# Quantum security proofs using semi-classical oracles

Andris Ambainis
University of Latvia

Mike Hamburg
Rambus Security Division

Dominique Unruh
University of Tartu

February 18, 2019

**Abstract.** We present an improved version of the one-way to hiding (O2H) Theorem by Unruh, J ACM 2015. Our new O2H Theorem gives higher flexibility (arbitrary joint distributions of oracles and inputs, multiple reprogrammed points) as well as tighter bounds (removing square-root factors, taking parallelism into account). The improved O2H Theorem makes use of a new variant of quantum oracles, semi-classical oracles, where queries are partially measured. The new O2H Theorem allows us to get better security bounds in several public-key encryption schemes.

## Contents

# 1 Introduction

Ever since it was first introduced in [BR93] as a proof technique for cryptographic proofs, the random oracle model has been widely used to analyze cryptographic schemes, especially when highly efficient, practical solutions are desired. In the post-quantum setting, however, we need to be careful how the random oracle is modeled. When the adversary makes a query, the input to the random oracle should not be measured [BDF+11]. That is, queries should be possible in superposition between different inputs (we then speak of a "quantum random oracle"). Otherwise, the random oracle model would be a very unrealistic idealization of the real world since a quantum adversary can evaluate, say, a hash function in superposition.

Unfortunately, proving the security in the quantum random oracle model is considerably more difficult than in the classical random oracle model. One example of a classical proof technique that is not easy to mimic is programming of the random oracle. In this technique, we run the adversary with access to a random oracle but we change the answer to certain queries during the execution. In a nutshell, as long as we can show that the probability of changing a value that the adversary has already queried is negligible, the adversary will not notice the programming, and the proof goes through. In the quantum setting, this does not make sense. The adversary could query the superposition of all inputs in its first query. Then any programming would change a value that has already been queried.

A technique that can solve this problem (at least in certain situations) is the One-Way to Hiding (O2H) Theorem from [Unr15b]. The O2H Theorem solves the reprogramming problem by showing, roughly speaking, that we can bound the probability that the adversary distinguishes between two oracles $G$ and $H$ (the original and the reprogrammed oracle) in terms of the probability that the adversary can guess the location where the oracle is reprogrammed (we speak of the "guessing game"). This conceptually simple theorem has proven powerful in a number of security proofs for post-quantum secure encryption schemes and other constructions (see our overview in Section 1.2). However, the O2H Theorem has a number of limitations that limit its applicability, or give bad bounds in concrete security proofs.

In this work, we present a new version of the O2H Theorem that improves on the state of the art in a number of aspects:

- **Non-uniform random oracles.** The random oracle that is reprogrammed does not have to be a uniformly random function. We allow any distribution of oracles, e.g., invertible permutations, ideal ciphers, etc.

- **Multiple reprogrammed points.** We can reprogram the oracle in more than a single point. That is, we can reprogram the random oracle at a set of positions $S$ and then bound the probability that the adversary detects this reprogramming with a single application of the O2H Theorem.

- **Arbitrary joint distributions.** We allow the distribution of reprogrammed locations and of the adversary's input to be arbitrarily correlated with the distribution

of the random oracle. This is especially important if the reprogrammed location depends on the random oracle (e.g., reprogramming $H(x)$ where $x := H(r)$ for random $r$).

- **Tighter bounds for guessing games.** Our O2H Theorem bounds the difference of the square-roots of the adversary probabilities between two games. In many cases involving guessing games (i.e., where we intend to show that the probability of a certain event is negligible) this leads to bounds that are quadratically better.

- **Tighter bounds using semi-classical oracles.** We introduce a new technique, called semi-classical oracles. By applying the O2H Theorem to games involving semi-classical oracles, we can again get better bounds in some cases. (Whether some advantage is gained depends very much on the specific proof in which the O2H Theorem is used.)

- **Query depth.** Our O2H Theorem distinguishes query number $q$ and query depth $d$. Thus, for cases in which the adversary has a high parallelism, we get better bounds (and for sequential adversaries nothing is lost by setting $d := q$).

One crucial novelty in our O2H Theorem is the use of "semi-classical oracles". In a nutshell, a semi-classical oracle is an oracle that only measures whether the adversary queried a given "forbidden" input, but does not measure anything beyond that. (In contrast, a quantum oracle does not measure anything, and a classical oracle measures everything.) So, for example, if the adversary queries a superposition of non-measured inputs, nothing is measured.

Our O2H Theorem bounds the distinguishing probability between two oracles $G$ and $H$ again in terms of the success probability in a "guessing game" where the adversary has to query an oracle on one of the forbidden inputs on which $G$ and $H$ differ. But in contrast to the original O2H Theorem, the adversary is given a semi-classical oracle in the guessing game! (In the original O2H Theorem, the adversary is given a quantum oracle.) Using a semi-classical oracle, the guessing game can be expressed more simply since it is well-defined whether the forbidden input has been queried or not. (In the original O2H Theorem, we instead have to stop at a random query and measure whether that particular query queries the forbidden input. This makes the description of the game more complex, and the random selection of a single query is the reason why the original O2H Theorem gives worse bounds.)

We stress that the semi-classical oracles are purely a proof technique and occur in intermediate games in proofs involving the new O2H Theorem. The final security results still hold in the quantum random oracle model, not in some "semi-classical random oracle model".

In this work, we introduce semi-classical oracles, state and prove the new O2H Theorem (together with a query complexity result about searching in semi-classical oracles), and demonstrate its usefulness by elementary examples and by exploring the impact on the security bounds of existing encryption schemes.

**Organization.** In Section 1.1 we shortly discuss some related work, and in Section 1.2 we discuss the impact of our result on existing cryptographic schemes. Section 2 presents basic notation. Our notion of semi-classical oracles is introduced in Section 3. We also state our main theorems in Section 3, the proofs are deferred to Section 5 (after the examples). We present examples how to use the new technique in Section 4. In Appendix A, we give a proof of optimality of one of our results. In Appendix B, we revisit the security proof from [TU16] for the security of a variant of Fujisaki-Okamoto and show how to rework it with our new technique (with better bounds).

### 1.1 Related work

**Variants of the O2H Theorem.** Variants of the O2H Theorem were introduced in [Unr15b, Unr14, Unr15a, JZC+18, Eat17], see the beginning of Section 1.2 for more details.

**Other proof techniques for the quantum random oracle model.** [BBHT98] showed that Grover search is optimal with respect to worst-case complexity ([Zal99] when parallelism is considered). [Unr15a, HRS16] generalized this to the average-case which implies that finding preimages of the random oracle is hard. [BDF+11] introduced "history-free reductions" which basically amounts to replacing the random oracle by a different function right from the start. [Zha12b] showed that random oracles can be simulated using $2q$-wise independent functions. Based on this, [Unr15a] introduces a technique for extracting preimages of the random oracle. [Zha12b] introduces the "semi-constant distributions" technique that allows us to program the random oracle in many random locations with a challenge value without the adversary noticing. [Zha12a] improves on this with the "small-range distribution" technique that allows us to simulate random oracles using random looking functions with a small range. [Zha15] shows that random oracles are indistinguishable from random permutations, and as a consequence that random oracles are collision resistant (this is generalized by [TTU16, EU18, BES18] to the case of non-uniformly distributed functions). Collision-resistance of the random oracle is generalized to the "collapsing property" which allows us to show that measuring the output of the random oracle effectively measures the input. More general methods for problems in quantum query complexity (not limited to random oracles) include the polynomial method [BBC+01] and the adversary method [Amb02]. [ARU14] shows that the difficulties of using the quantum random oracle are not just a matter of missing proof techniques, but that in certain cases classically secure schemes are not secure in the quantum random oracle model.

**Cryptosystems whose security proof is based on O2H Theorems.** See Section 1.2.

### 1.2 Impact on existing cryptosystems

Above, we explained why our new O2H Theorem can lead to better bounds. We will also illustrate that point with a few simple examples in Section 4. However, to better judge the impact on realistic cryptosystems, we need to ask the question how the bounds achieved by existing security proofs improve.

We are aware of the following results in the quantum random oracle model that employ some variant of the original O2H Theorem from [Unr15b]: [Unr15b] introduced the O2H Theorem to build revocable timed-release encryption schemes, [Unr14] introduced an "adaptive" version of the O2H Theorem[1] to analyze a quantum position verification protocol, [Unr15a] made the O2H Theorem even more adaptive and used this for the design of non-interactive zero-knowledge proof systems and signature schemes (and this in turn is the basis for various follow-up schemes such as [YAJ+17, GPS17, CDG+17, DRS18, CHR+18, BEF18]). [Unr17] uses the O2H variant from [Unr15a] to prove security of Fiat-Shamir [FS87], both as a proof system and as a signature scheme. [Eat17] uses a variant of the O2H Theorem for proving security of Leighton-Micali signatures [LM95] (their variant generalizes [Unr15b] in some aspects but only works when the position where the oracle is programmed is information-theoretically hidden). [SY17] uses the O2H Theorem for constructing PRFs and MACs. [TU16] was the first paper to employ the O2H Theorem for designing public key encryption schemes: it proved the security of variants of the Fujisaki-Okamoto transform [FO13] and the OAEP transform [BR95] (introducing one extra hash value in the ciphertext for "key confirmation"). [HHK17] modularized and improved the Fujisaki-Okamoto variant from [TU16], also using key confirmation. [SXY18] proved security of a construction without key confirmation, still using the O2H Theorem. [JZC+18] introduced a variant of the O2H Theorem that allows some of the oracles and inputs given to the adversary to be non-uniformly distributed, subject to the independence of certain random variables, and uses it to prove the security of further public-key encryption schemes. Unfortunately, the precise meaning of the independence requirement in [JZC+18] is unclear, it might be unsatisfiable.[2] (Since our O2H Theorem can also handle non-uniform inputs, it might be that it can serve as a drop-in replacement in the proofs in [JZC+18].) [JZM19] proves security of public-key encryption schemes with explicit rejection; an earlier version [JZM] of [JZM19] used the O2H Theorem from [JZC+18], the current version uses our new O2H Theorems. [HKSU18] analyzes public-key encryption and authenticated key exchange schemes, using the original O2H Theorem from [Unr15b] in the first revision, but improving the bounds using our new O2H Theorem.

Thus, O2H Theorems might be one of the most widely used proof technique for

[1] Which allows to reprogram the random oracle at a location that is influenced by the adversary.

[2] The requirement is that $x$ is uniformly distributed given $\mathcal{O}(x')$ for all $x' \neq x$. The formal meaning of this is hard to pin down because the requirement says that $x$ is supposed to be uniform given a set of random variables (namely $\{\mathcal{O}(x')\}_{x' \neq x}$) where the choice which random variables are in that set depends in turn on $x$, but $x$ is a random variable itself and thus has no fixed value. We can formalize the requirement as "$x$ is uniform given $\mathcal{O}(x := \bot)$" (i.e., we remove the point $x$ from $\mathcal{O}$). But $x$ cannot be uniform given $\mathcal{O}(x := \bot)$ since $\mathcal{O}(x := \bot)$ determines $x$. So the conditions in the O2H variant from [JZC+18] may be unsatisfiable.

| Setting | Bound | Queries |
|---|---|---|
| **Targhi-Unruh [TU16]** | | |
| **old** O2H, one-way | $\varepsilon_{sym} + q^{9/5}2^{-\gamma/5} + q^{3/2}\varepsilon^{1/4} + q^{3/2}2^{-n_1/4}$ | $q^6 \approx 1/\varepsilon$ |
| **new** O2H, IND-CPA | $\varepsilon_{sym} + q^{9/5}2^{-\gamma/5} + qq_{dec}^{1/2}\varepsilon^{1/2} + q^{3/2}q_{dec}2^{-n/2}$ | $q^2q_{dec} \approx 1/\varepsilon$ |
| **new** O2H, one-way | $\varepsilon_{sym} + q^{9/5}2^{-\gamma/5} + q^{3/2}q_{dec}\varepsilon^{1/2}$ | $q^3q_{dec}^2 \approx 1/\varepsilon$ |
| **Hövelmanns-Kiltz-Schäge-Unruh [HKSU18]** | | |
| **old** O2H, IND-CPA | $q\varepsilon^{1/2} + q2^{-n/2}$ | $q^2 \approx 1/\varepsilon$ |
| **new** O2H, IND-CPA | $q^{1/2}\varepsilon^{1/2} + q2^{-n/2}$ | $q \approx 1/\varepsilon$ |
| **Jiang-Zhang-Ma [JZM19]** | | |
| **old** O2H, one-way | $q\varepsilon^{1/2}$ | $q^2 \approx 1/\varepsilon$ |
| **new** O2H, one-way | $q\varepsilon^{1/2}$ | $q^2 \approx 1/\varepsilon$ |
| **new** O2H, IND-CPA | $q^{1/2}\varepsilon^{1/2}$ | $q \approx 1/\varepsilon$ |

The "setting" column says whether the proof uses the old/new O2H and whether it is based on one-wayness or IND-CPA security of the underlying public-key encryption scheme.
The "bound" column gives the bound on the advantage of the adversary against IND-CCA security, up to constant factor. (In the case of [TU16] a hybrid public-key encryption scheme is constructed, in the other cases a KEM.) $\varepsilon$ is the advantage of the reduced adversary against the one-wayness or IND-CPA security of the underlying public-key scheme, respectively. (A complete description would contain the runtime of that adversary. For this overview this is not relevant since in all cases, that runtime did not change when switching to the new O2H Theorem.) $\varepsilon_{sym}$ is the advantage against the underlying symmetric encryption scheme. $q$ is the number of queries (random oracle + decryption queries), $q_{dec}$ only the decryption queries. $\gamma$ is the min-entropy of ciphertexts and $n$ the plaintext length of the underlying public-key scheme.
The "queries" column summarizes the effect of queries compared to the security of the underlying public-key scheme (see the explanation in the text, higher exponent is worse).
For simplicity, we give the bounds for the case where no decryption errors occur.

