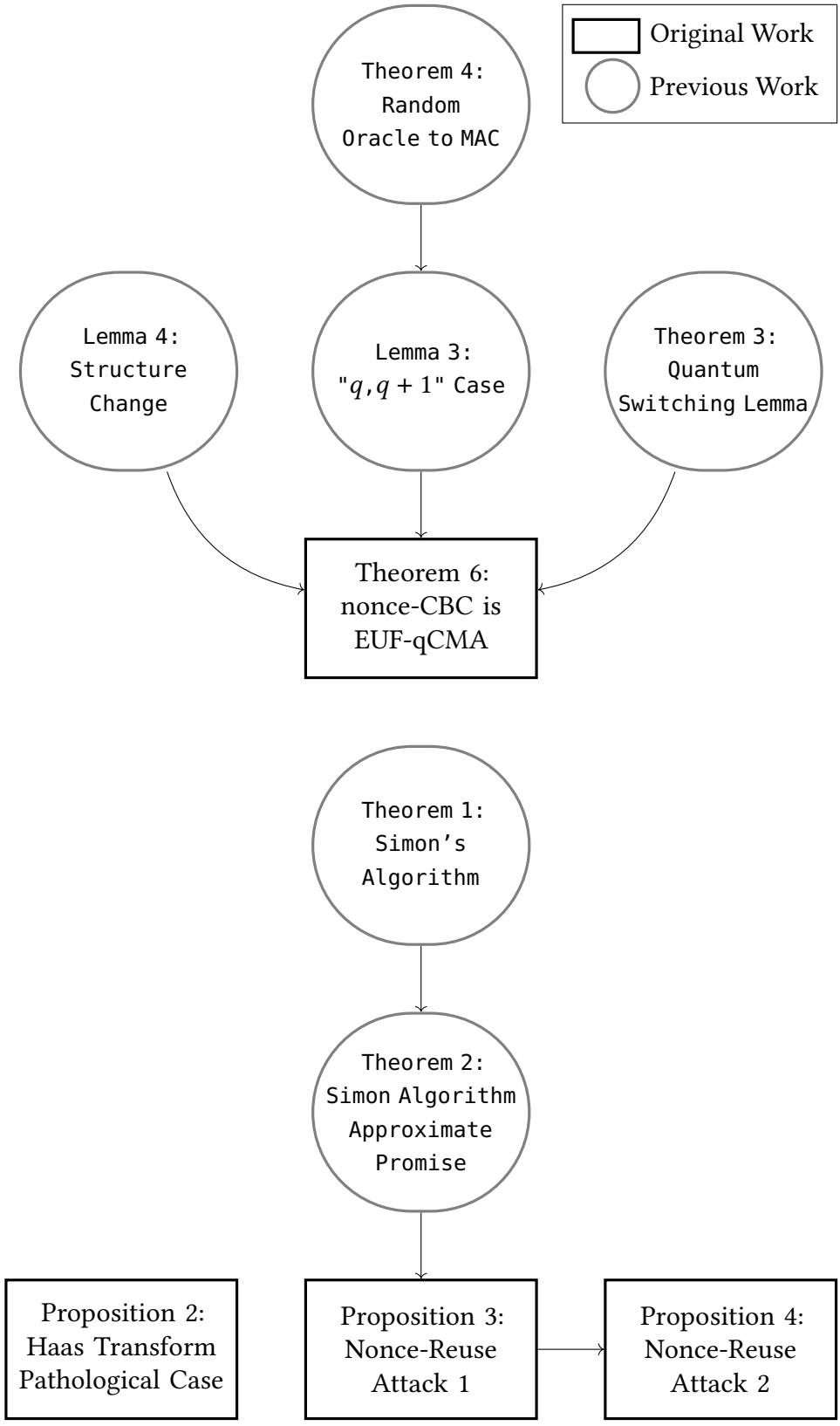


Figure 1.3.: Theorem Dependencies Overview

2. Preliminaries

We start this chapter by defining the notations used in the rest of the work. We then give the definitions for the primitives and security notions that we will encounter. Furthermore, we give a formal description of the quantum-setting. In particular, we describe the quantum oracle access. For the basics of quantum computation, we refer the reader to [NC00].

Notations. In the whole thesis, we define $n \in \mathbb{N}$ to be the security parameter. We write in uppercase cursive letters $\mathcal{K}, \mathcal{X}, \mathcal{Y}$ to denote sets. We usually use families of sets parametrized by the security parameter such as $(\mathcal{X}_n)_n$, but we will omit explicitly writing it when it is clear from context. It is then understood that $\mathcal{X}$ stands for $(\mathcal{X}_n)_n$.

We say that a function f is *polynomially-bounded* if and only if there exists a polynomial function p and an $x^* \in \mathbb{R}$, such that for every $|x| \geq |x^*|$, we have $|f(x)| \leq |p(x)|$. We write $\mathrm{poly}(n)$ to denote an unspecified pollynomially-bounded function. We say that a function g is *negligible* if and only if, for any polynomial p, there exists an $x^* \in \mathbb{R}$, such that for every $|x| \geq |x^*|$, we have $|g(x)| \leq \left(\frac{1}{p(x)}\right)$. We write $\mathrm{negl}(n)$ to denote an unspecified negligible function in the security parameter. We sometimes omit the security parameter n.

We use cursive letters $\mathcal{A}, \mathcal{B}$ or $\mathcal{D}$ to denote algorithms. We sometimes call them *adversary* or *distinguisher*. They can be defined to be *deterministic polynomial-time* (DPT), *probabilistic polynomial-time* (PPT) or *quantum polynomial-time* (QPT). For an algorithm $\mathcal{A}$, and an oracle O, we write $\alpha \leftarrow \mathcal{A}^O$ the event that $\mathcal{A}$ runs relative to the oracle O and outputs α. If the adversary has quantum access to its oracle, we denote that by putting the oracle in a ket, e.g $\alpha \leftarrow \mathcal{A}^{|O\rangle}$. The integer q will always denote the numbers of queries an algorithm $\mathcal{A}$ makes to an oracle O. If not specified otherwise, q will be a polynomial number in n.

We use the letter E to denote a block cipher. Let $k \in \mathcal{K}$ be an element of the key space of E. We may write $E(k, \cdot)$ or $E_k(\cdot)$ interchangeably to denote a call to E with key k. Likewise, for any keyed algorithm, e.g a signing algorithm Sign, we may write interchangeably $\mathsf{Sign}(k, \cdot)$ or $\mathsf{Sign}_k(\cdot)$. If an algorithm S takes a nonce N as an additional input, we may write $S_N(\cdot)$ or $S(\cdot; N)$.

2.1. Security Definitions

This section defines the primitives we use in this paper. We give their classical security definition for context, before giving their quantum security definitions in Section 2.2.3.

Oracle distinguishing advantage For two oracles O_1 and O_2, we define the distinguishing advantage of an oracle aided algorithm $\mathcal{A}$ by,

$$\mathsf{Adv}^{\mathrm{dist}}_{O_1,O_2}(\mathcal{A}) := \left|\Pr\left[b \leftarrow \mathcal{A}^{O_1} : b = 1\right] - \Pr\left[b \leftarrow \mathcal{A}^{O_2} : b = 1\right]\right|$$

where the probabilities are over the randomness of the oracles O_1, O_2 and adversary $\mathcal{A}$.

If for any *polynomial-time* algorithm $\mathcal{A}$, the distinguishing advantage $\mathsf{Adv}^{\mathrm{dist}}_{O_1,O_2}(\mathcal{A})$ is negligible in the security parameter, we call O_1 and O_2 *computationally indistinguishable*.
If for any *unbounded* algorithm $\mathcal{A}$, the distinguishing advantage $\mathsf{Adv}^{\mathrm{dist}}_{O_1,O_2}(\mathcal{A})$ is negligible in the security parameter, we call O_1 and O_2 *statistically* or *information theoretically* indistinguishable.

The following definitions are based on [Gag17, Section 3.1]. A (family of) *pseudorandom functions* (PRF) is a family indexed by $k \in \mathcal{K}$ of efficiently computable functions $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that, without knowledge of k, it is classically computationally indistinguishable from the collection of all function from $\mathcal{X}$ to $\mathcal{Y}$ (denoted by $\mathcal{Y}^{\mathcal{X}}$). We write $F_k : \mathcal{X} \to \mathcal{Y}$ to denote the member of the family indexed by k.

Definition 2.1.1 (Pseudorandom Function (PRF))**.** A (family of) *pseudorandom functions* (PRF) from $\mathcal{X}$ to $\mathcal{Y}$ with key space $\mathcal{K}$ is a DPT algorithm $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that for any classic PPT algorithm $\mathcal{D}$ it holds that,

$$\mathsf{Adv}^{\mathrm{PRF}}_{F,RF}(\mathcal{D}) := \mathsf{Adv}^{\mathrm{dist}}_{F,RF}(\mathcal{D}) = \left|\Pr_{k \xleftarrow{\$} \mathcal{K}}\left[1 \leftarrow \mathcal{A}^{F_k}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{RF}\right]\right| \leq \mathrm{negl}$$

where RF is a random oracle, and the probabilities are over the choice of k, the randomness of RF and of $\mathcal{D}$.

A (family of) *pseudorandom permutations* (PRP) is a (family of) pseudorandom functions that is also an invertible permutation on some space $\mathcal{X}$. We write $P_k : \mathcal{X} \to \mathcal{X}$ to denote the member of the family indexed by k.

Definition 2.1.2 (Pseudorandom Permutation (PRP))**.** A (family of) *pseudorandom permutations* (PRP) on $\mathcal{X}$ with key space $\mathcal{K}$ is a pair of DPT algorithm $P, P^{-1} : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ such that:

- For all keys $k \in \mathcal{K}$, we have that P_k, P_k^{-1} are permutations on $\mathcal{X}$;
- For all keys $k \in \mathcal{K}$, we have that $(P_k)^{-1} = (P_k^{-1})$; and
- for any classical PPT algorithm $\mathcal{D}$ it holds:

$$\mathsf{Adv}^{\mathrm{PRF}}_{P,RP}(\mathcal{D}) := \mathsf{Adv}^{\mathrm{dist}}_{P,RP}(\mathcal{D}) = \left|\Pr_{k \xleftarrow{\$} \mathcal{K}}\left[1 \leftarrow \mathcal{A}^{P_k}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{RP}\right]\right| \leq \mathrm{negl}$$

where RP is a random permutation oracle, and the probabilities are over the choice of k, the randomness of RP and of $\mathcal{D}$.

In this work, we will interchangeably use the term *block cipher* for a pseudorandom permutation. We now give the well known random permutation/random function switching lemma. Block ciphers are defined to behave as pseudorandom permutations, but oftentimes, it is easier to handle pseudorandom functions in security proofs. This lemma bridges this gap.

Lemma 1 (RF/RP Switching Lemma [BR04])**.** *Let $RP : \mathcal{X} \to \mathcal{X}$ be a random permutation oracle, and let $RF : \mathcal{X} \to \mathcal{X}$ be a random function oracle. Let $\mathcal{A}$ be a classical adversary that makes at most q oracle queries. Then,*

$$\left|\Pr\left[\mathcal{A}^{RP} = 1\right] - \Pr\left[\mathcal{A}^{RF} = 1\right]\right| \leq \frac{q^2}{2|\mathcal{X}|}$$

We will also encounter *tweakable block ciphers*. This is a concept introduced by Liskov, Rivest and Wagner in [LRW02]. This can be seen as a block cipher that receives an additional input called tweak. Conceptually, this tweak should be "cheaper" to change than the key, and should have no requirement to stay secret for the safety of the scheme.

Definition 2.1.3 (Tweakable Block Cipher)**.** A *tweakable block cipher* with message space $\mathcal{M}$, key space $\mathcal{K}$, and tweak space $\mathcal{T}$, is a pair of DPT algorithm $\tilde{E}, \tilde{E}^{-1} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ such that:

- For all key-tweak pairs $(k, t) \in \mathcal{K} \times \mathcal{T}$, we have that $\tilde{E}_k(t, \cdot), \tilde{E}_k^{-1}(t, \cdot)$ are permutations on $\mathcal{X}$;
- For all key-tweak pairs $(k, t) \in \mathcal{K} \times \mathcal{T}$ we have that $(\tilde{E}_k)^{-1}(t, \cdot) = (\tilde{E}_k^{-1})(t, \cdot)$; and
- for any classic PPT algorithm $\mathcal{D}$ it holds:

$$\mathrm{Adv}^{\widetilde{\mathrm{qPRP}}}_{\tilde{E},\widetilde{RP}}(\mathcal{D}) := \mathrm{Adv}^{\mathrm{dist}}_{\tilde{E},\widetilde{RP}}(\mathcal{D}) = \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} \left[1 \leftarrow \mathcal{A}^{\tilde{E}}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\widetilde{RP}}\right] \right| \leq \mathrm{negl}$$

where $\widetilde{RP}^t$ is a family of independently drawn random permutation indexed by the tweak t. We may write $\tilde{E}_k^t(\cdot)$ instead of $\tilde{E}_k(t, \cdot)$.

This master thesis concerns itself with *message authentication codes* (MAC). These are schemes that allow authentication in the symmetric-key setting, by pairing any sent message with an authentication tag.

