Online-Extractability in the Quantum Random-Oracle Model

Jelle Don[1], Serge Fehr[1,2], Christian Majenz[1,4], and Christian Schaffner[3,4]

[1] Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands
[2] Mathematical Institute, Leiden University, Netherlands
[3] Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, Netherlands
[4] QuSoft, Amsterdam, Netherlands
`jelle.don@cwi.nl`, `serge.fehr@cwi.nl`, `christian.majenz@cwi.nl`, `c.schaffner@uva.nl`

Abstract. We show the following generic result. Whenever a quantum query algorithm in the quantum random-oracle model outputs a classical value t that is promised to be in some tight relation with $H(x)$ for some x, then x can be efficiently extracted with almost certainty. The extraction is by means of a suitable simulation of the random oracle and works *online*, meaning that it is *straightline*, i.e., without rewinding, and *on-the-fly*, i.e., during the protocol execution and without disturbing it.

The technical core of our result is a new commutator bound that bounds the operator norm of the commutator of the unitary operator that describes the evolution of the compressed oracle (which is used to simulate the random oracle above) and of the measurement that extracts x.

We show two applications of our generic online extractability result. We show *tight* online extractability of commit-and-open Σ-protocols in the quantum setting, and we offer the first non-asymptotic post-quantum security proof of the *textbook* Fujisaki-Okamoto transformation, i.e, without adjustments to facilitate the proof.

1 Introduction

Background. *Extractability* plays an important role in cryptography. In an extractable protocol, on a high level, an algorithm $\mathcal{A}$ sends messages that depend on some secret s, and while the secret remains private in an honest run of the protocol, an *extractor* can learn s via some form of enhanced access to $\mathcal{A}$. The probably most prominent example is that of (zero-knowledge) *proofs* (or *arguments*) *of knowledge*, for which, by definition, there must exist an extractor that manages to extract a witness from any successful yet possibly dishonest prover. Another example are *extractable commitments*, which have a wide range of applications. Hash-based extractable commitments are extremely simple to construct and prove secure in the random-oracle model (ROM) [Pas03]. Indeed, when the considered hash function H is modelled as a random oracle, the hash input x for the commitment $c = H(x)$, where $x = s\|r$ consists of the actual secret s and randomness r, can be extracted simply by finding a query x to the random oracle that yielded c as an output.

The general notion of extractability comes in different flavors. The most well-known example is extraction by *rewinding*. Here, the extractor is allowed to run $\mathcal{A}$ several times, on the same private input and using different randomness. This is the notion usually considered in the context of proofs/arguments of knowledge. In some contexts, extraction via rewinding access is not possible. For example, the UC security model prohibits the simulator to rewind the adversary. In other occasions, rewinding may be possible but not desirable due to a loss of efficiency, which stems from having to run $\mathcal{A}$ multiple times. In comparison, so-called *straightline* extraction works with a single ordinary run of $\mathcal{A}$, without rewinding. Instead, the extractor is then assumed to know some trapdoor information, or it is given enhanced control over some part of the setting. For instance, in the above construction of an extractable commitment, the extractor is given "read access" to $\mathcal{A}$'s random-oracle queries.

Another binary criterion is whether the extraction takes place *on-the-fly*, i.e., during the run of the protocol, or *after-the-fact*, i.e., at the end of the execution. For instance, in the context of proving CCA security for an encryption scheme, to simulate decryption queries without knowing the secret key, it is necessary to extract the plaintext for a queried ciphertext on-the-fly; otherwise, the attacker may abort and not produce the output for which the reduction is waiting.

The extractability of our running example of an extractable commitment in the ROM is *both*, straightline and on-the-fly; we refer to this combination as *online* extraction. This is what we are aiming for in this work: online extractability of (general) hash-based commitments, but now with *post-quantum security*.

For post-quantum security, the ROM needs to be replaced by the *quantum* random-oracle model (QROM) [BDF+11], to reflect the fact that attackers can implement hash functions on a quantum computer. Here, adversaries have quantum superposition access to the random oracle. Many ROM techniques fail in the QROM due to fundamental features of quantum information, such as the so-called *no-cloning principle*. In particular, it is impossible to maintain a query transcript (a fact sometimes referred to as the *recording barrier*), and so one cannot simply "search for a query x to the random oracle", as was exploited for the (classical) RO-security of the extractable-commitment example.

A promising step in the right direction is the compressed-oracle technique, recently developed by Zhandry [Zha19]. This technique enables to maintain *some sort* of a query transcript, but now in the form of a quantum state. This state can be inspected via quantum measurements, offering the possibility to learn some information about the interaction history of an algorithm $\mathcal{A}$ and the random oracle. However, since quantum measurements disturb the state to which they are applied, and this disturbance is often hard to control, this inspection of the query transcript can *per-se*, i.e., without additional argumentation, only be done at the end of the execution (see the Related Work paragraph for more on this).

Our Results. Our main contribution is the following generic extractability result in the QROM. We consider an arbitrary quantum query algorithm $\mathcal{A}$ in the QROM, which announces during its execution some classical value t that is supposed to be equal to $f(x, H(x))$ for some x. Here, f is an arbitrary fixed function, subject to that it must tie t sufficiently to x and $H(x)$, e.g., there must not be too many y's with $f(x, y) = t$; a canonical example is the function $f(x, y) = y$ so that t is supposed to be $t = H(x)$. In general, it is helpful to think of $t = f(x, H(x))$ as a commitment to x. We then show that x can be *efficiently extracted* with almost certainty. The extraction works *online* and is by means of a simulator $\mathcal{S}$ that simulates the quantum random oracle, but which additionally offers an *extraction interface* that produces a guess $\hat{x}$ for x when queried with t. The simulation is statistically indistiguishable from the real quantum random oracle, and $\hat{x}$ is such that whenever $\mathcal{A}$ outputs x with $f(x, H(x)) = t$ at some later point, $\hat{x} = x$ except with negligible probability, while $\hat{x} = \emptyset$ (some special symbol) indicates that $\mathcal{A}$ will not be able to output such an x.

The simulator $\mathcal{S}$ simulates the random oracle using Zhandry's compressed-oracle technique, and extraction is done via a suitable measurement of the compressed oracle's internal register. The technical core of our result is a new bound for the operator norm $\|[O, M]\|$ of the commutator of O, the unitary operator that describes the evolution of the compressed oracle, and of M, the measurement that is used to extract x. This commutator bound allows us to show that the extraction measurement disturbs the behavior of the compressed oracle only by a negligible amount, and so can indeed be performed *on-the-fly*.

We emphasize that even though the existence of the simulator with its extraction interface is proven using the compressed-oracle technique, our presentation is in terms of a black-box simulator $\mathcal{S}$ with certain interfaces and with certain promises on its behavior, abstracting away all the (mainly internal) quantum workings. This makes our generic result applicable (e.g. for the applications discussed below) without the need to understand the underlying quantum aspects.

A first concrete application of our generic result is in the context of so-called commit-and-open Σ-protocols. These are (typically honest-verifier zero-knowledge) interactive proofs of a special form, where the prover first announces a list of commitments and is then asked to open a subset of them, chosen at random by the verifier. We show that, when implementing the commitments with a typical hash-based commitment scheme (like committing to s by $H(s\|r)$ with a random r), such Σ-protocols allow for *online* extraction of a witness in the QROM, with a *smaller security loss* than witness extraction via rewinding.

Equipped with our extractable RO-simulator $\mathcal{S}$, the idea for the above online extraction is very simple: we simulate the random oracle using $\mathcal{S}$ and use its extraction interface to extract the prover's commitments from the first message of the Σ-protocol. As we work out in detail, this procedure gives rise to an online witness extractor that has a polynomial additive overhead in running time compared to the considered prover, and that outputs a valid witness with a probability that is *linear* in the difference of the prover's success probability and the trivial cheating probability, up to an additive error. Using rewinding techniques, on the other hand, incurs a *square-root* loss in success probability classically and a *cube-root* loss quantumly for special-sound Σ-protocols, and typically an even worse loss in case of weaker soundness guarantees, like a k-th-root loss classically and a $(2k+1)$-th-root loss quantumly for k-sound protocols. Furthermore, we show that the dominating additive loss of our reduction is necessary in general, due to attacks on the computational binding property of the random-oracle-based commitments. Along the way, we set up a definitional framework for generalized special soundness notions that might be of independent interest.

A second application of our extractable RO-simulator is a security reduction for the Fujisaki-Okamoto (FO) transformation. We offer the first *non-asymptotic* post-quantum security proof of the *textbook* FO transformation [FO99]. Prior post-quantum security proofs were either asymptotic and without concrete security bounds (like [Zha19]), or had to adjust the transformation to facilitate the proof (like [HHK17]). In particular, all prior non-asymptotic post-quantum security proofs either consider a FO variant that employs an *implicit-rejection* routine, i.e., the decapsulation algorithm outputs a pseudo-random key upon an invalid ciphertext rather than a rejection message, or have to resort to an additional so-called "key confirmation" hash [TU16] that is appended to the ciphertex, thus increasing the ciphertext size. Beyond its theoretical relevance of showing that no adjustment is necessary to admit a post-quantum security proof, the security of the original unmodified FO transformation with explicit rejection in particular ensures that the conservative variant with implicit rejection remains secure even when the decapsulation algorithm is not implemented carefully enough and admits a side-channel attack that reveals information on whether the submitted ciphertext is valid or not.

The core idea of our proof for the textbook FO transformation is to use the extractability of the RO-simulator to handle the decryption queries. Indeed, letting $f(x,y)$ be the encryption $Enc_{pk}(x;y)$ of the message x under the randomness y, a "commitment" $t = f(x, H(x))$ is then precisely the encryption of x under the derandomized scheme, and so the extraction interface recovers x.

Related Work. The compressed-oracle technique has proven to be a powerful tool for lifting classical ROM proofs to the QROM setting. Examples are [LZ19a, CFHL20] for quantum query complexity lower bounds and [HM20] for space-time trade-off bounds, [CMS19] for the security of succinct arguments, [AMRS20] for quantum-access security, and [BHH$^+$19] for a new "double-sided" O2H lemma in the context of the FO transformation. In all these cases, the argument exploits the possibility to extract information on the interaction history of the algorithm $\mathcal{A}$ and the (compressed) oracle *after-the-fact*, i.e., at the very end of the run.

So far, arguments that rely on measuring (the internal state of) the compressed oracle *on-the-fly*, which then causes the state to change, do so by controlling how much *the state* is disturbed by the measurement. In some cases, the disturbance is significant yet asymptotically good enough for the considered application, causing "only" a polynomial blow-up of a negligible error term, as, e.g., in [LZ19b] for proving the security of the Fiat-Shamir transformation. In other cases, it is shown that the measurement outcome is almost certain and therefore the disturbance is negligible, as in [Zha19, CMSZ19] for proving indifferentiability. In those cases, the goal of the measurement is to ensure that the state remains in some "good" subspace rather than learning information about the interaction history.

The approach in our work here differs in that we do not aim at controlling the disturbance of the state, but instead we control the effect that the disturbance will have on future query responses, and thus on $\mathcal{A}$. Technically, this is achieved by bounding the operator norm of the said commutator $[O, M]$.

2 Preliminaries

For Sect. 3 and 4 (only), we assume some familiarity with the mathematics of quantum information as well as with the compressed-oracle technique of [Zha19]. Below, we summarize the concepts that will be of particular importance.

For a function or algorithm f, we slightly abuse notation and write Time$[f]$ to denote the time complexity of (an algorithm computing) f.

2.1 Mathematical Preliminaries

Let $\mathcal{H}$ be a finite-dimensional complex Hilbert space. We use the standard bra-ket notation for the vectors in $\mathcal{H}$ and its dual space. We write $\||\varphi\rangle\|$ for the (Euclidean) norm $\||\varphi\rangle\| = \sqrt{\langle\varphi|\varphi\rangle}$ of $|\varphi\rangle \in \mathcal{H}$. Furthermore, for an operator $A \in \mathcal{L}(\mathcal{H})$, we denote by $\|A\|$ its *operator norm*, i.e., $\|A\| = \max_{|\psi\rangle} \|A|\psi\rangle\|$, where the max is over all $|\psi\rangle \in \mathcal{H}$ with norm 1. We assume the reader to be familiar with basic properties of these norms, like triangle inequality, $\||\varphi\rangle\langle\psi|\| = \||\varphi\rangle\|\||\psi\rangle\|$, $\|A|\varphi\rangle\| \le \|A\|\||\varphi\rangle\|$, $\|AB\| \le \|A\|\|B\|$, etc. Less well known may be the inequality[1]

$$\||\varphi\rangle\langle\psi| - |\psi\rangle\langle\varphi|\| \le \||\varphi\rangle\|\||\psi\rangle\| \,. \tag{1}$$

Another basic yet important property that we will exploit is the following.

Lemma 2.1. *Let A and B be operators in $\mathcal{L}(\mathcal{H})$ with $A^\dagger B = 0$ (i.e., they have orthogonal images)and $AB^\dagger = 0$ (i.e., they have orthogonal supports). Then, $\|A+B\| \le \max\{\|A\|, \|B\|\}$.*

Exploiting that $\|A \otimes B\| = \|A\|\|B\|$, the following is a direct consequence of Lemma 2.1.

Corollary 2.2. *If $A = \sum_x |x\rangle\langle x| \otimes A^x$, i.e., A is a controlled operator,hen $\|A\| \le \max_x \|A^x\|$.*

Definition 2.3. *For operators $A, B \in \mathcal{L}(\mathcal{H})$, the* commutator *is defined as $[A,B] := AB - BA$.*

Some obvious properties of the commutator are:

$$[B,A] = -[A,B] = [A, \mathbb{1} - B] \qquad \text{and} \qquad [A \otimes \mathbb{1}, B \otimes C] = [A,B] \otimes C\,, \tag{2}$$

as well as

$$[AB, C] = A[B,C] + [A,C]B \tag{3}$$

Combining the right equality in (2) with basic properties of the operator norm, if $\|C\| \le 1$, e.g., if C is a unitary of a projection, we have

$$\|[A \otimes \mathbb{1}, B \otimes C]\| = \|[A,B]\|\|C\| \le \|[A,B]\| \,. \tag{4}$$

It is common in quantum information science to write A_X to emphasize that the operator A acts on *register* X, i.e., on a Hilbert space $\mathcal{H}_X$ that is labeled by the letter/symbol X. It is then understood that when applied to registers X and Y, say, A_X acts as A on register X and as identity $\mathbb{1}$ on register Y, i.e., A_X is identified with $A_X \otimes \mathbb{1}_Y$. Property (4) would then e.g. be written as $\|[A_X, B_X \otimes C_Y]\| \le \|[A_X, B_X]\|$. In this work, we will write or not write these subscripts emphasizing the register(s) at our convenience; typically we write them when the argument crucially depends on the registers, and we may omit them otherwise.

Another important matrix norm is the *Schatten-1* or *trace norm*, $\|A\|_1 = \mathrm{tr}\big[\sqrt{A^\dagger A}\big]$. For density matrices ρ and σ, the *trace distance* is then defined as $\delta(\rho,\sigma) = \frac{1}{2}\|\rho - \sigma\|_1$. By equation (9.110) in [NC11] and a short calculation, any norm-1 vectors $|\varphi\rangle$ and $|\psi\rangle$ satsify

$$\delta(|\varphi\rangle\langle\varphi|, |\psi\rangle\langle\psi|) \le \||\varphi\rangle - |\psi\rangle\| \,. \tag{5}$$

For probability distributions p and q, we write $\delta(p,q)$ for the *total variational distance*; this is justified as $\|\rho_0 - \rho_1\|_1 = \delta(p_0, q_1)$ for $\rho_i = \sum_x p_i(x)|x\rangle\langle x|$, $i = 0, 1$. In case of a hybrid classical-quantum state, consisting of a randomized classical value x that follows a distribution p and of a quantum register W with a state ρ^x_W that depends on x, we write $[x, W] = \sum_x p(x)|x\rangle\langle x| \otimes \rho^x_W$.[2] When the distribution p and the density operators ρ^x_W are implicitly given by a game (or experiment) $\mathcal{G}$ then we may write $[x,W]_{\mathcal{G}}$, in particular when considering and comparing different such games. For instance, we write $\delta\big([x,W]_{\mathcal{G}}, [x,W]_{\mathcal{G}'}\big)$ for the trace distance of the respective density matrices in game $\mathcal{G}$ and in game $\mathcal{G}'$.

[1] It is immediate for normalized $|\phi\rangle$ and $|\psi\rangle$ when expanding both vectors in an orthonormal basis containing $|\varphi\rangle$ and $\frac{|\psi\rangle - \langle\varphi|\psi\rangle|\varphi\rangle}{\sqrt{1 - |\langle\varphi|\psi\rangle|^2}}$, and the general case then follows by homogeneity of the norms.