Figure 1: Security bounds of different Fujisaki-Okamoto variants with new and old O2H Theorems.

cryptosystems involving quantum random oracles. We expect that our improvement of the O2H Theorem allows us to derive better security bounds for most of the above schemes. To give some evidence to this hypothesis, we report on the advantages gained by using our improvement in three of the works above, namely Targhi-Unruh [TU16], Hövelmanns-Kiltz-Schäge-Unruh [HKSU18], and Jiang-Zhang-Ma [JZM19].

In case of [JZM19], an earlier draft [JZM] used the O2H variant from [JZC+18], while the current version [JZM19] already uses our new O2H Theorem. Since the O2H variant from [JZC+18] was introduced to handle the case where not all oracles and adversary inputs are independent, this demonstrates that our O2H Theorem can handle this case, too. (Besides giving tighter bounds.) Similarly, the first eprint version of [HKSU18] used the original O2H Theorem from [Unr15b], while the second version was updated to use our new O2H Theorem.

The old and new bounds are are summarized in Figure 1. The figure lists the advantages against IND-CCA security for different settings. Since it is difficult to compare the various formulas, in the column "queries", we summarize the relationship

between query number and attack probability: Assuming that the terms involving $\varepsilon$, the advantage against the underlying public-key encryption scheme, dominate all other terms, how many queries does one have to make to break the scheme (with constant probability)? E.g., given an advantage $q\sqrt{\varepsilon}$, we need $q \approx \varepsilon^{-1/2}$ queries for a successful attack, so we write $q^2 \approx 1/\varepsilon$ in that case.

Furthermore, in Appendix B, we reprove the security of the Fujisaki-Okamoto variant from [TU16] using our O2H Theorem. That result is particularly interesting because of its heavy use of the O2H Theorem. This allows us to make use of several of the new features of our O2H Theorem.

- It uses "nested invocations" of the O2H Theorem. That is, first the O2H Theorem is applied as usual to a pair of games, leading to a guessing game in which we need to show that the guessing probability $P_{\text{guess}}$ of the adversary is negligible. But then the O2H Theorem is applied again to prove this. Since the bound obtained by the O2H Theorem contains a square root over $P_{\text{guess}}$, the nested application of the O2H Theorem introduces nested square roots, i.e., a forth root. This leads to a particularly bad bound in [TU16].
  In contrast, our new O2H Theorem allows us to directly bound the difference of the square roots of the success probabilities of the adversary in two games. This means that in a nested invocation, when we analyze $P_{\text{guess}}$, the O2H Theorem directly tells us how $\sqrt{P_{\text{guess}}}$ changes (instead of how $P_{\text{guess}}$ changes). This avoids the nested square root.
- It uses the adaptive version of the O2H Theorem (from [Unr14]). While our O2H Theorem is not adaptive (in the sense that the input where the oracle is reprogrammed has to be fixed at the beginning of the game), it turns out that in the present case our new O2H Theorem can replace the adaptive one. This is because our new O2H Theorem allows us to reprogram the oracle at a large number of inputs (not just a single one). It turns out we do not need to adaptively choose the one input to reprogram, we just reprogram all potential inputs. At least in the proof from [TU16], this works without problems.

We restate the proof from [TU16] both under the assumption that the underlying public-key encryption scheme is one-way and under the assumption that it is IND-CPA secure. While in the original proof, we get essentially the same bound no matter which of the two assumptions we use, with the new O2H Theorem, the resulting bounds are much better when using IND-CPA security (but there is also an improvement in the one-way case).

The resulting bounds are given in Figure 1 as well. We see that the biggest improvement is in the case of IND-CPA security, where the dependence on the query number changed from the sixth power to cubic.

We also noticed a mistake in the proof,[3] which we fixed in our proof. (We do not know if the fix carries over to the original proof.)

But our analysis also shows some potential for future research on the O2H Theorem.

[3] In Game 7 in [TU16], a secret $\delta^*$ is encrypted using a one-time secure encryption scheme, and the final step in the proof concludes that therefore $\delta^*$ cannot be guessed. However, Game 7 contains an oracle $Dec^{**}$ that in turn accesses $\delta^*$ directly, invalidating that argument.

The proof from [TU16] constructs a plaintext extractor $Dec^{**}$ that is relatively inefficient because it iterates through a large number of possible candidate keys. Thus the number of oracle queries performed by $Dec^{**}$ (namely, $O(qq_{dec})$) by far outweighs the number of oracle queries performed by the adversary (namely, $O(q)$). This large number of queries negatively influences the bounds obtained when applying the new O2H Theorem. However, the $O(qq_{dec})$ queries performed by $Dec^{**}$ are all classical, only $O(q)$ quantum queries are made. Our O2H Theorem treats classical and quantum queries the same. A variant of the O2H Theorem that gives better bounds when only a small fraction of the queries are quantum would lead to improvements in the bounds obtained here. We leave this as a problem for future work.

## 2 Preliminaries

For basics of quantum computing, we refer to a standard textbook such as [NC00].

Given a function $f : X \to Y$, we model a quantum-accessible oracle $\mathcal{O}$ for $f$ as a unitary transformation $U_f$ operating on two registers $Q, R$ with spaces $\mathbb{C}^X$ and $\mathbb{C}^Y$, respectively, where $U_f : |q, r\rangle \mapsto |q, r \oplus f(x)\rangle$, where $\oplus$ is some involutive group operation (e.g., XOR if $Y$ is a set of bitstrings).

A quantum oracle algorithm is an algorithm that can perform classical and quantum computations, and that can query classical and/or quantum-accessible oracles. We allow an oracle algorithm $A$ to perform oracle queries in parallel. We say $A$ is a $q$-query algorithm if it performs at most $q$ oracle queries (counting parallel queries as separate queries), and has query depth $d$ if it invokes the oracle at most $d$ times (counting parallel queries as one query). For example, if $A$ performs 5 parallel queries followed by 7 parallel queries, we have $q = 12$ and $d = 2$.

The distinction between query number and query depth is important because realistic brute-force attacks are highly parallel. It's easy to do $2^{64}$ hash queries on parallel machines — the Bitcoin network does this several times a minute — but it would take millennia to do them sequentially. Query depth is also important because early quantum computers are likely to lose coherency quickly, limiting them to shallow circuits. Our model does not capture this limitation because it does not differentiate between a deep quantum computation and several shallow ones with measurements between. But we hope that future work can account for coherency using a notion of query depth.

We will make use of the well-known fact that any quantum oracle algorithm $A^{\mathcal{O}}(z)$ can be transformed into a *unitary* quantum oracle algorithm with constant factor computational overhead and the same query number and query depth. Such an algorithm has registers $Q_A$ (for its state), and $Q_1, \ldots, Q_n$ and $R_1, \ldots, R_n$ for query inputs and outputs, respectively. It starts with an initial state $|\Psi\rangle$ (that may depend on the input $z$). Then, $A$ alternatingly applies a fixed unitary $U$ on all registers (independent of $z$ and $\mathcal{O}$), and performs parallel queries. Parallel queries apply the oracle $\mathcal{O}$ to $Q_i, R_i$ for each $i = 1, \ldots, n$. (I.e., if $\mathcal{O}$ is implemented by $U_f$, we apply $U_f \otimes \cdots \otimes U_f$ between $U$-applications.) Finally, the classical output of $A^{\mathcal{O}}(z)$ is the result of a projective measurement on the final state of $A$. This implies that in many situations, we can assume

our algorithms to be unitary without loss of generality.

## 3 Semi-classical oracles

Classical oracles measure both their input and their output, whereas quantum-accessible oracles measure neither. We define semi-classical oracles, which measure their output but not their input. Formally, a semi-classical oracle $\mathcal{O}_f^{SC}$ for a function $f$ with domain $X$ and codomain $Y$ is queried with two registers: an input register $Q$ with space $\mathbb{C}^X$ and an output register $R$ with space $\mathbb{C}^Y$.

When queried with a value $|x\rangle$ in $Q$, the oracle performs a measurement of $f(x)$. Formally, it performs the measurements corresponding to the projectors $M_y : y \in Y$ where $M_y := \sum_{x \in S : f(x)=y} |x\rangle\langle x|$. The oracle then initializes the $R$ register to $|y\rangle$ for the measured $y$.

In this paper, the function $f$ is always the indicator function $f_S$ for a set $S$, where $f_S(x) = 1$ if $x \in S$ and 0 otherwise. For brevity, we overload the notation $\mathcal{O}_S^{SC}$ to be the semiclassical oracle for this index function.

To illustrate this, let us see what happens if the adversary performs the same query with a quantum oracle, a classical oracle, and a semi-classical oracle implementing the indicator function for $S$, respectively: Say the adversary sends the query $\sum_x 2^{-n/2}|x\rangle|0\rangle$, and say $S = \{x_0\}$. When querying a quantum oracle, the oracle returns the state $\sum_x 2^{-n/2}|x\rangle|f_S(x)\rangle = 2^{-n/2}|x\rangle|1\rangle + \sum_{x \neq x_0} 2^{-n/2}|x\rangle|0\rangle$. When querying a classical oracle, the resulting state will be $|x\rangle|f_S(x)\rangle$ for a uniformly random $x$. But when querying a semi-classical oracle, with probability $1 - 2^{-n}$, the resulting state is $\sum_{x \neq x_0} \frac{1}{\sqrt{2^n - 1}}|x\rangle|0\rangle$, and with probability $2^{-n}$, the resulting state is $|x_0\rangle|1\rangle$. In particular, the superposition between all $|x\rangle$ that are not in $S$ is preserved!

In the execution of a quantum algorithm $A^{\mathcal{O}_S^{SC}}$, let $\mathsf{Find}$ be the event that $\mathcal{O}_S^{SC}$ ever returns $|1\rangle$. This is a well-defined classical event because $\mathcal{O}_S^{SC}$ measures its output. This event is called $\mathsf{Find}$ because if it occurs, the simulator could immediately stop execution and measure the input register $Q$ to obtain a value $x \in S$.

If $H$ is some other quantum-accessible oracle with domain $X$ and codomain $Y$, we define $H \setminus S$ ("$H$ punctured on $S$") as an oracle which, on input $x$, first queries $\mathcal{O}_S^{SC}(x)$ and then $H(x)$. The following lemma shows why this is called "puncturing": when $\mathsf{Find}$ doesn't occur, the outcome of $A^{H \setminus S}$ is independent of $H(x)$ for all $x \in S$. Those values are effectively removed from $H$'s domain.

**Lemma 1** *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S.\ G(x) = H(x)$. Let $z$ be a random bitstring. ($S, G, H, z$ may have arbitrary joint distribution.)*

*Let $A$ be a quantum oracle algorithm (not necessarily unitary).*

*Let $E$ be an arbitrary (classical) event.*

*Then* $\Pr[E \wedge \neg\mathsf{Find} : x \leftarrow A^{H \setminus S}(z)] = \Pr[E \wedge \neg\mathsf{Find} : x \leftarrow A^{G \setminus S}(z)]$.

Semi-classical oracles allow us to split the O2H Theorem into two parts. The first

part bounds how much a quantum adversary's behavior changes when a random oracle is punctured on $S$ based on $\Pr[\mathsf{Find}]$:

**Theorem 1 (Semi-classical O2H)** *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S.\ G(x) = H(x)$. Let $z$ be a random bitstring. ($S, G, H, z$ may have arbitrary joint distribution.)*

*Let $A$ be an oracle algorithm of query depth $d$ (not necessarily unitary).*

*Let*

$$\begin{aligned} P_{\text{left}} &:= \Pr[b = 1 : b \leftarrow A^H(z)] \\ P_{\text{right}} &:= \Pr[b = 1 : b \leftarrow A^G(z)] \\ P_{\text{find}} &:= \Pr[\mathsf{Find} : A^{G\setminus S}(z)] \overset{\text{Lem. 1}}{=} \Pr[\mathsf{Find} : A^{H\setminus S}(z)] \end{aligned} \tag{1}$$

*Then*

$$|P_{\text{left}} - P_{\text{right}}| \le 2\sqrt{(d+1)\cdot P_{\text{find}}} \quad \text{and} \quad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \le 2\sqrt{(d+1)\cdot P_{\text{find}}}$$

*The theorem also holds with bound $\sqrt{(d+1)P_{\text{find}}}$ for the following alternative definitions of $P_{\text{right}}$:*

$$P_{\text{right}} := \Pr[b = 1 : b \leftarrow A^{H\setminus S}(z)], \tag{2}$$

$$P_{\text{right}} := \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)], \tag{3}$$

$$P_{\text{right}} := \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)], \tag{4}$$

$$P_{\text{right}} := \Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{H\setminus S}(z)], \tag{5}$$

$$P_{\text{right}} := \Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]. \tag{6}$$

In this theorem, we give $A$ only access to a single oracle ($G$ or $H$). In many settings, there may be additional oracles that $A$ has access to. It may not be obvious at the first glance, but Theorem 1 applies in that case, too. Since there is no assumption on the runtime of $A$, or on the size of $z$, nor on the number of queries made to the additional oracles, additional oracles can simply be encoded as part of $z$. That is, if we want to consider an adversary $A^{H,F}()$, we can instead write $A^H(F)$ where $F$ is a complete (exponential size) description of $F$.

The proof of Theorem 1 is given in Section 5.2.