Definition 2.1.4 (Message Authentication Code (MAC))**.** A *message authentication code* (MAC), with keyspace $\mathcal{K}$, message space $\mathcal{M}$, and tag space $\mathcal{T}$, is a tuple of PPT algorithms $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Verify})$ such that,

- $\mathrm{Gen}(1^n) \to k$ generates a secret key $k \in \mathcal{K}$
- $\mathrm{Sign}(k, m) \to t$ for any $m \in \mathcal{M}$ outputs a tag $t \in \mathcal{T}$

- $\text{Verify}(k, m, t) \to b$ is a deterministic algorithm that either outputs $b = 1$ to accept the message-tag pair or $b = 0$ to reject it.

In the classical setting, we consider MACs that are secure under *chosen message attack* (CMA). Here the attacker has access to a tag oracle. The usual adversary goal is to find a valid forgery for any message in the message space. We call this goal *existential unforgeability* (EUF).

Definition 2.1.5 ($\text{Exp}^{\text{EUF-CMA}}_{\Pi,\mathcal{A}}(n)$)**.** Let $\Pi = (\text{Gen}, \text{Sign}, \text{Verify})$ be a MAC scheme. Let $n \in \mathbb{N}$ be the security parameter and let $\mathcal{A}$ be a PPT adversary. We define the EUF-CMA security experiment by,

$\text{Exp}^{\text{EUF-CMA}}_{\Pi,\mathcal{A}}(n)$	$\text{Sign}_k(m)$
$Q := \emptyset$	$t \leftarrow \text{Sign}(k, m)$
$k \leftarrow \text{Gen}(1^n)$	$Q := Q \cup m$
$(m^*, t^*) \leftarrow \mathcal{A}^{\text{Sign}_k}$	**return** t
return $\text{Verify}(k, m^*, t^*) \wedge (m^* \notin Q)$	

We say that the MAC Π is EUF-CMA secure if for any PPT adversary $\mathcal{A}$,

$$\text{Adv}^{EUF-CMA}_{\Pi}(\mathcal{A}) := \Pr[\text{Exp}^{\text{EUF-CMA}}_{\Pi,\mathcal{A}}(n) = 1] \leq \text{negl}(n)$$

2.2. The Quantum Setting

The term "*quantum security*" has been sometimes used inconsistently in the literature. To avoid any confusion, we use the classification of quantum models introduced by Gargliardoni in [Gag17].

Q0: The classical setting
This is the classical setting, where the adversary has no access to any kind of quantum computation. This encompasses all the definitions given above in Section 2.1.

Q1: Local quantum computations, classical queries
In this setting, an adversary may have access to a quantum computer and do local quantum computations. On the other hand, any queries to an oracle have to be made classically. Security in this model is called *post-quantum security* or sometimes *standard security*.

Q2: Local quantum computations, quantum queries
In this setting, an adversary is given quantum superposition access to its oracles. Speaking loosely, this means that an adversary may query a superposition of inputs, and the oracle will respond with the corresponding superposition of outputs. We define this formally in section 2.2.1. We call security in this model *quantum security*.

Notice that in the Q2 model, the adversary is allowed to make quantum queries to a classical algorithm. This may seem like an unlikely situation at first. But consider a

hypothetical future situation where a quantum computer runs an encryption algorithm. It could be possible that the scheme runs a classical protocol as a subroutine. In that case, you would require the use of Q2 secure schemes.

A further motivating example for the relevance of the Q2 setting is given in [GHS16, "frozen smart-card" example]. Consider a small device, such as an RFID tag or a smart-card, running a purely classical encryption scheme (such as AES) on classical inputs, and outputting classical values. A potential future adversary could attempt a new kind of side-channel attack on the device, by putting it in conditions that would make it "take on" a quantum behavior, i.e answering queries in superposition with answers in superposition. This would be akin to fault-injection side-channel attacks we see nowadays, where an adversary might freeze a smart-card to make it take on some faulty behavior.

A quantum-secure algorithm could be implemented in a black-box manner, and remain secure long into the future with no additional hardware requirements. On the contrary, quantum-insecure schemes could only be used in cases where no such side-channel attacks would be realistic (e.g queries must go through a classical network).

2.2.1. Quantum Oracle and Quantum Adversary

In this section, we define our model for quantum adversaries. Moreover, we give the quantum oracle access model used in thesis. We base this section on [Bha+20, Section 2.2].

In this thesis, a quantum adversary $\mathcal{A}$ that makes q queries to its oracle is modelled as a sequence of unitary operators $(U_0, \dots U_q)$, where each U_i is an unitary operator on an s-qubit quantum system. We will see U_0 and U_q as an initialization and finalization processes respectively. For $1 \leq i \leq q-1$, U_i is the work after the i-th query. We assume that $\mathcal{A}$'s quantum state is a vector of a Hilbert space

$$\mathcal{H}_{\mathcal{A}} = \mathcal{H}_{\text{query}} \otimes \mathcal{H}_{\text{answer}} \otimes \mathcal{H}_{\text{work}}$$

where $\mathcal{H}_{\text{query}}$, $\mathcal{H}_{\text{answer}}$, $\mathcal{H}_{\text{work}}$ correspond to $\mathcal{A}$'s query, answer, and offline work registers respectively.

A quantum oracle is a unitary operator that acts on the adversary's $\mathcal{H}_{\text{query}} \otimes \mathcal{H}_{\text{answer}}$ register. In this thesis, we use the standard oracle definition.

Definition 2.2.1 (Quantum oracle)**.** Let $f : \{0,1\}^m \to \{0,1\}^n$ be a function. Then, the quantum oracle of f is defined as the unitary operator

$$\mathcal{O}_f : |x, y\rangle \to |x, y \oplus f(x)\rangle$$

where $|x\rangle\,|y\rangle$ are registers provided by the adversary and correspond to its query and answer registers respectively.

Note that if f is a random function, the same randomness is used to answer every message in the superposition. The oracle only draws a new random value between subsequent queries.

Assume we run $\mathcal{A}$ relative to the oracle O_f. Then the unitary operators U_i and oracle calls O_f act sequentially on the initial state $|0^l\rangle$. The final state is then,

$$U_q O_f U_{q-1} \dots O_f U_0 |0^l\rangle \tag{2.1}$$

This final state is then measured and $\mathcal{A}$ returns the result as its output. In this thesis, we will always use the adversary as a black-box algorithm. In particular, we do not consider the state of the work register $\mathcal{H}_{\text{work}}$. We only care about the adversary's final output. Note that this is always a classical value. Hence, we omit writing the $\mathcal{H}_{\text{work}}$ register from now on.

2.2.2. Simon's Algorithm

Many of the published attacks in the Q2 setting use Simon's algorithm [Sim94] to solve the following problem in polynomial-time.

Definition 2.2.2 (Simon problem)**.** Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function such that for an $s \in \{0,1\}^n$, for all $x, y \in \{0,1\}^n$, $f(x) = f(y)$ if and only if $x \oplus y \in \{0^n, s\}$. The aim of the problem is to find s.

Theorem 1 ([Sim94])**.** *After $n+\alpha$ iterations, Simon's algorithm solves Simon's problem with probability $1 - 2^{-\alpha}$.*

In each round, Simon's algorithm recovers a vector orthogonal to the period s. Hence, in $O(N)$ queries, the algorithm provides a full rank system of linear equations. This allows to compute the period s using linear algebra techniques such as Gaussian elimination.

The attacks based on Simon's algorithm build a periodic function from their oracle. They then run Simon's algorithm to recover the period. This can then either directly leads to a distinguishing attack, or the period contains sensitive information that allows a forgery attack.

In some cases, the function f is not perfectly periodic and has unwanted "parasitic" collisions. In that case, Kaplan et. al [Kap+16] have shown that Simon's algorithm may still be able to find the period, provided the number of additional collisions is not too high.

For $f : \{0,1\}^n \to \{0,1\}^n$ such that $f(x) = f(x \oplus s)$ for all x, consider,

$$\epsilon(f, s) := \max_{t \in \{0,1\}^n \setminus \{0,s\}} \Pr_x[f(x) = f(x \oplus t)] \tag{2.2}$$

Theorem 2 (Simon's algorithm with approximate promise [Kap+16])**.** *If $\epsilon(f,s) \le p_0$, then Simon's algorithm returns s in $c \cdot n$ queries, with probability at least $1 - \left(2\left(\frac{1+p_0}{2}\right)^c\right)^n$.*

Kaplan et al. further note that choosing $c \ge 3/(1-p_0)$ makes the error probability decrease exponentially.

2.2.3. Quantum Security Definitions

We now give the security definitions adapted to the Q2 setting. This mostly follows [Gag17, section 5.2]. For most definitions it is only a matter of allowing quantum adversaries, and giving them quantum access to the oracles. However, we will see that some primitives demand a more subtle approach.

Definition 2.2.3 (Quantum-Secure Pseudorandom Function (qPRF))**.** A (family of) *quantum-secure pseudorandom functions* (qPRF) from $\mathcal{X}$ to $\mathcal{Y}$ with key space $\mathcal{K}$ is a DPT algorithm $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ such that for any QPT algorithm $\mathcal{A}$ it holds:

$$\mathsf{Adv}^{\mathrm{qPRF}}_{F,RF}(\mathcal{A}) \coloneqq \mathsf{Adv}^{\mathrm{dist}}_{F,RF}(\mathcal{A}) = \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} \left[1 \leftarrow \mathcal{A}^{|F_k\rangle} \right] - \Pr \left[1 \leftarrow \mathcal{A}^{|RF\rangle} \right] \right| \leq \mathrm{negl}$$

where $|RF\rangle$ is a quantum random oracle, and the probabilities are over the choice of k, the randomness of RF and of $\mathcal{A}$.

Definition 2.2.4 (Quantum-Secure Pseudorandom Permutation (qPRP))**.** A (family of) *quantum-secure pseudorandom permutations* (qPRP) on $\mathcal{X}$ with key space $\mathcal{K}$ is a pair of DPT algorithm $P, P^{-1} : \mathcal{X} \to \mathcal{X}$ such that:

- For all keys $k \in \mathcal{K}$, we have that P_k, P_k^{-1} are permutations on $\mathcal{X}$;
- For all keys $k \in \mathcal{K}$, we have that $(P_k)^{-1} = (P_k^{-1})$; and
- for any QPT algorithm $\mathcal{A}$ it holds:

$$\mathsf{Adv}^{\mathrm{qPRF}}_{P,RP}(\mathcal{A}) \coloneqq \mathsf{Adv}^{\mathrm{dist}}_{P,RP}(\mathcal{A}) = \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} \left[1 \leftarrow \mathcal{A}^{|P_k\rangle} \right] - \Pr \left[1 \leftarrow \mathcal{A}^{|RP\rangle} \right] \right| \leq \mathrm{negl}$$

  where $|RP\rangle$ is a quantum random permutation oracle, and the probabilities are over the choice of k and the randomness of RP and of $\mathcal{A}$.

Definition 2.2.5 (Quantum-secure Tweakable Block Cipher)**.** A *Quantum-secure tweakable block cipher* (qTBC) with message space $\mathcal{M}$, key space $\mathcal{K}$, and tweak space $\mathcal{T}$, is a pair of DPT algorithm $\tilde{E}, \tilde{E}^{-1} : \mathcal{K} \times \mathcal{T}\mathcal{M} \to \mathcal{M}$ such that:

1. For all key-tweak pair $(k, t) \in \mathcal{K} \times \mathcal{T}$, we have that $\tilde{E}_k(t, \cdot), \tilde{E}_k^{-1}(t, \cdot)$ are permutations on $\mathcal{X}$;
2. For all key-tweak pair $(k, t) \in \mathcal{K} \times \mathcal{T}$, we have that $(\tilde{E}_k)^{-1}(t, \cdot) = (\tilde{E}_k^{-1})(t, \cdot)$; and

3. for any classic QPT algorithm $\mathcal{D}$ it holds:

$$\mathsf{Adv}^{\mathrm{PRF}}_{\tilde{E},\widetilde{RP}}(\mathcal{D}) \coloneqq \mathsf{Adv}^{\mathrm{dist}}_{\tilde{E},\widetilde{RP}}(\mathcal{D}) = \left| \Pr_{k \xleftarrow{\$} \mathcal{K}} \left[1 \leftarrow \mathcal{A}^{|\tilde{E}\rangle}\right] - \Pr\left[1 \leftarrow \mathcal{A}^{|\widetilde{RP}\rangle}\right] \right| \leq \mathrm{negl}$$

where $\widetilde{RP}$ is a family of independently drawn random permutations indexed by the tweak t. The probabilities are over the choice of k and the randomness of $\tilde{RP}$ and of $\mathcal{D}$.

While the definition of a MAC does not change compared to Definition 2.1.4, we need to carefully approach translating EUF-CMA security to the Q2 setting. In the classical setting, we require the forgery to be made for a fresh message. In the quantum case, any query to the tag oracle can be a superposition of all the messages in the message space. As such, an adversary could submit different queries in superposition designed to have a high probability of collapsing to the same message. Boneh and Zhandry define the EUF-qCMA model in [BZ13b, Definition 2.1]. They require an adversary that makes q queries, to provide $q+1$ distinct and valid classical message-tag pairs to win the experiment. Notice that for a classical adversary, this corresponds to the EUF-CMA security.