[2] In this equality and at other occasions, we use the same letter, here x, for the considered *random variable* as well as for a *particular value*.

2.2 The (Compressed) Random Oracle

The (quantum) random-oracle model. In the *random-oracle model*, a cryptographic hash function $H : \mathcal{X} \to \mathcal{Y}$ is treated as an external oracle RO that the adversary needs to query on $x \in \mathcal{X}$ in order to learn $H(x)$. The random oracle answers these queries by means of a uniformly random function $H : \mathcal{X} \to \mathcal{Y}$. For concreteness, we restrict here to $\mathcal{Y} = \{0,1\}^n$; on the other hand, we do not further specify the domain $\mathcal{X}$ except that we assume it to have an efficiently computable order, so one may well think of $\mathcal{X}$ as $\mathcal{X} = \{1, \dots, M\}$ for some positive $M \in \mathbb{Z}$ or as bit strings of bounded size. We then often write $RO(x)$ instead of $H(x)$ in order to emphasize that $H(x)$ is obtained by querying the random oracle and/or to emphasize the randomized nature of H.

In the *quantum* random oracle model (QROM), a quantum algorithm $\mathcal{A}$ may make *superposition queries* to RO, meaning that the oracle acts as unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus H(x)\rangle$.

The compressed oracle. We recall here (some version of) the *compressed* oracle, as introduced in [Zha19], which offers a powerful tool for QROM proofs. For this purpose, we consider the multi-register $D = (D_x)_{x\in\mathcal{X}}$, where the state space of D_x is given by $\mathcal{H}_{D_x} = \mathbb{C}[\{0,1\}^n \cup \{\perp\}]$, meaning that it is spanned by an orthonormal set of vectors $|y\rangle$ labelled by $y \in \{0,1\}^n \cup \{\perp\}$. The initial state is set to be $|\perp\rangle_D := \bigotimes_x |\perp\rangle_{D_x}$. Consider the unitary F defined by

$$F|\perp\rangle = |\phi_0\rangle\ , \quad F|\phi_0\rangle = |\perp\rangle \quad \text{and} \quad F|\phi_y\rangle = |\phi_y\rangle\ \ \forall\, y \in \{0,1\}^n \setminus \{0^n\}\,,$$

where $|\phi_y\rangle := H|y\rangle$ with H the Walsh-Hadamard transform on $\mathbb{C}[\{0,1\}^n] = (\mathbb{C}^2)^{\otimes n}$. Exploiting the relation $|y\rangle = 2^{-n/2}\sum_\eta (-1)^{\eta\cdot y}|\phi_\eta\rangle$, we see that

$$F|y\rangle = |y\rangle + 2^{-n/2}\left(|\perp\rangle - |\phi_0\rangle\right)\ . \tag{6}$$

When the oracle is queried, a unitary O_{XYD}, acting on the query registers X and Y and the oracle register D, is applied, given by

$$O_{XYD} = \sum_x |x\rangle\langle x|_X \otimes O^x_{YD_x}\,,$$

with

$$O^x_{YD_x} = F_{D_x}\mathrm{CNOT}_{YD_x}F_{D_x} \tag{7}$$

where $\mathrm{CNOT}_{YD_x}|y\rangle|y_x\rangle = |y \oplus y_x\rangle|y_x\rangle$ for $y, y_x \in \{0,1\}^n$ and acts as identity on $|\perp\rangle|y\rangle$.

Efficient representation of the compressed oracle. Following [Zha19], one can make the (above variant of the) compressed oracle efficient. Indeed, by applying the standard classical sparse encoding to quantum states with the right choice of basis, one can *efficiently* maintain the state D, compute the unitary O_{XYD}, and extract information from D. More details are given in Sect. A in the appendix. For simplicity, we will express things in the remainder of the paper in terms of the inefficient variant of the compressed oracle, but we stress that by the said means all relevant unitaries and measurements can be efficiently computed.

3 Main Technical Result: A Commutator Bound

Our main technical result is a bound on the operator norm of the commutator $[O_{XYD}, M_{DP}]$ of the unitary O_{XYD}, which describes the evolution of the compressed oracle, and the (purified) measurement M_{DP}. Informally, this measurement checks if there is a pair (x, y) in the database satisfying a given relation. If yes, it outputs x, otherwise it outputs $\emptyset$.[3] A small bound on this commutator means that performing this measurement during the runtime of an oracle algorithm $\mathcal{A}$ interacting with a (compressed) random oracle, has little effect.

[3] If there are multiple x's that satisfy the relation, then, by construction, the measurement outputs the *smallest* one.

3.1 Setup and the Technical Statement

Throughout this section, we consider an arbitrary but fixed relation $R \subset \mathcal{X} \times \{0,1\}^n$. A crucial parameter of the relation R is the number of y's that fulfill the relation together with x, maximized over all possible $x \in \mathcal{X}$:

$$\Gamma_R := \max_{x\in\mathcal{X}} \left|\left\{y \in \{0,1\}^n \middle| (x,y) \in R\right\}\right| . \tag{8}$$

Given the relation R, we consider the following projectors:

$$\Pi^x_{D_x} := \sum_{\substack{y \text{ s.t.} \\ (x,y)\in R}} |y\rangle\langle y|_{D_x} \quad \text{and} \quad \Pi^\emptyset_D := \mathbb{1}_D - \sum_{x\in\mathcal{X}} \Pi^x_{D_x} = \bigotimes_{x\in\mathcal{X}} \bar\Pi^x_{D_x} \tag{9}$$

with $\bar\Pi^x_{D_x} := \mathbb{1}_{D_x} - \Pi^x_{D_x}$. Informally, $\Pi^x_{D_x}$ checks whether register D_x contains a value $y \neq \perp$ such that $(x,y) \in R$. We then define the measurement $\mathcal{M} = \mathcal{M}^R$ to be given by the projectors

$$\Sigma^x := \bigotimes_{x'<x} \bar\Pi^{x'}_{D_{x'}} \otimes \Pi^x_{D_x} \quad \text{and} \quad \Sigma^\emptyset := \mathbb{1} - \sum_{x'} \Sigma^{x'} = \bigotimes_{x'} \bar\Pi^{x'}_{D_{x'}} = \Pi^\emptyset \tag{10}$$

where x ranges over all $x \in \mathcal{X}$. Informally, a measurement outcome x means that register D_x is the first that contains a value y such that $(x,y) \in R$; outcome $\emptyset$ means that no register contains such a value. For technical reasons, we consider the *purified* measurement $M_{DP} = M^R_{DP} \in \mathcal{L}(\mathcal{H}_D \otimes \mathcal{H}_R)$ given by the unitary[4]

$$M_{DP} := \sum_{x\in\mathcal{X}\cup\{\emptyset\}} \Sigma^x \otimes \mathsf{X}^x : |\varphi\rangle_D|w\rangle_P \mapsto \sum_{x\in\mathcal{X}\cup\{\emptyset\}} \Sigma^x|\varphi\rangle_D|w+x\rangle_P \,. \tag{11}$$

The following main technical result is a bound on the norm of the commutator $[O_{XYD}, M_{DP}]$.

Theorem 3.1. *For any relation $R \subset \mathcal{X} \times \{0,1\}^n$ and Γ_R as defined in Eq. (8), the purified measurement M_{DP} defined in Eq. (11) almost commutes with the oracle unitary O_{XYD}:*

$$\big\| [O_{XYD}, M_{DP}] \big\| \le 8 \cdot 2^{-n/2}\sqrt{2\Gamma_R} \,.$$

We remark that Lemma 8 in [CMS19] (with the subsequent discussion there) also provides a bound on the norm of a commutator involving O_{XYD}; however, there are various differences that make the two bounds incomparable, as far as we can see. E.g., we consider a specific *measurement* whereas Lemma 8 in [CMS19] is for a rather general *projector*.

Corollary 3.2. *For any state vector $|\psi\rangle \in \mathcal{H}_{WXYDP}$, with W an arbitrary additional register, the state vectors $|\psi'\rangle := O_{XYD}M_{DP}|\psi\rangle$ and $|\psi''\rangle := M_{DP}O_{XYD}|\psi\rangle$ satisfy*

$$\delta\big(|\psi'\rangle\langle\psi'|, |\psi''\rangle\langle\psi''|\big) \le 8 \cdot 2^{-n/2}\sqrt{2\Gamma_R} \,.$$

Proof. By elementary properties and applying Theorem 3.1, we have that

$$\big\| |\psi'\rangle - |\psi''\rangle \big\| = \big\| (O_{XYD}M_{DP} - M_{DP}O_{XYD})|\psi\rangle \big\| \le \big\| [O_{XYD}, M_{DP}] \big\| \le 8 \cdot 2^{-n/2}\sqrt{2\Gamma_R} \,,$$

and the claim on the trace distance then follows from (5).

3.2 The Proof

We prove the Theorem 3.1 by means of the following two lemmas.

Lemma 3.3. *Let F and $O^x_{YD_x}$ be the unitaries introduced in Sect. 2.2, and let $\Pi^x_{D_x}$ and $\Pi^\emptyset_D$ be as in (9). Set $\Gamma_x := \left|\left\{y \in \{0,1\}^n \middle| (x,y) \in R\right\}\right|$. Then*

$$\big\| [F_{D_x}, \Pi^x_{D_x}] \big\| \le 2^{-n/2}\sqrt{2\Gamma_x} \,, \qquad \textit{as well as}$$

$$\big\| [O^x_{YD_x}, \Pi^x_{D_x}] \big\| \le 2 \cdot 2^{-n/2}\sqrt{2\Gamma_x} \quad \textit{and} \quad \big\| [O^x_{YD_x}, \Pi^\emptyset_D] \big\| \le 2 \cdot 2^{-n/2}\sqrt{2\Gamma_x} \,.$$

[4] Both in X^x and in $w + x$ we understand $x \in \mathcal{X} \cup \{\emptyset\}$ to be encoded as an element in $\mathbb{Z}/(|\mathcal{X}|+1)\mathbb{Z}$, $\dim(\mathcal{H}_P) = d := |\mathcal{X}| + 1$, and $\mathsf{X} \in \mathcal{L}(\mathcal{H}_P)$ is the generalized Pauli of order d that maps $|w\rangle$ to $|w+1\rangle$.

The bound on $\|[F,\Pi^x]\|$ can be considered a compact reformulation of (a variant of) Lemma 39 in [Zha19]. We state it here in this form, and (re-)prove it in the appendix (Sect. B), for convenience and completeness. The conceptually new ingredient to the proof of Theorem 3.1 is then Lemma 3.4 below.

Lemma 3.4. *The purified measurement M_{DP} defined in Equation (11) satisfies*

$$\big\|[F_{D_x}, M_{DP}]\big\| \le 3\big\|[F_{D_x}, \Pi_D^x]\big\| + \big\|[F_{D_x}, \Pi_D^\emptyset]\big\| \qquad \textit{and}$$

$$\big\|[O^x_{YD_x}, M_{DP}]\big\| \le 3\big\|[O^x_{YD_x}, \Pi_D^x]\big\| + \big\|[O^x_{YD_x}, \Pi_D^\emptyset]\big\| .$$

Proof. We do the proof for the second claim. The first is proven exactly the same way: the sole property we exploit from $O^x_{YD_x}$ is that it acts only on the D_x register within D, which holds for F_{D_x} as well. Let

$$\bar{\Delta}^\xi := \bigotimes_{\xi'<\xi} \bar{\Pi}^{\xi'}_{D_{\xi'}}$$

be the projection that accepts if no register $D_{\xi'}$ with $\xi' < \xi$ contains a value y' with $(\xi', y') \in R$, and let Δ^ξ be the complement. We then have, using that Π^ξ and $\bar{\Delta}^\xi$ act on disjoint registers,

$$\Sigma^\xi = \bar{\Delta}^\xi \otimes \Pi^\xi = \Pi^\xi \bar{\Delta}^\xi = \bar{\Delta}^\xi \Pi^\xi . \tag{12}$$

We also observe that, with respect to the Loewner order, $\bar{\Delta}^{\xi'} \le \bar{\Delta}^\xi$ for $\xi' < \xi$. Taking it as understood that $O^x_{YD_x}$ acts on registers Y and D_x, we can write

$$[O^x, M_{DP}] = \sum_\xi [O^x, \Sigma^\xi] \otimes \mathsf{X}^\xi + [O^x, \Sigma^\emptyset] \otimes \mathsf{X}^\emptyset . \tag{13}$$

Exploiting basic properties of the operator norm and recalling that $\Sigma^\emptyset = \Pi_D^\emptyset$, we see that the norm of the last term is bounded by $\|[O^x, \Sigma^\emptyset]\| = \|[O^x, \Pi^\emptyset]\|$.

To deal with the sum in (13), we use $\mathbb{1} = \Delta^\xi + \bar{\Delta}^\xi$ to further decompose

$$[O^x, \Sigma^\xi] = \bar{\Delta}^\xi [O^x, \Sigma^\xi] \bar{\Delta}^\xi + \bar{\Delta}^\xi O^x, \Sigma^\xi] \Delta^\xi + \Delta^\xi [O^x, \Sigma^\xi] \bar{\Delta}^\xi + \Delta^\xi [O^x, \Sigma^\xi] \Delta^\xi . \tag{14}$$

We now analyze the four different terms. For the first one, using (12) we see that

$$\bar{\Delta}^\xi [O^x, \Sigma^\xi] \bar{\Delta}^\xi = \bar{\Delta}^\xi \big(O^x \Sigma^\xi - \Sigma^\xi O^x\big) \bar{\Delta}^\xi = \bar{\Delta}^\xi O^x \Pi^\xi \bar{\Delta}^\xi - \bar{\Delta}^\xi \Pi^\xi O^x \bar{\Delta}^\xi = \bar{\Delta}^\xi [O^x, \Pi^\xi] \bar{\Delta}^\xi ,$$

which vanishes for $\xi \ne x$, since then O^x and Π^ξ act on different registers and thus commute. For $\xi = x$, its norm is upper bounded by $\|[O^x, \Pi^x]\|$.

We now consider the second term; the third one can be treated the same way by symmetry, and the fourth one vanishes, as will become clear immediately from below. Using (12) and $\bar{\Delta}^\xi \Delta^\xi = 0$, so that $\bar{\Delta}^\xi \Sigma^\xi = 0$, we have

$$\bar{\Delta}^\xi [O^x, \Sigma^\xi] \Delta^\xi = \bar{\Delta}^\xi \big(O^x \Sigma^\xi - \Sigma^\xi O^x\big) \Delta^\xi = \Sigma^\xi O^x \Delta^\xi =: N_\xi . \tag{15}$$

Looking at (13), we want to control the norm of the sum $N := \sum_\xi N_\xi \otimes X^\xi$. To this end, we show that N_ξ and $N_{\xi'}$ have orthogonal images and orthogonal support, i.e., $N_{\xi'}^\dagger N_\xi = 0 = N_{\xi'} N_\xi^\dagger$, for all $\xi \ne \xi'$. We first observe that if $x \ge \xi$ then O^x commutes with Δ^ξ, since they act on different registers then, and thus

$$N_\xi = \Sigma^\xi O^x \Delta^\xi = \Sigma^\xi \Delta^\xi O^x = \Pi^\xi \bar{\Delta}^\xi \Delta^\xi O^x = 0 ,$$

exploiting once more that $\bar{\Delta}^\xi \Delta^\xi = 0$. Therefore, we only need to consider $N_\xi, N_{\xi'}$ for $\xi, \xi' > x$ (see Fig. 1 top left), where we may assume $\xi > \xi'$. For the orthogonality of the images, we observe that

$$\Pi^{\xi'} \bar{\Delta}^\xi = 0 \tag{16}$$

by definition of $\bar{\Delta}^\xi$ as a tensor product with $\bar{\Pi}^{\xi'}$ being one of the components. Therefore,

$$(\Sigma^{\xi'})^\dagger \Sigma^\xi = \Sigma^{\xi'} \Sigma^\xi = \bar{\Delta}^{\xi'} \Pi^{\xi'} \bar{\Delta}^\xi \Pi^\xi = 0 ,$$

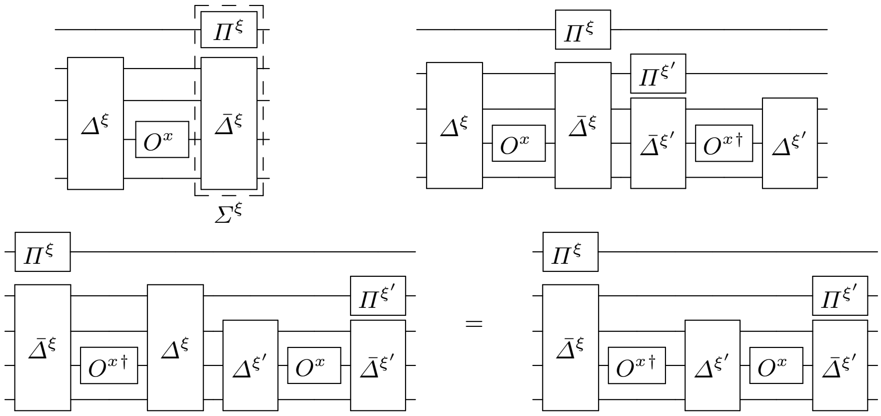

Fig. 1. The operators N_ξ (top left), $N_{\xi'}^\dagger N_\xi$ (top right), and $N_{\xi'} N_\xi^\dagger$ (bottom), for $x < \xi' < \xi$.

and $N_{\xi'}^\dagger N_\xi = 0$ follows directly (see also Fig. 1 top right). For the orthogonality of the supports, we recall that $\bar\Delta^{\xi'} \le \bar\Delta^\xi$, and thus $\Delta^{\xi'} \ge \Delta^\xi$, from which it follows that $\Delta^\xi \Delta^{\xi'} = \Delta^{\xi'}$. $N_{\xi'} N_\xi^\dagger = 0$ then follows by exploiting (16) again (see Fig. 1 bottom).