The second part relates $\Pr[\mathsf{Find}]$ to the guessing probability:

**Theorem 2 (Search in semi-classical oracle)** *Let $A$ be any quantum oracle algorithm making at most $q$ queries to a semi-classical oracle with domain $X$. Let $S \subseteq X$ and $z \in \{0,1\}^*$. ($S, z$ may have arbitrary joint distribution.)*

*Let $B$ be an algorithm that on input $z$ chooses $i \xleftarrow{\$} \{1, \dots, d\}$; runs $A^{\mathcal{O}^{SC}_{\varnothing}}(z)$ until (just before) the $i$-th query; then measures all query input registers in the computational basis and outputs the set $T$ of measurement outcomes.*

*Then*

$$\Pr[\mathsf{Find} : A^{\mathcal{O}^{SC}_S}(z)] \le 4d \cdot \Pr[S \cap T \neq \varnothing : T \leftarrow B(z)] \tag{7}$$

The proof is given in Section 5.3.

In the simple but common case that the input of $A$ is independent of $S$, we get the following corollary:

**Corollary 1** *Suppose that $S$ and $z$ are independent, and that $A$ is a $q$-query algorithm. Let $P_{\max} := \max_{x \in X} \Pr[x \in S]$. Then*

$$\Pr[\mathsf{Find} : A^{\mathcal{O}_S^{SC}}(z)] \leq 4q \cdot P_{\max}. \tag{8}$$

For example, for uniform $x \in \{1, \ldots, N\}$, $A^{\mathcal{O}_{\{x\}}^{SC}}$ finds $x$ with probability $\leq 4q/N$.

*Proof.* Since the query depth of $A$ does not occur in the lemma, we can assume that $A$ does not perform parallel queries. Then the output $T$ of $B$ in Theorem 2 has $|T| \leq 1$, and $d = q$. Thus $\Pr[S \cap T \neq \varnothing : T \leftarrow B(z)]$ is simply the probability that $B(z)$ outputs an element of $S$. Hence $\Pr[S \cap T \neq \varnothing : T \leftarrow B(z)] \leq P_{\max}$. Then by Theorem 2, $\Pr[\mathsf{Find} : A^{\mathcal{O}_S^{SC}}(z)] \leq 4q \cdot P_{\max}$. □

### 3.1 Regular O2H, revisited

Note that the use of semi-classical oracles in Theorem 1 is entirely optional. If we use variant (1) and apply Theorem 2 to $P_{\text{find}}$, we get a variant of Theorem 1 that does not involve semi-classical oracles. The result is essentially the following Theorem 3. However, proving Theorem 3 directly gives a better bound: $2d\sqrt{P_{guess}}$ instead of $4d\sqrt{P_{guess}}$.

**Theorem 3 (One-way to hiding, probabilities)** *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S. G(x) = H(x)$. Let $z$ be a random bitstring. ($S, G, H, z$ may have arbitrary joint distribution.)*

*Let $A$ be quantum oracle algorithm with query depth $d$ (not necessarily unitary).*

*Let $B^H$ be an oracle algorithm that on input $z$ does the following: pick $i \xleftarrow{\$} \{1, \ldots, d\}$, run $A^H(z)$ until (just before) the $i$-th query, measure all query input registers in the computational basis, output the set $T$ of measurement outcomes.*

*Let*

$$\begin{aligned} P_{\text{left}} &:= \Pr[b = 1 : b \leftarrow A^H(z)] \\ P_{\text{right}} &:= \Pr[b = 1 : b \leftarrow A^G(z)] \\ P_{\text{guess}} &:= \Pr[S \cap T \neq \varnothing : T \leftarrow B^H(z)] \end{aligned}$$

*Then*

$$|P_{\text{left}} - P_{\text{right}}| \leq 2d\sqrt{P_{\text{guess}}} \qquad \textit{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2d\sqrt{P_{\text{guess}}}$$

*The same result holds with $B^G$ instead of $B^H$ in the definition of $P_{\text{guess}}$.*

As we said already, except for a factor of 2 in the bound, this is an immediate corollary of Theorem 1 and Theorem 2. To get the slightly better bound in Theorem 3, we use a direct proof. See Section 5.4.

The original O2H [Unr15b, Lemma 6.2] is an immediate consequence of Theorem 3: Pick $H$ uniformly, pick $x, y$ uniformly, set $G(\cdot) := y$, $I := \{x\}$ and $z := (x, H(x))$. Then $P_{\text{left}}$ and $P_{\text{guess}}$ are as in the original O2H. $P_{\text{right}}$ is $\Pr[b = 1 : b \leftarrow A^{H(x:=y)}(x, H(x))]$, but this is the same as $\Pr[b = 1 : b \leftarrow A^H(x, y)]$.

This also implies that using Theorem 1 and Theorem 2 instead can never give worse bounds than the original O2H, except by a factor of 2.

# 4 Examples how to use the O2H Theorems

To illustrate the use of the theorems from the previous section, we give two illustrative examples: hardness of searching in a sparse random function, and hardness of inverting a random oracle with leakage (in the sense that an only computationally secret encryption of the preimage is given to the adversary).

## 4.1 Hardness of searching in a sparse random function

Consider the following setting: $H : X \to \{0, 1\}$ is a random function where for each $x$, $H(x) = 1$ with probability $\leq \lambda$ (not necessarily independently). What is the probability to find $x$ with $H(x) = 1$ in $q$ queries? We will prove an upper bound.

We solve this problem using the semi-classical O2H technique introduced by Theorem 1. Let $A$ be a $q$-query algorithm with depth $d$. We want to bound $\Pr[H(x) = 1 : x \leftarrow A^H()]$. We do this by a series of games.

**Game 1** *$x \leftarrow A^H()$. Measure $x$. Then $A$ wins if $H(x) = 1$.*

We would like to apply Theorem 1 to this game. But it doesn't work well to apply it to $A^H$ because $H$ is also used outside of $A$. Therefore, we use a different but obviously equivalent game:

**Game 2** *Define $\hat{A}^H()$ to run $x \leftarrow A^H()$; measure $x$; and return $b := H(x)$. Game 2 runs $b \leftarrow \hat{A}^H()$. Then $A$ wins if $b = 1$.*

Note that $\hat{A}$ is a $(q + 1)$-query algorithm with depth $d + 1$.

We can apply the semi-classical O2H Theorem (Theorem 1), variant (4)[4] to this game,

[4]Theorem 1 gives us different options how to define the right game. Conceptually simplest is variant (1) (it does not involve a semi-classical oracle in the right game), but it does not apply in all situations. The basic idea behind all variants is the same, namely that the adversary gets access to an oracle $G$ that behaves differently on the set $S$ of marked elements.

where $G := 0$ (the constant zero function) and $S := \{x : H(x) = 1\}$. This gives us:

$$\left|\sqrt{\underbrace{\Pr[b = 1 : \text{Game 2}]}_{P_{\text{left}}}} - \sqrt{\underbrace{\Pr[b = 1 \wedge \neg\mathsf{Find} : \text{Game 3}]}_{P_{\text{right}}}}\right| \leq \sqrt{(d+2)\underbrace{\Pr[\mathsf{Find} : \text{Game 3}]}_{P_{\text{find}}}} \quad (9)$$

with

**Game 3** *Run* $b \leftarrow \hat{A}^{G\setminus S}()$. *Then* $A$ *wins if* $b = 1$ *and not* Find.

which is equivalent to

**Game 4** $x \leftarrow A^{G\setminus S}()$; *set* $b \leftarrow (G \setminus S)(x)$. *Then* $A$ *wins if* $b = 1$ *and not* Find.

What has happened so far? We have used the O2H Theorem to rewrite a game with access to an oracle $H$ (Game 1) into the same game with a different oracle $G = 0$ (Game 4) ("right game"). The new oracle is considerably simpler: in this specific case, it is all zero. The difference between the two games is bounded by (9) in terms of how hard it is to find an element in the set $S$ (the "marked elements"), i.e., a position where $G$ and $H$ differ (the "finding game"). This is the typical way of applying an O2H Theorem: Replace the oracle $H$ by something simpler, continue the game-based proof from the right game, and additionally perform a second game-based proof to bound the probability of finding a marked element in the finding game.

However, there are several crucial differences to the use of prior O2H lemmas (e.g., [Unr15b]). First, prior O2H Theorems required $G$ and $H$ to be uniformly random functions, and to differ only at a single location $x$. But here $H$ is not assumed to be uniform, and it differs from $G$ at more than a single input (i.e. at the entire set $S$). This allows us to analyze search problems with multiple targets.

Second, (9) has square roots on the left-hand side. This is optional: Theorem 1 also gives a bound without square roots. In our example, since $P_{\text{right}}$ is very small, the square-root variant gives smaller bounds for $P_{\text{left}}$.

Third, the finding game is expressed using semi-classical oracles. This is never a limitation because we can always replace the semi-classical oracles by quantum-accessible ones using Theorem 2 (which then gives bounds comparable to the O2H from [Unr15b]). However, as we will see in the next section, in some cases semi-classical oracles give better bounds.

In our case, we trivially have $\Pr[G(x) = 1 \wedge \neg\mathsf{Find} : \text{Game 4}] = 0$ since $G = 0$.

However, analyzing $\Pr[\mathsf{Find} : \text{Game 3}]$ is less trivial. At the first glance, it seems that having access to the oracle $G = 0$ yields no information about $S$, and thus finding an element of $S$ is down to pure luck, and cannot succeed with probability greater than $(q+1)\lambda$. But in fact, computing $G \setminus S$ requires measuring whether each query is in $S$. The measurement process can leak information about $S$. Appendix A shows that at least in some cases, it is possible to find elements of $S$ with greater probability than $(q+1)\lambda$.

Fortunately, we have a result for this situation, namely Corollary 1, which shows that $\Pr[\mathsf{Find} : \text{Game } 4] \leq 4(q+1)\lambda$.

Plugging this into (9), we get

$$\Pr[H(x) = 1 : \text{Game } 1] \leq 4(d+2)(q+1)\lambda.$$

Without the square roots on the left-hand side of (9), we would get only the bound $\sqrt{4(d+2)(q+1)\lambda}$.

We summarize what we have proven in the following lemma:

**Lemma 2 (Search in unstructured function)** *Let $H$ be a random function, drawn from a distribution such that $\Pr[H(x) = 1] \leq \lambda$ for all $x$. Let $A$ be a $q$-query adversary with query depth $d$. Then $\Pr[H(x) = 1 : b \leftarrow A^H()] \leq 4(d+2)(q+1)\lambda$.*

While this is a simple consequence of our O2H technique, we are not aware that this bound was already presented in the literature. While [Zal99] already showed a trade-off between parallelism and query number in unstructured quantum search. However, our result gives an explicit (and tight) success probability and applies even to functions whose outputs are not i.i.d. For the special case of no-parallelism ($d = q$) and i.i.d. functions, the best known bound was [HRS16, Theorem 1] which we improve upon by a factor of 2. Additionally, our lemma allows the different outputs of $H$ to be correlated while prior results require them to be independent.

### 4.2 Hardness of inverting a random oracle with leakage

The previous example considered a pure query-complexity problem, searching in a random function. It can easily be solved with other techniques (giving slightly different bounds). Where O2H Theorems shine is the combination of computational hardness and random oracles. The following example illustrates this.

Let $E$ be a randomized algorithm taking input from a space $X$, such that it is difficult to distinguish the distributions

$$\mathcal{D}_1 := \{(x, E(x)) : x \xleftarrow{\$} X\} \quad \text{and} \quad \mathcal{D}_0 := \{(x_1, E(x_2)) : (x_1, x_2) \xleftarrow{\$} X\}$$

For a quantum algorithm $B$, define its $E$-distinguishing advantage as

$$\mathrm{Adv}_{\mathrm{IND}-E}(B) := \left| \begin{array}{l} \Pr\left[1 \leftarrow B(x,e) : (x,e) \leftarrow \mathcal{D}_1\right] \\ \quad - \Pr\left[1 \leftarrow B(x,e) : (x,e) \leftarrow \mathcal{D}_0\right] \end{array} \right|$$

For example, $E$ could be IND-CPA-secure encryption. Let $H : X \to Y$ be a random oracle which is independent of $E$. How hard is it to invert $H$ with a leakage of $E$? That is, given a quantum oracle algorithm $A$, we want to bound

$$\mathrm{Adv}_{\text{OW-LEAK-}E}(A) := \Pr\left[A^H(H(x), E(x)) = x : x \xleftarrow{\$} X\right]$$

We can do this using a series of games. For brevity, we will go into slightly less detail than in Section 4.1. Let $w_i$ be the probability that the adversary wins Game $i$.

**Game 0 (Original)** $x \xleftarrow{\$} X; x' \leftarrow A^H(H(x), E(x))$. *The adversary wins if* $x' = x$.

Now choose a random $y \xleftarrow{\$} Y$, and set a different random oracle $G := H(x := y)$ which is the same as $H$ on every input except $S := \{x\}$. We can define a new game where the adversary has access to $G \setminus S$:

**Game 1 (Punctured, first try)** $x \xleftarrow{\$} X; x' \leftarrow A^{G\setminus\{x\}}(H(x), E(x))$. *The adversary wins if* $x' = x$ *and not* Find.

Applying Theorem 1 variant (4), we find that

$$\left| \sqrt{\underbrace{\Pr[x' = x : \text{Game } 0]}_{P_{\text{left}}=w_0}} - \sqrt{\underbrace{\Pr[x' = x \wedge \neg\mathsf{Find} : \text{Game } 1]}_{P_{\text{right}}=w_1}} \right| \leq \sqrt{\underbrace{(d+1)\Pr[\mathsf{Find} : \text{Game } 1]}_{P_{\text{find}}}}$$

Unlike in Section 4.1, this time we do not have a trivial bound for $w_1$. We could bound it in terms of distinguishing advantage against $E$. But let's instead try to make this game more like the ones in Section 4.1: we can cause the adversary to Find instead of winning. To do this, we just apply an extra hash operation. Let $\hat{A}^H(y, e)$ be the algorithm which runs $x' \leftarrow A^H(y, e)$; computes $H(x')$ and ignores the result; and then returns $x'$. Then $\hat{A}$ performs $q + 1$ queries at depth $d + 1$. This gives us a new game:

**Game 2 (Original with extra hash)** $x \xleftarrow{\$} X; x' \leftarrow \hat{A}^H(H(x), E(x))$. *The adversary wins if* $x' = x$.