Definition 2.2.6 ($\mathrm{Exp}^{\text{EUF-qCMA}}_{\Pi,\mathcal{A}}(n)$)**.** Let $\Pi = (\mathrm{Gen}, \mathrm{Sign}, \mathrm{Verify})$ be a MAC scheme. Let $n \in \mathbb{N}$ be the security parameter and q a polynomial number in n. Let $\mathcal{A}$ be a QPT adversary that makes q quantum queries to its oracle. Then we define the EUF-qCMA security experiment by,

$\mathrm{Exp}^{\text{EUF-qCMA}}_{\Pi,\mathcal{A}}(n,q)$

$k \leftarrow \mathrm{Gen}(1^n)$

$S = \{(m_1,t_1),(m_2,t_2),\dots(m_{q+1},t_{q+1})\} \leftarrow \mathcal{A}^{|\mathrm{Sign}_k\rangle}$

return $\mathrm{isForgeSet}(S)$

Sign_k

$|m\rangle\,|y\rangle \rightarrow |m\rangle\,|y \oplus \mathrm{Sign}(k,m)\rangle$

$\mathrm{isForgeSet}(S)$

$Valid \coloneqq 1$

$Distinct \coloneqq 1$

for $i = 1$ **to** $q+1$:

  $Valid \coloneqq Valid \wedge \mathrm{Verify}(k, m_i, t_i)$

  $Distinct \coloneqq Distinct \wedge (\nexists j \neq i, (m_i,t_i) = (m_j,t_j))$

return $Valid \wedge Distinct$

We say that the MAC Π is *existentially unforgeable under quantum chosen-message attack* (EUF-qCMA) if for any QPT adversaries $\mathcal{A}$ that makes q queries to its oracle,

$$\mathsf{Adv}^{\text{EUF-qCMA}}_{\Pi}(\mathcal{A}) \coloneqq \Pr[\mathrm{Exp}^{\text{EUF-qCMA}}_{\Pi,\mathcal{A}}(n,q) = 1] \leq \mathrm{negl}(n)$$

2.3. Nonces

In the classical setting, the use of nonces is a well known concept. A nonce is an additional input to the primitive that is meant to be used only once (hence the name). It is used in deterministic algorithms to give variance to successive runs of a primitive on the same input. The nonce does usually not need to be kept secret for the security of a scheme. Moreover, we even let the adversary choose the nonces in the security games.

The nonce version of any classical experiment proceeds similarly to the regular version of it. If in the regular experiment, the PPT adversary $\mathcal{A}$ sends a query m, then in the nonce experiment, $\mathcal{A}$ additionally chooses a nonce N and queries the tuple (N, m). The oracle verifies if the nonce is new before sending the corresponding answer. If the nonce is repeated, then the experiment is aborted and the adversary loses.

The use of nonces is trickier in the quantum setting. As discussed previously for MACs, it is not clear how to define freshness for nonces when the adversary is allowed to make queries in superposition. An attempt at giving an appropriate model has been made in [Haa20]. Haas introduces the *no quantum nonce reuse* (NQNR) model. It allows nonces to be queried in superposition, but they have to me measured at the latest, at the end of the interaction. If the same nonce is measured twice, then the experiment is lost.

We argue that this model is too artificial for our purposes. We will instead require the nonces to be classical and distinct i.e we assume the adversary to be nonce-respecting in the classical sense. This is the model used by Bhaumik et al. in [Bha+20]. If an oracle implements a scheme f that admits a nonce, we model this as a family of unitary operators $(\mathcal{O}_{f,N})_N$ indexed by the nonce. We now define the nonce version of the EUF-qCMA security experiment.

Definition 2.3.1 ($\mathrm{Exp}^{\text{nonce-EUF-qCMA}}_{\Pi,\mathcal{A}}(n)$)**.** Let $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be a MAC scheme that uses nonces. Let $\mathcal{A}$ be a QPT adversary that makes q quantum queries to its oracle. The nonce-EUF-qCMA security game proceeds in two phases.

1. For the i'th query, the adversary $\mathcal{A}$ chooses a fresh classic value N_i as a nonce. The oracle then answers with the unitary operator Sign_{N_i} that is defined by

$$|x\rangle\,|y\rangle \to |x\rangle\,|y \oplus \mathsf{Sign}(k, x; N_i)\rangle$$

2. $\mathcal{A}$ produces $q+1$ classic tuple (N, M, T) with any N of its choice and wins the game if they are all valid and distinct.

We say that the MAC Π is *nonce existentially unforgeable under quantum chosen-message attack* (nonce-EUF-qCMA) if for any QPT adversaries $\mathcal{A}$ that makes q queries to its oracle,

$$\mathrm{Adv}^{\text{nonce-EUF-qCMA}}_{\Pi}(\mathcal{A}) \coloneqq \Pr[\mathrm{Exp}^{\text{nonce-EUF-qCMA}}_{\Pi,\mathcal{A}}(n, q) = 1] \leq \mathrm{negl}(n)$$

2.4. Derivatives of Binary Functions

Definition 2.4.1. For a function $f : \{0,1\}^n \to \{0,1\}^n$, we define the derivative of f at point $a \in \{0,1\}^n$ as

$$\Delta_a f(x) = f(x \oplus a) \oplus f(x)$$

As the derivative of f is also a function from $\{0,1\}^n$ to $\{0,1\}^n$, we can define the i'th ($i > 1$) as derivative of f at points $(a_1, a_2 \dots a_i)$ as

$$\Delta^{(i)}_{a_1,a_2\dots a_i} f(x) = \Delta_{a_i}(\Delta^{(i-1)}_{a_1,a_2\dots a_{i-1}} f(x))$$

where the 0'th derivative is f itself. The following proposition from [Lai94, Proposition 3] gives the general form of the derivative of a binary function.

Proposition 1. *Let $L[a_1, \dots, a_i]$ be the set of all 2^i possible linear combinations of $a_1, \dots a_i$. Then,*

$$\Delta^{(i)}_{a_1,\dots,a_i} f(x) = \sum_{c \in L[a_1,\dots,a_i]} f(x \oplus c)$$

3. Security Proofs in the Quantum Setting

In this chapter, we discuss some of the specific issues to proving security in the Q2 setting. Security proofs are a combination of two elements: a hardness assumption and a reduction. This also holds in the quantum setting, although we now need a quantum hardness assumption *and* a reduction that takes into account the abilities of a quantum adversary. We discuss these in Section 3.1 and in Section 3.2 respectively. In Section 3.3, we give the quantum tools that will be used in our proof in Chapter 4.

3.1. Hardness Assumptions

Concrete hardness assumptions are computational problems that are believed to be very difficult to solve. For our purposes a "hard" problem is one that is exponential in time (sometimes also in space) complexity. Identifying these problems as hard is a communal process. The confidence in a hard problem is build through years of research in which no significant contradiction to the hardness of the problem has been found (e.g a polynomial-time solution).

Quantum computers come as a new variable in this conversation. Since they behave in fundamentally different ways to a classical computer, every previously used hardness assumption has to be reassessed. Famously, Shor's algorithm [Sho97] is able to solve both the integer factorization problem as well as the discrete logarithm problem in polynomial time. On the other hand, problems such as the *learning with error* problem [Reg05] or the *general decoding of linear codes* problem [BMT78] both, for now, stand up to the quantum cryptanalysis efforts.

A shocking example relevant to our work is the case of the one-time pad. Famously, this scheme provides statistical security in the classical setting, and is the underlying idea behind stream ciphers. However, a quantum adversary can distinguish a one-time pad from a random function in a single query.

Lemma 2 ([Bon19])**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a function that implements a one time pad, i.e*

$$f(x) = x \oplus r, \quad r \xleftarrow{\$} \{0,1\}^n$$

There exists a quantum algorithm that makes a single query, that can distinguish between f and random function with probability $1 - \frac{1}{2^{n-1}}$

It is common for classical proofs to reduce the security of a scheme to a one-time pad. In the quantum setting, this is not an option. This a case where the classical intuition is very misleading when trying to prove quantum-security.

In this thesis, we often work with *generic* hardness assumptions. These are the building blocks for theoretical cryptography such as one-way functions (OWF), pseudorandom functions (PRF) or secret-key encryption schemes (SKE). We directly define these objects as having certain properties. This allows for a more general approach to security proofs.

As opposed to the *concrete* hardness assumption, these can be simply translated to the Q2 setting. We only need to tweak the definitions. As we have seen in Section 2.2.3, this is usually only a matter of allowing quantum adversaries and providing them with quantum oracle access.

3.2. Reduction

A reduction is an algorithm that transforms a problem into another one. In the context of cryptography, it is used to transform an adversary that breaks the security of a scheme into the an algorithm that solves a hardness assumption.

In the Q1 setting, most classical reductions are still valid. If the reduction treats the adversary as a black-box, then it does not matter if the adversary is quantum or classical. More specifically, if the reduction algorithm only interacts with the adversary through classical queries and answers, then the reduction can be fully classical. This means that as long as the hardness assumption is adapted to the Q1 setting appropriately, a classical proof will also hold in the Q1 setting. Note that this does not hold true if the reduction algorithm tries more sophisticated interactions with the adversary such as rewinding. The rewinding technique essentially corresponds to the cloning of the adversary's state. In the quantum setting, this would mean cloning a quantum state. However, the well known "no-cloning theorem" [WZ82] forbids this. As such the rewinding technique is not directly applicable in the quantum setting, and proofs that require it do not translate well. As we will not need this technique in this work, we refer the reader to [ARU14] for an in-depth discussion on this subject.

In the Q2 setting, the reduction algorithm, by definition, has to treat quantum queries. This means that its interaction with the adversary is inherently quantum. This affects some of the most used classical reduction techniques such as the "bad flag analysis" or the "lazy sampling" of a random oracle.

3.2.1. Lazy Sampling and Bad Flag Analysis

Lazy sampling is a powerful classical proof technique. It allows for the efficient implementation of random oracles. The idea is as follows. Whenever a query is submitted, the oracle checks if the value has already been queried previously. If the query is fresh, it outputs a random value. If the query is not fresh, the oracle outputs the same value as it did before.

Lazy sampling allows for many useful reduction techniques. When a reduction algorithm implements a random oracle using lazy sampling, it can learn the inputs the adversary queries and their corresponding outputs or even program the output of the oracle. Furthermore, it is often used for bad flag analysis.

The bad flag analysis is another commonly used classical proof technique. It is used to show indistinguishability between two oracles. The technique works in three steps. We define a bad event, which is typically conditioned on the queries the oracle receive, e.g the adversary queries two different inputs that cause a collision at the output. We then show that as long as the event does not happen, the oracles behave identically. Finally, we show that the event has a negligible probability of happening.

In the Q2 setting, both techniques suffer from similar problems. Let $F : \mathcal{X} \to \mathcal{Y}$ be a random function. A quantum adversary can submit a query that is a superposition of the whole message space $\mathcal{X}$. In that case, it is unclear how to do lazy sampling. Should the oracle answer by sampling $|\mathcal{X}|$ values? This solution is obviously not efficient for an exponentially sized $\mathcal{X}$. A further problem comes when the adversary makes a second query. Indeed, to assure the consistency between subsequent queries, the oracle has to "remember" the previous query/answer pairs. However, the only way to learn anything about a quantum query is to measure it. Measuring the query register can be detected by the adversary. In this case, the adversary may abort its run as it notices that it is interacting with a reduction algorithm instead of the normal protocol. Both these issues also make it difficult to know whether a problematic value has been queried triggering a bad event in the bad flag analysis.

Example. Let $RF : \mathcal{X} \to \mathcal{X}$ be a random function and let $RP : \mathcal{X} \to \mathcal{X}$ be a random permutation. A common application of both lazy sampling and bad flag analysis is the proof of the well-known Switching Lemma 1. We want to show that an adversary cannot efficiently distinguish between RF and RP. Against a classical adversary, the proof is straight forward. We implement both oracles with lazy sampling. If the randomly drawn value has already been drawn before, we set a "Bad" flag. For RF this does nothing. For RP, a new value is drawn from the pool of values that have not been drawn yet. This is depicted in Figure 3.1. Now it is easy to see that, after q queries, a collision happens with probability $\Pr[\text{Bad}] \leq \frac{q^2}{2|\mathcal{X}|}$. The claimed bound follows. A full proof of this lemma can be found in [BR04].

As discussed above, we cannot implement the oracles in this way in the quantum setting. Neither lazy sampling nor bad flag analysis is directly applicable here. Nevertheless, a generalisation of the switching lemma to the quantum setting has been proven, although it required different techniques. We give it in Section 3.3.

3.3. Other Quantum Tools

In this section we give the quantum proof tools that we will use. As mentioned above, a generalisation of the RF/RP switching lemma exists for the quantum setting. This was shown by Zhandry in [Zha13].

Theorem 3 (Quantum RF/RP Switching Lemma as in [HI20, Theorem 1])**.** *Let RF and RP denote quantum oracles of a random function from $\mathcal{X}$ to $\mathcal{X}$ and a random permutation on $\mathcal{X}$*

$RF(x)$	$RP(x)$
if $T[x] \neq \perp$	**if** $T[x] \neq \perp$
return $T[x]$	**return** $T[x]$
$y \xleftarrow{\$} \mathcal{X}$	$y \xleftarrow{\$} \mathcal{X}$
if $U[y] \neq \perp$	**if** $U[y] \neq \perp$
Bad $:= 1$	Bad $:= 1$
	$y \xleftarrow{\$} \mathcal{X} \setminus \text{keys}(U)$
$T[x] := y$	$T[x] := y$
$U[y] = x$	$U[y] = x$
return $T][x]$	**return** $T][x]$

Figure 3.1.: Lazy sampling implementation of a random function and a random permutation

respectively. Let $\mathcal{A}$ be a QPT adversary that makes at most q quantum queries. Then,

$$\mathsf{Adv}^{\text{dist}}_{RF,RP}(\mathcal{A}) \leq O\left(\frac{q^3}{|\mathcal{X}|}\right)$$

Let $H : \mathcal{X} \to \mathcal{Y}$ be a random oracle. In the classical setting, an adversary can only guess the value of $H(x)$ for a fresh value x. Therefore, its success probability is at most $\frac{1}{|\mathcal{Y}|}$. In the quantum setting, this assertion is not as clear, since a single query can "touch" the whole domain. The following theorem by Boneh and Zhandry [BZ13a, Theorem 4.1] shows that a quantum adversary does not have a significant advantage over a classical one.