These orthogonality properties for the images and supports of the N_ξ immediately extend to $N_\xi \otimes X^\xi$, so we have

$$\|N\| \le \max_{\xi > x} \|N_\xi \otimes \mathsf{X}^\xi\| \le \max_{\xi > x} \|N_\xi\|$$

by Lemma 2.1. Recall from (15) that $N_\xi = \bar\Delta^\xi [\Sigma^\xi, O^x] \Delta^\xi$. Furthermore, we exploit that, by definition, Σ^ξ is in tensor-product form and O^x acts trivially on all components in this tensor product except for the component $\bar\varPi^x$, so that $[\Sigma^\xi, O^x] = [\bar\varPi^x, O^x]$ by property (4). Thus,

$$\|N_\xi\| \le \|[\Sigma^\xi, O^x]\| = \|[\bar\varPi^x, O^x]\| = \|[\varPi^x, O^x]\|\,.$$

Using the triangle inequality with respect to the sum versus the last term in (13), and another triangle inequality with respect to the decomposition (14), we obtain the claimed inequality. □

The proof of Theorem 3.1 is now an easy consequence.

Proof (of Theorem 3.1). Since O_{XYD} is a control unitary $O_{XYD} = \sum_x |x\rangle\langle x| \otimes O^x_{YD_x}$, controlled by $|x\rangle$, while M_{DP} does not act on register X, it follows that

$$\big\|[O_{XYD}, M_{DP}]\big\| \le \max_x \big\|[O^x_{YD_x}, M_{DP}]\big\|\,.$$

The claim of the theorem now follows by combining Lemma 3.4 with Lemma 3.3. □

3.3 A First Immediate Application

As an immediate application of the commutator bound of Theorem 3.1, we can easily derive the following generic query-complexity bound for finding x with $(x, H(x)) \in R$ and $\varGamma_R$ as defined in Eq. (8).

Proposition 3.5. *For any algorithm $\mathcal{A}$ that makes q queries to the random oracle RO,*

$$\Pr_{x \leftarrow \mathcal{A}^{RO}} \big[\big(x, RO(x)\big) \in R\big] \le 152 q^2 \varGamma_R / 2^n\,. \tag{17}$$

Proof. By Lemma 5 in [Zha19], we have that

$$\sqrt{\Pr_{x \leftarrow \mathcal{A}^H}[(x, RO(x)) \in R]} \le \sqrt{\Pr_{x' \leftarrow G^R}[x' \ne \emptyset]} + 2^{-n/2}, \tag{18}$$

where G^R is the following procedure/game: (1) run $\mathcal{A}$ using the compressed oracle, and (2) apply the measurement $\mathcal{M}^R$ to obtain $x' \in \mathcal{X} \cup \{\emptyset\}$, which is the same as preparing a register P, applying $M_{DP} = M^R_{DP}$, and measuring P.

In other words, writing $|\psi\rangle_{WXY}$ for the initial state of $\mathcal{A}$ and V_{WXY} for the unitary applied between any two queries (which we may assume to be fixed without loss of generality), and setting $U_{WXYD} := V_{WXY}O_{XYD}$, $\Pi_P := \mathbb{1}_P - |\emptyset\rangle\langle\emptyset|_P$ and $|\Psi\rangle := |\psi\rangle_{WXY} \otimes |\bot\rangle_D^{\otimes|\mathcal{X}|} \otimes |0\rangle_P$, we have, omitting register subscripts,

$$\begin{aligned}\sqrt{\Pr[x' \neq \emptyset]} &= \big\|\Pi M U^q|\Psi\rangle\big\| \leq \sum_{i=1}^{q} \big\|\Pi U^{i-1}[M,U]U^{q-i}|\Psi\rangle\big\| + \big\|\Pi U^q M|\Psi\rangle\big\| \\ &\leq q\big\|[M_{DP}, O_{XYD}]\big\| + \big\|\Pi_P M_{DP}|\Psi\rangle\big\| = q\big\|[M_{DP}, O_{XYD}]\big\| \leq 8\cdot 2^{-n/2}q\sqrt{2\Gamma_R}\,,\end{aligned}$$

where the last equation exploits that $\Pi_P M_{DP}$ applied to $|\bot\rangle_D^{\otimes|\mathcal{X}|} \otimes |0\rangle_P$ vanishes, and the final inequality is by Theorem 3.1. Observing $(8\sqrt{2}+1)^2 = 129 + 16\sqrt{2} \approx 151.6$ finishes the proof. □

Applied to $R = \mathcal{X} \times \{0^n\}$, where $\Gamma_R = 1$, we recover the famous lower bound for search in a random function. In essence, our commutator bound replaces the "progress-measure" argument in the search-lower-bound proof from [Zha19].

Corollary 3.6. *For any algorithm $\mathcal{A}$ that makes q queries to the random oracle RO,*

$$\Pr_{x \leftarrow \mathcal{A}^{RO}}[RO(x) = 0^n] \leq 152q^2/2^n. \tag{19}$$

4 Extraction of Random-Oracle Based Commitments

Throughout this Sect. 4, let $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{T}$ be an arbitrary fixed function with $\mathcal{Y} = \{0,1\}^n$. For a hash function $H : \mathcal{X} \to \mathcal{Y}$, which will then be modelled as a random oracle RO, we will think and sometimes speak of $f(x, H(x))$ as a *commitment* of x (though we do not require it to be a commitment scheme in the strict sense). Typical examples are $f(x,y) = y$ and $f(x,y) = \mathsf{Enc}_{pk}(x;y)$, where the latter is the encryption of x under public key pk with randomness y.

4.1 Informal Problem Description

Consider a query algorithm $\mathcal{A}^{RO}$ in the random oracle model, which, during the course of its run, announces some $t \in \mathcal{T}$. This t is supposed to be $t = f(x, RO(x))$ for some x, and, indeed, $\mathcal{A}^{RO}$ may possibly reveal x later on, i.e., open the commitment Intuitively, in order for the required relation between x and t to hold, we expect that $\mathcal{A}^{RO}$ *first* has to query RO on x and only *then* can output t; thus, one may hope to be able to extract x from RO *early on*, i.e., at the time $\mathcal{A}^{RO}$ announces t.

This is clearly true when $\mathcal{A}$ is restricted to classical queries, simply by checking all the queries made so far. This observation was first made and utilized by Pass [Pas03] and only requires looking at the query transcript (it can be done in the *non-programmable* ROM). As the extractor does not change the course of the experiment, it is in particular also suitable in situations where it is necessary to extract an opening on the fly, i.e., while guaranteeing that $\mathcal{A}$ still proceeds to produce its output (e.g. for multiple-committer parallel extraction [ABG+20]).

In the setting considered here, $\mathcal{A}^{RO}$ may query the random oracle in *superposition* over various choices of x, making it impossible to maintain a classical query transcript. On the positive side, since the output t is required to be classical, $\mathcal{A}^{RO}$ has to perform a measurement before announcing t, enforcing such a superposition to collapse.[5] We show here that early extraction of x is indeed possible in this quantum setting as well.

Note that if the goal is to extract *the same* x as $\mathcal{A}$ will (potentially) output, which is what we aim for, then we must naturally assume that it is hard for $\mathcal{A}$ to find $x \neq x'$ that are both consistent with the same t, i.e., we must assume the commitment to be binding. Formally, for the upcoming discussion in this section to be meaningful,e will think of $\Gamma(f)$ and $\Gamma'(f)$, defined as follows, to be small compared to $|\mathcal{Y}| = 2^n$. When f is fixed, we simply write Γ and Γ'.

Definition 4.1. *For $f : \mathcal{X} \times \{0,1\}^n \to \mathcal{T}$, we define*

$$\Gamma(f) := \max_{x,t} |\{y \mid f(x,y) = t\}| \quad \textit{and} \quad \Gamma'(f) := \max_{x \neq x', y'} |\{y \mid f(x,y) = f(x',y')\}|\,.$$

[5] We can also think of this measurement being done by the interface that receives t.

For the example $f(x, y) = y$, we have $\Gamma(f) = 1 = \Gamma'(f)$. For the example $f(x, y) = \mathsf{Enc}_{pk}(x; y)$, they both depend on the choice of the encryption scheme but typically are small, e.g. $\Gamma(f) = 1$ if Enc is injective as a function of the randomness y and $\Gamma'(f) = 0$ if there are no decryption errors.

Remark 4.2. We note that the ratio $\Gamma(f)/2^n$ remains unaffected when n is increased, i.e., if $\tilde{n} \geq n$ and $\tilde{f} : \mathcal{X} \times \{0,1\}^{\tilde{n}} \to \mathcal{T}$ is given by $\tilde{f}(x, y\|y') := f(x, y)$ for all $x \in \mathcal{X}$, $y \in \{0,1\}^n$ and $y' \in \{0,1\}^{\tilde{n}-n}$, then $\Gamma(\tilde{f})/2^{\tilde{n}} = \Gamma(f)/2^n$, because the additional $\tilde{n} - n$ bits of y' do not affect the conditions on $\tilde{f}$ in Definition 4.1, so both numerator and denominator of the fraction get multiplied by $2^{\tilde{n}-n}$. The same holds for $\Gamma'(f)/2^n$.

4.2 The Extractable RO-Simulator $\mathcal{S}$

Towards formalizing the above goal, we introduce a simulator $\mathcal{S}$ that replaces RO and tries to extract x early on, right after $\mathcal{A}$ announces t. In more detail, $\mathcal{S}$ acts as a black-box oracle with two interfaces, the *RO-interface* $\mathcal{S}.RO$ providing access to the simulated random oracle, and the *extraction interface* $\mathcal{S}.E$ providing the functionality to extract x early on (see Fig. 3, left). In principle, both interfaces can be accessed quantumly, i.e., in superposition over different classical inputs, but in our applications we only use classical access to $\mathcal{S}.E$. We stress that $\mathcal{S}$ is per-se *stateful* and thus may change its behavior from query to query.

Formally, the considered simulator $\mathcal{S}$ is defined to work as follows. It simulates the random oracle and answers queries to $\mathcal{S}.RO$ by means of the compressed oracle. For the $\mathcal{S}.E$ interface, upon a classical input $t \in \mathcal{T}$, $\mathcal{S}$ applies the measurement $\mathcal{M}^t := \mathcal{M}^{R_t}$ from (10) for the relation $R_t := \{(x, y) \,|\, f(x, y) = t\}$ to obtain $\hat{x} \in \mathcal{X} \cup \{\emptyset\}$, which it then outputs (see Fig. 2). In case of a *quantum* query to $\mathcal{S}.E$, the above is performed coherently: given the query registers TP, the unitary $\sum_t |t\rangle\langle t|_T \otimes M^{R_t}_{DP}$ is applied to TPD, and registers TP are then returned.

The extractable RO-oracle $\mathcal{S}$:
Initialization: $\mathcal{S}$ prepares its internal register D to be in state $\|\bot\rangle_D := \bigotimes_x \|\bot\rangle_{D_x}$.
$\mathcal{S}.RO$-query: Upon a (quantum) RO-query, with query registers XY, $\mathcal{S}$ applies O_{XYD} to registers XYD.
$\mathcal{S}.E$-query: Upon a classical extraction-query with input t, $\mathcal{S}$ applies $\mathcal{M}^t$ to D and returns the outcome $\hat{x}$.

Fig. 2. The (inefficient version of the) simulator $\mathcal{S}$, restricted to classical extraction queries.

We note that, as described here, the simulator $\mathcal{S}$ is inefficient, having to maintain an exponential number of qubits; however, using the sparse representation of the internal state D, as discussed in the appendix, Sect. A, $\mathcal{S}$ can well be made efficient without affecting its query-behavior (see Theorem 4.3 for details).

The following statement captures the core properties of $\mathcal{S}$. We refer to two subsequent queries as being *independent* if they can in principle be performed in either order, i.e., if the input to one query does not depend on the output of the other. More formally, e.g., two $\mathcal{S}.RO$ queries are independent if they can be captured by first preparing the two in-/output registers XY and $X'Y'$, and then doing the two respective queries with XY and $X'Y'$. The commutativity claim then means that the order does not matter. Furthermore, whenever we speak of a *classical* query (to $\mathcal{S}.RO$ or to $\mathcal{S}.E$), we consider the obvious classical variant of the considered query, with a classical input and a classical response. Finally, the almost commutativity claims are in terms of the trace distance of the (possibly quantum) output of any algorithm interacting with $\mathcal{S}$ arbitrarily and doing the two considered independent queries in one or the other order.

Theorem 4.3. *The extractable RO-simulator $\mathcal{S}$ constructed above, with interfaces $\mathcal{S}.RO$ and $\mathcal{S}.E$, satisfies the following properties.*

1. *If $\mathcal{S}.E$ is unused, $\mathcal{S}$ is perfectly indistinguishable from the random oracle RO.*

2.a *Any two subsequent independent queries to $\mathcal{S}.RO$ commute. In particular, two subsequent* classical *$\mathcal{S}.RO$-queries with the same input x give identical responses.*

2.b *Any two subsequent independent queries to $\mathcal{S}.E$ commute. In particular, two subsequent* classical *$\mathcal{S}.E$-queries with the same input t give identical responses.*

2.c Any two subsequent independent queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$ $8\sqrt{2\Gamma(f)/2^n}$-almost-commute.

3.a Any classical query $\mathcal{S}.RO(x)$ is idempotent.[6]

3.b Any classical query $\mathcal{S}.E(t)$ is idempotent.

4.a If $\hat{x} = \mathcal{S}.E(t)$ and $\hat{h} = \mathcal{S}.RO(\hat{x})$ are two subsequent classical queries then $f(\hat{x}, \hat{h}) = t$ or $\hat{x} = \emptyset$.

4.b If $h = \mathcal{S}.RO(x)$ and $\hat{x} = \mathcal{S}.E(f(x, h))$ are two subsequent classical queries then $\hat{x} \neq \emptyset$.

Furthermore, the total runtime of $\mathcal{S}$, when implemented using the sparse representation of the compressed oracle described in Sect. A, is bounded as

$$T_{\mathcal{S}} = O\big(q_{RO} \cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2\big)\,,$$

where q_E and q_{RO} are the number of queries to $\mathcal{S}.E$ and $\mathcal{S}.RO$, respectively.

Proof. All the properties follow rather directly by construction of $\mathcal{S}$. Indeed, without $\mathcal{S}.E$-queries, $\mathcal{S}$ is simply the compressed oracle, known to be perfectly indistinguishable from the random oracle, confirming 1. Property 2.a follows from the fact that the unitaries O_{XYD} and $O_{X'Y'D}$, acting on the same register D but on distinct query registers, commute. For 2.b, the claim follows from the commutativity of the unitaries M^t_{DP} and $M^{t'}_{DP'}$. 2.c is a direct consequence of our main technical result Theorem 3.1 (in the form of Cor. 3.2). 3.a follows from the fact that a classical $\mathcal{S}.RO$ query with input x acts as a measurement of register D_x in the computational basis, which is, as any projective measurement, idempotent. Thus, so is the measurement $\mathcal{M}^t$, confirming 3.b. 4.a holds by definition of $\mathcal{M}^t$: if the measurement outcome is $\hat{x} \neq \emptyset$ then the post-measurement state of $D_{\hat{x}}$ is supported by $|y_{\hat{x}}\rangle$'s with $f(\hat{x}, y_{\hat{x}}) = t$. Similarly for 4.b, observing that after the query $h = \mathcal{S}.RO(x)$, the state of D_x has collapsed to $|h\rangle$. Finally, the runtime estimate can be obtained as follows: the second summand is due to the compressed oracle itself, processing the ith $\mathcal{S}$.RO-query takes time $O(i)$. The first term is an estimate using Lemma A.1 in the the appendix, of the cost of performing q_E measurements of the form $\mathcal{M}^t$, on databases of size at most q. □

4.3 Two More Properties of $\mathcal{S}$

On top of the above basic features of our extractable RO-simulator $\mathcal{S}$, we show the following two additional, more technical, properties, which in essence capture that the extraction interface cannot be used to bypass query hardness results.