Clearly $w_2 = w_0$. The new punctured game is also similar:

**Game 3 (Punctured, extra hash)** $x \xleftarrow{\$} X; x' \leftarrow \hat{A}^{G\setminus\{x\}}(H(x), E(x))$. *The adversary wins if* $x' = x$ *and not* Find.

Applying Theorem 1 variant (4) as before gives

$$|\sqrt{w_3} - \sqrt{w_2}| \leq \sqrt{(d+2)\Pr[\mathsf{Find} : \text{Game } 3]} \tag{10}$$

But the adversary cannot win Game 3: the extra hash query triggers Find if $x' = x$, and the adversary does not win if Find. Therefore $w_3 = 0$. Plugging this into (10) and squaring both sides gives:

$$w_0 = w_2 \leq (d+2)\Pr[\mathsf{Find} : \text{Game } 3] \tag{11}$$

It remains to bound the right-hand side. We first note that in Game 3, the value $H(x)$ is only used once, since the adversary does not have access to $H(x)$: it only has access to $G$, which is the same as $H$ everywhere except $x$. So Game 3 is the same as if $H(x)$ is replaced by a random value:

**Game 4 (No $H(x)$)** *Set* $x \xleftarrow{\$} X; y \xleftarrow{\$} Y; \hat{A}^{G\setminus\{x\}}(y, E(x))$. *We don't care about the output of* $\hat{A}$*, but only whether it* Find*s.*

Clearly $\Pr[\mathsf{Find} : \text{Game } 4] = \Pr[\mathsf{Find} : \text{Game } 3]$. Finally, we apply the indistinguishability assumption by comparing to the following game:

**Game 5 (IND-$E$ challenge)** $(x_1, x_2) \xleftarrow{\$} X; y \xleftarrow{\$} Y; \hat{A}^{G\setminus\{x_1\}}(y, E(x_2))$.

Let $B(x, e)$ be an algorithm which chooses $y \xleftarrow{\$} Y$; runs $\hat{A}^{G\setminus\{x\}}(y, e)$; and returns 1 if Find and 0 otherwise. Then $B$ runs in about the same time as $A$ plus $(q+1)$ comparisons. If $(y, e)$ are drawn from $\mathcal{D}_1$, then this experiment is equivalent to Game 4, and it they are drawn from $\mathcal{D}_0$ then it is equivalent to Game 5. Therefore $B$ is a distinguisher for $E$ with advantage exactly

$$\mathrm{Adv}_{\mathrm{IND}-E}(B) = |\Pr[\mathsf{Find} : \text{Game } 5] - \Pr[\mathsf{Find} : \text{Game } 4]| \tag{12}$$

Furthermore, in Game 5, the oracle $G$ is punctured at $x_1$, which is uniformly random and independent of everything else in the game. So by Theorem 2,

$$\Pr[\mathsf{Find} : \text{Game } 5] \leq 4(q+1)/\mathrm{card}(X)$$

Combining this with (11) and (12), we have

$$\mathrm{Adv}_{\text{OW-LEAK-}E}(A) \leq (d+2)\mathrm{Adv}_{\mathrm{IND}-E}(B) + \frac{4(d+2)(q+1)}{\mathrm{card}(X)}$$

This is a much better bound than we would have gotten without using semi-classical oracles (i.e., using Theorem 3 or the O2H Theorem from [Unr15b]). In front of $\mathrm{Adv}_{\mathrm{IND}-E}(B)$, we only have the factor $d+2$. In contrast, if we had applied Theorem 2 directly after using Theorem 1, then we would have gotten a factor of $O(qd)$ in front of $\mathrm{Adv}_{\mathrm{IND}-E}(B)$. If we had used the O2H from [Unr15b], then we would have gotten an even greater bound of $O(q\sqrt{\mathrm{Adv}_{\mathrm{IND}-E}(B) + 1/\mathrm{card}(X)})$. However, this bound with semi-classical oracles assumes indistinguishability, whereas an analysis with Theorem 3 would only require $E$ to be one-way.

# 5 Proofs

## 5.1 Auxiliary lemmas

The fidelity $F(\sigma, \tau)$ between two density operators is $\mathrm{tr}\sqrt{\sqrt{\sigma}\tau\sqrt{\sigma}}$, the trace distance $\mathrm{TD}(\sigma, \tau)$ is defined as $\frac{1}{2}\mathrm{tr}|\sigma - \tau|$, and the Bures distance $B(\tau, \sigma)$ is $\sqrt{2 - 2F(\tau, \sigma)}$.

**Lemma 3** *For states* $|\Psi\rangle, |\Phi\rangle$ *with* $\||\Psi\rangle\| = \||\Phi\rangle\| = 1$*, we have*

$$F(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|) \geq 1 - \frac{1}{2}\||\Psi\rangle - |\Phi\rangle\|^2$$

*so that*

$$B(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|) \leq \||\Psi\rangle - |\Phi\rangle\|$$

*Proof.* We have

$$\begin{aligned}\||\Psi\rangle - |\Phi\rangle\|^2 = (\langle\Psi| - \langle\Phi|)(|\Psi\rangle - |\Phi\rangle) = \||\Psi\rangle\|^2 + \||\Phi\rangle\|^2 - \langle\Psi|\Phi\rangle - \langle\Phi|\Psi\rangle \\ = 2 - 2\Re(\langle\Psi|\Phi\rangle) \geq 2 - 2|\langle\Psi|\Phi\rangle| \overset{(*)}{=} 2 - 2F(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|)\end{aligned}$$

where $\Re$ denotes the real part, and $(*)$ is by definition of the fidelity $F$ (for pure states). Thus $F(|\Psi\rangle\langle\Psi|, |\Phi\rangle\langle\Phi|) \geq 1 - \frac{1}{2}\||\Psi\rangle - |\Phi\rangle\|^2$ as claimed. The second inequality follows from the definition of Bures distance. □

**Lemma 4 (Distance measures vs. measurement probabilities)** *Let $\rho_1, \rho_2$ be density operators (with $\operatorname{tr}\rho_i = 1$). Let $M$ be a binary measurement (e.g., represented as a POVM). Let $P_i$ be the probability that $M$ returns 1 when measuring $\rho_i$.*

*Then*

$$\sqrt{P_1 P_2} + \sqrt{(1-P_1)(1-P_2)} \geq F(\rho_1, \rho_2) \tag{13}$$

*Also,*

$$\left|\sqrt{P_1} - \sqrt{P_2}\right| \leq B(\rho_1, \rho_2). \tag{14}$$

*Furthermore,*

$$|P_1 - P_2| \leq \mathrm{TD}(\rho_1, \rho_2) \leq B(\rho_1, \rho_2). \tag{15}$$

*Proof.* In this proof, given a probability $P$, let $\bar{P} := 1 - P$. Let $\mathcal{E}$ be the superoperator that maps $\rho$ to the classical bit that contains the result of measuring $\rho$ using $M$. That is, for every density operator $\rho$ with $\operatorname{tr}\rho = 1$, $\mathcal{E}(\rho) = \begin{pmatrix} p & 0 \\ 0 & \bar{p} \end{pmatrix}$ where $p$ is the probability that $M$ returns 1 when measuring $\rho$.

Then $\rho_i' := \mathcal{E}(\rho_i) = \begin{pmatrix} P_i & 0 \\ 0 & \bar{P}_i \end{pmatrix}$ for $i = 1, 2$. We then have

$$\begin{aligned}F(\rho_1, \rho_2) \overset{(*)}{\leq} F(\rho_1', \rho_2') \overset{(**)}{=} \left\|\sqrt{\rho_1'}\sqrt{\rho_2'}\right\|_{\mathrm{tr}} \\ = \operatorname{tr}\begin{pmatrix} \sqrt{P_1 P_2} & 0 \\ 0 & \sqrt{\bar{P}_1 \bar{P}_2} \end{pmatrix} = \sqrt{P_1 P_2} + \sqrt{\bar{P}_1 \bar{P}_2}\end{aligned}$$

where $(*)$ is due to the the monotonicity of the fidelity [NC00, Thm. 9.6], and $(**)$ is the definition of fidelity. This shows (13). To prove (14), we compute:

$$\begin{aligned}\left(\sqrt{P_1} - \sqrt{P_2}\right)^2 &= P_1 + P_2 - 2\sqrt{P_1 P_2} \\ &\leq P_1 + P_2 - 2\sqrt{P_1 P_2} + \left(\sqrt{\bar{P}_1} - \sqrt{\bar{P}_2}\right)^2 \\ &= 2 - 2\sqrt{P_1 P_2} - 2\sqrt{\bar{P}_1 \bar{P}_2} \overset{(13)}{\leq} 2 - 2F(\rho_1, \rho_2) \overset{(*)}{=} B(\rho_1, \rho_2)^2\end{aligned}$$

where $(*)$ is by definition of the Bures distance. This implies (14).

The first inequality in (15) is well-known (e.g., [NC00, Thm. 9.1]). For the second part, we calculate

$$\begin{aligned}\mathrm{TD}(\rho,\tau) &\overset{(*)}{\leq} \sqrt{1-F(\rho,\tau)^2} = \sqrt{\frac{1+F(\rho,\tau)}{2}}\cdot\sqrt{2-2F(\rho,\tau)} \\ &= \sqrt{\frac{1+F(\rho,\tau)}{2}}\cdot B(\rho,\tau) \overset{(**)}{\leq} B(\rho,\tau)\end{aligned}$$

Here the inequality marked $(*)$ is shown in [NC00, (9.101)], and $(**)$ is because $0 \leq F(\rho,\tau) \leq 1$. □

## 5.2 Proof of Theorem 1

In the following, let $H : X \to Y$, $S \subseteq X$, $z \in \{0,1\}^*$.

**Lemma 5 (O2H in terms of pure states)** *Fix $H, S, z$. Let $A^H(z)$ be a unitary quantum oracle algorithm of query depth $d$. Let $Q_A$ denote the register containing all of $A$'s state.*

*Let $L$ be a quantum register with space $\mathbb{C}^{2^d}$ (for the "query log").*

*Let $B^{H,S}(z)$ be the unitary algorithm on registers $Q_A, L$ that operates like $A^H(z)$, except:*

- *It initializes the register $L$ with $|0\dots0\rangle$.*
- *When $A$ performs its $i$-th set of parallel oracle queries on input/output registers $(Q_1,R_1),\dots,(Q_n,R_n)$ that are part of $Q_A$, $B$ instead first applies $U_S$ on $(Q_1,\dots,Q_n,L)$ and then performs the oracle queries. Here $U_S$ is defined by:*

$$U_S|x_1,\dots,x_n\rangle|l\rangle := \begin{cases}|x_1,\dots,x_n\rangle|l\rangle & (\text{every } x_j \notin S),\\ |x_1,\dots,x_n\rangle|\mathrm{flip}_i(l)\rangle & (\text{any } x_j \in S)\end{cases}$$

*Let $|\Psi_{\mathrm{left}}\rangle$ denote the final state of $A^H(z)$, and $|\Psi_{\mathrm{right}}\rangle$ the final state of $B^{H,S}(z)$.*

*Let $\tilde{P}_{\mathrm{find}}$ be the probability that a measurement of $L$ in the state $|\Psi_{\mathrm{right}}\rangle$ returns $\neq 0$. (Formally, $\big\|(I\otimes(I-|0\rangle\langle 0|))|\Psi_{\mathrm{right}}\rangle\big\|^2$.)*

*Then*

$$\big\||\Psi_{\mathrm{left}}\rangle\otimes|0\rangle - |\Psi_{\mathrm{right}}\rangle\big\|^2 \leq (d+1)\tilde{P}_{\mathrm{find}}.$$

*Proof.* We first define a variant $B_{\mathrm{count}}$ of the algorithm $B$ that, instead of keeping a log of the successful oracle queries (as $B$ does in $L$), just counts the number of successful oracle queries (in a register $C$). Specifically:

Let $C$ be a quantum register with space $\mathbb{C}^{\{0,\dots,d\}}$, i.e., $C$ can store states $|0\rangle,\dots,|d\rangle$. Let $B^{H,S}_{\mathrm{count}}(z)$ be the unitary algorithm on registers $Q_A, S$ that operates like $A^H(z)$, except:

- It initializes the register $C$ with $|0\rangle$.

- When $A$ performs its $i$-th set of parallel oracle queries on input/output registers $((Q_1, R_1), \ldots)$ that are part of $Q_A$, $B$ instead first applies $U'_S$ on $(Q_1, \ldots, Q_n), C$ and then performs the oracle queries. Here $U'_S$ is defined by:

$$U'_S|x_1, \ldots, x_n\rangle|c\rangle := \begin{cases} |x_1, \ldots, x_n\rangle|c\rangle & (\text{every } x_j \notin S), \\ |x_1, \ldots, x_n\rangle|c+1 \bmod d+1\rangle & (\text{any } x_j \in S) \end{cases}$$

Note that the $\bmod d+1$ part of the definition of $U'_S$ has no effect on the behavior of $\tilde{B}$ because $U_S$ is applies only $d$ times. However, the $\bmod d+1$ is required so that $U_S$ is unitary.

Consider the state $|\Psi_{\text{count}}\rangle$ at the end of the execution $B^{H,S}_{\text{count}}(z)$. This may be written

$$|\Psi_{\text{count}}\rangle = \sum_{i=0}^{d} |\Psi'_i\rangle|i\rangle_C. \tag{16}$$

for some (non-normalized) states $|\Psi'_i\rangle$ on $Q_A$.