Theorem 4. *Let $\mathcal{A}$ be a quantum algorithm that makes q queries to a random oracle $H : \mathcal{X} \to \mathcal{Y}$. The probability that $\mathcal{A}$ is able to produce $k > q$ distinct pairs (x_i, y_i) such that $y_i = H(x_i)$ for all $i \in [k]$ is at most*

$$\frac{1}{|\mathcal{Y}|} \sum_{r=0}^{q} \binom{k}{r} (n-1)^r$$

.

For our purposes, we are interested in the case where $|\mathcal{Y}|$ is exponentially large and $k = q + 1$. This case is shown in [BZ13a, Equation (4.1)].

Lemma 3. *Let $|\mathcal{Y}| = 2^n$ and $k = q + 1$. Then,*

$$\frac{1}{|\mathcal{Y}|} \sum_{r=0}^{q} \binom{k}{r} (n-1)^r \leq \frac{q+1}{2^n}$$

Boneh and Zhandry then use this to show that any secure qPRF is also an EUF-qCMA secure MAC. This is given by the following theorem [BZ13a, Theorem 5.1].

Theorem 5. *If $PRF : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ is a quantum-secure pseudorandom function and $1/|\mathcal{Y}|$ is negligible, then $S(k, m) = PRF(k, m)$ is an EUF-qCMA-secure MAC, and,*

$$\mathsf{Adv}_{S}^{EUF\text{-}qCMA} \leq \mathsf{Adv}_{PRF}^{qPRF} + \frac{q+1}{|\mathcal{Y}|}$$

4. A Quantum Secure Nonce-based MAC

In his master thesis [Haa20], Haas proposes a generic transformation, making any EUF-CMA secure MAC scheme into a quantum secure nonce-EUF-qCMA MAC scheme [Haa20]. In this chapter, we first give a formalized description of this transformation. We then show that there exists a pathological case where this transformation does not make the MAC scheme quantum secure. We also underline the sensitive nature of nonce based schemes in the quantum setting by applying the transformation to a qPRP. We show that if the nonce is allowed to be reused, then the transformed qPRP suffers from a superposition attack. Finally, we argue that while the transformation is not secure in general, it still works for many practical use cases. We illustrate this by applying it to the CBC-MAC and proving the nonce-EUF-qCMA security of the transformed version.

4.1. Description

Haas describes in [Haa20, Theorem 2] a transformation for any tweakable block cipher based MAC. We give here a slightly modified version of it using block ciphers. In Haas' original version, multiple different but fixed tweaks are used. We instead use different keys for the block cipher where a different tweak would have been used in the original version. In the original version, the tweaks are chosen internally, therefore, modifying the transformation to use differently keyed block ciphers is equivalent to the original scheme in terms of security. Any security proof for our version also applies to Haas' version. This is strictly a generalization, as the transformation still applies to tweakable block cipher based schemes.

Definition 4.1.1 (Haas Transform)**.** Let $\Pi = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Verify}')$ be a deterministic EUF-CMA secure MAC over $(\mathcal{K} = \{0,1\}^{\mathrm{poly}(n)}, \mathcal{M} = \{0,1\}^{l \cdot n}, \mathcal{T} = \{0,1\}^n)$. Assume that the signing algorithm is of the form,

$$\mathsf{Sign}'(k, m) = \Phi_{k,m}(E_k(\Psi(m_0)), m_1 \dots m_{l-1})$$

where m is an l-blocks message $m = m_0 \dots m_{l-1}$, E_k is a quantum-secure block cipher, Ψ is a public permutation, and $\Phi_{k,m}$ is the rest of the structure of the algorithm. We further require Ψ^{-1} to be efficiently computable.

We denote the transformed MAC by $\Pi_N = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$. The new generation algorithm Gen runs Gen′ twice to obtain $K = (k_0, k_1)$. The new signing algorithm takes a nonce $N \in \{0,1\}^n$ as an additional input. The signing algorithm is then,

$$\mathsf{Sign}(k_0, k_1, N, m) = \Phi_{k,m}(E_{k_1}(E_{k_0}(N) \oplus \Psi(m_0)), m_1 \dots m_{l-1})$$

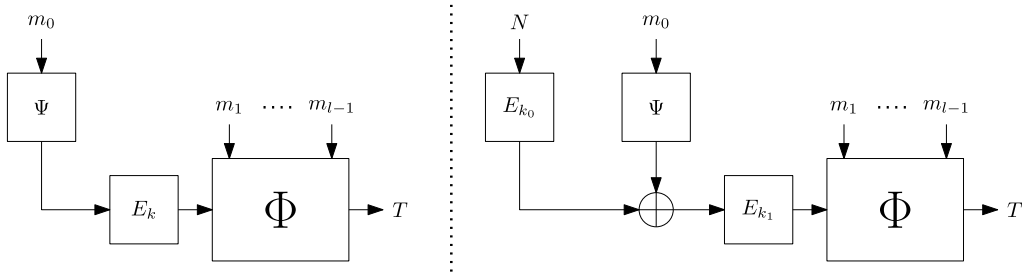


Figure 4.1.: Haas Transform

For ease of notation, we will omit to write the input to Ψ, the second input of Φ as well as its indices, and write

$$\mathrm{Sign}_N(k_0, k_1, m) = \Phi(E_{k_1}(E_{k_0}(N) \oplus \Psi))$$

As we require the signing algorithm to be deterministic, we can use the canonical verification algorithm, i.e we run the signing algorithm again and check if the submitted tag is equal to the output. The transform is depicted in Figure 4.1.

We show that the transform preserves the classical security of the transformed scheme in Appendix A.

Example. Consider the CBC-MAC for messages in $\{0,1\}^{3n}$. The signing algorithm is:

$$\mathrm{CBC}'(k, m_0 \| m_1 \| m_2) = E_k(E_k(E_k(m_0) \oplus m_1) \oplus m_2)$$

where E_k is a block cipher. In this case, Ψ is the identity function, and $\Phi(x, m_0, m_1, m_2) = E_k(E_k(x \oplus m_1) \oplus m_2)$. After the transformation, the new signing algorithm is:

$$\mathrm{CBC}_N(k_0, k_1, m_0 \| m_1 \| m_2) = E_{k_1}(E_{k_1}(E_{k_1}(m_0 \oplus E_{k_0}(N)) \oplus m_1) \oplus m_2)$$

where $K = (k_0, k_1)$

4.2. Pathological Case

Haas gave a generic transformation for any MAC without restriction on its structure. In this section, we give a pathological case, showing that this transformation does not achieve quantum security for any MAC in general.

Proposition 2. *The Haas transform given in Definition 4.1.1 does not produce nonce-EUF-qCMA secure MACs in general.*

Proof. Let $MAC' = (\mathrm{Gen}', \mathrm{Sign}'', \mathrm{Verify}')$ be a deterministic EUF-CMA secure MAC that accepts inputs of variable length. Assume further that it suffers from a key recovery attack against quantum adversaries. We then define $MAC = (\mathrm{Gen}', \mathrm{Sign}', \mathrm{Verify})$ to be a second MAC scheme with,

$$\begin{aligned}\mathsf{Sign}' : \mathcal{K} \times \{0,1\}^{ln} &\to \{0,1\}^{ln} \times \{0,1\}^n \times \{0,1\}^n \\ (k, m_0 \| m_1 \ldots \| m_{l-1}) &\mapsto ((m_0 \| m_1 \| \ldots \| m_{l-1}), \sigma_0', \sigma_1')\end{aligned}$$

where $\sigma_0' = \mathsf{Sign}''(k, m_1 \| \ldots \| m_{l-1})$ and $\sigma_1' = \mathsf{Sign}''(k, m_0 \| m_1 \| \ldots \| m_{l-1})$. Verify is the canonical verifying algorithm.

It is clear that *MAC* is also an EUF-CMA secure MAC scheme. We now apply the transformation to it. MAC_N, the transformed version of *MAC* has the following signing algorithm,

$$\begin{aligned}\mathsf{Sign}_N : \mathcal{K} \times \mathcal{K} \times \{0,1\}^{ln} &\to \{0,1\}^{ln} \times \{0,1\}^n \times \{0,1\}^n \\ (k_0, k_1, m_0 \| m_1 \ldots \| m_{l-1}) &\mapsto ((m_0 \oplus E_{k_0}(N) \| m_1 \| \ldots \| m_{l-1}), \sigma_0, \sigma_1)\end{aligned}$$

where $\sigma_0 = \mathsf{Sign}''(k_1, m_1 \| \ldots \| m_l)$ and $\sigma_1 = \mathsf{Sign}''(k_1, m_0 \oplus E_{k_0}(N) \| m_1 \| \ldots \| m_{l-1})$.

We now show that MAC_N is not nonce-EUF-qCMA secure (definition 2.3.1). Let $\mathcal{A}$ be a quantum adversary that makes q tag queries to MAC' and recovers the secret key with non-negligible probability ϵ. Then there exists the following quantum adversary $\mathcal{B}$ that wins the nonce-EUF-qCMA security game against MAC_N.

1. Use $\mathcal{A}$ to get the secret key k_1. Answer $\mathcal{A}$'s i'th query $m_i = m_{1,i} \| \ldots \| m_{l-1,i}$ by choosing a random $m_{0,i}$ and a fresh N_i, and querying $(N_i, m = m_{0,i} \| m_{1,i} \| \ldots \| m_{l-1,i})$ to MAC_N.

2. Get the first q tuples $(N, m_0 \| m_1 \| \ldots \| m_{l-1}, \mathsf{Sign}_N(k_0, k_1, m_0 \| m_1 \| \ldots \| m_{l-1})$ by measuring the q queries in the computational basis, as usual in the EUF-qCMA game.

3. Choose a pair $(m_{0,i}, N_i)$ from any of the q queries. Then choose $(m_1^* \| \ldots \| m_{l-1}^*$ such that $m* = (m_{0,i} \| m_1^* \| \ldots \| m_{l-1}^*)$ is a fresh message. Using the recovered key, we can sign $(m_1^* \| \ldots \| m_{l-1}*)$ and obtain,

   $$\sigma_0 = \mathsf{Sign}''(k_1, m_1^* \| \ldots \| m_{l-1}^*)$$

   As we used $m_{0,i}$ and N_i in a previous query, we know the value of $m_{0,i} \oplus N$. Therefore we can sign $(m_{0,i} \oplus E_{k_0}(N_i) \| m_1^* \| \ldots \| m_{l-1}^*)$ and get,

   $$\sigma_1 = \mathsf{Sign}''(k_1, m_{0,i} \oplus E_{k_0}(N_i) \| m_1^* \| \ldots \| m_{l-1}^*)$$

4. Finally submit $(N_i, m^*, \sigma_0, \sigma_1)$ as well as the q tuples from step 2.

Note that we do not violate the freshness condition of the nonces from the nonce-EUF-qCMA game. Freshness is only required when querying. A forgery is allowed to be issued for an old nonce. Therefore, $\mathcal{B}$ wins with probability at least ϵ, and it follows that MAC_N is not nonce-EUF-qCMA secure. □

4.3. Nonce-Reuse Attacks

In this section, we consider a trivial application of the transformation. We do so to illustrate an attack that can occur when nonces are allowed to be reused. Let S' be defined by,

$$S'(k, m) = E_k(m)$$

where E is a qPRP. Then $\Pi = (\text{Gen}', S', \text{Verify}')$ is an EUF-CMA secure MAC. Since E_k is assumed to be a qPRP, it follows from Theorem 3 and Theorem 5 that Π is already an EUF-qCMA secure MAC. Yet, we will show that the transformed version is not quantum-secure against nonce-abusing adversaries.

Towards this, we now apply the transformation to Π. The signing algorithm for $\Pi_N = (\text{Gen}, S, \text{Verify})$ is given by:

$$S_N(K, m) = E_{k_1}(E_{k_0}(N) \oplus m) \tag{4.1}$$

We show that this scheme is *not* nonce-reuse resistant. Indeed, we give an efficient attack against both the qPRF and the EUF-qCMA security of Π_N. Consider our scheme $\Pi_N = (\text{Gen}, S, \text{Verify})$ in a setting where the adversary is freely allowed to reuse nonces. We show that there exists a quantum algorithm $\mathcal{A}$ making q queries to the oracle, that distinguishes S from a random function. Therefore Π_N is not a qPRF. Further, we show that Π_N is not an EUF-qCMA secure MAC either. Indeed, there exists a quantum algorithm $\mathcal{B}$ that makes $q+1$ queries and is able to output $q+2$ distinct and valid message-tag pairs.

Proposition 3. *Let $\Pi_N = (Gen, S, Verify)$ be defined as described above. Then Π_N is not a qPRF. More precisely, there exists a nonce-abusing adversary $\mathcal{A}$ that makes q quantum queries and wins the qPRF game against Π with non-negligible probability.*

Proof. We define $\mathcal{A}$ to be a quantum adversary that makes q queries to an oracle. In the qPRF game (definition 2.2.3), the adversary gets access to an oracle $\mathcal{O}$. At the beginning of the interaction, it is decided by coin flip if this oracle is the signing oracle S or a random function. The algorithm then proceeds as follows.