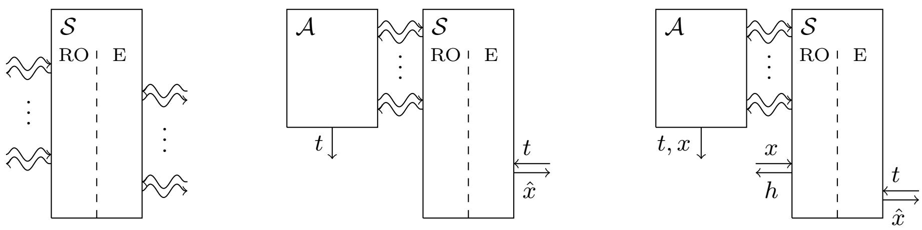

Fig. 3. The extractable RO-simulator $\mathcal{S}$, with its $\mathcal{S}.RO$ and $\mathcal{S}.E$ interfaces, distinguished here by queries from the left and right. Waved arrows denote quantum queries, straight arrows denote classical queries.

The first property is easiest to understand in the context of the example $f(x, y) = y$, where $\mathcal{S}.E(t)$ tries to extract a hash-preimage of t, and where the relations R and R' in Prop. 4.4 below then coincide. In this case, recall from Prop. 3.5 that, informally, if Γ_R is small then it is hard to find $x \in \mathcal{X}$ so that $t := RO(x)$ satisfies $(x, t) \in R$. The statement below ensures that this hardness cannot be bypassed by first selecting a "good" hash value t (e.g. $t := t_\circ$ for the canonical example $R = \mathcal{X} \times \{t_\circ\}$) and then trying to extract a preimage by means of $\mathcal{S}.E$: while 4.b of Theorem 4.3 ensures that $\hat{x} := \mathcal{S}.E(t)$ either satisfies $\mathcal{S}.RO(\hat{x}) = t$ or $\hat{x} = \emptyset$, Prop. 4.4 will actually ensure that $\hat{x} = \emptyset$ most likely.

[6] I.e., applying it twice in a rowhas the same effect on the state of $\mathcal{S}$ as applying it once.

Proposition 4.4. *Let $R' \subseteq \mathcal{X} \times \mathcal{T}$ be a relation. Consider a query algorithm $\mathcal{A}$ that makes q queries to the $\mathcal{S}.RO$ interface of $\mathcal{S}$ but no query to $\mathcal{S}.E$, outputting some $t \in \mathcal{T}$. Let $\hat{x}$ then be obtained by making an additional query to $\mathcal{S}.E$ on input t (see Fig. 3, middle). Then*

$$\Pr_{\substack{t \leftarrow \mathcal{A}^{\mathcal{S}.RO} \\ \hat{x} \leftarrow \mathcal{S}.E(t)}}[(\hat{x}, t) \in R'] \leq 128 \cdot q^2 \Gamma_R / 2^n \,,$$

where $R \subseteq \mathcal{X} \times \mathcal{Y}$ is the relation $(x, y) \in R \Leftrightarrow (x, f(x, y)) \in R'$ and Γ_R as in (8).

Proof. The considered experiment (given by the circuit in Fig. 4, left) is like the experiment G^R in the proof of Prop. 3.5, the only difference being that in G^R the measurement $\mathcal{M}^R$ is applied to register D to obtain x' (see Fig. 4, middle), while here it is the measurement $\mathcal{M}^t$ that is applied to obtain $\hat{x}$. Since both measurements are defined by means of projections that are diagonal in the same basis $\{|\mathbf{y}\rangle\}$ with $|\mathbf{y}\rangle$ ranging over $\mathbf{y} \in (\mathcal{Y} \cup \{\perp\})^{\mathcal{X}}$, we may equivalently measure D in that basis to obtain $\mathbf{y}$ (see Fig. 4, left), and let $\hat{x}$ be minimal so that $f(\hat{x}, y_{\hat{x}}) = t$ (and $\hat{x} = \emptyset$ if no such value exists), and let x' be minimal so that $(x', y_{x'}) \in R$ (and $x' = \emptyset$ if no such value exists). By the respective definitions of $\mathcal{M}^t$ and $\mathcal{M}^R$, both, the variables $\hat{x}, t$ and x', t then have the same distributions as in the respective original two games. But now, we can consider their joint distribution and argue that

$$\Pr[(\hat{x}, t) \in R'] = \Pr[(\hat{x}, f(\hat{x}, y_{\hat{x}})) \in R'] = \Pr[(\hat{x}, y_{\hat{x}}) \in R] \leq \Pr[\exists\, x : (x, y_x) \in R] = \Pr[x' \neq \emptyset] \,,$$

The bound on $\Pr[x' \neq \emptyset]$ from the proof of Prop. 3.5 concludes the proof. □

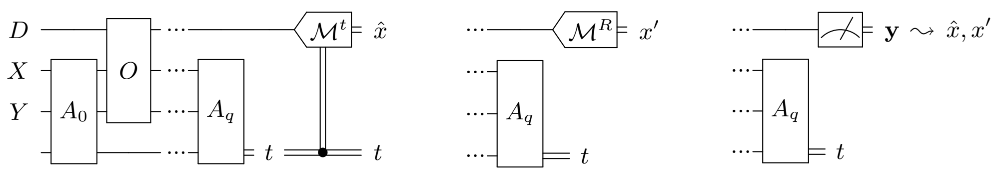

Fig. 4. Quantum circuit diagrams for the experiments in the proof of Prop. 4.4.

In a somewhat similar spirit, the following ensures that if it is hard in the QROM to find x and x' with $f(x, RO(x)) = f(x', RO(x'))$ then this hardness cannot be bypassed by, say, first choosing x, querying $h = \mathcal{S}.RO(x)$, computing $t := f(x, h)$, and then extracting $\hat{x} := \mathcal{S}.E(t)$. The latter will most likely give $\hat{x} = x$, except, intuitively, if $\mathcal{S}.RO$ has additionally been queried on a colliding x'.

Proposition 4.5. *Consider a query algorithm $\mathcal{A}$ that makes q queries to $\mathcal{S}.RO$ but no query to $\mathcal{S}.E$, outputting some $t \in \mathcal{T}$ and $x \in \mathcal{X}$. Let h then be obtained by making an additional query to $\mathcal{S}.RO$ on input x, and $\hat{x}$ by making an additional query to $\mathcal{S}.E$ on input t (see Fig. 3, right). Then*

$$\Pr_{\substack{t,\, x \,\leftarrow\, \mathcal{A}^{\mathcal{S}.RO} \\ h \,\leftarrow\, \mathcal{S}.RO(x) \\ \hat{x} \,\leftarrow\, \mathcal{S}.E(t)}}[\hat{x} \neq x \wedge f(x, h) = t] \leq 40e^2(q+2)^3 \Gamma'(f)/2^n \,.$$

More generally, if $\mathcal{A}$ outputs ℓ-tuples $\mathbf{t} \in \mathcal{T}^\ell$ and $\mathbf{x} \in \mathcal{X}^\ell$, and $\mathbf{h} \in \mathcal{Y}^\ell$ is obtained by querying $\mathcal{S}.RO$ component-wise on $\mathbf{x}$, and $\hat{\mathbf{x}} \in (\mathcal{X} \cup \{\emptyset\})^\ell$ by querying $\mathcal{S}.E$ component-wise on $\mathbf{t}$, then

$$\Pr_{\substack{\mathbf{t},\, \mathbf{x} \,\leftarrow\, \mathcal{A}^{\mathcal{S}.RO} \\ \mathbf{h} \,\leftarrow\, \mathcal{S}.RO(\mathbf{x}) \\ \hat{\mathbf{x}} \,\leftarrow\, \mathcal{S}.E(\mathbf{t})}}[\exists\, i : \hat{x}_i \neq x_i \wedge f(x_i, h_i) = t] \leq 40e^2(q+\ell+1)^3 \Gamma'(f)/2^n \,.$$

The proof is similar in spirit to the proof of Prop. 4.4, but relying on the hardness of collision finding (Lemma C.1) rather than on (the proof of) Prop. 3.5, and so is moved to the appendix (Section B).

Remark 4.6. The claim of Prop. 4.5 stays true when the queries $\mathcal{S}.RO(x_i)$ are not performed as *additional* queries *after* the run of $\mathcal{A}$ but are explicitly *among* the q queries that are performed by $\mathcal{A}$ *during* its run. One way to see this is to use 2.a and 3.a of Theorem 4.3to re-do these queries once more after the run of $\mathcal{A}$, which does not affect the subsequent $\mathcal{S}.E$-queries. Alternatively, we observe that the proof does not exploit that these queries are performed at the end, which additionally shows that in this case the ℓ-term on the right hand side of the bound vanishes, i.e., scales as $(q+1)^3$ rather than as $(q+\ell+1)^3$.

4.4 Early Extraction

We consider here the following concrete setting. Let $\mathcal{A}$ be a two-round query algorithm, interacting with the random oracle RO and behaving as follows. At the end of the first round, $\mathcal{A}^{RO}$ outputs some $t \in \mathcal{T}$, and at the end of the second round, it outputs some $x \in \mathcal{X}$ that is supposed to satisfy $f(x, RO(x)) = t$; on top, $\mathcal{A}^{RO}$ may have some additional (possibly quantum) output W (see Fig. 5, left).

We now show how the extractable RO-simulator $\mathcal{S}$ provides the means to extract x early on, i.e., right after $\mathcal{A}$ has announced t. To formalize this claim, we consider the following experiment, which we denote by $G^{\mathcal{A}}_{\mathcal{S}}$. The RO-interface $\mathcal{S}.RO$ of $\mathcal{S}$ is used to answer all the oracle queries made by $\mathcal{A}$. In addition, as soon as $\mathcal{A}$ outputs t, the interface $\mathcal{S}.E$ is queried on t to obtain $\hat{x} \in \mathcal{X} \cup \{\emptyset\}$, and after $\mathcal{A}$ has finished, $\mathcal{S}.RO$ is queried on $\mathcal{A}$'s final output x to generate h; see Fig. 5 (right).

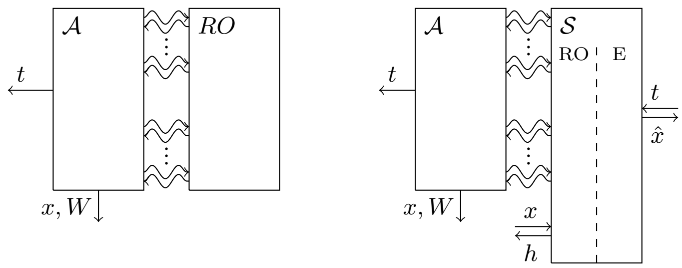

Fig. 5. The original execution of $\mathcal{A}^{RO}$ (left), and the experiment $G^{\mathcal{A}}_{\mathcal{S}}$ with RO simulated by $\mathcal{S}$ (right).

Informally, we want that $\mathcal{A}$ does not notice any difference when RO is replaced by $\mathcal{S}.RO$, and that $\hat{x} = x$ whenever $f\big(x, h\big) = t$, while $\hat{x} = \emptyset$ implies that $\mathcal{A}$ will fail to output x with $f\big(x, h\big) = t$. This situation is captured by the following statement.

Corollary 4.7. *The extractable RO-simulator $\mathcal{S}$ is such that the following holds. For any $\mathcal{A}$ that outputs t after q_1 queries and $x \in \mathcal{X}$ and W after an additional q_2 queries, it holds that*

$$\delta\big([t, x, RO(x), W]_{\mathcal{A}^{RO}}, [t, x, h, W]_{G^{\mathcal{A}}_{\mathcal{S}}}\big) \leq 8(q_2 + 1)\sqrt{2\Gamma/2^n} \qquad \textit{and}$$

$$\Pr_{G^{\mathcal{A}}_{\mathcal{S}}}\big[x \neq \hat{x} \wedge f(x, h) = t\big] \leq 8(q_2 + 1)\sqrt{2\Gamma/2^n} + 40e^2(q + 2)^3\Gamma'/2^n\,,$$

where $q = q_1 + q_2$.

Proof. The first claim follows from the fact that the trace distance vanishes when $\mathcal{S}.E(t)$ is performed at the very end, after the $\mathcal{S}.RO(x)$-query, in combination with the (almost-)commutativity of the two interfaces (Theorem 4.3, 2.a to 2.c). Similarly, the second claim follows from Prop. 4.5 when considering the $\mathcal{S}.E(t)$ query to be performed at the very end, in combination with the (almost-)commutativity of the interfaces again. □

The statements above extend easily to *multi*-round algorithms $\mathcal{A}^{RO}$ that output $t_1, \ldots, t_\ell$ in (possibly) different rounds, and $x_1, \ldots, x_\ell \in \mathcal{X}$ and some (possibly quantum) output W at the end of the run. We then extend the definition of $G^{\mathcal{A}}_{\mathcal{S}}$ in the obvious way: $\mathcal{S}.E$ is queried on each output t_i to produce $\hat{x}_i$, and at the end of the run $\mathcal{S}.RO$ is queried on each of the final outputs $x_1, \ldots, x_\ell$ of $\mathcal{A}$ to obtain $\mathbf{h} = (h_1, \ldots, h_\ell) \in \mathcal{Y}^\ell$. As a minor extension, we allow some of the x_i to be $\perp$, i.e., $\mathcal{A}^{RO}$ may decide to not output certain x_i's; the $\mathcal{S}.RO$ query on x_i is then not done and h_i is set to $\perp$ instead, and we declare that $RO(\perp) = \perp$ and $f(\perp, h_i) \neq t_i$. To allow for a compact notation, we write $RO(\mathbf{x}) = (RO(x_1), \ldots, RO(x_\ell))$ for $\mathbf{x} = (x_1, \ldots, x_\ell)$.

Corollary 4.8. *The extractable RO-simulator $\mathcal{S}$ is such that the following holds. For any $\mathcal{A}$ that makes q queries in total, it holds that*

$$\delta\big([\mathbf{t}, \mathbf{x}, RO(\mathbf{x}), W]_{\mathcal{A}^{RO}}, [\mathbf{t}, \mathbf{x}, \mathbf{h}, W]_{G^{\mathcal{A}}_{\mathcal{S}}}\big) \leq 8\ell(q + \ell)\sqrt{2\Gamma/2^n} \qquad \textit{and}$$

$$\Pr_{G^{\mathcal{A}}_{\mathcal{S}}}\big[\exists\, i : x_i \neq \hat{x}_i \wedge f(x_i, h_i) = t_i\big] \leq 8\ell(q + 1)\sqrt{2\Gamma/2^n} + 40e^2(q + \ell + 1)^3\Gamma'/2^n.$$

Proof. The first claim follows from the fact that the trace distance vanishes when the $\mathcal{S}.E(t_i)$-queries are performed at the very end, after all $\mathcal{S}.RO(x_i)$-queries, in combination with the (almost-)commutativity of the interfaces. Similarly, the second claim follows from (the more general second part of) Prop. 4.5 when considering the $\mathcal{S}.E(t_i)$-queries to be performed at the very end, in combination with the (almost-)commutativity of the interfaces again. □

5 Application I: Extractability of Commit-And-Open Σ-protocols

5.1 Commit-and-Open Σ-protocols

We assume the reader to be familiar with the concept of an interactive proof system for a language $\mathcal{L}$ or a relation R, and specifically with the notion of a *Σ-protocol*. We briefly discuss here the following special class of Σ-protocols.