Consider the linear (but not unitary) map $N' : |x\rangle|y\rangle \mapsto |x\rangle|0\rangle$. Obviously, $N'$ commutes with the oracle queries and with the unitary applied by $A$ between queries (since those unitaries do not operate on $C$.) Furthermore $N'U'_S = N'$, and the initial state of $B_{\text{count}}$ is invariant under $N'$. Thus $N'|\Psi_{\text{count}}\rangle$ is the same as the state we get if we execute $B_{\text{count}}$ without the applications of $U'_S$. But that state is $|\Psi_{\text{left}}\rangle|0\rangle_C$ because the only difference between $B_{\text{count}}$ and $A$ is that $B_{\text{count}}$ initializes $C$ with $|0\rangle$ and applies $U'_S$ to it.

So we have

$$\sum_{i=0}^{d} |\Psi'_i\rangle|0\rangle_C = N|\Psi_{\text{count}}\rangle = |\Psi_{\text{left}}\rangle|0\rangle_C$$

and hence

$$|\Psi_{\text{left}}\rangle = \sum_{i=0}^{d} |\Psi'_i\rangle. \tag{17}$$

The state $|\Psi_{\text{right}}\rangle$ is a state on $Q_A, L$ and thus can be written as

$$|\Psi_{\text{right}}\rangle = \sum_{l \in \{0,1\}^q} |\Psi_l\rangle|l\rangle_L \tag{18}$$

for some (non-normalized) states $|\Psi_l\rangle$ on $Q_A$.

Furthermore, both $|\Psi_{\text{count}}\rangle$ and $|\Psi_{\text{right}}\rangle$, when projected onto $|0\rangle$ in register $C/L$, respectively, result in the same state, namely the state corresponding to no query to $\mathcal{O}^{SC}_S$ succeeding. By (16) and (18), the result of that projection is $|\Psi_0\rangle|0\rangle_L$ and $|\Psi'_0\rangle|0\rangle_C$, respectively. Hence

$$|\Psi_0\rangle = |\Psi'_0\rangle. \tag{19}$$

Furthermore, the probability that no query succeeds is the square of the norm of that state. Hence

$$\big\||\Psi_0\rangle\big\|^2 = 1 - \tilde{P}_{\text{find}}. \tag{20}$$

We have

$$\sum_{i=0}^{d}\big\||\Psi'_i\rangle\big\|^2=\sum_{i=0}^{d}\big\||\Psi'_i\rangle|i\rangle_C\big\|^2=\Big\|\sum_{i=0}^{d}|\Psi'_i\rangle|i\rangle_C\Big\|^2\overset{(16)}{=}\big\||\Psi_{\text{count}}\rangle\big\|^2=1.$$

$$\sum_{l\in\{0,1\}^d}\big\||\Psi_l\rangle\big\|^2=\sum_{l\in\{0,1\}^d}\Big\||\Psi_l\rangle|l\rangle_L\Big\|^2=\Big\|\sum_{l\in\{0,1\}^d}|\Psi_l\rangle|l\rangle_L\Big\|^2\overset{(18)}{=}\big\||\Psi_{\text{right}}\rangle\big\|^2=1.$$

Thus

$$\sum_{i=1}^{d}\big\||\Psi'_i\rangle\big\|^2=1-\big\||\Psi'_0\rangle\big\|^2\overset{(20)}{=}\tilde{P}_{\text{find}},\qquad\sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}}\big\||\Psi_l\rangle\big\|^2=1-\big\||\Psi_0\rangle\big\|^2\overset{(20)}{=}\tilde{P}_{\text{find}}.\tag{21}$$

Therefore

$$\begin{aligned}
&\Big\||\Psi_{\text{right}}\rangle-|\Psi_{\text{left}}\rangle|0\rangle_L\Big\|^2\overset{(18)}{=}\Big\|\big(|\Psi_0\rangle-|\Psi_{\text{left}}\rangle\big)|0\rangle+\sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}}|\Psi_l\rangle|l\rangle\Big\|^2\\
&=\Big\||\Psi_0\rangle-|\Psi_{\text{left}}\rangle\Big\|^2+\sum_{\substack{l\in\{0,1\}^d\\ l\neq 0}}\big\||\Psi_l\rangle\big\|^2\overset{(21)}{=}\Big\||\Psi_0\rangle-|\Psi_{\text{left}}\rangle\Big\|^2+\tilde{P}_{\text{find}}\\
&\overset{(19),(17)}{=}\Big\|\sum_{i=1}^{d}|\Psi'_i\rangle\Big\|^2+\tilde{P}_{\text{find}}\overset{(*)}{\leq}\Big(\sum_{i=1}^{d}\Big\||\Psi'_i\rangle\Big\|\Big)^2+\tilde{P}_{\text{find}}\overset{(**)}{\leq}d\cdot\sum_{i=1}^{d}\Big\||\Psi'_i\rangle\Big\|^2+\tilde{P}_{\text{find}}\\
&\overset{(21)}{=}d\tilde{P}_{\text{find}}+\tilde{P}_{\text{find}}=(d+1)\tilde{P}_{\text{find}}.
\end{aligned}$$

Here $(*)$ uses the triangle inequality, and $(**)$ the AM-QM (or Jensen's) inequality. This is the inequality claimed in the lemma. □

**Lemma 6 (O2H in terms of mixed states)** *Let $H, I, z$ be random. (With some joint distribution.)*

*Let $A$ be an algorithm with query depth $d$. Let $B$ and $P_{\text{find}}$ be as in Theorem 1.*

*Let $\rho_{\text{left}}$ denote the final state of $A$.*

*Let $\rho_{\text{right}}$ denote the final state of $Q_A, L$, where $Q_A$ is the register used for its state by $B$ (or $A$), and $L$ is a register that contains the log of the responses of $\mathcal{O}_I^{SC}$. If the $i$-th query to $\mathcal{O}_I^{SC}$ returns $\ell_i$, then $L$ contains $|\ell_1 \dots \ell_q\rangle$ at the end of the execution of $B$.*

*Then $F(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}}) \geq 1 - \frac{1}{2}(d+1)P_{\text{find}}$ and $B(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}}) \leq \sqrt{(d+1)P_{\text{find}}}$.*

*Proof.* Without loss of generality, we can assume that $A$ is unitary: If $A$ is not unitary, we can construct a unitary variant of $A$ that uses an extra auxiliary register $Z$, and later trace out that register again from the states $\rho_{\text{left}}$ and $\rho_{\text{right}}$.

Let $\left|\Psi_{\text{left}}^{HIz}\right\rangle$ be the state $\left|\Psi_{\text{left}}\right\rangle$ from Lemma 5 for specific values of $H, I, z$. And analogously for $\left|\Psi_{\text{right}}^{HIz}\right\rangle$ and $\tilde{P}_{\text{find}}^{HIz}$.

Then $\rho_{\text{left}} = \mathrm{Exp}_{HIz}[\big|\Psi^{HIz}_{\text{left}}\big\rangle\big\langle\Psi^{HIz}_{\text{left}}\big|]$

Furthermore, if we define $\rho'_{\text{right}} := \mathrm{Exp}_{HIz}[|\Psi^{HIz}_{\text{right}}\rangle\langle\Psi^{HIz}_{\text{right}}|]$, then $\rho_{\text{right}} = \mathcal{E}_L(\rho'_{\text{right}})$ where $\mathcal{E}_L$ is the quantum operation that performs a measurement in the computational basis on the register $L$.

And $P_{\text{find}} = \mathrm{Exp}_{HIz}[\tilde{P}^{HIz}_{\text{find}}]$.

Then

$$\begin{aligned}
F(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}}) &= F\big(\mathcal{E}_L(\rho_{\text{left}} \otimes |0\rangle\langle 0|), \mathcal{E}_L(\rho'_{\text{right}})\big) \\
&\overset{(*)}{\geq} F\big(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho'_{\text{right}}\big) \\
&= F\left(\operatorname*{Exp}_{HIz}\Big[\big|\Psi^{HIz}_{\text{left}}\big\rangle\big\langle\Psi^{HIz}_{\text{left}}\big| \otimes |0\rangle\langle 0|\Big], \operatorname*{Exp}_{HIz}\Big[\big|\Psi^{HIz}_{\text{right}}\big\rangle\big\langle\Psi^{HIz}_{\text{right}}\big|\Big]\right) \\
&\overset{(**)}{\geq} \operatorname*{Exp}_{HIz}\Big[F\left(\big|\Psi^{HIz}_{\text{left}}\big\rangle\big\langle\Psi^{HIz}_{\text{left}}\big| \otimes |0\rangle\langle 0|, \big|\Psi^{HIz}_{\text{right}}\big\rangle\big\langle\Psi^{HIz}_{\text{right}}\big|\right)\Big] \\
&\overset{\text{Lem. 3}}{\geq} 1 - \tfrac{1}{2}\operatorname*{Exp}_{HIz}\Big[\big\||\Psi^{HIz}_{\text{left}}\rangle \otimes |0\rangle - |\Psi^{HIz}_{\text{right}}\rangle\big\|^2\Big] \\
&\overset{\text{Lem. 5}}{\geq} 1 - \frac{1}{2}\operatorname*{Exp}_{HIz}\Big[(d+1)\tilde{P}^{HIz}_{\text{find}}\Big] = 1 - \tfrac{1}{2}(d+1)P_{\text{find}}.
\end{aligned}$$

Here $(*)$ follows from the monotonicity of the fidelity [NC00, Thm. 9.6], and $(**)$ follows from the joint concavity of the fidelity [NC00, (9.95)]. This shows the first bound from the lemma.

The Bures distance $B$ is defined as $B(\rho,\tau)^2 = 2(1 - F(\rho,\tau))$. Thus

$$\begin{aligned}
B(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}})^2 &= 2(1 - F(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}})) \\
&\leq 2(1 - (1 - \tfrac{1}{2}(d+1)P_{\text{find}})) = (d+1)P_{\text{find}},
\end{aligned}$$

hence $B(\rho_{\text{left}}, \rho_{\text{right}}) \leq \sqrt{(d+1)P_{\text{find}}}$. □

**Theorem 1 (Semi-classical O2H – restated)** *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S.\ G(x) = H(x)$. Let $z$ be a random bitstring. ($S, G, H, z$ may have arbitrary joint distribution.)*

*Let $A$ be an oracle algorithm of query depth $d$ (not necessarily unitary).*

*Let*

$$\begin{aligned}
P_{\text{left}} &:= \Pr[b = 1 : b \leftarrow A^H(z)] \\
P_{\text{right}} &:= \Pr[b = 1 : b \leftarrow A^G(z)] \\
P_{\text{find}} &:= \Pr[\mathsf{Find} : A^{G\setminus S}(z)] \overset{\text{Lem. 1}}{=} \Pr[\mathsf{Find} : A^{H\setminus S}(z)]
\end{aligned} \tag{1}$$

*Then*

$$|P_{\text{left}} - P_{\text{right}}| \leq 2\sqrt{(d+1)\cdot P_{\text{find}}} \quad \textit{and} \quad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2\sqrt{(d+1)\cdot P_{\text{find}}}$$

*The theorem also holds with bound* $\sqrt{(d+1)P_{\text{find}}}$ *for the following alternative definitions of* $P_{\text{right}}$*:*

$$P_{\text{right}} := \Pr[b = 1 : b \leftarrow A^{H\setminus S}(z)], \tag{2}$$

$$P_{\text{right}} := \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)], \tag{3}$$

$$P_{\text{right}} := \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)], \tag{4}$$

$$P_{\text{right}} := \Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{H\setminus S}(z)], \tag{5}$$

$$P_{\text{right}} := \Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]. \tag{6}$$

*Proof.* We first prove the theorem using the definition of $P_{\text{right}}$ from (2).

Let $M$ be the measurement that measures, given the the register $Q_A, L$, what the output $b$ of $A$ is. Here $Q_A$ is the state space of $A$, and $L$ is the additional register introduced in Lemma 6. (Since $A$ obtains $b$ by measuring $Q_A$, such a measurement $M$ exists.)

Let $P_M(\rho)$ denote the probability that $M$ returns 1 when measuring a state $\rho$.

Then $P_{\text{left}} = P_M(\rho_{\text{left}} \otimes |0\rangle\langle 0|)$ and $P_{\text{right}} = P_M(\rho_{\text{right}})$ where $\rho_{\text{left}}$ and $\rho_{\text{right}}$ are defined in Lemma 6.

Then

$$\begin{aligned}
\left|P_{\text{left}} - P_{\text{right}}\right| &= \left|P_M(\rho_{\text{left}} \otimes |0\rangle\langle 0|) - P_M(P_{\text{right}})\right| \\
&\overset{\text{Lem. 4}}{\leq} B(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}}) \\
&\overset{\text{Lem. 6}}{\leq} \sqrt{(d+1)P_{\text{find}}} \\
\left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| &= \left|\sqrt{P_M(\rho_{\text{left}} \otimes |0\rangle\langle 0|)} - \sqrt{P_M(P_{\text{right}})}\right| \\
&\overset{\text{Lem. 4}}{\leq} B(\rho_{\text{left}} \otimes |0\rangle\langle 0|, \rho_{\text{right}}) \\
&\overset{\text{Lem. 6}}{\leq} \sqrt{(d+1)P_{\text{find}}}.
\end{aligned}$$

This shows the theorem with the definition of $P_{\text{right}}$ from (2).

Now we show the theorem using the definition of $P_{\text{right}}$ from (3). Let $M$ instead be the measurement that measures whether $b = 1$ and $L$ contains $|0\rangle$ (this means $\mathsf{Find}$ did not happen). Then $P_{\text{left}} = P_M(\rho_{\text{left}} \otimes |0\rangle\langle 0|)$ and $P_{\text{right}} = P_M(\rho_{\text{right}})$, and the rest of the proof is as in the case of (2).