- Chose $N, N' \in \{0,1\}^n$, two different nonces. Then let f be the following function.

  $$f(M) = \mathcal{O}(N, M) \oplus \mathcal{O}(N', M)$$

  This function can be efficiently implemented with two calls to the oracle. If the oracle implements S, then the function has a period $p = E_{k_0}(N) \oplus E_{k_0}(N')$. Indeed, for every $M \in \{0,1\}^n$ we have,

$$\begin{aligned}
&f(M \oplus p) \\
&= E_{k_1}\Big(\underbrace{E_{k_0}(N)} \oplus M \oplus \underbrace{E_{k_0}(N)} \oplus E_{k_0}(N')\Big) \oplus E_{k_1}\Big(\underbrace{E_{k_0}(N')} \oplus M \oplus E_{k_0}(N) \oplus \underbrace{E_{k_0}(N')}\Big) \\
&= E_{k_1}\Big(M \oplus E_{k_0}(N')\Big) \oplus E_{k_1}\Big(M \oplus E_{k_0}(N)\Big) \\
&= f(M)
\end{aligned}$$

- Use Simon's algorithm (theorem 2) on f. If the algorithm gives a period s then $\mathcal{A}$ returns that the oracle O is the signing oracle S. Else, $\mathcal{A}$ returns that the oracle O is a random function.

A random function has a period only with negligible probability. So if Simon's algorithm returns a period, then the distinguisher $\mathcal{A}$ is right with overwhelming probability. Theorem 2 requires that we bound the number of periods that f could have apart from p. We show that $\epsilon(f,p) = \max_{t\in\{0,1\}^n\setminus\{0,p\}} \Pr_x[f(x) = f(x\oplus t] < \frac{1}{2}$, assuming E to behave as a random permutation. Assume towards contradiction that there exists a $t \in \{0,1\}^n \setminus \{0,p\}$ such that $\Pr_x[f(x) = f(x\oplus t] \geq \frac{1}{2}$, i.e

$$\Pr_x[E_{k_1}(E_{k_0}(N)\oplus x) \oplus E_{k_1}(E_{k_0}(N')\oplus x) \oplus E_{k_1}(E_{k_0}(N)\oplus x\oplus t) \oplus E_{k_1}(E_{k_0}(N')\oplus x\oplus t)] \geq \frac{1}{2}.$$

We can then make a change of variables, replacing x with $X = x \oplus E_{k_0}(N)$. X is still distributed uniformly at random. Therefore we have,

$$\Pr_X[E_{k_1}(X) \oplus E_{k_1}(X\oplus p) \oplus E_{k_1}(X\oplus t) \oplus E_{k_1}(X\oplus p\oplus t)] \geq \frac{1}{2}.$$

It follows from Proposition 1 that this corresponds to a higher order differential for E_{k_1} with probability $\frac{1}{2}$. This only happens with negligible probability for a random permutation, which leads to a contradiction. Therefore Simon's algorithm returns p with non-negligible probability when O is the signing oracle S. It follows that $\mathcal{A}$ distinguishes between a random oracle and S with non-negligible probability and Π_N is not a qPRF when we allow nonces to be reused.

□

Proposition 4. *Let $\Pi = (Gen, S, Verify)$ be defined as described above. Then Π is not an EUF-qCMA secure MAC. More precisely, there exists a nonce-abusing adversary $\mathcal{A}$ that makes $q+1$ quantum queries and wins the EUF-qCMA game against Π with non-negligible probability.*

Proof. We define $\mathcal{B}$ to be a quantum adversary that makes $q+1$ queries to an oracle. In the EUF-qCMA game (definition 2.2.6), the adversary gets access to the signing oracle S and must output $q+2$ distinct and valid message-tag pairs. The algorithm then proceeds as follows.

- Choose two arbitrary different values $N, N' \in \{0,1\}^n$. Then, retrieve $p = E_{k_0}(N) \oplus E_{k_0}(N')$ using Simon's algorithm as described in proposition 3.
- Query $(N, M_0 \oplus p)$ to the tag oracle, with M_0 a fresh message, and receive answer τ. We have,

$$\begin{aligned}\tau = S(N, M_0\oplus p) &= E_{k_1}(E_{k_0}(N)\oplus M_0\oplus p)\\ &= E_{k_1}(E_{k_0}(N)\oplus M_0\oplus E_{k_0}(N)\oplus E_{k_0}(N'))\\ &= E_{k_1}(E_{k_0}(N')\oplus M_0)\\ &= S(N', M_0)\end{aligned}$$

- Recover the first q message-tag pairs by measuring every query made during the execution of Simon's algorithm in the first step and submit them.
- Submit $\big((N', M_0), \tau\big)$ as well as $\big((N, M_0 \oplus p), \tau\big)$ as the remaining two message-tag pairs.

We showed in Proposition 3 that Simon's algorithm revocers p with non-negligible probability. It follows that $\mathcal{B}$ is able to output $q+2$ distinct and valid message-query pairs with non-negligible probability. As such, Π_N is not EUF-qCMA secure when we allow the nonces to be reused. □

Both these attacks underline the sensitive nature of nonces for this transformation. Note that we started of with $S'(k, m) = E_k(m)$, an already secure qPRP. This already provided a secure EUF-qCMA MAC. The transformation opened it up to an avenue of attack that did not exist before.

4.4. Case Study: The Transformed CBC-MAC

Up until this point, we have only seen negative results for the Haas transform. However, we argue that for specific applications, and against nonce-respecting adversaries, the transformation does produce quantum-secure MACs. In this section, we apply it to the CBC-MAC. This scheme suffers from a superposition attack [SS16]. This seems to be a good candidate for the transform, as the sequential structure means that no attack such as the one from Section 4.2 can exist. In fact, we are able to prove the nonce-EUF-qCMA security of the transformed CBC-MAC.

Let E be a pseudo-random permutation over $\{0,1\}^n$ with key space $\mathcal{K}$. The CBC-MAC = $(\text{Gen}', CBC, \text{Verify}')$ for messages of length l is defined by,

$$\begin{aligned} CBC \colon \mathcal{K} \times \{0,1\}^{ln} &\to \{0,1\}^n \\ (k, M) &\mapsto x_l \end{aligned}$$

where $x_0 = 0, \quad x_i = E_k(x_{i-1} \oplus m_i)$ and $M = m_1 \| m_2 \| \ldots \| m_l$.

CBC-MAC is represented in Figure 1.1. It has been proven to be EUF-CMA secure in [BKR00]. Consider the transformed version of it, nonce-CBC = $(\text{Gen}, CBC_N, \text{Verify})$ with CBC_N given by,

$$\begin{aligned} CBC_N \colon \mathcal{K} \times \{0,1\}^{ln} \times \{0,1\}^n &\to \{0,1\}^n \\ (k_0, k_1, M; N) &\mapsto x_l. \end{aligned}$$

where $x_0 = E_{k_0}(N), \quad x_i = E_{k_1}(x_{i-1} \oplus m_i), \quad \text{CBC}_N(k, M) = x_l$ and $M = m_1 \| m_2 \| \ldots \| m_l$.

The transformed CBC-MAC is shown in Figure 1.2. We show that this is a nonce-EUF-qCMA secure MAC. The classical security proof for the CBC-MAC first shows the PRF security of the scheme [BKR00]. We do not take this approach as it would require some very technical proving techniques such as Zhandry's *compressed oracle* [Zha18]. Instead we directly prove the nonce-EUF-qCMA security. The crux of the proof is that the nonce-EUF-qCMA game (definition 2.3.1) requires the adversary to output $q+1$ *classical* message-tag pairs. Hence, we are able to use mostly classical arguments.

Theorem 6. *If E is a qPRP, then the transformed scheme nonce-CBC = (Gen, CBC_N, Verify) is nonce-EUF-qCMA secure MAC. More precisely, for any nonce-respecting quantum adversary $\mathcal{A}$ that makes at most q tag queries,*

$$\mathsf{Adv}^{\textit{nonce-EUF-qCMA}}_{CBC_N} \leq \frac{c \cdot l + 1}{2^n} + \mathsf{negl}(n)$$

Proof. Let $\mathcal{A}$ be a nonce-respecting quantum adversary that makes at most q queries. We consider games G_0, G_1. We write W_i to denote the event that $\mathcal{A}$ wins the game G_i.

Game G_0. This is the nonce-EUF-qCMA game (definition 2.3.1) with nonce-CBC = (Gen, CBC_N, Verify). The oracle first draws keys k_0, k_1 and then answers any query $(M = m_1 \| m_2 \ldots \| m_l, N)$ with,

$$O(M;N) = CBC_N(k_0,k_1,M;N) E_{k_1}(E_{k_1}(\ldots E_{k_1}(E_{k_1}(m_1 \oplus E_{k_0}(N)) \oplus m_2)\ldots) \oplus m_l)$$

where N must be a fresh classical value. If the adversary queries a nonce twice, then the oracle aborts the game.

Game G_1. This game proceeds as G_0, only we slightly modify the oracle. Let $(\tilde{\Pi}_N)_{N\in\{0,1\}^n}$ be a family of independently drawn random permutations indexed by $N \in \{0,1\}^n$. At the beginning of the game, we draw a random value $t \xleftarrow{\$} [l]$. We replace the call to E in the t'th round of the CBC-MAC with a call to $\tilde{\Pi}_N)_N$. More precisely, the oracle answers any query $(M = m_1 \| m_2 \| \ldots \| m_l, N)$ by performing the following steps,

- Compute the first $t-1$ rounds of the signing algorithm. We denote this with,

$$h_t(M;N) := E_{k_1}(E_{k_1}(\ldots E_{k_1}(E_{k_1}(m_1 \oplus E_{k_0}(N)) \oplus m_2)\ldots) \oplus m_{t-1})$$

- Compute $\tilde{\Pi}_N(h_t(M;N) \oplus m_t)$
- Compute the remaining steps of the signing algorithm. The output of the oracle is given by

$$O(M;N) = E_{k_1}(E_{k_1}(\ldots E_{k_1}(E_{k_1}(\tilde{\Pi}_N(h_t(M;N) \oplus m_t)) \oplus m_{t+1})\ldots) \oplus m_l)$$

$B^{|(\tilde{\Pi}_N)_N\rangle}$

$t \xleftarrow{\$} [l]$

$k_0, k_1 \xleftarrow{\$} \mathcal{K}$

$S \leftarrow \mathcal{A}^{CBC_t^{(\tilde{\Pi}_N)_N}}$

$CBC_t^{(\tilde{\Pi}_N)_N}(M;N)$

$x \coloneqq E_{k_0}(N)$

for $i = 1$ **to** $t-1$:

  $x \coloneqq E_{k_1}(x \oplus m_i)$

$x \coloneqq \tilde{\Pi}_N(x \oplus m_t)$

for $i = t+1$ **to** l :

  $x \coloneqq E_{k_1}(x \oplus m_i)$

return x

Figure 4.2.: Simulator $\mathcal{B}$

We denote this oracle by $CBC_t^{(\tilde{\Pi}_N)_N}$. It follows from Lemma 4 that,

$$|\Pr[W_0] - \Pr[W_1]| \leq \mathsf{negl}(n) \tag{4.2}$$

We now bound $\Pr[W_1]$. Let $\mathcal{B}$ be a quantum algorithm that has oracle access to $(\tilde{\Pi}_N)_{N\in\{0,1\}^n}$. Algorithm $\mathcal{B}$ simulates game G_1 for $\mathcal{A}$. It does so by drawing keys k_0 and k_1. It then can perfectly answer the adversary's query by computing the calls to E and using its $(\tilde{\Pi}_N)_{N\in\{0,1\}^n}$ oracle. This is shown in figure 4.2. We define

$$\mathcal{S} = \{(N^{(1)}, M^{(1)}, T^{(1)}), \ldots, (N^{(q)}, M^{(q)}, T^{(q)}), (N^{(q+1)}, M^{(q+1)}, T^{(q+1)})\}$$

to be the set containing the $q+1$ classical nonce-message-tag tuples $\mathcal{A}$ outputs as forgeries when playing in game G_1. Let further $\mathcal{I} = \{N_i' | 1 \leq i \leq q\}$ denote the set of all nonces used during the q queries. Notice that both set are completely classical. We define the following events.

- Event_A: There exists an i such that $N^{(i)} \notin \mathcal{I}$.
- Event_B There exists $i \neq j$ such that $N^{(i)} = N^{(j)}$.

Notice that these events partition all possible valid sets S. Indeed, all tuples in S are distinct, and $|S| = q+1$ but $|\mathcal{I}| = q$. It follows that,

$$\Pr[W_2] \leq \Pr[\text{Event}_A] + \Pr[\text{Event}_B] \tag{4.3}$$

Consider a tuple $(N^{(i)}, M^{(i)}, T^{(i)}) \in S$. If the set is valid, $\mathcal{B}$ can efficiently compute $h_t(M^{(i)}; N^{(i)})$ using only E_{k_0} and E_{k_1}. It can also efficiently compute $\tilde{\Pi}_{N^{(i)}}(h_t(M^{(i)}; N^{(i)}) \oplus m_t^{(i)})$ without using its $\tilde{\Pi}_N$ oracle, by using $E_{k_1}^{-1}$ on $T^{(i)}$. Using these facts, we now bound each event.