Here, we consider the notion of a *commit-and-open* Σ-protocol, which is as follows. The prover begins by sending commitments $a_1, ..., a_\ell$ to the prover, computed as $a_i = H(x_i)$ for $x_1, ..., x_\ell \in \mathcal{X}$, where $H : \mathcal{X} \to \{0,1\}^n$ is a hash function, and where we assume for concreteness that $\mathcal{X}$ consists of bitstrings of bounded size. Here, x_i can either be the actual message m_i to be committed, or m_i concatenated with randomness. The verifier answers by sending a challenge c, which is a subset $c \subseteq [\ell] = \{1, ..., \ell\}$, picked uniformly at random from a challenge set $C \subseteq 2^{[\ell]}$, upon which the prover sends the response $z = (x_i)_{i \in c}$. Finally, the verifier checks whether $H(x_i) = a_i$ for every $i \in c$, computes an additional verification predicate $V(c, z)$ and outputs 1 if both check out, 0 otherwise. Such (usually zero-knowledge) protocols have been known since the concept of zero-knowledge proofs was developed [BCC88, GMW91].

Commit-and-open Σ-protocols are (classically) extractable in a straight-forward manner as soon as a witness can be computed from sufficiently many of the x_i's: rewind the prover a few times until it has opened every commitment a_i at least once.[7] There is, however, an alternative (classical) *online* extractor if the hash function H is modelled as a random oracle: simply look at the query transcript of the prover to find preimages of the commitments $a_1, ..., a_\ell$. As the challenge is chosen independently, the extractability and collision resistance of the commitments implies that for a prover with a high success probability, the ℓ extractions succeed simultaneously with good probability. This is roughly how the proof of online extractability of the ZK proof system for graph 3-coloring by Goldreich, Micali and Wigderson [GMW91], instantiated with random-oracle based commitments, works that was announced in [Pas03] and shown in [Pas04] (Prop. 5).

Equipped with our extractable RO-simulator $\mathcal{S}$, we can mimmic the above in the quantum setting. Indeed, the only change is that the look-ups in the transcript are replaced with the additional interface of the simulator $\mathcal{S}$. Cor. 4.8 can then be used to prove the success of extraction using essentially the same extractor as in the classical case.

5.2 Notions of Special Soundness

The property that allows such an extraction is most conveniently expressed in terms of special soundness and its variants. Because there are, next to special and k-soundness, a number of additional variants in the literature (e.g. in the context of Picnic2/Picnic3 [KZ20] or MQDSS [CHR+16]), we begin by formulating a generalized notion of special soundness that captures in a broad sense that a witness can be computed from correct responses to "*sufficiently many*" challenges.[8] While the notions introduced below can be formulated for arbitrary public-coin interactive proof systems, we present them here tailored to commit-and-open Σ-protocols.

In the remainder, Π is thus assumed to be an arbitrary commit-and-open Σ-protocol for a relation R with associated language $\mathcal{L}$, and C is the challenge space of Π. Furthermore, we consider a non-empty, monotone increasing set $\mathfrak{S}$ of subsets $S \subseteq C$, i.e., such that $S \in \mathfrak{S} \wedge S \subseteq S' \Rightarrow S' \in \mathfrak{S}$, and we let $\mathfrak{S}_{\min} := \{S \in \mathfrak{S} \,|\, S_\circ \subsetneq S \Rightarrow S_\circ \notin \mathfrak{S}\}$ consist of the minimal sets in $\mathfrak{S}$.

[7] Naturally, we can assume $[\ell] = \bigcup_{c \in C} c$

[8] Using the language from secret sharing, we consider an arbitrary access structure $\mathfrak{S}$, while the k-soundness case corresponds to a threshold access structure.

Definition 5.1. *Π is called* $\mathfrak{S}$-sound *if there exists an efficient algorithm $\mathcal{E}_{\mathfrak{S}}(I, x_1, \ldots, x_\ell, S)$ that takes as input an instance $I \in \mathcal{L}$, strings $x_1, \ldots, x_\ell \in \mathcal{X}$ and a set $S \in \mathfrak{S}_{\min}$, and outputs a witness for I whenever $V(c, (x_i)_{i \in c}) = 1$ for all $c \in S$, and outputs $\perp$ otherwise.*[9]

Note that there is no correctness requirement on the x_i's with $i \notin \bigcup_{c \in S} c$; thus, those x_i's may just as well be set to be empty strings.

This property generalizes k-soundness, which is recovered for $\mathfrak{S} = \mathfrak{T}_k := \{S \subseteq C \,|\, |S| \geq k\}$, but it also captures more general notions. For instance, the r-fold parallel repetition of a k-sound protocol is not k-sound anymore, but it is $\mathfrak{T}_k^{\vee r}$-sound with $\mathfrak{T}_k^{\vee r}$ consisting of those subsets of challenge-sequences $(c_1, \ldots, c_r) \in C^r$ for which the restriction to at least one of the positions is a set in $\mathfrak{T}_k$. This obviously generalizes to the parallel repetition of an arbitrary $\mathfrak{S}$-sound protocol, with the parallel repetition then being $\mathfrak{S}^{\vee r}$-sound with

$$\mathfrak{S}^{\vee r} := \{S \subseteq C^r \,|\, \exists\, i : S_i \in \mathfrak{S}\},$$

where $S_i := \{c \in C \,|\, \exists\, (c_1, ..., c_r) \in S : c_i = c\}$ is the i-th *marginal* of S.

For our result to apply, we need a strengthening of the above soundness condition where $\mathcal{E}_{\mathfrak{S}}$ has to find the set S himself. This is clearly the case for $\mathfrak{S}$-sound protocols that have a *constant sized* challenge space C, but also for the parallel repetition of $\mathfrak{S}$-sound protocols with a constant sized challenge space. Formally, we require the following strengthened notion of $\mathfrak{S}$-sound protocols.

Definition 5.2. *Π is called* $\mathfrak{S}$-sound* *if there exists an efficient algorithm $\mathcal{E}^*_{\mathfrak{S}}(I, x_1, \ldots, x_\ell)$ that takes as input an instance $I \in \mathcal{L}$ and strings $x_1, \ldots, x_\ell \in \mathcal{X}$, and outputs a witness for I whenever there exists $S \in \mathfrak{S}$ with $V(c, (x_i)_{i \in c}) = 1$ for all $c \in S$, and outputs $\perp$ otherwise.*

$\mathfrak{S}$-sound Σ-protocols may — and often do — have the property that a dishonest prover can pick any set $\hat{S} = \{\hat{c}_1, \ldots, \hat{c}_m\} \notin \mathfrak{S}$ of challenges $\hat{c}_i \in C$ and then prepare $\hat{x}_1, \ldots, \hat{x}_\ell$ in such a way that $V(c, (\hat{x}_i)_{i \in c}) = 1$ if $c \in \hat{S}$, i.e., after having committed to $\hat{x}_1, \ldots, \hat{x}_\ell$ the prover can successfully answer challenge c if $c \in \hat{S}$. We call this a *trivial* attack. The following captures the largest success probability of such a trivial attack, maximized over the choice of $\hat{S}$:

$$p_{triv}^{\mathfrak{S}} := \frac{1}{|C|} \max_{\hat{S} \notin \mathfrak{S}} |\hat{S}|\,. \tag{20}$$

When there is no danger of confusion, we omit the superscript $\mathfrak{S}$. Looking ahead, our result will show that for any prover that does better than the trivial attack by a non-negligible amount, online extraction is possible. For special sound Σ-protocols, $p_{triv} = 1/|C|$, and for k-sound Σ-protocols, $p_{triv} = (k-1)/|C|$. Furthermore, our definition of $\mathfrak{S}$-soundness allows a straightforward parallel repetition lemma on the combinatorial level providing an expression for p_{triv} of parallel-repeated Σ-protocols.

Lemma 5.3. *Let Π be an $\mathfrak{S}$-sound Σ-protocol. Then $p_{triv}^{\mathfrak{S}^{\vee r}} = \left(p_{triv}^{\mathfrak{S}}\right)^r$.*

Proof. To prove the lemma, we simplify

$$p_{triv}^{\mathfrak{S}^{\vee r}} = \frac{1}{|C|^r} \max_{\hat{S} \notin \mathfrak{S}^{\vee r}} |\hat{S}| = \frac{1}{|C|^r} \max_{\substack{\hat{S} \subseteq C^r: \\ \forall i: \hat{S}_i \notin \mathfrak{S}}} |\hat{S}| = \frac{1}{|C|^r} \left(\max_{\hat{S} \notin \mathfrak{S}} |\hat{S}| \right)^r = \left(p_{triv}^{\mathfrak{S}}\right)^r.$$

□

5.3 Online Extractability in the QROM

We are now ready to define our extractor and prove that it succeeds. Equipped with the results from the previous section, the intuition is very simple. Given a (possibly dishonest) prover $\mathcal{P}$, running the considered Σ-protocol in the QROM, we use the simulator $\mathcal{S}$ to answer $\mathcal{P}$'s queries to the random oracle but also to extract the commitments $a_1, \ldots, a_\ell$, and if the extracted $\hat{x}_1, \ldots, \hat{x}_\ell$ satisfy the verification predicate V for sufficiently many challenges, we can compute a witness by applying $\mathcal{E}^*_{\mathfrak{S}}$.

The following relates the success probability of this extraction procedure to the success probability of the (possibly dishonest) prover.

[9] The restriction for S to be in $\mathfrak{S}_{\min}$, rather than in $\mathfrak{S}$, is only to avoid an exponentially sized input while asking $\mathcal{E}_{\mathfrak{S}}$ to be efficient. When C is constant in size, we may admit any $S \in \mathfrak{S}$.

Theorem 5.4. *Let Π be an $\mathfrak{S}$-sound* commit-and-open Σ-protocol where the first message consists of ℓ commitments. Then it admits an online extractor $\mathcal{E}$ in the QROM that succeeds with probability*

$$\Pr[\mathcal{E} \text{ succeeds}] \geq \frac{1}{1-p_{triv}}\big(\Pr[\mathcal{P}^{RO} \text{ succeeds}] - p_{triv} - \varepsilon\big) \qquad \textit{where}$$

$$\varepsilon = 8\sqrt{2}\,\ell(2q+\ell+1)/\sqrt{2^n} + 40e^2(q+\ell+1)^3/2^n$$

and p_{triv} is defined in Eq. (20). For $q \geq \ell+1$, the bound simplifies to

$$\varepsilon \leq 34\ell q/\sqrt{2^n} + 2365q^3/2^n\,.$$

Furthermore, the running time of $\mathcal{E}$ is bounded as $T_{\mathcal{E}} = T_{\mathcal{P}_1} + T_{\mathcal{E}^*_{\mathfrak{S}}} + O(q_1^2)$, where $T_{\mathcal{P}_1}$ and $T_{\mathcal{E}^*_{\mathfrak{S}}}$ are the respective runtimes of $\mathcal{P}_1$ and $\mathcal{E}^*_{\mathfrak{S}}$.

Recall that $p_{triv} = (k-1)/|C|$ for k-soundness, giving a corresponding bound.

Proof. We begin by describing the extractor $\mathcal{E}$. In a first step, using $\mathcal{S}.RO$ to answer $\mathcal{P}$'s queries, $\mathcal{E}$ runs the prover $\mathcal{P}$ until it announces $a_1,\ldots,a_\ell$, and then it uses $\mathcal{S}.E$ to extract $\hat{x}_1,...,\hat{x}_\ell$. I.e., $\mathcal{E}$ acts as $\mathcal{S}$ in Cor. 4.8 for the function $f(x,h) = h$ and runs the game $G^{\mathcal{P}}_{\mathcal{S}}$ to the point where $\mathcal{S}.E$ outputs $\hat{x}_1,...,\hat{x}_\ell$ on input $a_1,\ldots,a_\ell$. As a matter of fact, for the purpose of the analysis, we assume that $G^{\mathcal{P}}_{\mathcal{S}}$ is run until the end, with the challenge c chosen uniformly at random, and where $\mathcal{P}$ then outputs x_i for all $i \in c$ (and $\perp$ for $i \notin c$) at the end of $G^{\mathcal{P}}_{\mathcal{S}}$; we also declare that $\mathcal{P}$ additionally outputs c and $a_1,\ldots,a_\ell$ at the end. Then, upon having obtained $\hat{x}_1,...,\hat{x}_\ell$, the extractor $\mathcal{E}$ runs $\mathcal{E}^*_{\mathfrak{S}}$ on $\hat{x}_1,...,\hat{x}_\ell$ to try to compute a witness. By definition, this succeeds if $\hat{S} := \{\hat{c} \in C \,|\, V(\hat{c},(\hat{x}_i)_{i\in\hat{c}}) = 1\}$ is in $\mathfrak{S}$.

It remains to relate the success probability of $\mathcal{E}$ to that of the prover $\mathcal{P}^{RO}$. By the first statement of Cor. 4.8, writing $\mathbf{x}_c = (x_i)_{i\in c}$, $RO(\mathbf{x}_c) = (RO(x_i))_{i\in c}$, $\mathbf{a}_c = (a_i)_{i\in c}$, etc., we have

$$\begin{aligned}\Pr[\mathcal{P}^{RO} \text{ succeeds}] &= \Pr_{\mathcal{P}^{RO}}[V(c,\mathbf{x}_c) = 1 \wedge RO(\mathbf{x}_c) = \mathbf{a}_c] \\ &\leq \Pr_{G^{\mathcal{P}}_{\mathcal{S}}}[V(c,\mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c] + \delta_1\end{aligned} \tag{21}$$

with $\delta_1 = 8\sqrt{2}\,\ell(q+\ell)/\sqrt{2^n}$. Omitting the subscript $G^{\mathcal{P}}_{\mathcal{S}}$ now,

$$\begin{aligned}&\Pr[V(c,\mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c] \\ &\quad\leq \Pr[V(c,\mathbf{x}_c) = 1 \wedge \mathbf{h}_c = \mathbf{a}_c \wedge \mathbf{x}_c = \hat{\mathbf{x}}_c] + \Pr[\mathbf{h}_c = \mathbf{a}_c \wedge \mathbf{x}_c \neq \hat{\mathbf{x}}_c] \\ &\quad\leq \Pr[V(c,\hat{\mathbf{x}}_c) = 1] + \Pr[\exists\, j \in c : x_j \neq \hat{x}_j \wedge h_j = a_j] \\ &\quad\leq \Pr[V(c,\hat{\mathbf{x}}_c) = 1] + \delta_2\end{aligned} \tag{22}$$

with $\delta_2 = 8\sqrt{2}\,\ell(q+1)/\sqrt{2^n} + 40e^2(q+\ell+1)^3/2^n$, where the last inequality is by the second statement of Cor. 4.8, noting that, by choice of f, the event $h_j = a_j$ is equal to $f(x_j,h_j) = a_j$. Recalling the definition of $\hat{S}$,

$$\begin{aligned}\Pr[V(c,\hat{\mathbf{x}}_c) = 1] = \Pr[c \in \hat{S}] &\leq \Pr[\hat{S} \in \mathfrak{S}] + \Pr[c \in \hat{S} \,|\, \hat{S} \notin \mathfrak{S}]\Pr[\hat{S} \notin \mathfrak{S}] \\ &\leq \Pr[\mathcal{E} \text{ succeeds}] + p_{triv}(1 - \Pr[\mathcal{E} \text{ succeeds}])\end{aligned} \tag{23}$$

where the final inequality exploits that c is chosen at random and independent of $\hat{x}_1,\ldots,\hat{x}_\ell$, and thus is independent of the event $\hat{S} \notin \mathfrak{S}$. Combining (21), (22) and (23), we obtain

$$\Pr[\mathcal{P}^{RO} \text{ succeeds}] \leq \Pr[\mathcal{E} \text{ succeeds}] + p_{triv}(1 - \Pr[\mathcal{E} \text{ succeeds}]) + \delta_1 + \delta_2$$

and solving for $\Pr[\mathcal{E} \text{ succeeds}]$ gives the claimed bound. □

5.4 Tightness

The bound given by Theorem 5.4 is tight in the sense that the extraction success probability is proportional to the advantage of a malicious prover over the trivial success probability, up to a negligible additive error term. On top, the additive error term is asymptotically tight: ε remains negligible in n for $q = 2^{\alpha n}$ with any $\alpha < \frac{1}{3}$, while with $q = 2^{n/3}$ queries a collision in the hash function can be found with constant success probability [BHT98, Zha15], breaking the binding property of the commitment scheme upon which typical soundness proofs for commit-and-open Σ-protocols rely.

It is even not too hard to find relevant examples of commit-and-open Σ-protocols where a collision-finding attack not only invalidates the soundness proof but leads to an actual attack against extractability. Consider e.g. the Σ-protocol ZKBoo that underlies the signature scheme Picnic. Here, the prover commits to three messages m_1, m_2, m_3 as $a_i = H(m_i, r_i)$ for random strings r_1, r_2, r_3, and where the m_i's are the respective views of the three parties in an "in-the-head" execution of a 3-party-computation protocol. The challenge space is $C = \{\{1,2\},\{1,3\},\{2,3\}\}$, which means that the prover is then asked to open two out of the three commitments. Now consider the following attack. The attacker can easily find pairs (m_1, m_2), (m_1', m_3) and (m_2', m_3'), so that each pair consists of two mutually consistent views of the considered 3-party-computation protocol. Now the only thing the attacker has to do is to find three collisions in the hash function of the form $a_i = H(m_i, r_i) = H(m_i', r_i')$, $i = 1, 2, 3$. This can be done using e.g. the BHT algorithm [BHT98] if r_i are sufficiently long. The attacker now sends (a_1, a_2, a_3), receives a challenge and responds with the appropriate preimages of the two commitments indicated by the challenge.