Now we show the theorem using the definition of $P_{\text{right}}$ from (5). Let $M$ instead be the measurement that measures whether $b = 1$ or $L$ contains $|x\rangle$ for $x \neq 0$ (this means $\mathsf{Find}$ did happen). Then $P_{\text{left}} = P_M(\rho_{\text{left}} \otimes |0\rangle\langle 0|)$ and $P_{\text{right}} = P_M(\rho_{\text{right}})$, and the rest of the proof is as in the case of (2).

Now we show the theorem using the definition of $P_{\text{right}}$ from (4). This follows immediately by case (3), and the fact that $\Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)] = \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]$ by Lemma 1.

Now we show the theorem using the definition of $P_{\text{right}}$ from (6). By Lemma 1,

$$\Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)] = \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)] \tag{22}$$

$$\Pr[\text{true} \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)] = \Pr[\text{true} \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]. \tag{23}$$

From (23), we get (by considering the complementary event):

$$\Pr[\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)] = \Pr[\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]. \tag{24}$$

Adding (22) and (24), we get

$$\Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{H\setminus S}(z)] = \Pr[b = 1 \vee \mathsf{Find} : b \leftarrow A^{G\setminus S}(z)]. \tag{25}$$

Then case (6) follows from case (5) and the fact (25).

Now we show the theorem using the definition of $P_{\text{right}}$ from (1). Let

$$\begin{aligned} P_{\text{mid}} &:= \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{H\setminus S}(z)], \\ P'_{\text{mid}} &:= \Pr[b = 1 \wedge \neg\mathsf{Find} : b \leftarrow A^{G\setminus S}(z)], \\ P'_{\text{find}} &:= \Pr[\mathsf{Find} : A^{G\setminus S}(z)]. \end{aligned}$$

By the current lemma, case (3) (which we already proved), we have

$$|P_{\text{left}} - P_{\text{mid}}| \le \sqrt{(d+1)P_{\text{find}}}, \qquad |P_{\text{left}} - P_{\text{mid}}| \le \sqrt{(d+1)P_{\text{find}}},$$

and by case (4), we also get

$$|P_{\text{right}} - P'_{\text{mid}}| \le \sqrt{(d+1)P'_{\text{find}}}, \qquad |P_{\text{right}} - P'_{\text{mid}}| \le \sqrt{(d+1)P'_{\text{find}}},$$

Note that in the second case, we invoke the current lemma with $G$ and $H$ exchanged, and our $P_{\text{right}}$ is their $P_{\text{left}}$.

By Lemma 1, $P_{\text{mid}} = P'_{\text{mid}}$ and by (24), $P_{\text{find}} = P'_{\text{find}}$. With this and the triangle inequality, we get

$$|P_{\text{left}} - P_{\text{right}}| \le 2\sqrt{(d+1)P_{\text{find}}}, \qquad |P_{\text{left}} - P_{\text{right}}| \le 2\sqrt{(d+1)P_{\text{find}}}.$$

as required. □

### 5.3 Proof of Theorem 2

In the following, let $S \subseteq X$, $z \in \{0,1\}^*$.

**Lemma 7** *Fix $S, z$ ($S, z$ are not randomized in this lemma.) Let $A^H(z)$ be a unitary oracle algorithm with query depth $d$.*

*Let $B$ be an oracle algorithm that on input $z$ does the following: pick $i \xleftarrow{\$} \{1, \dots, d\}$, runs $A^{\mathcal{O}^{SC}_{\varnothing}}(z)$ until (just before) the $i$-th query, measure all query input registers in the computational basis, output the set $T$ of measurement outcomes.*

*Then*

$$\Pr[\mathsf{Find} : A^{\mathcal{O}^{SC}_{S}}(z)] \le 4q \cdot \Pr[S \cap T \neq \varnothing : T \leftarrow B(z)].$$

*Proof.* Let $|\Psi_i\rangle$ be the (non-normalized) state of $A^{\mathcal{O}_S^{SC}}(z)$ right after the $i$-th query in the case that the first $i$ queries return 0. That is, $\||\Psi_i\rangle\|^2$ is the probability that the first $i$ queries return 0, and $|\Psi_i\rangle/\||\Psi_i\rangle\|$ is the state conditioned on that outcome. Let $|\Psi'_i\rangle$ be the corresponding state of $A^{\mathcal{O}_\varnothing^{SC}}(z)$, that is, $|\Psi'_i\rangle$ is the state just after the $i$th query (or before, since queries to $\mathcal{O}_\varnothing^{SC}$ do not affect the state). Note that $|\Psi_0\rangle = |\Psi'_0\rangle$ is the initial state of $A(z)$ (independent of the oracle).

From the state $|\Psi_i\rangle$, the algorithm $A$ first applies a fixed unitary $U$ that depends only on $A$. Then it queries the semi-classical oracle $\mathcal{O}_S^{SC}$.

Let $P_S$ be the orthogonal projector projecting the query input registers $Q_1, \dots, Q_n$ onto states $|T\rangle$ with $S \cap T \neq \varnothing$, formally $P_S := \sum_{T \text{ s.t. } S\cap T\neq\varnothing} |T\rangle\langle T|$. Thus $\|P_S U|\Psi_i\rangle\|^2$ is the probability of measuring $T$ with $S \cap T \neq \varnothing$ in registers $Q_1, \dots, Q_n$ given the state $U|\Psi_i\rangle$.

Then the $i$-th query to $\mathcal{O}_S^{SC}$ applies $I - P_S$ to $|\Psi_i\rangle$. Therefore $|\Psi_{i+1}\rangle = (I - P_S)U|\Psi_i\rangle$.

Let $p_i = 1 - \||\Psi_i\rangle\|^2$ be the probability that one of the first $i$ queries returns 1, and let

$$\begin{aligned} r_i &:= p_i + 2\||\Psi_i\rangle - |\Psi'_i\rangle\|^2 = 1 - \||\Psi_i\rangle\|^2 + 2\||\Psi_i\rangle\|^2 - 4\Re\langle\Psi'_i|\Psi_i\rangle + 2\underbrace{\||\Psi'_i\rangle\|^2}_{=1} \\ &= 3 - 4\Re\langle\Psi'_i|\Psi_i\rangle + \||\Psi_i\rangle\|^2 \end{aligned} \tag{26}$$

Notice that $r_0 = 0$ since $|\Psi_0\rangle = |\Psi'_0\rangle$ and $\||\Psi_0\rangle\| = 1$. During the query, $U|\Psi_i\rangle$ is changed to $U|\Psi_i\rangle - P_S U|\Psi_i\rangle$, and $U|\Psi'_i\rangle$ stays the same, so that

$$\begin{aligned} |\Psi_{i+1}\rangle &= U|\Psi_i\rangle - P_S U|\Psi_i\rangle \\ |\Psi'_{i+1}\rangle &= U|\Psi'_i\rangle \end{aligned}$$

Therefore,

$$\begin{aligned} \||\Psi_{i+1}\rangle\|^2 &= \|U|\Psi_i\rangle\|^2 - \big\langle\Psi_i\big|U^\dagger P_S U\big|\Psi_i\big\rangle - \big\langle\Psi_i\big|U^\dagger P_S^\dagger U\big|\Psi_i\big\rangle + \big\langle\Psi_i\big|U^\dagger P_S^\dagger P_S U\big|\Psi_i\big\rangle \\ &= \||\Psi_i\rangle\|^2 - \big\langle\Psi_i\big|U^\dagger P_S U\big|\Psi_i\big\rangle \end{aligned} \tag{27}$$

because $P_S$ is a projector and thus $P_S^\dagger P_S = P_S^\dagger = P_S$. Likewise,

$$\begin{aligned} \langle\Psi'_{i+1}|\Psi_{i+1}\rangle &= \big\langle\Psi'_i\big|U^\dagger U\big|\Psi_i\big\rangle - \big\langle\Psi'_i\big|U^\dagger P_S U\big|\Psi_i\big\rangle \\ &= \langle\Psi'_i|\Psi_i\rangle - \big\langle\Psi'_i\big|U^\dagger P_S U\big|\Psi_i\big\rangle \end{aligned} \tag{28}$$

Let

$$g_i := \langle\Psi'_i|U^\dagger P_S U|\Psi'_i\rangle$$

be the probability that the algorithm $B$ returns $T$ with $S \cap T \neq \varnothing$ when measured at the $i$-th query.

We calculate

$$\begin{aligned} r_{i+1} - r_i &\overset{(26)}{=} -4\Re\langle\Psi'_{i+1}|\Psi_{i+1}\rangle + \||\Psi_{i+1}\rangle\|^2 + 4\Re\langle\Psi'_i|\Psi_i\rangle - \||\Psi_i\rangle\|^2 \\ &\overset{(27),(28)}{=} 4\Re\langle\Psi'_i|U^\dagger P_S U|\Psi_i\rangle - \langle\Psi_i|U^\dagger P_S U|\Psi_i\rangle \\ &= 4\langle\Psi'_i|U^\dagger P_S U|\Psi'_i\rangle - \underbrace{\langle 2\Psi'_i - \Psi|U^\dagger P_S U|2\Psi'_i - \Psi_i\rangle}_{\geq 0} \\ &\leq 4\langle\Psi'_i|U^\dagger P_S U|\Psi'_i\rangle = 4g_{i+1} \end{aligned}$$

Since $r_0 = 0$, by induction we have

$$\Pr[\mathsf{Find} : A^{\mathcal{O}_S^{SC}}(z)] = p_d \leq r_d \leq 4\sum_{i=1}^{d} g_i = 4d \cdot \Pr\left[S \cap T \neq \varnothing : T \leftarrow B(z)\right]$$

as claimed. □

**Theorem 2 (Search in semi-classical oracle – restated)** *Let $A$ be any quantum oracle algorithm making at most $q$ queries to a semi-classical oracle with domain $X$. Let $S \subseteq X$ and $z \in \{0,1\}^*$. ($S, z$ may have arbitrary joint distribution.)*

*Let $B$ be an algorithm that on input $z$ chooses $i \xleftarrow{\$} \{1,\dots,d\}$; runs $A^{\mathcal{O}_\varnothing^{SC}}(z)$ until (just before) the $i$-th query; then measures all query input registers in the computational basis and outputs the set $T$ of measurement outcomes.*

*Then*

$$\Pr[\mathsf{Find} : A^{\mathcal{O}_S^{SC}}(z)] \leq 4d \cdot \Pr[S \cap T \neq \varnothing : T \leftarrow B(z)] \tag{7}$$

*Proof.* Immediate from Lemma 7 by using the fact that $A$ can always be transformed into a unitary oracle algorithm, and by averaging. □

### 5.4 Proof of Theorem 3

In the following, let $G, H : X \to Y$, $S \subseteq X$, $z \in \{0,1\}^*$.

**Lemma 8 (One-way to hiding, pure states)** *Fix $G, H, S, z$ satisfying $\forall x \notin S.\ G(x) = H(x)$. ($G, H, S, z$ are not randomized in this lemma.) Let $A^H(z)$ be a unitary quantum oracle algorithm with query depth $d$. Let $Q_A$ denote the register containing all of $A$'s state.*

*Let $B$ be an oracle algorithm that on input $z$ does the following: pick $i \xleftarrow{\$} \{1,\dots,d\}$, run $A^H(z)$ until (just before) the $i$-th query, measure all query input registers in the computational basis, output the set $T$ of measurement outcomes.*

*Let $|\Psi_{\text{left}}\rangle$ be the final state of $A$ after running $A^H(z)$. And let $|\Psi_{\text{right}}\rangle$ be the final state of $A$ after running $A^G(z)$.*

*Let*

$$P_{\text{guess}} := \Pr[S \cap T \neq \varnothing : T \leftarrow B^H(z)]$$

*Then $\big\||\Psi_{\text{left}}\rangle - |\Psi_{\text{right}}\rangle\big\| \leq 2d\sqrt{P_{\text{guess}}}$.*

*Proof.* The state of $A$ is composed of three quantum systems $A, Q, R$ where $Q$,$R$ are the query and the response register for oracle queries. (That is, $Q$ consists of a number of registers $Q_1, \dots, Q_n$ where $r$ is the maximum number of queries performed in parallel, and $R$ consists of corresponding registers $R_1, \dots, R_n$.) Then an execution of $A^H(z)$ leads to the final state $(UO_H)^q|\Psi_0\rangle$ where $|\Psi_0\rangle$ is an initial state that depends on $z$ (but not on $G$, $H$, or $S$), $O_H : |a, q_1, \dots, q_n, r_1, \dots, r_n\rangle \mapsto |a, q_1, \dots, q_n, r_1 \oplus H(q_1), \dots, r_n \oplus H(q_n)\rangle$ is an oracle query, and $U$ is $A$'s state transition operation. (And analogously for $A^G$.)

We define $|\Psi_H^i\rangle := (UO_H)^i|\Psi_0\rangle$ and similarly $|\Psi_G^i\rangle$. Then $|\Psi_{\text{left}}\rangle = |\Psi_H^d\rangle$ and $|\Psi_{\text{right}}\rangle = |\Psi_G^d\rangle$.

And in our notation, we can describe $B$ as follows: $B^H(x)$ picks $i \xleftarrow{\$} \{1, \dots, d\}$ and $y \xleftarrow{\$} Y$, measures the quantum system Q of the state $|\Psi_H^{i-1}\rangle$ (this gives a list $T$ of inputs), and outputs the result $T$. Thus

$$P_{\text{guess}} = \tfrac{1}{q}\big\|P_S|\Psi_H^{i-1}\rangle\big\|^2 = \sum_{i=1}^{q} \tfrac{1}{q} B_i \qquad \text{with} \qquad B_i := \big\|P_S|\Psi_H^{i-1}\rangle\big\|^2. \tag{29}$$

Here $P_S$ is the orthogonal projector projecting $Q$ onto states $|T\rangle$ with $S \cap T \neq \varnothing$, formally $P_S := \sum_{T \text{ s.t. } S\cap T\neq\varnothing} |T\rangle\langle T|$. (I.e., $\|P_S|\Psi_H^{i-1}\rangle\|^2$ is the probability of measuring $T$ with $S \cap T \neq \varnothing$ in register $Q$ given the state $|\Psi_H^{i-1}\rangle$.)