We first bound the probability that Event_A happens. Assume that $\mathcal{A}$ outputs a tuple $(N^{(i)}, M^{(i)}, T^{(i)})$ such that $N^{(i)} \notin \mathcal{I}$. It follows from the above discussion that $\mathcal{B}$ can use the tuple to compute the pair,

$$\left(X = h_t(M^{(i)}; N^{(i)}), \tilde{\Pi}_{N^{(i)}}(X)\right)$$

without any query to the permutation $\tilde{\Pi}_{N^{(i)}}$. Notice that the permutation $\tilde{\Pi}_{N^{(i)}}$ has not been queried to compute the tuple either, as $N^{(i)} \notin \mathcal{I}$. Since $\tilde{\Pi}_{N^{(i)}}$ is an independently drawn random permutation, this happens with probability at most,

$$\Pr[\text{Event}_A] \leq \frac{1}{2^n} \tag{4.4}$$

We now bound the probability that Event_B happens. Assume that $\mathcal{A}$ outputs two distinct tuples $(N^{(i)}, M^{(i)}, T^{(i)})$ and $(N^{(j)}, M^{(j)}, T^{(j)})$ such that $N^{(i)} = N^{(j)}$. Since the tuples are distinct, there exists a u such that,

$$h_u(M^{(i)}; N^{(i)}) \oplus m_u^{(i)} \neq h_u(M^{(j)}; N^{(j)}) \oplus m_u^{(j)}$$

.

With probability $1/l$, we guessed u when drawing t and $t = u$. This means that $\mathcal{B}$ can build two distinct pairs

$$\left(X_0 = \left(h_t(M^{(i)}; N^{(i)}) \oplus m_t^{(i)}\right), \tilde{\Pi}_{N^{(i)}}(X_0)\right) \quad \text{and} \quad \left(X_1 = \left(h_t(M^{(j)}; N^{(i)}) \oplus m_t^{(j)}\right), \tilde{\Pi}_{N^{(i)}}(X_1)\right)$$

without any additional $\tilde{\Pi}_N^{(i)}$ query. Remember that $\tilde{\Pi}_{N^{(i)}}$ has been queried at most once, since it is indexed by the nonce. Since it is a random permutation, it follows from lemma 3 and from Theorem 3 that,

$$\Pr[\text{Event}_B] \leq \frac{l \cdot c}{2^n} \tag{4.5}$$

where c is a constant from Theorem 3 and l is the loss that comes from guessing u. We can now use Equation (4.2), Equation (4.3), Equation (4.4) and Equation (4.5) to derive the claimed bound.

$$\begin{aligned}
\text{Adv}^{\text{nonce-EUF-qCMA}}_{CBC_N} &\leq |\Pr[W_0] - \Pr[W_1]| + \Pr[W_1] && (4.6)\\
&\leq \text{negl}(n) + \Pr[\text{Event}_A] + \Pr[\text{Event}_B] && (4.7)\\
&\leq \frac{l \cdot c + 1}{2^n} + \text{negl}(n) && (4.8)
\end{aligned}$$

□

In the theorem, we used the fact that we can swap a call to E within the structure of the CBC-MAC with a family of random permutation indexed by the nonce.

Lemma 4. *Using the same notation as in the proof of Theorem 6, for any quantum adversary $\mathcal{A}$ that makes at most q queries, G_0 and G_1 are indistinguishable. More precisely,*

$$\Pr[W_0] - \Pr[W_1]| \leq \text{negl}(n)$$

This result is used without proof in [AR17, Theorem 4]. We will not show it here either. We leave this for a future work.

5. Heuristics for Designing Quantum Secure Protocols

In this chapter, we compare the architecture of protocols that suffer from quantum attacks with their quantum-secure counterparts. By doing so, we identify secure design strategies and pitfalls. We synthesize this into heuristics for the conception of quantum secure protocols. We focus our approach on the use of nonces in block cipher based schemes. Towards this, we first review some unsecure construction in Section 5.1, to expose the structural flaws of the algorithm. We then look at protocols that were designed to be their quantum-secure replacements in Section 5.2.

5.1. Unsecure Designs

In this section, we first take a look at some unsecure constructions. We divide these into deterministic protocols and nonce protocols. Note that the nonce protocols are in fact also deterministic, but they use nonces to introduce a part of pseudorandomness to each query. We analyze the published attacks against these schemes, and formalize the flaws they exploit.

5.1.1. Deterministic Protocols

We first consider the CBC-MAC. This MAC has many variants, as the original scheme is only secure for prefix-free messages. We work on the fixed-length version, which has been proven classically secure in [BKR00]. However, similar attacks exists for other variants [Kap+16].

The CBC-MAC is a MAC for messages of fixed length $l \cdot n$, where n is the input size of the underlying block cipher E. We give its definition again here. For any message message $M = m_1 \| m_2 \| \dots \| m_l$, the signing algorithm is given by,

$$x_0 = 0, \quad x_i = E_k(x_{i-1} \oplus m_i), \quad \text{CBC-MAC}(k, M) = x_l$$

The CBC-MAC signing algorithm is represented in Figure 1.1. Santoli and Schaffner have found the following attack [SS16] against the EUF-qCMA security of the scheme. Choose two distinct values $\alpha_0, \alpha_1 \in \{0,1\}^n$ and some integer j such that $1 \leq j \leq l-1$. Consider the following function,

$$f(b\|x) = \text{CBC-MAC}(\alpha_b \| 0^{(j-1)\dot{n}} \| x \| 0^{(l-j-1)\cdot n}) = E_k^{(l-j)}(E_k^j(\alpha_b) \oplus x)$$

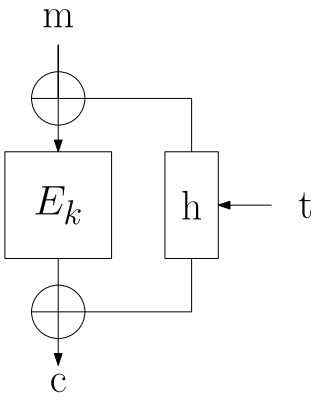


Figure 5.1.: The LRW Tweakabl Block Cipher

where CBC-MAC(M) is the oracle provided in the EUF-qCMA security game. We can construct f with two calls to the oracle. Notice that $f(b\|x) = f(b'\|y)$ if and only if $x = y \oplus 1\|\left(E_k^j(\alpha_0) \oplus E_k^j(\alpha_1)\right)$.

It follows that we can recover $\left(E_k^j(\alpha_0) \oplus E_k^j(\alpha_1)\right)$ in polynomial time using Simon's algorithm. This is then enough to produce forgeries. We have,

$$E_k^{(l-j)}\Big(E_k^j(\alpha_0) \oplus x\Big) = E_k^{(l-j)}\Big(E_k^j(\alpha_0) \oplus x \oplus \left(E_k^j(\alpha_0) \oplus E_k^j(\alpha_1)\right)\Big)$$

In their paper, Santoli and Schaffner extend this attack further, but we will not cover this, since this suffices to illustrate our point. CBC-MAC is a sequential construction, that uses a "Horner-type" structure for efficiency. This means that every round of the protocol has the form,

$$E_k\Big(E_k(\cdot) \oplus m_i\Big) \tag{5.1}$$

We observe a similar structure when considering the LRW construction. This is a tweakable block cipher construction, introduced by Liskov, Rivest and Wagner [LRW02]. Let h be an almost-universal hash function. Then the tweakable block cipher is given by

$$\tilde{E}_k^t(m) = \underbrace{E_k\Big(m \oplus h(t)\Big)} \oplus h(t) \tag{5.2}$$

The LRW algorithm is represented in Figure 5.1. We recognise that the underlined part has a similar structure to the one described in Equation (5.1). This leads to a very similar attack as on the CBC-MAC found by Kaplan et al. [Kap+16]. Choose two distinct tweaks t, t'. Consider the function f given by,

$$f(x) = \tilde{E}_k^t(x) \oplus \tilde{E}_k^{t'}(x) \tag{5.3}$$

$$= E_k\Big(x \oplus h(t)\Big) \oplus h(t) \oplus E_k\Big(x \oplus h(t')\Big) \oplus h(t') \tag{5.4}$$

This function can be built with two calls to $\tilde{E}_k^t$. Notice that it admits a period $p = h(t) \oplus h(t')$. This allows to build an efficient distinguisher against the tweakable block cipher security of the LRW construction.

We have identified a design flaw. For efficiency reasons, the above problematic structure is commonly used. Going forward, we will call a structure of the following form a "Horner-structure:

$$E_k\Big(\gamma \oplus x\Big) \tag{5.5}$$

where γ is any value and x can be queried in quantum superposition. The crux of both of the previous attacks is that γ can be fixed to be two different values γ_0, γ_1. This then allows to build a function that has a period $p = \gamma_0 \oplus \gamma_1$. This is then used to continue the attack. Notice that γ may be a classical value.

5.1.2. Nonce Protocols

Consider the OCB3 protocol published by Rogaway [KR11]. It is an authenticated encryption scheme closely related to the LRW construction. Let F_k be a qPRF. Let Δ_i be some value that depends on i and define $\Delta_i^N = \Delta_i \oplus F_k(N)$. We do not detail the generation of Δ_i further as it is not relevant to the attack. For any message $M = m_1 \| m_2 \| \dots \| m_l$ with associated data $A = a_1 \| a_2 \| \dots \| a_{l'}$, the protocol outputs a ciphertext $C = c_1 \| c_2 \| \dots \| c_l$ and a tag T such that,

$$c_i = E_k(m_i \oplus \Delta_i^N), \quad T = \bigoplus_i E_k(a_i \oplus \Delta_i) \oplus E_k\left(\bigoplus_i m_i \oplus \Delta_{l+1}^N\right) \tag{5.6}$$

OCB3's structure is represented in Figure 5.2. Bhaumik et al. describe two attacks against the scheme [Bha+20]. The first one focuses on the authentication part. Let f_N be a function that queries an empty message and two variable identical associated data blocks with a nonce N to the $OCB3_N$ oracle and outputs the tag. We have,

$$f_N(x) = E_k(\Delta_{l+1}^N) \oplus E_k(x \oplus \Delta_1) \oplus E_k(x \oplus \Delta_2)$$

This function has a period $p = \Delta_1 \oplus \Delta_2$. The issue is that Simon's algorithm requires multiple queries to the *same* f_N to recover the period. Since N must always be fresh for every query, this is not possible. However, notice that the period does not depend on the nonce. This means that a single round of Simon's algorithm, which requires only one query to f_N, returns a vector orthogonal to p for any N. Hence, we can run Simon's algorithm and let it query a different f_N in each round. This allows us to recover the period p in polynomial time. Bhaumik et al. then use the period to produce forgeries.

Looking back at the structure of the authentication part of OCB3 described in Equation (5.6), we recognise multiple instances of the problematic Horner-structure we identified in Equation (5.5).

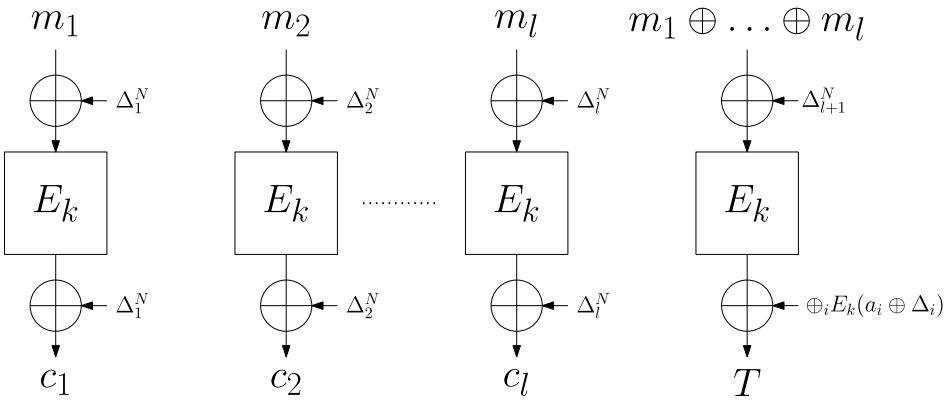


Figure 5.2.: The OCB3 Authenticated Encryption

$$T = \underbrace{\bigoplus_i E_k(a_i \oplus \Delta_i)}_{\mathbf{A}} \oplus \underbrace{E_k\left(\bigoplus_i m_i \oplus \Delta_{l+1}^N\right)}_{\mathbf{B}} \tag{5.7}$$

The instance from part **B** contains the nonce. In the notation from Equation (5.5), $\gamma = \Delta_{l+1}^N$ cannot be fixed as it depends on the nonce. Hence, an attack on this part of the authentication scheme seems unlikely. The period would depend on the nonce. On the other hand, notice that the instances in part **A** do not depend on the nonce. This is the weakness the attack exploits. Indeed, while part **B** cannot be directly attacked, it does not modify the period of part **A**. It essentially serves as a random permutation on part **A**, and a permutation of a periodic function does not modify the period. We identify a second design flaw. The period of a function cannot be broken from outside of the structure of the function.

We now consider the attack on the encryption part of OCB3 given by Bhaumik et al. Let g_N be a function that queries the same block twice in a single message for a nonce N and XORs the corresponding encryption block, i.e

$$g_N(x) = E_k(x \oplus \Delta_i^N) \oplus E_k(x \oplus \Delta_j^N)$$

for some $1 \leq i, j \leq l$. Remember that $\Delta_i^N = \Delta_i \oplus F_k(N)$. It follows that,

$$g_N(x) = E_k(x \oplus \Delta_i \oplus F_k(N)) \oplus E_k(x \oplus \Delta_j \oplus F_k(N))$$

The function admits a period $p = \Delta_i \oplus \Delta_j$. Again, the period does not depend on the nonce. It follows that Simon's algorithm can recover the period and the attack continues from there.