5.5 Application to Fiat Shamir Signatures

In the appendix (Sect. D) we discuss the impact on Fiat Shamir signatures, in particular on the round-3 signature candidate Picnic [CDG+17] in the NIST standardization process for post-quantum cryptographic schemes. In short, one crucial part in the chain of arguments to prove security of Fiat Shamir signatures is to prove that the underlying Σ-protocol is a proof of knowledge. For post-quantum security, so far this step relied on Unruh's rewinding lemma, which leads (after suitable generalization), to a $(2k+1)$-th root loss for a k-sound protocols. For commit-and-open Σ-protocols, Theorem 5.4 can replace Unruhs rewinding lemma when working in the QROM, making this step in the chain of arguments tight up to unavoidable additive errors.

As an example, Theorem 5.4 implies a sizeable improvement over the current best QROM security proof of Picnic2 [CDG+17, KZ20, CDG+19]. Indeed, Unruh's rewinding lemma implies a 6-th root loss for the variant of special soundness the underlying Σ-protocol possesses [DFMS19], while Theorem 5.4 is tight.

6 Application II: QROM-Security of Textbook Fujisaki-Okamoto

6.1 The Fujisaki-Okamoto Transformation

The Fujisaki-Okamoto (FO) transform [FO99] is a general method to turn any public-key encryption scheme secure against *chosen-plaintext attacks* (CPA) into a key-encapsulation mechanism (KEM) that is secure against *chosen-ciphertext attacks* (CCA). We can start either from a scheme with one-way security against CPA attacks (OW-CPA) or from one with indistinguishability against CPA attacks (IND-CPA), and in both cases obtain an IND-CCA secure KEM. We recall that a KEM establishes a shared key, which can then be used for symmetric encryption.

We include the (standard) formal definitions of a public-key encryption scheme and of a KEM in the appendix, Section E, and we recall the notions of δ-correctness and γ-spreadness there. In addition, we define a relaxed version of the latter property, *weak* γ-spreadness (see Definition E.4), where the ciphertexts are only required to have high min-entropy when averaged over key generation.[10]. The security games for OW-CPA security of a public-key encryption scheme and for IND-CCA security of a KEM are given in Fig. 6.

The formal specification of the FO transformation, mapping a public-key encryption scheme $\mathsf{PKE} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and two suitable hash functions H and G (which will then be modeled as random oracles) into a key encapsulation mechanism $\mathsf{FO}[\mathsf{PKE}, H, G] = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$, is given in Fig. 7.

[10] This seems relevant e.g. for lattice-based schemes, where the ciphertext has little (or even no) entropy for certain very unlikely choices of the key (like being all 0)

$\underline{\textbf{GAME } \mathsf{OW\text{-}CPA}}$	$\underline{\textbf{GAME } \mathsf{IND\text{-}CCA\text{-}KEM}}$	$\underline{\textsc{Decaps}(c)}$
$(pk, sk) \leftarrow \mathsf{Gen}$	$(pk, sk) \leftarrow \mathsf{Gen}$	$K := \mathsf{Decaps}_{sk}(c)$
$m^* \xleftarrow{\$} \mathcal{M}$	$b \xleftarrow{\$} \{0,1\}$	**return** K
$c^* \leftarrow \mathsf{Enc}_{pk}(m^*)$	$(K_0^*, c^*) \leftarrow \mathsf{Encaps}(pk)$	
$m' \leftarrow \mathcal{A}(pk, c^*)$	$K_1^* \xleftarrow{\$} \mathcal{K}$	
return $m' == m^*$	$b' \leftarrow \mathcal{A}^{\textsc{Decaps}}(c^*, K_b^*)$	
	return $b' == b$	

Fig. 6. Games for OW-CPA security of a PKE and IND-CCA security of a KEM. In the latter, $\mathcal{A}$ is not allowed to query c^* to $\textsc{Decaps}$.

$\underline{\mathsf{Gen}}$	$\underline{\mathsf{Encaps}(pk)}$	$\underline{\mathsf{Decaps}_{sk}(c)}$
$(sk, pk) \leftarrow \mathsf{Gen}$	$m \xleftarrow{\$} \mathcal{M}$	$m := \mathsf{Dec}_{sk}(c)$
return (sk, pk)	$c \leftarrow \mathsf{Enc}_{pk}(m; H(m))$	**if** $m = \perp$ **or** $\mathsf{Enc}_{pk}(m; H(m)) \neq c$ **return** $\perp$
	$K := G(m)$	**else return** $K := G(m)$
	return (K, c)	

Fig. 7. The KEM $\mathsf{FO}[\mathsf{PKE}, H, G]$, obtained by applying the FO transformation [FO99] to PKE.

6.2 Post-Quantum Security of FO in the QROM

Our main contribution here is the following security result for the FO transformation in the QROM. In contrast to most of the previous works on the topic, our result applies to the *standard* FO transformation, without any adjustments. Next to being CPA secure, we require the underlying public-key encryption scheme to be so that ciphertexts have a lower-bounded amount of min-entropy (resulting from the encryption randomness), captured by the aforementioned spreadness property. This seems unavoidable for the FO transformation with explicit rejection and without any adjustment, like an additional key confirmation hash (as e.g. in [TU16]).

Theorem 6.1. *Let* PKE *be a δ-correct public-key encryption scheme satisfying weak γ-spreadness. Let $\mathcal{A}$ be any* $\mathsf{IND\text{-}CCA}$ *adversary against* $\mathsf{FO}[\mathsf{PKE}, H, G]$*, making $q_D \geq 1$ queries to the decapsulation oracle* $\textsc{Decaps}$ *and q_H and q_G queries to $H : \mathcal{M} \to \mathcal{R}$ and $G : \mathcal{M} \to \mathcal{K}$, respectively, where H and G are modeled as random oracles. Let $q := q_H + q_G + 2q_D$. Then, there exists a* $\mathsf{OW\text{-}CPA}$ *adversary $\mathcal{B}$ against* PKE *with*

$$\mathsf{ADV}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} \leq 2q\sqrt{\mathsf{ADV}^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}}[\mathcal{B}]} + 84q\sqrt{(q+2)^3\delta} + 12q\sqrt{q_H q_D} \cdot 2^{-\gamma/4}\,.$$

Furthermore, $\mathcal{B}$ has a running time $T_{\mathcal{B}} \leq T_{\mathcal{A}} + O\big(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q^2\big)$.

We start with a proof outline, which is somewhat simplified in that it treats $\mathsf{FO}[\mathsf{PKE}, H, G]$ as an encryption scheme rather than as a KEM. We will transform the adversary $\mathcal{A}$ of the IND-CCA game into a OW-CPA adversary against the PKE in a number of steps. There are two main challenges to overcome. (1) We need to switch from the *deterministic* challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; H(m^*))$ that $\mathcal{A}$ attacks to a *randomized* challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; r^*)$ that $\mathcal{B}$ is then supposed to attack. We do this switch by re-programming $H(m^*)$ to a random value right after the computation of c^*, which is equivalent to keeping H but choosing a random r^* for computing c^*. For reasons that we explain later, we do this switch from H to its re-programmed variant, denoted $H^\diamond$, in two steps, where the first step (from **Game 0** to **1**) will be "for free", and the second step (from **Game 1** to **2**) is argued using the O2H lemma ([Unr14], we use the version given in [AHU19], Theorem 3). (2) We need to answer decryption queries without knowing the secret key. At this point our extractable RO-simulator steps in. We replace $H^\diamond$, modelled as a random oracle, by $\mathcal{S}$, and we use its extraction interface to extract m from any correctly formed encryption $c = \mathsf{Enc}_{pk}(m; H^\diamond(m))$ and to identify incorrect ciphertexts.

One subtle issue in the argument above is the following. The O2H lemma ensures that we can find m^* by measuring one of the queries to the random oracle. However, given that also the decryption oracle makes queries to the random oracle (for performing the re-encryption check), it could be the case that one of those decryption queries is the one selected by the O2H extractor. This situation is problematic since, once we switch to $\mathcal{S}$ to deal with the decryption queries, some of these queries will be dropped (namely when $\mathcal{S}.E(c) = \emptyset$). This is problematic because, per-se, we cannot exclude that this is the one query that will give us m^*. We avoid this problem by our two-step approach for switching from H to $H^\diamond$, which ensures that the only ciphertext c that would bring us in the above unfortunate situation is the actual (randomized) *challenge ciphertext* $c^* = \mathsf{Enc}_{pk}(m^*; r^*)$, which is not submitted by the specification of the security game.

<u>GAME SETUP G_0-G_8</u>

$(pk, sk) \leftarrow \mathsf{Gen}$ $\quad /\!\!/ G_0$-G_7
$(b, m^*) \xleftarrow{\$} \{0,1\} \times \mathcal{M}$ $\quad /\!\!/ G_0$-G_7
$c^* := \mathsf{Enc}_{pk}(m^*; H(m^*))$ $\quad /\!\!/ G_0$-G_7
$\mathbf{input}(pk, c^* = \mathsf{Enc}_{pk}(m^*))$ $\quad /\!\!/ G_8$
$c^\diamond := \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$ $\quad /\!\!/ G_0$-G_6
$K_0^* := G(m^*)$ $\quad /\!\!/ G_0$-G_2
$K_1^* \xleftarrow{\$} \mathcal{K}$
$j \xleftarrow{\$} J_\mathcal{A} \cup J_{D(c^\diamond)}$ $\quad /\!\!/ G_3$-G_6
$j \xleftarrow{\$} J$ $\quad /\!\!/ G_7$-G_8

<u>MAIN PHASE G_0-G_2</u>

$b' \leftarrow \mathcal{A}^{\text{DECAPS},H,G}(c^*, K_b^*)$ $\quad /\!\!/ G_0$-G_1
$b' \leftarrow \mathcal{A}^{\text{DECAPS},H^\diamond,G^\diamond}(c^*, K_b^*)$ $\quad /\!\!/ G_2$
$\mathbf{return}\ b' == b$

<u>MAIN PHASE G_3-G_8</u>

$m' \leftarrow \mathcal{MA}_j^{\text{DECAPS},H^\diamond,G^\diamond}(c^*, K_1^*)$ $\quad /\!\!/ G_3$
$m' \leftarrow \mathcal{MA}_j^{\text{DECAPS},\mathcal{S}.RO,G^\diamond}(c^*, K_1^*)$ $\quad /\!\!/ G_4$-G_5
$m' \leftarrow \mathcal{EA}_j^{\text{DECAPS},\mathcal{S}.RO,G^\diamond}(c^*, K_1^*)$ $\quad /\!\!/ G_6$-G_8
$\mathbf{while}\ i \in I\ \mathbf{do}$ $\quad /\!\!/ G_4$
$\quad \hat{m}_i \leftarrow \mathcal{S}.E(c_i)$ $\quad /\!\!/ G_4$
$\mathbf{return}\ m'$

<u>DECAPS(c) G_0-G_5</u>

$m := \mathsf{Dec}_{sk}(c)$ $\quad /\!\!/ G_0$-G_5
$\mathbf{if}\ m = \perp\ \mathbf{return}\ \perp$ $\quad /\!\!/ G_0$-G_5
$h := H(m), g := G(m)$ $\quad /\!\!/ G_0$
$\mathbf{if}\ c = c^\diamond$ $\quad /\!\!/ G_1$
$\quad h := H(m), g := G(m)$ $\quad /\!\!/ G_1$
$\mathbf{else}$ $\quad /\!\!/ G_1$
$\quad h := H^\diamond(m), g := G^\diamond(m)$ $\quad /\!\!/ G_1$
$h := H^\diamond(m), g := G^\diamond(m)$ $\quad /\!\!/ G_2$-G_3
$h := \mathcal{S}.RO(m), g := G^\diamond(m)$ $\quad /\!\!/ G_3$-G_5
$\mathbf{if}\ \mathsf{Enc}_{pk}(m; h) \neq c$ $\quad /\!\!/ G_0$-G_5
$\quad \mathbf{return}\ \perp$ $\quad /\!\!/ G_0$-G_5
$\mathbf{else\ return}\ K := g$ $\quad /\!\!/ G_0$-G_5
$\hat{m} \leftarrow \mathcal{S}.E(c)$ $\quad /\!\!/ G_5$

<u>DECAPS(c) G_6-G_8</u>

$m := \mathsf{Dec}_{sk}(c)$ $\quad /\!\!/ G_6$-G_7
$\mathbf{query}\ \mathcal{S}.RO(m)$ $\quad /\!\!/ G_6$-G_7
$\hat{m} \leftarrow \mathcal{S}.E(c)$ $\quad /\!\!/ G_6$-G_8
$\mathbf{if}\ \hat{m} = \perp\ \mathbf{return}\ \perp$ $\quad /\!\!/ G_6$-G_8
$\mathbf{else\ return}\ K := G^\diamond(\hat{m})$ $\quad /\!\!/ G_6$-G_8

Fig. 8. Games 0 to 8. H and G are independent random oracles; $H^\diamond$ and $G^\diamond$ coincide with H and G, respectively, except that $H^\diamond(m^*)$ and $G^\diamond(m^*)$ are freshly chosen. We consider the oracle queries to $H^\diamond$ (respectively to $\mathcal{S}.RO$ later on) and to $G^\diamond$ to be labeled by indices $j \in J$, where $J = J_\mathcal{A} \cup J_D$ decomposes this set into those queries made by $\mathcal{A}$ and those made by DECAPS, respectively, and $J_{D(c^\diamond)} \subseteq J_D$ consists of DECAPS' queries upon input $c^\diamond$. Similarly, we consider the queries to DECAPS to be indexed by $i \in I$, with c_i then being the corresponding ciphertext. Since $\mathcal{A}$ is not allowed to query c^* to DECAPS, we have $c_i \neq c^*\ \forall i \in I$. For $j \in J$, $\mathcal{MA}_j^{\text{DECAPS}}$ denotes the execution of $\mathcal{A}^{\text{DECAPS}}$ up to the query indexed by j, and followed by measuring this query and outputting the result. $\mathcal{EA}_j^{\text{DECAPS}}$ coincides with $\mathcal{MA}_j^{\text{DECAPS}}$, except that if $j \in J_D$ then it outputs the corresponding $\hat{m}_i$ instead. The colors are meant to help the reader track (the use of) some variables and concepts that occur in different places across the code.

Proof (of Theorem 6.1). **Games 0** to **8** below show how to turn $\mathcal{A}$ into $\mathcal{B}$ (see also Figure 8). We first analyze the sequence of hybrids for a fixed key pair (sk, pk). Let therefore $\mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\text{KEM}}$ be A's advantage for key pair (sk, pk). In addition, for a fixed pair (sk, pk), let δ_{sk} be the maximum probability of a decryption error and g_{sk} be the maximum probability of any ciphertext, so that $\mathbb{E}[\delta_{sk}] \leq \delta$ and $\mathbb{E}[g_{sk}] \leq 2^{-\gamma}$, with the expectation over $(sk, pk) \leftarrow \mathsf{Gen}$ (we can assume without loss of generality that pk is included in sk).

Game 0 is the IND-CCA game for KEMs, except that we replace the random oracles G and H with a single random oracle F, by setting $H(x) := F(0||x)$ and $G(x) := F(1||x)$.[11] When convenient, we still refer to $F(0\|\cdot)$ as H and $F(1\|\cdot)$ as G. This change does not affect the view of the adversary nor the outcome of the game; therefore,

$$\Pr[b = b' \text{ in } \mathbf{Game\ 0}] = \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\text{KEM}}.$$

[11] These assignments seem to suggest that $\mathcal{R} = \mathcal{K}$, which may not be the case. Indeed, we understand here that $F : \mathcal{M} \to \{0,1\}^n$ with n large enough, and $F(0||x)$ and $F(1||x)$ are then cut down to the right size.