Let $D_i := \big\||\Psi_H^i\rangle - |\Psi_G^i\rangle\big\|^2$. We have $D_0 = \big\||\Psi_0\rangle - |\Psi_0\rangle\big\|^2 = 0$, and for $i \geq 1$ we have:

$$\begin{aligned}
D_i &= \big\|UO_H|\Psi_H^{i-1}\rangle - UO_G|\Psi_G^{i-1}\rangle\big\|^2 \\
&\overset{(*)}{=} \big\|(O_H|\Psi_H^{i-1}\rangle - O_G|\Psi_H^{i-1}\rangle) + (O_G|\Psi_H^{i-1}\rangle - O_G|\Psi_G^{i-1}\rangle)\big\|^2 \\
&\overset{(**)}{\leq} \big\|(O_H - O_G)|\Psi_H^{i-1}\rangle\big\|^2 + \big\|O_G(|\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle)\big\|^2 \\
&\qquad + 2\big\|(O_H - O_G)|\Psi_H^{i-1}\rangle\big\| \cdot \big\|O_G(|\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle)\big\| \\
&\overset{(***)}{=} \big\|(O_H - O_G)P_S|\Psi_H^{i-1}\rangle\big\|^2 + \big\||\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle\big\|^2 \\
&\qquad + 2\big\|(O_H - O_G)P_S|\Psi_H^{i-1}\rangle\big\| \cdot \big\||\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle\big\| \\
&\overset{(****)}{\leq} 4\underbrace{\big\|P_S|\Psi_H^{i-1}\rangle\big\|^2}_{=B_i} + \underbrace{\big\||\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle\big\|^2}_{=D_{i-1}} \\
&\qquad + 4\underbrace{\big\|P_S|\Psi_H^{i-1}\rangle\big\|}_{=\sqrt{B_i}} \cdot \underbrace{\big\||\Psi_H^{i-1}\rangle - |\Psi_G^{i-1}\rangle\big\|}_{=\sqrt{D_i}} \\
&= 4B_i + D_{i-1} + 4\sqrt{B_i D_{i-1}} = (\sqrt{D_{i-1}} + 2\sqrt{B_i})^2.
\end{aligned} \tag{30}$$

Here $(*)$ uses that $U$ is unitary. And $(**)$ uses the inequality $\|a+b\|^2 \leq \|a\|^2 + \|b\|^2 + 2\|a\| \cdot \|b\|$. And $(***)$ uses that $(O_H - O_G)P_S = O_H - O_g$ since $G = H$ outside of $S$ (this can be verified by checking on all basis states $|a, q_1, \dots, r_1, \dots\rangle$), and that $O_G$ is unitary. And $(****)$ follows since $O_H - O_G$ has operator norm $\leq 2$.

From (30), we get $\sqrt{D_i} \le \sqrt{D_{i-1}} + 2\sqrt{B_i}$. This implies (with $D_0 = 0$) that

$$\sqrt{D_d} \le 2\sum_{i=1}^{d} \sqrt{B_i} = 2d \sum_{i=1}^{d} \tfrac{1}{d}\sqrt{B_i} \overset{(*)}{\le} 2d\sqrt{\sum_{i=1}^{d} \tfrac{1}{d} B_i} \overset{(29)}{=} 2d\sqrt{P_{\text{guess}}}$$

where $(*)$ follows from Jensen's inequality. By definition of $D_q$, this shows the lemma. □

**Lemma 9 (One-way to hiding, mixed states)** *Let $G, H, S, z$ be random satisfying $\forall x \notin S.\ G(x) = H(x)$. (With some joint distribution.)*

*Let $A$ be a quantum oracle algorithm with query depth $q$ (not necessarily unitary). Let $B$ and $P_{\text{guess}}$ be as in Theorem 3.*

*Let $\rho_{\text{left}}$ be the final state of $A^H(z)$ and let $\rho_{\text{right}}$ be the final state of $A^G(z)$*

*Then $F(\rho_{\text{left}}, \rho_{\text{right}}) \ge 1 - 2d^2 P_{\text{guess}}$ and $B(\rho_{\text{left}}, \rho_{\text{right}}) \le 2d\sqrt{P_{\text{guess}}}$.*

*Proof.* Without loss of generality, we can assume that $A$ is unitary during the execution, and applies a quantum operation $\mathcal{E}$ to its state in the last step. (Note that transforming an adversary $A$ into a unitary adversary $A'$ may change the internal state during the execution because additional auxiliary qubits are used to simulate measurements. However, this does not affect the probability $P_{\text{guess}}$ because $B$ does not measure those auxiliary qubits of $A'$.)

For fixed $G, H, S, z$, let $|\Psi^{HSz}_{\text{left}}\rangle, |\Psi^{GSz}_{\text{right}}\rangle, P^{HSz}_{\text{guess}}$ refer to the values $|\Psi_{\text{left}}\rangle, |\Psi_{\text{right}}\rangle, P_{\text{guess}}$ from Lemma 8 for those fixed $G, H, S, z$.

Let $\hat{\rho}_{\text{left}}$ and $\hat{\rho}_{\text{right}}$ refer to the state of $A$ before applying $\mathcal{E}$ in the games defining $\hat{\rho}_{\text{left}}$ and $\hat{\rho}_{\text{right}}$, respectively.

Then

$$\hat{\rho}_{\text{left}} = \operatorname*{Exp}_{GHSz}\left[|\Psi^{HSz}_{\text{left}}\rangle\langle\Psi^{HSz}_{\text{left}}|\right] \qquad \text{and}$$
$$\hat{\rho}_{\text{right}} = \operatorname*{Exp}_{GHSz}\left[|\Psi^{GSz}_{\text{right}}\rangle\langle\Psi^{GSz}_{\text{right}}|\right].$$

Thus we have

$$\begin{aligned}
F(\rho_{\text{left}}, \rho_{\text{right}}) &= F(\mathcal{E}(\hat{\rho}_{\text{left}}), \mathcal{E}(\hat{\rho}_{\text{right}})) \overset{(*)}{\ge} F(\hat{\rho}_{\text{left}}, \hat{\rho}_{\text{right}}) \\
&= F\Big(\operatorname*{Exp}_{HGSz}[|\Psi^{HSz}_{\text{left}}\rangle\langle\Psi^{HSz}_{\text{left}}|], \operatorname*{Exp}_{HGSz}[|\Psi^{GSz}_{\text{right}}\rangle\langle\Psi^{GSz}_{\text{right}}|]\Big) \\
&\overset{(**)}{\ge} \operatorname*{Exp}_{HGSz}[F\big(|\Psi^{HSz}_{\text{left}}\rangle\langle\Psi^{HSz}_{\text{left}}|, |\Psi^{GSz}_{\text{right}}\rangle\langle\Psi^{GSz}_{\text{right}}|\big)] \\
&\overset{\text{Lemma 3}}{\ge} \operatorname*{Exp}_{HGSz}\left[1 - \tfrac{1}{2}\left\||\Psi^{HSz}_{\text{left}}\rangle - |\Psi^{GSz}_{\text{right}}\rangle\right\|^2\right] \\
&\overset{\text{Lemma 8}}{\ge} \operatorname*{Exp}_{HGSz}\left[1 - \tfrac{1}{2}(4dP^{HSz}_{\text{guess}})\right] \overset{(***)}{=} 1 - 2d^2 P_{\text{guess}}.
\end{aligned}$$

Here $(*)$ follows from the monotonicity of the fidelity [NC00, Thm. 9.6], and $(**)$ follows from the joint concavity of the fidelity [NC00, (9.95)]. And $(***)$ follows since $P_{\text{guess}} = \text{Exp}_{HGSz}\big[P^{HSz}_{\text{guess}}\big]$.

The Bures distance $B$ is defined as $B(\rho,\tau)^2 = 2(1-F(\rho,\tau))$. Thus

$$B(\rho_{\text{left}},\rho_{\text{right}})^2 = 2(1-F(\rho_{\text{left}},\rho_{\text{right}})) \leq 2(1-(1-2d^2P_{\text{guess}})) = 4d^2P_{\text{guess}},$$

hence $B(\rho_{\text{left}},\rho_{\text{right}}) \leq 2d\sqrt{P_{\text{guess}}}$, as claimed. □

**Theorem 3 (One-way to hiding, probabilities – restated)** *Let $S \subseteq X$ be random. Let $G, H : X \to Y$ be random functions satisfying $\forall x \notin S.\, G(x) = H(x)$. Let $z$ be a random bitstring. ($S, G, H, z$ may have arbitrary joint distribution.)*

*Let $A$ be quantum oracle algorithm with query depth $d$ (not necessarily unitary).*

*Let $B^H$ be an oracle algorithm that on input $z$ does the following: pick $i \xleftarrow{\$} \{1,\dots,d\}$, run $A^H(z)$ until (just before) the $i$-th query, measure all query input registers in the computational basis, output the set $T$ of measurement outcomes.*

*Let*

$$\begin{aligned} P_{\text{left}} &:= \Pr[b = 1 : b \leftarrow A^H(z)] \\ P_{\text{right}} &:= \Pr[b = 1 : b \leftarrow A^G(z)] \\ P_{\text{guess}} &:= \Pr[S \cap T \neq \varnothing : T \leftarrow B^H(z)] \end{aligned}$$

*Then*

$$|P_{\text{left}} - P_{\text{right}}| \leq 2d\sqrt{P_{\text{guess}}} \qquad \textit{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq 2d\sqrt{P_{\text{guess}}}$$

*The same result holds with $B^G$ instead of $B^H$ in the definition of $P_{\text{guess}}$.*

*Proof.* The output bit $b$ of $A$ is the result of a measurement $M$ applied to its final state. Thus, with $\rho_{A,1}, \rho_{A,2}$ as in Lemma 9, $P_{\text{left}}, P_{\text{right}}$ is the probability that the measurement $M$ returns 1 when measuring $\rho_{\text{left}}, \rho_{\text{right}}$, respectively. By Lemma 4,

$$\left|P_{\text{left}} - P_{\text{right}}\right| \leq B(\rho_{\text{left}},\rho_{\text{right}}) \qquad \text{and} \qquad \left|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}\right| \leq B(\rho_{\text{left}},\rho_{\text{right}})$$

By Lemma 9, $B(\rho_{\text{left}},\rho_{\text{right}}) \leq 2d\sqrt{P_B}$. The corollary follows. □

**Acknowledgements.** Thanks to Daniel Kane, Eike Kiltz, and Kathrin Hövelmanns for valuable discussions. Unruh was supported by institutional research funding IUT2-1 of the Estonian Ministry of Education and Research, the United States Air Force Office of Scientific Research (AFOSR) via AOARD Grant "Verification of Quantum Cryptography" (FA2386-17-1-4022), the Mobilitas Plus grant MOBERC12 of the Estonian Research Council, and the Estonian Centre of Exellence in IT (EXCITE) funded by ERDF.

# A Optimality of Corollary 1

**Lemma 10** *If $S = \{x\}$ where $x \xleftarrow{\$} \{1, \ldots, N\}$, then there is a $q$-query algorithm $A^{\mathcal{O}_S^{SC}}$ such that*

$$\Pr[\mathsf{Find} : A^{\mathcal{O}_S^{SC}}()] \geq \frac{4q-3}{N} - \frac{8q(q-1)}{N^2}$$

*Proof.* The algorithm is as follows:

- Make the first query with amplitude $1/\sqrt{N}$ in all positions.
- Between queries, transform the state by the unitary $U := 2E/N - I$ where $E$ is the matrix containing 1 everywhere. That $U$ is unitary follows since $U^\dagger U = 4E^2/N^2 - 4E/N + I = I$ using $E^2 = NE$.

One may calculate by induction that the final non-normalized state has amplitude

$$\left(1 - \frac{2}{N}\right)^{q-1} \cdot \frac{1}{\sqrt{N}}$$

in all positions except for the $x$th one (where the amplitude is 0), so its squared norm is

$$1 - \Pr[\mathsf{Find}] = \left(1 - \frac{2}{N}\right)^{2q-2} \cdot \frac{1}{N} \cdot (N-1) = \left(1 - \frac{2}{N}\right)^{2q-2} \cdot \left(1 - \frac{1}{N}\right)$$

As a function of $1/N$, this expression's derivatives alternate on $[0, 1/2]$, so it is below its second-order Taylor expansion:

$$1 - \Pr[\mathsf{Find}] \leq 1 - \frac{4q-3}{N} + \frac{8q(q-1)}{N^2}$$

This completes the proof. □

# B Improved proof of the Targhi-Unruh transform

In this section we show how to adapt the security proof from [TU16] (of their variant of the Fujisaki-Okamoto transform) to the our new O2H Theorem. To make it easier to compare the original and the new proof, we stick as closely as possible to the original proof and its notation, reproducing text verbatim where the proof does not change. (In particular, the advantage of the adversary against the underlying schemes are written $\mathtt{negl}(\mathtt{n})^{\mathtt{sy}}$ and $\mathtt{negl}(\mathtt{n})^{\mathtt{asy}}$, not $\varepsilon_{sym}$ and $\varepsilon$ as in the comparison table in Figure 1. And the term $\omega(\log(n))$ corresponds to $\gamma$ in Figure 1, see footnote 5.)

We have extended the proof to compute security bounds both for the case that the underlying public-key encryption scheme is one-way, and the case that it is IND-CPA. (Since with the new O2H Theorem, we get different bounds in both cases.)