Looking at the structure of the encryption part of OCB3, each message block is individually encrypted using a Horner-structure,

$$c_i = E_k(m_i \oplus \Delta_i^N) \tag{5.8}$$
$$= E_k(m_i \oplus \underbrace{\Delta_i}_{\gamma_i} \oplus F_k(N)) \tag{5.9}$$

However, setting $\gamma_i = \Delta_i^N$ does not allow to build a periodic function, since $F_k(N)$ cannot be fixed. Instead, we can set $\gamma_i = \Delta_i$. In that case, $F_k(N)$ can simply be seen as a random permutation on $m_i \oplus \Delta_i$. By itself, this is not a periodic function yet. We cannot use the same trick as in the attack one the CBC-MAC, that allows to build a periodic function out of one Horner-structure. Indeed, this requires the adversary to be able to query γ in quantum superposition. Now, notice that the parallel-encryption structure of the OCB3 protocol essentially means, that one query of OCB3 gives you l instances of Horner-structures with the same nonce. This is what allows the attack to go through.

5.1.3. Identified Design Flaws

The Horner-structure. All the attacks we have seen rely on the ability to build a periodic function out of the oracle. This then allows them to recover sensitive information by using Simon's algorithm. We have identified a common structure within the protocols, that facilitates the construction of a periodic function. Any part of the algorithm with the following form is a potential weak point of the protocol,

$$E_k\Big(\gamma \oplus x\Big)$$

We have seen two different strategies that allow to build a periodic function from a Horner-structure.

1. If γ if a function of the form $\gamma(y)$, where y can be queried in quantum superposition, the adversary can choose two distinct y_0, y_1 and build a function of the form

$$f(b\|x) = E_k\Big(\gamma_(y_b) \oplus x\Big) \tag{5.10}$$

   where $\gamma_b = \gamma(y_b)$. This function has a period $p = 1\|(\gamma(y_0) \oplus \gamma(y_1))$, and can be built from a single oracle call.

2. The second strategy requires two oracle calls to build the function, but has looser requirements on γ. The adversary builds a function of the form,

$$f(x) = E_k\Big(\gamma \oplus x\Big) \oplus E_k\Big(\gamma' \oplus x\Big) \tag{5.11}$$

   where γ and γ' are two distinct values. This function has a period $p = \gamma \oplus \gamma'$. In this strategy, γ may be a classical value. We have seen in the case of LRW that the adversary uses two oracle queries to build the function. However, in the case of OCB, a single query already provides the two distinct Horner-structures. Notice that in that case, γ does not even depend on the input of the adversary.

Nonces Outside the Periodic Function Using a nonce to break up the periodicity of a function cannot be done from outside that function. Let f be a function that admits a period p. Let g be any deterministic function such that the oracle O given to the adversary is of the form

$$O(x_0, x_1; N) = g(f(x_0), x_1, N)$$

For a fixed x_1 and N, the oracle O also admits a period p. Now the input x_1 can be fixed, but the nonce, by definition, cannot. However, as seen in the attack on the authentication part of OCB3, Simon's algorithm can still recover p by querying a different function h_{N,x_1} in each round of the algorithm, with,

$$h_{N,x_1}(x) = g(f(x), x_1, N)$$

where N gets randomly drawn for each h_{N,x_1}. The caveat being if g produces more periods. In order for Simon's algorithm to return p with high probability, Theorem 2 states that the amount of parasitic periods must be bounded, i.e

$$\epsilon(h_{N,x_1}, p) \coloneqq \max_{t \in \{0,1\}^n \setminus \{0,p\}} \Pr_x[f(x) = f(x \oplus t)] \leq p_0 \tag{5.12}$$

for some p_0 specified in the theorem.

Parallelization. Protocols may use the nonce to break the period of a part of the scheme, e.g between Horner-structures. However, parallelization can lead to a situation where message blocks are being treated individually with the same nonce. As we have seen in the attack on the encryption part of the OCB3 protocol, this can allow an adversary to build a periodic function in spite of its dependence on a nonce. Parallel designs can facilitate the second type of attack on Horner-structures described in Equation (5.11).

5.2. Secure Designs

In this section, we review protocols that have been proven secure in the quantum setting. We scrutinise these schemes for the structural weaknesses we identified in the last section and identify the design features used to avoid the associated attacks.

5.2.1. The LRWQ

We start with the LRWQ construction. This is a tweakable block cipher designed by Hosoyamada and Iwata [HI20]. It is meant to be a quantum secure replacement for the LRW construction. Let k_0, k_1, k_2 be three independently drawn keys for the underlying block cipher E. Then the LRWQ tweakable block cipher is given by

$$\mathrm{LRWQ}^t_{k_1,k_2,k_3}(m) = E_{k_3}\big(E_{k_1}(m) \oplus E_{k_2}(t)\big)$$

The structure of the LRWQ tweakable block cipher is shown in Figure 5.3. Hosoyamada and Iwata have shown this construction to be a qPRP. When we compare the structure of

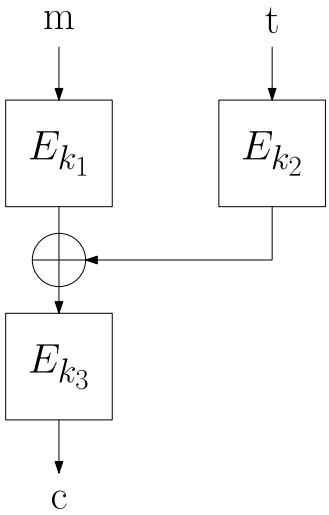


Figure 5.3.: The LRWQ Tweakable Block Cipher

the protocol to the structure of LRW from Equation (5.2), we note two corrections to its weaknesses.

The outer use of $h(t)$ has been removed. As discussed in Section 5.1.3, this does cannot break the periodicity of a function, and therefore is not needed for the LRWQ protocol.

More importantly, the periodic nature of the Horner-structure has been broken. This is done by encrypting the message block m with E_{k_1}. This "insulates" the variable part from the fixed γ part of the Horner-structure.

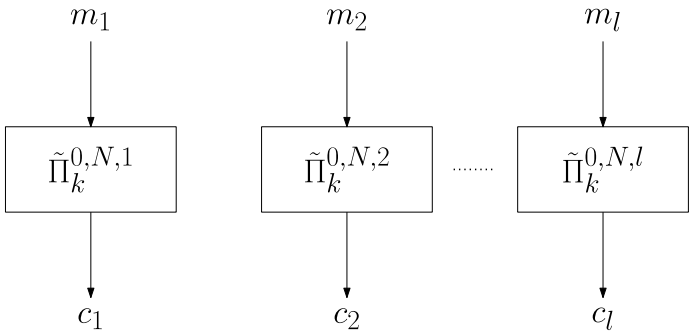
5.2.2. The Transformed CBC-MAC

We now look back at the CBC-MAC. This is a sequential design that has as a Horner-structure at every level. This is a core feature of the design as it allows the protocol to be very efficient. The protocol uses l calls to the block cipher E for an l-block message. Breaking the periodicity with the same means as for LRWQ would double that, as it would add $l - 1$ uses of E for every query. The transformed version from Section 4.4 instead uses nonces. We restate its definition. For any message message $M = m_1 \| m_2 \| \dots \| m_l$, the signing algorithm is given by,

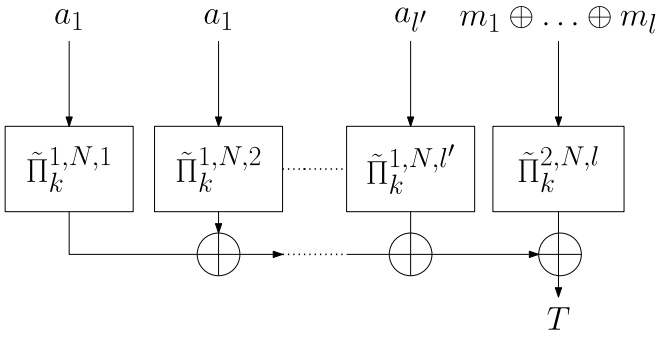
$$x_0 = E_{k_0}(N), \quad x_i = E_{k_1}(x_{i-1} \oplus m_i), \quad \mathrm{CBC}_N(k, M) = x_l$$

The transformed CBC-MAC is represented in Figure 1.2. We proved this design EUF-qCMA secure against nonce-respecting adversaries in Section 4.4. The transformation efficiently makes the CBC-MAC quantum secure, as it only requires a single additional use of the block cipher E. This synergises well with the sequential nature of the algorithm. Indeed, every Horner-structure x_i depends on all the Horner-structures x_j, with $j < i$, from the previous rounds. Consider the first Horner-structure x_1,

$$x_1 = E_{k_1}(E_{k_0}(N) \oplus m_1) \tag{5.13}$$



(a) Encryption



(b) Authentication

Figure 5.4.: QCB Authenticated Encryption

In the notation from equation (5.5), $\gamma = E_{k_0}(N)$ cannot be fixed. Hence, this level is not suitable for an attack. Additionally, x_1 depends on N and can not be fixed either, as the nonce N is encrypted. This means that the second Horner-structure x_2 is not suitable for an attack either. This propagates to every level making the transformed CBC-MAC quantum secure.

5.2.3. The QBC Authenticated Encryption

The QCB scheme is an authenticated encryption protocol designed by Bhaumik et al. [Bha+20]. It is meant to be a quantum-secure replacement for the OCB protocol. We give a slightly simplified version of the scheme to illustrate our point. It is based on a quantum-secure tweakable block cipher that accepts tweaks of the form (d, N, i), where $d \in \{0, 1, 2\}$ is a domain separator, N is a nonce, and $i \in [l]$ is the block index. Let $\widetilde{\Pi}_k^{(d,N,i)}$ be a quantum-secure tweakable block cipher as described above. Then for any message $M = m_1 \| m_2 \| \ldots \| m_l$ with associated data $A = a_1 \| a_2 \| \ldots \| a_{l'}$, the QCB algorithm outputs a ciphertext $C = c_1 \| c_2 \| \ldots \| c_l$ and a tag T such that,

$$c_i = \widetilde{\Pi}_k^{(0,N,i)}(m_i), \quad T = \bigoplus_i \widetilde{\Pi}_k^{(1,N,i)}(a_i) \oplus \widetilde{\Pi}_k^{(2,N,l+1)}(m_1 \oplus \cdots \oplus m_l) \tag{5.14}$$

We represent the QCB protocol in Figure 5.4. The structure of the QCB protocol is essentially identical to the structure of the OCB protocol. The scheme encrypts each message block/associated data block with the tweakable block cipher $\widetilde{\Pi}$ using the same

nonce. This directly patches the weakness of the OCB protocol. Indeed, every input from the adversary is individually "insulated" in the quantum-secure tweakable block cipher. The quantum attack is avoided by making the structure of QCB denser, similarly to the LRWQ scheme. The nonce and block index simply serve against the classical attacks as in the OCB protocol.

5.2.4. Secure Design Strategies

Dense Structure. A dense structure breaks the periodicity of the Horner-structures. This can be accomplished by encrypting the adversary's quantum inputs individually.

When applied to the LRWQ tweakable block cipher, this strategy has a big efficiency drawback. Indeed, since both the message and the tweak can to be queried in quantum superposition, both need to be individually encrypted. On the other hand, Bhaumik et al. are able to efficiently apply this strategy with QCB. They show in [Bha+20, Section 4.3] how to instantiate the protocol with a tweakable block cipher that only requires one call to the underlying block cipher E.

Nonce in the Horner-Structure. A second strategy consist in using nonces to break the periodicity of the Horner-structures. The protocol must be built in such a way that every Horner-structure of the protocol has the following form,

$$E_k\Big(\gamma_N \oplus x\Big)$$

This strategy is particularly effective with sequential designs. We have seen an example of its efficiency with the transformed CBC-MAC. However, as discussed in Section 5.1.3, this strategy may be ill suited to parallel designs.

6. Conclusion

In this thesis, we have studied the use of nonces in the design of quantum-secure protocols. Towards this, we reviewed the Haas transform [Haa20]. We have shown that it is not quantum-secure in general. Still, we applied it to the CBC-MAC scheme. We have shown that the transformed CBC-MAC is EUF-qCMA secure against nonce-respecting adversaries. This demonstrates that while the Haas transform may not be a secure generic transformation, it is a relevant design strategy against quantum attacks.

In order to build an array of such design strategies, we have surveyed existing quantum attacks. From this, we have identified parallel-encryption and Horner-structures to be structural weak points that a quantum attack may exploit. Parallel-encryption and Horner-structures are both commonly employed in algorithms for efficiency reasons. By comparing attacked protocols with their proven quantum-secure counterparts, we have extracted the design strategies that frustrate the previously mentioned attacks.

Going forward, the structural weaknesses and design strategies identified in this paper may serve as heuristic while building quantum-secure protocols. Furthermore, they may also help in cryptanalysing protocols in the Q2 setting.

Our security proof for the transformed CBC-MAC directly shows the EUF-qCMA security of the scheme. The security bound given is not tight. A future work may improve this bound. An interesting approach may be to use Zhandry's *compressed oracle technique* [Zha18] or one of its variant such as the *recording standard oracle with error* [HI19]. This could allow to prove the qPRF security of the scheme.