In **Game 1**, we introduce a new oracle $F^\diamond$ by setting $F^\diamond(0\|m^*) := r^\diamond$ and $F^\diamond(1\|m^*) := k^\diamond$ for uniformly random $r^\diamond \in \mathcal{R}$ and $k^\diamond \in \mathcal{K}$, while letting $F^\diamond(b\|m) := F(b\|m)$ for $m \neq m^*$ and $b \in \{0,1\}$. We note that while the *joint* behavior of $F^\diamond$ and F depends on the choice of the challenge message m^*, each one individually is a purely random function, i.e., a random oracle. In line with F, we write $H^\diamond$ for $F^\diamond(0\|\cdot)$ and $G^\diamond$ for $F^\diamond(1\|\cdot)$ when convenient.

Using these definitions, **Game 1** is obtained from **Game 0** via the following modifications. After m^* and c^* have been produced and before $\mathcal{A}$ is executed, we compute $c^\diamond := \mathsf{Enc}_{pk}(m^*; r^\diamond) = \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$, making a query to $H^\diamond$ to obtain $r^\diamond$. Furthermore, for every decapsulation query by $\mathcal{A}$, we let DECAPS use $H^\diamond$ and $G^\diamond$ instead of H and G for checking correctness of the queried ciphertexts c_i and for computing the key K_i, *except* when $c_i = c^\diamond$ (which we may assume to happen at most once), in which case DECAPS still uses H and G. We claim that

$$\Pr[b = b' \text{ in } \textbf{Game 1}] = \Pr[b = b' \text{ in } \textbf{Game 0}] = \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\text{KEM}}\,.$$

Indeed, for any decryption query c_i, we either have $\mathsf{Dec}_{sk}(c_i) =: m_i \neq m^*$ and thus $F^\diamond(b\|m_i) = F(b\|m_i)$, or else $m_i = m^*$; in the latter case we then either have $c_i = c^\diamond$, where nothing changes by definition of the game, or else $\mathsf{Enc}_{pk}(m^*; H(m^*)) = c^* \neq c_i \neq c^\diamond = \mathsf{Enc}_{pk}(m^*; H^\diamond(m^*))$, and hence the re-encryption check fails and $K_i := \perp$ in either case, without querying G or $G^\diamond$. Therefore, the input-output behavior of Decaps is not affected.

In **Game 2**, all oracle calls by Decaps (also for $c_i = c^\diamond$) and all calls by $\mathcal{A}$ are now to $F^\diamond$. Only the challenge ciphertext $c^* = \mathsf{Enc}_{pk}(m^*; H(m^*))$ is still computed using H, and thus with randomness $r^* = H(m^*)$ that is random and independent of m^* and $F^\diamond$. Hence, looking ahead, we can think of c^* as the input to the $\mathsf{OW\text{-}CPA}$ game that the to-be-constructed attacker $\mathcal{B}$ will attack. Similarly, $K_0^* = G(m^*)$ is random and independent of m^* and $F^\diamond$, exactly as K_1^* is, which means that $\mathcal{A}$ can only win with probability $\frac{1}{2}$.

By the O2H lemma ([AHU19], Theorem 3), the difference between the respective probabilities of $\mathcal{A}$ in guessing b in **Game 1** and **2** gives a lower bound on the success probability of a particular procedure to find an input on which F and $F^\diamond$ differ, and thus to find m^*. Formally,

$$\begin{aligned}
2(q_H + q_G + 2)&\sqrt{\Pr[m' = m^* \text{ in } \textbf{Game 3}]} \\
&\geq |\Pr[b' = b \text{ in } \textbf{Game 1}] - \Pr[b' = b \text{ in } \textbf{Game 2}]| \\
&= \frac{1}{2} + \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\text{KEM}} - \frac{1}{2} \\
&= \mathsf{ADV}_{sk}[\mathsf{A}]^{\mathsf{IND\text{-}CCA}}_{\text{KEM}}
\end{aligned}$$

where **Game 3** is identical to **Game 2** above, except that we introduce and consider a new variable m' (with the goal that $m' = m^*$), obtained as follows. Either one of the $q_H + q_G$ queries from $\mathcal{A}$ to $H^\diamond$ and $G^\diamond$ is measured, or one of the two respective queries from DECAPS to $H^\diamond$ and $G^\diamond$ upon a possible decryption query $c^\diamond$ is measured, and, in either case, m' is set to be the corresponding measurement outcome. The choice of which of these $q_H + q_G + 2$ queries to measure is done uniformly at random.[12]

We note that, since we are concerned with the measurement outcome m' only, it is irrelevant whether the game stops right after the measurement, or it continues until $\mathcal{A}$ outputs b'. Also, rather than actually measuring DECAPS' classical query to $H^\diamond$ or $G^\diamond$ upon decryption query $c_i = c^\diamond$ (if instructed to do so), we can equivalently set $m' := m_i = \mathsf{Dec}_{sk}(c^\diamond)$.

For **Game 4**, we consider the function $f : \mathcal{M} \times \mathcal{R} \to \mathcal{C}$, $(m, r) \mapsto \mathsf{Enc}_{pk}(m; r)$, and we replace the random oracle $H^\diamond$ with the extractable RO-simulator $\mathcal{S}$ from Theorem 4.3. Furthermore, *at the very end* of the game, we invoke the extractor interface $\mathcal{S}.E$ to compute $\hat{m}_i := \mathcal{S}.E(c_i)$ for each c_i that A queried to DECAPS in the course of its run. By the first statement of Theorem 4.3, given that the $\mathcal{S}.E$ queries take place only *after* the run of $\mathcal{A}$,

$$\Pr[m' = m^* \text{ in } \textbf{Game 4}] = \Pr[m' = m^* \text{ in } \textbf{Game 3}]\,.$$

Furthermore, applying Prop. 4.4 for $R' := \{(m, c) : \mathsf{Dec}_{sk}(c) \neq m\}$ and union bound, we get that

$$\Pr[\forall i : \hat{m}_i = m_i \vee \hat{m}_i = \emptyset] \geq 1 - \varepsilon_1$$

[12] If this choice instructs to measure DECAPS's query to $H^\diamond$ or to $G^\diamond$ for the decryption query $c^\diamond$, but there is no decryption query $c_i = c^\diamond$, $m' := \perp$ is output instead.

for $\varepsilon_1 = 128 q_D (q_H + q_D)^2 \delta_{sk}$. Similarly, applying Prop. 4.5 with Remark 4.6, we get

$$\Pr[\forall i : \hat{m}_i = m_i \vee \mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i] \geq 1 - \varepsilon_2$$

where $\varepsilon_2 := 40e^2(q_H + q_D + 2)^3 \delta_{sk}$. We conclude that the event

$$P := \big[\forall i : \hat{m}_i = m_i \ \vee \ (\hat{m}_i = \emptyset \ \wedge \ \mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i)\big]$$

holds except with probability $\varepsilon_1 + \varepsilon_2$. Thus

$$\Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 4}] \geq \Pr[m' = m^* \text{ in } \textbf{Game 4}] - (\varepsilon_1 + \varepsilon_2)\,.$$

In **Game 5**, we query $\mathcal{S}.E(c_i)$ *at runtime*, that is, as part of the DECAPS procedure upon input c_i. By 2.c of Theorem 4.3, each swap of a $\mathcal{S}.RO$ with a $\mathcal{S}.E$ query affects the final probability by at most $8\sqrt{2\Gamma(f)/|\mathcal{R}|} = 8\sqrt{2g_{sk}}$. Thus

$$\Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 5}] \geq \Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 4}] - \varepsilon_3$$

with $\varepsilon_3 := q_D \cdot (q_H + q_D) \cdot 8\sqrt{2g_{sk}}$.

In **Game 6**, DECAPS uses $\hat{m}_i$ instead of m_i to compute K_i. That is, it sets $K_i := \bot$ if $\hat{m}_i = \emptyset$ and $K_i := G^\diamond(\hat{m}_i)$ otherwise. Also, if instructed to output $m' := m_i$ where $c_i = c^\diamond$, then the output is set to $m' := \hat{m}_i$ instead. In all cases, DECAPS still queries $\mathcal{S}.RO(m_i)$, so that the interaction pattern between DECAPS and $\mathcal{S}.RO$ remains as in **Game 5**.

Here, we note that if the event

$$P_i := \big[\hat{m}_i = m_i \ \vee \ (\hat{m}_i = \emptyset \ \wedge \ \mathsf{Enc}_{pk}(m_i; \mathcal{S}.RO(m_i)) \neq c_i)\big]$$

holds for a given i then the above change will not affect DECAPS' response K_i, and thus also not the probability for P_{i+1} to hold as well. Therefore, by induction, $\Pr[P \text{ in } \textbf{Game 6}] = \Pr[P \text{ in } \textbf{Game 5}]$, and since conditioned on the event P the two games are identical, we have

$$\Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 6}] = \Pr[m' = m^* \ \wedge \ P \text{ in } \textbf{Game 5}].$$

In **Game 7**, instead of obtaining m' by measuring a random query of $\mathcal{A}$ to either $\mathcal{S}.RO$ or G, or outputting $\hat{m}_i$ with $c_i = c^\diamond$, here m' is obtained by measuring a random query of $\mathcal{A}$ to either $\mathcal{S}.RO$ or G, or outputting $\hat{m}_i$ for a *random* $i \in \{1, \ldots, q_D\}$, where the former case is chosen with probability $(q_H + q_G)/(q_H + q_G + 2q_D)$ and the latter with probability $2q_D/(q_H + q_G + 2q_D)$. Since conditioned on the first case being chosen or the latter with $i = i_\diamond$, **Game 7** coincides with **Game 6**, we have

$$\Pr[m' = m^* \text{ in } \textbf{Game 7}] \geq \frac{q_H + q_G + 2}{q_H + q_G + 2q_D} \cdot \Pr[m' = m^* \text{ in } \textbf{Game 6}]\,.$$

In **Game 8**, we observe that the response to the query $\mathcal{S}.RO(m^*)$, introduced in **Game 1** in order to compute $c^\diamond$, and the responses to the queries that DECAPS makes to $\mathcal{S}.RO$ on input m_i do not affect the game anymore, and thus we can drop all these queries, or, equivalently, move them to the very end of the execution of the game. Invoking once again 2.c of Theorem 4.3, we then get

$$\Pr[m' = m^* \text{ in } \textbf{Game 8}] \geq \Pr[m' = m^* \text{ in } \textbf{Game 7}] - \varepsilon_4\,,$$

for $\varepsilon_4 = (q_D + 1) \cdot q_H \cdot 8\sqrt{2g_{sk}}$.

With these queries now dropped, we observe that **Game 8** works without knowledge of the secret key sk, and thus constitutes a $\mathsf{OW\text{-}CPA}$ attacker $\mathcal{B}$ against PKE, which takes as input a public key pk and an encryption c^* of a random message $m^* \in \mathcal{M}$, and outputs m^* with the given probability, i.e, $\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} \geq \Pr[m' = m^* \text{ in } \textbf{Game 8}]$. We note that the oracle $G^\diamond$ can be simulated using standard techniques.

Backtracking all the above (in)equalities and setting $\varepsilon_{12} := \varepsilon_1 + \varepsilon_2$, $q_{HG} := q_H + q_G$ etc. and $q := q_H + q_G + 2q_D$, we get the following bound:

$$\begin{aligned}
\mathsf{ADV}_{sk}[\mathcal{A}]^{\mathsf{IND\text{-}CCA}}_{\mathrm{KEM}} &\leq 2(q_{HG} + 2)\sqrt{\frac{q_{HG} + 2q_D}{q_{HG} + 2}\big(\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} + \varepsilon_4\big) + \varepsilon_{12} + \varepsilon_3} \\
&\leq 2(q_{HG} + 2q_D)\sqrt{\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}} + \varepsilon_{34}} + 2(q_{HG} + 2)\sqrt{\varepsilon_{12}} \\
&\leq 2q\Big(\sqrt{\mathsf{ADV}_{sk}[\mathsf{B}]^{\mathsf{OW\text{-}CPA}}_{\mathrm{PKE}}} + \sqrt{\varepsilon_{34}} + \sqrt{\varepsilon_{12}}\Big)\,.
\end{aligned}$$

Additionally,

$$\sqrt{\varepsilon_{12}} = \sqrt{128 q_D q_{HD}^2 \delta_{sk} + 40e^2(q_{HD}+2)^3\delta_{sk}} \leq 42\sqrt{(q+2)^3\delta_{sk}}$$

and

$$\sqrt{\varepsilon_{34}} = \sqrt{8\big(q_D(q_H+q_D)+(q_D+1)q_H\big)\sqrt{2^{-(\gamma-1)}}} \leq 6\sqrt{q_H q_D}\cdot g_{sk}^{1/4}\,.$$

Taking the expectation over $(sk, pk) \leftarrow \mathsf{Gen}$ and applying Jensen's inequality, we get the claimed bound. Finally, we note that the runtime of $\mathcal{B}$ is given by $T_{\mathcal{B}} = T_{\mathcal{A}} + T_{\text{DECAPS}} + T_G + T_{\mathcal{S}}$, where apart from its oracle queries DECAPS runs in time linear in q_D, and $\mathcal{S}$ can be simulated in time

$$T_{\mathcal{S}} = O\big(q_{RO}\cdot q_E \cdot \mathrm{Time}[f] + q_{RO}^2\big) = O\big(q_H \cdot q_D \cdot \mathrm{Time}[\mathsf{Enc}] + q^2\big)$$

by Theorem 4.3, and similarly for G. □

7 Acknowledgement

The authors thank Andreas Hülsing and Kathrin Hövelmanns for helpful discussions. JD was funded by ERC-ADG project 740972 (ALGSTRONGCRYPTO). SF was partly supported by the EU Horizon 2020 Research and Innovation Program Grant 780701 (PROMETHEUS). CM was funded by a NWO VENI grant (Project No. VI.Veni.192.159). CS was supported by a NWO VIDI grant (Project No. 639.022.519).

References

ABG$^+$20. Amit Agarwal, James Bartusek, Vipul Goyal, Dakshita Khurana, and Giulio Malavolta. Post-quantum multi-party computation, 2020.

AHU19. Andris Ambainis, Mike Hamburg, and Dominique Unruh. Quantum security proofs using semi-classical oracles. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 269–295, Cham, 2019. Springer International Publishing.

AMRS20. Gorjan Alagic, Christian Majenz, Alexander Russell, and Fang Song. Quantum-access-secure message authentication via blind-unforgeability. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020*, pages 788–817, Cham, 2020. Springer International Publishing.

BCC88. Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156 – 189, 1988.

BDF$^+$11. Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer.

BHH$^+$19. Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. Tighter proofs of cca security in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography*, pages 61–90, Cham, 2019. Springer International Publishing.

BHT98. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Cláudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN'98: Theoretical Informatics*, pages 163–169, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

CDG$^+$17. Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 1825–1842, New York, NY, USA, 2017. ACM.

CDG$^+$19. Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, and Greg Zaverucha. The picnic signature scheme, design document v2.1, 2019.

CFHL20. Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. Cryptology ePrint Archive, Report 2020/1305, 2020. https://eprint.iacr.org/2020/1305, to appear in *Advances in Cryptology – EUROCRYPT 2021*.

Cha19. André Chailloux. Tight quantum security of the Fiat-Shamir transform for commit-and-open identification schemes with applications to post-quantum signature schemes. Cryptology ePrint Archive, Report 2019/699, 2019. https://eprint.iacr.org/2019/699.

CHR+16. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass mq-based identification to mq-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 135–165, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

CMS19. Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *Theory of Cryptography*, pages 1–29, Cham, 2019. Springer International Publishing.

CMSZ19. Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. Cryptology ePrint Archive, Report 2019/428, 2019. https://eprint.iacr.org/2019/428.

DFM20. Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round Fiat-Shamir and more. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 602–631, Cham, 2020. Springer International Publishing.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 356–383, Cham, 2019. Springer International Publishing.

DKL+18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018(1):238–268, Feb. 2018.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, pages 53–68, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

GHHM20. Alex B. Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the QROM. Cryptology ePrint Archive, Report 2020/1361, 2020. https://eprint.iacr.org/2020/1361.

GMW91. Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, July 1991.

HHK17. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin, editors, *Theory of Cryptography*, pages 341–371, Cham, 2017. Springer International Publishing.

HM20. Yassine Hamoudi and Frédéric Magniez. Quantum time-space tradeoff for finding multiple collision pairs, 2020.

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 552–586, Cham, 2018. Springer.

KZ20. Daniel Kales and Greg Zaverucha. Improving the performance of the picnic signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 154–188, 2020.

LZ19a. Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*, pages 189–218, Cham, 2019. Springer International Publishing.

LZ19b. Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 326–355, Cham, 2019. Springer International Publishing.

NC11. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

Pas03. Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 316–337, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

Pas04. Rafael Pass. *Alternative variants of zero-knowledge proofs*. PhD thesis, KTH Stockholm, 2004.

TU16. Ehsan Ebrahimi Targhi and Dominique Unruh. Post-quantum security of the Fujisaki-Okamoto and OAEP transforms. In Martin Hirt and Adam Smith, editors, *Theory of Cryptography*, pages 192–216, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

Unr12. Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 135–152, Berlin, Heidelberg, 2012. Springer.

Unr14. Dominique Unruh. Revocable quantum timed-release encryption. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 129–146, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Zha15. Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7-8):557–567, 2015.

Zha19. Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 239–268, Cham, 2019. Springer International Publishing.

Appendix

A Efficient representation of the compressed oracle.

By the techniques of [Zha19], it is possible to make the (considered variant of the) compressed oracle efficient. Concretely, by means of a suitable encoding, it is possible to *efficiently* maintain the quantum state of the register D of the compressed oracle, compute the unitary O_{XYD}, and extract information from the state of D. We briefly describe this procedure below.

Writing $\bar{\mathcal{Y}} = \{0,1\}^n \cup \{\perp\}$, consider the following standard sparse encoding scheme

$$\mathsf{SparseEnc}^q : \bar{\mathcal{Y}}^{\mathcal{X}} \to \mathcal{D} = (\mathcal{X} \times \bar{\mathcal{Y}})^q\,,$$

which maps any "database" $\mathbf{y} = (y_x)_{x\in\mathcal{X}}$ with at most q non-$\perp$ entries to the "compressed database"

$$\mathsf{SparseEnc}^q(\mathbf{y}) = \big((x_1, y_{x_1}), \ldots, (x_s, y_{x_s}), (0,\perp), \ldots, (0,\perp)\big)$$

of pairs (x, y_x) with $y_x \neq \perp$, sorted as $x_1 < \cdots < x_s$, and padded with $(0,\perp)$s. Naturally, we then set

$$\big|\mathsf{SparseEnc}^q(\mathbf{y})\big\rangle = |x_1\rangle|y_{x_1}\rangle\cdots|x_s\rangle|y_{x_s}\rangle|0\rangle|\perp\rangle\cdots|0\rangle|\perp\rangle \in \big(\mathbb{C}[\mathcal{X}]\otimes\mathbb{C}[\bar{\mathcal{Y}}]\big)^{\otimes q}$$

for any such $\mathbf{y}$. The crucial observations now are:

1. Using the representation $H^{\otimes|\mathcal{X}|}|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ for the state of register D after q queries, the evolution of the compressed oracle, given by O_{XYD}, is an efficiently quantum computable isometry (this was shown by Zhandry, but is also easy to see from scratch). Here and below, H is the Walsh-Hadamard transform on $\mathbb{C}[\{0,1\}^n] = (\mathbb{C}^2)^{\otimes n}$, extended to act as identity on $|\perp\rangle$.
2. Using the representation $|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ instead, it follows from basic theory of quantum computation that for any classical function f with domain $\bar{\mathcal{Y}}^{\mathcal{X}}$ and that is classically efficiently computable using the representation $\mathbf{y} \mapsto \mathsf{SparseEnc}^q(\mathbf{y})$, the unitary $U : |\mathbf{y}\rangle|z\rangle \mapsto |\mathbf{y}\rangle|z+f(\mathbf{y})\rangle$ is efficiently quantum computable.
3. $|\mathbf{y}\rangle \mapsto |\mathsf{SparseEnc}^q(\mathbf{y})\rangle$ commutes with applying Walsh-Hadamards to the $\mathbb{C}[\bar{\mathcal{Y}}]$-components. Therefore, one can efficiently switch between the two representations above, simply by applying $H^{\otimes q}$ to the corresponding registers of $|\mathsf{SparseEnc}^q(\mathbf{y})\rangle$.

Thus, using either of the two representations for representing the internal state of the oracle, both the evolution of the oracle and the typical unitaries or measurements used to "read out" information are efficiently quantum computable. For example, checking if $y_x = \perp$ for a given $x \in \mathcal{X}$, or if there exists $x \in \mathcal{X}$ for which x and y_x satisfy some given (efficiently computable) relation, etc. Formally:

Lemma A.1. *Let $f : (\{0,1\}^n \cup \{\perp\})^{|\mathcal{X}|} \to \mathcal{T}$ be a function such that $\tilde{f} = f \circ \mathsf{SparseDec}^q$ can be computed in polynomial time in q. Then the measurement $\{\tilde{\Pi}^t\}_{t\in\mathcal{T}}$ given by the projections*

$$\tilde{\Pi}^t = \sum_{\mathbf{y}:\tilde{f}(\mathbf{y})=t} |\mathbf{y}\rangle\langle\mathbf{y}|$$

can be implemented in time linear in $\mathrm{Time}[\tilde{f}]$ *and thus in quantum polynomial time in q.*

B Supplementary proofs

B.1 Proof of Lemma 3.3

Recalling from (6) that $F|y\rangle = |y\rangle + 2^{-n/2}|\delta\rangle$ with $|\delta\rangle := |\perp\rangle - |\phi_0\rangle$, we have

$$[F, |y\rangle\langle y|] = F|y\rangle\langle y| - |y\rangle\langle y|F = 2^{-n/2}|\delta\rangle\langle y| - 2^{-n/2}|y\rangle\langle\delta|\,.$$

From this, it follows that

$$[F, \Pi^x] = \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} [F, |y\rangle\langle y|] \le 2^{-n/2}\, |\delta\rangle \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} \langle y| - 2^{-n/2} \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} |y\rangle\langle\delta|$$

and thus, using (1), that

$$\|[F, \Pi^x]\| \le 2^{-n/2}\, \||\delta\rangle\| \left\| \sum_{\substack{y\in\{0,1\}^n \\ (x,y)\in R}} \langle y| \right\| \le 2^{-n/2}\sqrt{2}\sqrt{\Gamma_x}\,.$$

For the second bound, let $C_{YD_x} = \mathrm{CNOT}$ with CNOT as in (7), with the understanding that D_x is the control register and Y the target. Recall from (7) that $O^x_{YD_x} = F_{D_x} C_{YD_x} F_{D_x}$. Thus, using (3) twice and omitting the registers, we obtain

$$[O^x, \Pi^x] = F[CF, \Pi^x] + [F, \Pi^x]CF = FC[F, \Pi^x] + F[C, \Pi^x]F + [F, \Pi^x]CF\,.$$

Finally, we notice that $[C_{YD_x}, \Pi^x_{D_x}] = 0$, since projections on the control register of a CNOT commute with the CNOT. The claimed bound now follows from the derived bound on $[F, \Pi^x]$ together with Equation (4).

The third bound follows by recalling that $\Pi^{\emptyset}_D = \bigotimes_{x'} \bar{\Pi}^{x'}_{D_{x'}}$ is a tensor-product for which $O^x_{YD_x}$ acts trivially on all the components except for the component $\bar{\Pi}^x_{D_x}$, so with Equation (4) we obtain,

$$\|[O^x_{YD_x}, \Pi^{\emptyset}_D]\| \le \|[O^x_{YD_x}, \bar{\Pi}^x_{D_x}]\| = \|[O^x_{YD_x}, \Pi^x_{D_x}]\|\,,$$

which completes the proof. □

B.2 Proof of Proposition 4.5

The left circuit in Fig. 9 defines (the distribution of) the considered variables $x, \hat{x}, h, t$. We also consider the circuit that applies the measurement $\{\Pi^{col}, \Pi^{\neg col}\}$ instead of $\mathcal{M}^t$, where Π^{col} is as in Lemma C.1 and $\Pi^{\neg col} = \mathbb{1} - \Pi^{col}$ (Fig. 9, middle). Since the projections defining either measurement are all diagonal in the basis $\{|\mathbf{y}\rangle\}$, we may equivalently measure register D in that basis (Fig. 9, right), and then set $\hat{x}$ to be the smallest element $\mathcal{X}$ so that $f(\hat{x}, y_{\hat{x}}) = t$ (with $\hat{x} = \emptyset$ if no such element exists) and consider the event *col* given by $\exists\, x' \neq x'' : f(x', y_{x'}) = f(x'', y_{x''})$. By the respective definitions of $\mathcal{M}^t$ and Π^{col}, both, the variables $\hat{x}, x, h, t$ and the event and variable *col* and x, h, t then have the same distributions as in the respective original two games. But now, we can consider their joint distribution and argue that

$$\Pr[\hat{x} \neq x \wedge f(x,h) = t] \le \Pr[\hat{x} \neq x \,|\, f(x,h) = t \wedge \neg col] + \Pr[col]\,.$$

We now observe that right before the considered measurement, by definition of O, the state of D is supported by vectors $|\mathbf{y}\rangle$ with $y_x = h$, and so the measurement outcome $\mathbf{y}$ satisfies $y_x = h$. Therefore, the first term vanishes by definition of *col* and $\hat{x}$, while $\Pr[col]$ is bounded by $40e^2(q+2)^3\Gamma'(f)/2^n$, using Lemma C.1. □

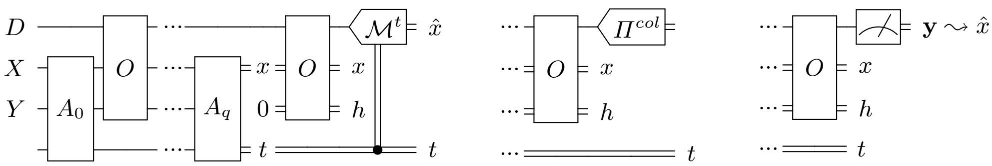

Fig. 9. Quantum circuit diagrams for the experiments in the proof of Prop. 4.5.

C Hardness of collision finding

The following can be easily extracted from the derivation of the general collision-finding bound Theorem 5.29 from [CFHL20]. It expresses that, for any algorithm with bounded query complexity, it is unlikely that one encounters a collision within the superposition oracle.

Lemma C.1. *Let $f : \mathcal{X} \times \{0,1\}^n \to \mathcal{T}$, and let Π^{col} be the projection into the space spanned by $|\mathbf{y}\rangle \in \mathcal{H}_D$ for $\mathbf{y} = (y_x)_{x\in\mathcal{X}} \in (\mathcal{Y} \cup \{\perp\})^{\mathcal{X}}$ such that there exist $x \neq x'$ with $y_x, y_{x'} \neq \perp$ and $f(x, y_x) = f(x', y_{x'})$. Then, for any oracle algorithm $\mathcal{A}$ with query complexity q, at the end of the execution the state ρ of the compressed oracle is such that*

$$\mathrm{tr}(\Pi^{col}\rho) \leq 40e^2q^2(q+1)\Gamma'(f)/2^n \,,$$

where $\Gamma'(f) = \max_{x\neq x',y'} |\{y \mid f(x,y) = f(x',y')\}|$ and $e \approx 2.718$ is Euler's number.

D Application to Fiat Shamir Signatures

Σ-protocols are commonly used to obtain non-interactive zero-knowledge proofs and digital signatures via the Fiat Shamir (FS) transform. Here, the random challenges are (possibly after a suitable number of parallel repetitions) replaced by the hash of the first message in the 3-round protocol, thus making the protocol non-interactive. To construct a digital signature scheme (DSS), the message to be signed is included in the hash argument.[13]

The post-quantum security of FS signatures has recently drawn additional attention. This is mainly because FS signatures are some of the most promising candidates for replacing RSA and elliptic curve signatures which can be broken by quantum adversaries. Indeed, two out of the 6 round-3 candidate DSSs in the NIST standardization process for post-quantum cryptographic schemes, CRYSTALS Dilithium [DKL+18] and Picnic [CDG+17], are FS signature schemes. In the QROM,[14] the chain of arguments for reducing the UF-CMA security of a FS signature scheme $\mathsf{Sig}[\Sigma]$ to the i) honest-verifier zero-knowledge, and ii) (some variant of the) special soundness, properties of the underlying Σ-protocol Σ as follows (also depicted in Fig. 10).

- First, the UF-CMA security of $\mathsf{Sig}[\Sigma]$ is reduced to plain unforgeability (UF-NMA), using the HVZK property of Σ [KLS18, GHHM20].
- The UF-NMA property of $\mathsf{Sig}[\Sigma]$ follows from the extractability of the Fiat Shamir transformation $\mathsf{FS}[\Sigma]$ of Σ.
- The extractability of $\mathsf{FS}[\Sigma]$ is then reduced to the extractability of Σ [DFMS19, LZ19b, DFM20].
- Finally, the extractability of Σ is reduced to the (variant of) special soundness of Σ [Unr12].

UF-CMA of $\mathsf{Sig}[\Sigma]$	$\Leftarrow$	UF-NMA of $\mathsf{Sig}[\Sigma]$	$\Leftarrow$	Extractability of $\mathsf{FS}[\Sigma]$	$\Leftarrow$	Extractability of Σ	$\Leftarrow$	Spec. soundness of Σ

Fig. 10. Chain of arguments for proving security of FS signatures.

Prior to this work, the last step (arguing extractability from special soundness) has relied on Unruh's rewinding lemma [Unr12], which leads, e.g., after suitable generalization, to a $2k+1$-th root loss for a k-sound Σ. For commit-and-open Σ-protocols, Theorem 5.4 can replace Unruhs rewinding lemma for commit-and-open protocols in the QROM, making the last step above tight up to unavoidable additive errors.

As an example, Theorem 5.4 implies a sizeable improvement over the current best QROM security proof of Picnic2 [CDG+17, KZ20, CDG+19]. Indeed, Unruh's rewinding lemma implies a

[13] For FS DSS, the relation R needs to admit an efficient generator of hard instances.
[14] The typical ROM reductions proceed similarly

6-th root loss for the variant of special soundness the underlying Σ-protocol possesses [DFMS19], while Theorem 5.4 is tight.

We note that for commit-and-open Σ-protocols, there is hope for further improvements by means of combining the last two steps and doing a *direct* analysis of $\mathsf{FS}[\Sigma]$. Indeed, [Cha19] suggests such an approach, but the analysis provided there there still relies on some unproven assumption.

E Public-Key Encryption and Key Encapsulation

Following the presentation of [HHK17] in general lines, we recall the formal definition of a public-key encryption scheme.

Definition E.1 (Public-Key Encryption). *A* public-key encryption scheme PKE *consists of algorithms* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$*, a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$ and a set of random coins $\mathcal{R}$, such that for any $m \in \mathcal{M}$, $r \in \mathcal{R}$*

$$(sk, pk) \leftarrow \mathsf{Gen}\,, \quad \mathcal{C} \ni c \leftarrow \mathsf{Enc}_{pk}(m; r) \quad \textit{and} \quad \mathsf{Dec}_{sk}(c) \in \mathcal{M} \cup \{\bot\}\,.$$

For a given public-key encryption scheme, it may be useful to consider the probability of encountering decryption failures.

Definition E.2 (δ-correctness). *A public-key encryption scheme is δ-correct if*

$$\mathop{\mathbb{E}}_{(sk,pk)\leftarrow\mathsf{Gen}}\left[\max_{m\in\mathcal{M}} \Pr\big[\mathsf{Dec}_{sk}(c) \neq m : c \leftarrow \mathsf{Enc}_{pk}(m)\big]\right] \leq \delta$$

where the probability is over the randomness of the encryption.

Another important property of encryption schemes is the min-entropy of a ciphertext given the plaintext, measured by their *γ-spreadness*.

Definition E.3 (γ-spreadness). *A public-key encryption scheme is γ-spread if*

$$\min_{\substack{m\in\mathcal{M}\\(sk,pk)}} \left(-\log \max_{c\in\mathcal{C}} \Pr\big[c = \mathsf{Enc}_{pk}(m)\big]\right) \geq \gamma\,,$$

where the probability is over the randomness of the encryption, and the minimum is over all key pairs that have positive probability of being produced by Gen*.*

The above definition can be relaxed to an *expectation* over the choice of pk, when the expectation is done inside the negative logarithm.

Definition E.4 (weak γ-spreadness). *A public-key encryption scheme is weakly γ-spread if*

$$-\log \mathop{\mathbb{E}}_{(sk,pk)\leftarrow\mathsf{Gen}}\left[\max_{\substack{m\in\mathcal{M}\\c\in\mathcal{C}}} \Pr\big[c = \mathsf{Enc}_{pk}(m)\big]\right] \geq \gamma\,,$$

where again the probability is over the randomness of the encryption.

A key-encapsulation mechanism (KEM) is defined as follows:

Definition E.5 (Key Encapsulation Mechanism). *A* key encapsulation mechanism KEM *consists of algorithms* $(\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ *and a key space $\mathcal{K}$, where*

$$(sk, pk) \leftarrow \mathsf{Gen}\,, \quad (K, c) \leftarrow \mathsf{Encaps}(pk) \quad \textit{and} \quad \mathsf{Decaps}_{sk}(c) \in \mathcal{K} \cup \{\bot\}\,.$$