Bibliography

[Ala+20] Gorjan Alagic et al. “Quantum-Access-Secure Message Authentication via Blind-Unforgeability”. In: *Advances in Cryptology – EUROCRYPT 2020*. Springer International Publishing, 2020, pp. 788–817. DOI: 10.1007/978-3-030-45727-3_27. URL: https://doi.org/10.1007%2F978-3-030-45727-3_27.

[AR17] Gorjan Alagic and Alexander Russell. “Quantum-Secure Symmetric-Key Cryptography Based on Hidden Shifts”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2017, pp. 65–93. DOI: 10.1007/978-3-319-56617-7_3. URL: https://doi.org/10.1007%2F978-3-319-56617-7_3.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. *Quantum Attacks on Classical Proof Systems - The Hardness of Quantum Rewinding*. Cryptology ePrint Archive, Report 2014/296. https://ia.cr/2014/296. 2014.

[Bha+20] Ritam Bhaumik et al. *QCB: Efficient Quantum-secure Authenticated Encryption*. Cryptology ePrint Archive, Report 2020/1304. https://ia.cr/2020/1304. 2020.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. “The Security of the Cipher Block Chaining Message Authentication Code”. In: *Journal of Computer and System Sciences* 61.3 (2000), pp. 362–399. ISSN: 0022-0000. DOI: https://doi.org/10.1006/jcss.1999.1694. URL: https://www.sciencedirect.com/science/article/pii/S002200009991694X.

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg. “On the inherent intractability of certain coding problems (Corresp.)” In: *IEEE Transactions on Information Theory* 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[BN18] Xavier Bonnetain and María Naya-Plasencia. “Hidden Shift Quantum Cryptanalysis and Implications”. In: *Advances in Cryptology – ASIACRYPT 2018*. Vol. 11272. Lecture Notes in Computer Science. Springer, 2018, pp. 560–592. DOI: 10.1007/978-3-030-03326-2_19.

[Bon19] Xavier Bonnetain. “Hidden Structures and Quantum Cryptanalysis”. Theses. Sorbonne Université, Nov. 2019. URL: https://tel.archives-ouvertes.fr/tel-02400328.

[BR04] Mihir Bellare and Phillip Rogaway. *Code-Based Game-Playing Proofs and the Security of Triple Encryption*. mihir@cs.ucsd.edu 13498 received 30 Nov 2004, last revised 16 Dec 2006. 2004. URL: http://eprint.iacr.org/2004/331.

[BZ13a] Dan Boneh and Mark Zhandry. "Quantum-Secure Message Authentication Codes". In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 592–608. ISBN: 978-3-642-38348-9.

[BZ13b] Dan Boneh and Mark Zhandry. "Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World". In: *CRYPTO*. Springer, 2013, pp. 361–379. DOI: 10.1007/978-3-642-40084-1_21. URL: https://www.iacr.org/archive/crypto2013/80420154/80420154.pdf.

[Cza+19] Jan Czajkowski et al. *Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability*. 2019. DOI: 10.48550/ARXIV.1904.11477. URL: https://arxiv.org/abs/1904.11477.

[DH06] W. Diffie and M. Hellman. "New Directions in Cryptography". In: *IEEE Trans. Inf. Theor.* 22.6 (Sept. 2006), pp. 644–654. ISSN: 0018-9448. DOI: 10.1109/TIT.1976.1055638. URL: https://doi.org/10.1109/TIT.1976.1055638.

[Gag17] Tommaso Gagliardoni. "Quantum Security of Cryptographic Primitives". In: *CoRR* abs/1705.02417 (2017). arXiv: 1705.02417. URL: http://arxiv.org/abs/1705.02417.

[GHS16] Tommaso Gagliardoni, Andreas Hülsing, and Christian Schaffner. "Semantic Security and Indistinguishability in the Quantum World". In: *Lecture Notes in Computer Science* (2016), pp. 60–89. ISSN: 1611-3349. DOI: 10.1007/978-3-662-53015-3_3. URL: http://dx.doi.org/10.1007/978-3-662-53015-3_3.

[Gro96] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. STOC '96. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1996, pp. 212–219. ISBN: 0897917855. DOI: 10.1145/237814.237866. URL: https://doi.org/10.1145/237814.237866.

[GYZ17] Sumegha Garg, Henry Yuen, and Mark Zhandry. *New security notions and feasibility results for authentication of quantum data*. Cryptology ePrint Archive, Report 2017/538. https://ia.cr/2017/538. 2017.

[Haa20] Jonas Haas. "Superposition Attacks on Lightweight Message Authentication". MA thesis. Karlsruher Institut für Technologie, 2020.

[HI19] Akinori Hosoyamada and Tetsu Iwata. *4-Round Luby-Rackoff Construction is a qPRP: Tight Quantum Security Bound*. Cryptology ePrint Archive, Report 2019/243. https://ia.cr/2019/243. 2019.

[HI20] Akinori Hosoyamada and Tetsu Iwata. *Provably Quantum-Secure Tweakable Block Ciphers*. Cryptology ePrint Archive, Report 2020/1321. https://ia.cr/2020/1321. 2020.

[Kap+16] Marc Kaplan et al. *Breaking Symmetric Cryptosystems using Quantum Period Finding*. 2016. arXiv: 1602.05973 [quant-ph].

[KM10] Hidenori Kuwakado and Masakatu Morii. “Quantum distinguisher between the 3-round Feistel cipher and the random permutation”. In: *2010 IEEE International Symposium on Information Theory*. 2010, pp. 2682–2685. DOI: 10.1109/ISIT.2010.5513654.

[KM12] Hidenori Kuwakado and Masakatu Morii. “Security on the quantum-type Even-Mansour cipher”. In: *2012 International Symposium on Information Theory and its Applications*. 2012, pp. 312–316.

[KR11] Ted Krovetz and Phillip Rogaway. “The Software Performance of Authenticated-Encryption Modes”. In: May 2011, pp. 306–327. ISBN: 978-3-642-21701-2. DOI: 10.1007/978-3-642-21702-9_18.

[Lai94] Xuejia Lai. “Higher Order Derivatives and Differential Cryptanalysis”. In: *Communications and Cryptography: Two Sides of One Tapestry*. Ed. by Richard E. Blahut et al. Boston, MA: Springer US, 1994, pp. 227–233. ISBN: 978-1-4615-2694-0. DOI: 10.1007/978-1-4615-2694-0_23. URL: https://doi.org/10.1007/978-1-4615-2694-0_23.

[LRW02] Moses D. Liskov, Ronald L. Rivest, and David A. Wagner. “Tweakable Block Ciphers”. In: *CRYPTO*. 2002.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Reg05] Oded Regev. “On Lattices, Learning with Errors, Random Linear Codes, and Cryptography”. In: *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*. STOC ’05. Baltimore, MD, USA: Association for Computing Machinery, 2005, pp. 84–93. ISBN: 1581139608. DOI: 10.1145/1060590.1060603. URL: https://doi.org/10.1145/1060590.1060603.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://doi.org/10.1145/359340.359342.

[Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. ISSN: 1095-7111. DOI: 10.1137/s0097539795293172. URL: http://dx.doi.org/10.1137/S0097539795293172.

[Sim94] D. R. Simon. “On the Power of Quantum Computation”. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. SFCS ’94. USA: IEEE Computer Society, 1994, pp. 116–123. ISBN: 0818665807. DOI: 10.1109/SFCS.1994.365701. URL: https://doi.org/10.1109/SFCS.1994.365701.

[SS16] Thomas Santoli and Christian Schaffner. “Using Simon’s Algorithm to Attack Symmetric-Key Cryptographic Primitives”. In: (2016). DOI: 10.48550/ARXIV.1603.07856. URL: https://arxiv.org/abs/1603.07856.

[Unr15] Dominique Unruh. "Revocable Quantum Timed-Release Encryption". In: *J. ACM* 62.6 (Dec. 2015). ISSN: 0004-5411. DOI: 10.1145/2817206. URL: https://doi.org/10.1145/2817206.

[WZ82] William K. Wootters and Wojciech Zurek. "A single quantum cannot be cloned". In: *Nature* 299 (1982), pp. 802–803.

[Zha13] Mark Zhandry. "A Note on the Quantum Collision and Set Equality Problems". In: *CoRR* abs/1312.1027 (2013). arXiv: 1312.1027. URL: http://arxiv.org/abs/1312.1027.

[Zha18] Mark Zhandry. *How to Record Quantum Queries, and Applications to Quantum Indifferentiability*. Cryptology ePrint Archive, Report 2018/276. https://ia.cr/2018/276. 2018.

A. Appendix

In this section, we show that the transformation preserves the classical security of the transformed scheme.

Lemma 5. *Let Π be a deterministic EUF-CMA secure MAC based on a block cipher $E_k : \{0,1\}^n \to \{0,1\}^n$ as described in definition 4.1.1. Then Π_N is a nonce-EUF-CMA secure MAC.*

In particular, for any adversary $\mathcal{A}$ that makes at most q queries, there exists $\mathcal{B}_0$ and $\mathcal{B}_1$, two adversaries against the PRP security of E and EUF-CMA security of Π respectively, such that,

$$Adv_{\Pi_N}^{nonce\text{-}EUF\text{-}CMA}(\mathcal{A}) \leq Adv_E^{PRP}(\mathcal{B}_0) + Adv_{\Pi}^{EUF\text{-}CMA}(\mathcal{B}_1) + \frac{q^2}{2^{n+1}}$$

Proof. Let $\mathcal{A}$ be an efficient nonce-EUF-CMA adversary against Π_N that makes at most q queries to its challenger. We consider a series of games from G_0 to G_4. Also, for $j = 0, \ldots, 4$, we define W_j to be the event that $\mathcal{A}$ wins in game G_j.

G_0: This is the nonce-EUF-CMA game between $\mathcal{A}$ and its challenger C. We give description of the challenger's behavior.

C draws a random key $K = (k_0, k_1) \xleftarrow{\$} \mathcal{K}^2$ and uses it to answer queries in the following way,

$$(N, m) \to \Phi(E_{k_1}(E_{k_0}(N) \oplus \Psi))$$

Note that for each query, N must always be fresh or the challenger interrupts the game. By definition,

$$\Pr[W_0] = \mathrm{Adv}_{\Pi_N}^{\text{nonce-EUF-CMA}}(\mathcal{A}) \tag{A.1}$$

G_1: Let $P : \{0,1\}^n \to \{0,1\}^n$ be a random permutation. In this game, the challenger answers queries by using a call to $P(N)$ instead of $E_{k_0}(N)$. More precisely, tag queries are answered as follows,

$$(N, m) \to \Phi(E_{k_1}(P(N) \oplus \Psi))$$

.

Since E_k is a bloc-cipher,

$$|\Pr[W_0] - \Pr[W_1]| = \mathrm{Adv}_E^{\mathrm{PRP}}(\mathcal{B}_0) \tag{A.2}$$

G_2: Let $F : \{0,1\}^n \to \{0,1\}^n$ be a random function. In this game, the calls to the family of random permutations from G_1 are replaced by calls to F. The tag queries are answered as follows,

$$(N, m) \to \Phi(E_{k_1}(F(N) \oplus \Psi))$$

.

The following distinguishing advantage is given by the Switching Lemma [lemma 1].

$$|\Pr[W_1] - \Pr[W_2]| \leq \frac{q^2}{2^{n+1}} \tag{A.3}$$

G_3: In this game, the challenger does not use the given nonce N to answer queries. Instead, it directly draws a random value $r \xleftarrow{\$} 2^n$.

$$(N, m) \rightarrow \Phi(E_{k_1}(r \oplus \Psi)) \quad | \quad r \xleftarrow{\$} \{0,1\}^n$$

Since the adversary can never query the same nonce more than once, the games G_2 and G_3 behave exactly in the same way. It follows that,

$$\Pr[W_2] = \Pr[W_3] \tag{A.4}$$

Let Π' be a MAC that answers queries as in game G_4. Π being EUF-CMA secure implies that Π' is also EUF-CMA secure. Indeed, it is possible to perfectly simulate Π' from an oracle for Π. For any query $(N, m = m_0, m_1, \ldots m_{l-1})$, this is done by drawing a random value r and querying $\Psi^{-1}(r \oplus \Psi(m_0), m_1, \ldots m_{l-1})$ to Π. It follows that,

$$\Pr[W_3] = \mathrm{Adv}_{\Pi}^{\text{EUF-CMA}}(\mathcal{B}_1) \tag{A.5}$$

It follows from (A.1), (A.2), (A.3), (A.4) and (A.5) that,

$$\mathrm{Adv}_{\Pi_N}^{\text{nonce-EUF-CMA}}(\mathcal{A}) \leq \mathrm{Adv}_{E}^{\text{PRP}}(\mathcal{B}_0) + \mathrm{Adv}_{\Pi}^{\text{EUF-CMA}}(\mathcal{B}_1) + \frac{q^2}{2^{n+1}}$$

□

Note. For our proof to go through with this general descripion, we require Ψ to be a public permutation and Ψ^{-1} to be efficiently computable. Indeed, allowing Ψ to be any function introduces some bad edge cases: the security of Π could depend upon specific properties of Ψ that would be lost with $r \oplus \Psi(m_0)$. Now it seems that for any reasonable Π where this is not the case, you can allow Ψ to be any function. You could even have it take the secret key as input. You would then argue that $r \oplus \Psi(m_0)$ with a random r is itself random, and the proof would then go from there.