**Construction.** We combine an asymmetric encryption scheme with a symmetric encryption scheme by using three hash functions in order to gain an IND-CCA secure public encryption scheme $\Pi^{hy} = (Gen^{hy}, Enc^{hy}, Dec^{hy})$ in the quantum random oracle model.

Let $\Pi^{asy} = (Gen^{asy}, Enc^{asy}, Dec^{asy})$ be an asymmetric encryption scheme with the message space $\mathtt{MSP}^{asy} = \{0,1\}^{n_1}$ and the coin space $\mathtt{COIN}^{asy} = \{0,1\}^{n_2}$. Let $\Pi^{sy} = (Enc^{sy}, Dec^{sy})$ be a symmetric encryption scheme where $\mathtt{MSP}^{sy}$ and $\mathtt{KSP}^{sy} = \{0,1\}^m$ are its message space and key space, respectively. The parameters $n_1$, $n_2$ and $m$ depend on the security parameter $n$. We define three hash functions:

$$G : \mathtt{MSP}^{asy} \to \mathtt{KSP}^{sy},\ H : \{0,1\}^* \to \mathtt{COIN}^{asy} \text{ and } \ H' : \mathtt{MSP}^{asy} \to \mathtt{MSP}^{asy}.$$

These hash functions will be modeled as random oracles in the following.

The hybrid scheme $\Pi^{hy} = (Gen^{hy}, Enc^{hy}, Dec^{hy})$ is constructed as follows, with $\mathtt{MSP}^{hy}$ as its message space:

1. $Gen^{hy}$, the key generation algorithm, on input $1^n$ runs $Gen^{asy}$ to obtain a pair of keys $(pk, sk)$.
2. $Enc^{hy}$, the encryption algorithm, on input $pk$ and message $m \in \mathtt{MSP}^{hy} := \mathtt{MSP}^{sy}$ does the following:
   - Select $\delta \xleftarrow{\$} \mathtt{MSP}^{asy}$.
   - Compute $c \leftarrow Enc_a^{sy}(m)$, where $a := G(\delta)$.
   - Compute $e := Enc_{pk}^{asy}(\delta; h)$, where $h := H(\delta \| c)$.
   - Finally, output $(e, c, d)$ as $Enc_{pk}^{hy}(m; \delta)$, where $d := H'(\delta)$.
3. $Dec^{hy}$, the decryption algorithm, on input $sk$ and ciphertext $(e, c, d)$ does the following:
   - Compute $\hat{\delta} := Dec_{sk}^{asy}(e)$.
   - If $\hat{\delta} = \perp$: abort and output $\perp$.
   - Otherwise set $\hat{h} := H(\hat{\delta} \| c)$.
   - If $e \neq Enc_{pk}^{asy}(\hat{\delta}; \hat{h})$: abort and output $\perp$.
   - Else if $d = H'(\hat{\delta})$:
     - Compute $\hat{a} := G(\hat{\delta})$ and output $Dec_{\hat{a}}^{sy}(c)$.
   - Else output $\perp$.

Note that our construction is the same as the Fujisaki-Okamoto construction, except that we use an extra random oracle $H'$. Consequently, the ciphertext has one more component, the encryption algorithm has an additional instruction to compute $H'(\delta)$ and the decryption algorithm has an additional check corresponding to $H'$.

**Theorem 4** *The hybrid scheme $\Pi^{hy}$ constructed above is IND-CCA secure in the quantum random oracle model if $\Pi^{sy}$ is an one-time secure symmetric encryption scheme and $\Pi^{asy}$ is a well-spread*[5] *one-way/IND-CPA secure asymmetric encryption scheme.*

*Proof.* Let $A_{hy}$ be a quantum polynomial time adversary that attacks $\Pi^{hy}$ in the sense of IND-CCA in the quantum random oracle model. Let $q$ denote an upper bound on the queries to $H$, $G$, $H'$, and the decryption queries performed by $A_{hy}$. Let $q_{dec}$ be a bound on the decryption queries alone. (Since decryption queries are typically more "expensive", we will sometimes use $q_{dec}$ to highlight that a certain term depends only on decryption queries.) Let $\Omega_H$, $\Omega_G$, $\Omega_{H'}$ be the set of all function $H : \{0,1\}^* \to \{0,1\}^{n_2}$, $G : \{0,1\}^{n_1} \to \{0,1\}^m$ and $H' : \{0,1\}^{n_1} \to \{0,1\}^{n_1}$, respectively. The following game shows the chosen ciphertext attack by the adversary $A_{hy}$ in the quantum setting where the adversary $A_{hy}$ has quantum access to the random oracles $H$, $G$ and $H'$ and classical access to the decryption algorithm $Dec^{hy}$.

**Game 0:**

**let** $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $H' \xleftarrow{\$} \Omega_{H'}$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $(pk, sk) \leftarrow Gen^{asy}(1^n)$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^{hy}}(pk)$
**let** $b \xleftarrow{\$} \{0,1\}$, $c^* \leftarrow Enc^{sy}_{G(\delta^*)}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$, $d^* := H'(\delta^*)$
**let** $b' \leftarrow A_{hy}^{H,G,H',Dec^{hy}}(e^*, c^*, d^*)$
**return** $[b = b']$

In order to show that the success probability of Game 0 is close to 1/2, we shall introduce a sequence of games and compute the difference between their success probabilities. For simplicity, we omit the definitions of random variables that appear with the same distribution and without any changes in all of the following games. These random variables are: $H \xleftarrow{\$} \Omega_H, G \xleftarrow{\$} \Omega_G$, $\delta^* \xleftarrow{\$} \mathtt{MSP}^{asy}, (pk, sk) \leftarrow Gen^{asy}(1^n)$, and $b \xleftarrow{\$} \{0,1\}$.

In the next game, we replace the decryption algorithm $Dec^{hy}$ with $Dec^*$ where $Dec^*$ on $(e, c, d)$ does the following:

1. If $e^*$ is defined and $e = e^*$: return $\perp$.
2. Else do:
   - Compute $\hat{\delta} := Dec^{asy}_{sk}(e)$.
   - If $\hat{\delta} = \perp$: return $\perp$.
   - Otherwise set $\hat{h} := H(\hat{\delta} \| c)$.
   - If $e \neq Enc^{asy}_{pk}(\hat{\delta}; \hat{h})$: return $\perp$.
   - Else if $d = H'(\hat{\delta})$: compute $\hat{a} := G(\hat{\delta})$ and return $Dec^{sy}_{\hat{a}}(c)$.

[5] Meaning that a ciphertext has min-entropy at least $\omega(\log(n))$.

- Else: return $\bot$.

We have slightly changed the definition of $Dec^*$ in comparison with [TU16]. Namely, in [TU16], $Dec^*$ contained some queries to $H'$ whose output were ignored (they were needed to keep the oracle queries in sync between different games). We removed those because they turn out not to be needed here.

Therefore, Game 1 is as follows:

**Game 1:**

**let** $H' \xleftarrow{\$} \Omega_{H'}$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
**let** $c^* \leftarrow Enc^{sy}_{G(\delta^*)}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$
**let** $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, H'(\delta^*))$
**return** $[b = b']$

We prove that the probabilities of success in Game 0 and Game 1 have negligible difference. We can conclude the result by the fact that the asymmetric encryption scheme is well-spread. The following lemma is shown in [TU16]. (Games 0 and 1 here are identical to Games 0 and 1 in [TU16] except that we removed some oracle queries that have no effect.)

**Lemma 11 ([TU16])** *If the asymmetric encryption scheme $\Pi^{asy}$ is well-spread, then*

$$\Big|\Pr[1 \leftarrow Game\ 0] - \Pr[1 \leftarrow Game\ 1]\Big| \leq O\left(\frac{O(q^{9/5})}{2^{\omega(\log(n))/5}}\right) =: \ell(n). \tag{31}$$

**Game 1b:**

**let** $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G(\delta^*:=d^*),H'(\delta^*:=a^*),\widetilde{Dec}^*}(pk)$
**let** $c^* \leftarrow Enc^{sy}_{a^*}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; H(\delta^* \| c^*))$
**let** $b' \leftarrow A_{hy}^{H,G(\delta^*:=d^*),H'(\delta^*:=a^*),\widetilde{Dec}^*}(e^*, c^*, d^*)$
**return** $[b = b']$

Here $G(\delta^* := d^*)$ refers to the function $G$, except that $G(\delta^* := d^*)$ returns $d^*$ on input $\delta^*$. Analogously $H'(\delta^* := a^*)$. And $\widetilde{Dec}^*$ is $Dec^*$, except that all occurrences of $G$ and $H'$ are replaced by $G(\delta^* := d^*)$ and $H'(\delta^* := a^*)$.

Since $G$ and $H'$ are uniformly random, replacing them everywhere by $G(\delta^* := d^*)$ and $H'(\delta^* := a^*)$ (for fresh uniformly random $a^*, d^*$) does not change their distribution. And replacing invocations $G(\delta^*)$ and $H'(\delta^*)$ by $a^*$ and $d^*$ does not change the game either because $G(\delta^*)$ and $H'(\delta^*)$ give those outputs anyway. Thus

$$\Pr[1 \leftarrow Game\ 1] = \Pr[1 \leftarrow Game\ 1b]. \tag{32}$$

**Game 2:**

**let** $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G,H',Dec^*}(pk)$
**let** $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
**let** $b' \leftarrow A_{hy}^{H,G,H',Dec^*}(e^*, c^*, d^*)$
**return** $[b = b']$

In [TU16], it is shown that

$$\Pr[1 \leftarrow \mathit{Game}\ 2] = 1/2 \pm \mathsf{negl}(\mathsf{n})^{sy}. \tag{33}$$

Since that proof does not change in the present setting, we omit it here.

We use the O2H Theorem 1 to obtain an upper bound for $\Big|\Pr[1 \leftarrow \mathit{Game}\ 1b] - \Pr[1 \leftarrow \mathit{Game}\ 2]\Big|$. (In [TU16], the original O2H Theorem from [Unr15b] was used instead.)

The only difference between Game 1b and Game 2 is that Game 2 uses $G(\delta^* := d^*)$ and $H'(\delta^* := a^*)$ instead of $G$ and $H'$. We can therefore apply Theorem 1 to replace $G(\delta^* := d^*)$ and $H'(\delta^* := a^*)$ by $G$ and $H'$. Specifically, in Theorem 1, let $G$ be $(G, H')$ (the function that on input $\delta$ returns $(G(\delta), H'(\delta))$), let $H$ be $(G(\delta^* := d^*), H'(\delta^* := a^*))$, let $S := \{\delta\}$, let $z := \delta^*$, and let $A^{(G,H')}(\delta^*)$ be the algorithm that simulates Game 2 (picking $H$ itself). Then $P_{\text{right}} = \Pr[1 \leftarrow A^{(G,H')}(\delta^*)] = \Pr[1 \leftarrow \mathit{Game}\ 2]$ and $P_{\text{left}} = \Pr[1 \leftarrow A^{(G(\delta^*:=d^*),H'(\delta^*:=a^*))}(\delta^*)] = \Pr[1 \leftarrow \mathit{Game}\ 1b]$. And by Theorem 1, $|P_{\text{left}} - P_{\text{right}}| \leq \sqrt{O(q)P_{\text{find}}}$ where $P_{\text{find}} = \Pr[\mathsf{Find}_{GH'} : A^{(G,H')\setminus\{\delta^*\}}(\delta^*)] = \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 3]$. (We write $\mathsf{Find}_{GH'}$ instead of $\mathsf{Find}$ here to distinguish it from the event $\mathsf{Find}_H$ introduced below. $\mathsf{Find}_{GH'}$ refers to a $\mathsf{Find}$-event raised due to a $\delta^*$ query to $G'$ or $H'$.) Here Game 3 is as defined below, the result from replacing $G$ and $H'$ by punctured oracles $G \setminus \{\delta^*\}$ and $G' \setminus \{\delta^*\}$ in Game 2. Thus

$$|\Pr[1 \leftarrow \mathit{Game}\ 2] - \Pr[1 \leftarrow \mathit{Game}\ 1b]| \leq \sqrt{O(q)\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 3]}. \tag{34}$$

**Game 3:**

**let** $H' \xleftarrow{\$} \Omega_{H'}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^*\setminus\{\delta^*\}}(pk)$
**let** $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
**let** $b' \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^*\setminus\{\delta^*\}}(e^*, c^*, d^*)$

Here $Dec^* \setminus \{\delta^*\}$ denotes $Dec^*$ with all invocations of $G$ and $H'$ replaced by $G \setminus \{\delta^*\}$ and $H' \setminus \{\delta^*\}$.

Note that in comparison with [TU16], Game 3 here is a bit different: Our Game 3 simply is Game 2 with punctured oracles. In [TU16], Game 3 is an execution of Game 2 which stops at a randomly chosen query, measures that query, and then compares the outcome of that measurement with $\delta^*$.

In the next game, we replace the random oracle $H'$ with a $2q$-wise independent function. Random polynomials of degree $2q-1$ over the finite field $GF(2^{n_1})$ are $2q$-wise independent. Let $\Omega_{wise}$ be the set of all such polynomials.

**Game 4:**

**let** $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^*\setminus\{\delta^*\}}(pk)$
**let** $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^* \| c^*))$
**let** $b' \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^*\setminus\{\delta^*\}}(e^*, c^*, d^*)$

Due to a result by Zhandry [Zha12b], a $2q$-wise independent function $H'$ is perfectly indistinguishable from a random function when the adversary makes at most $q$ queries to $H'$. Therefore,

$$\Pr[1 \leftarrow \mathit{Game}\ 3] = \Pr[1 \leftarrow \mathit{Game}\ 4]. \tag{35}$$

We replace the decryption algorithm $Dec^* \setminus \{\delta^*\}$ with a new decryption algorithm $Dec^{**}$ in Game 5. $Dec^{**}$ has access to the description (as a polynomial) of $H'$. $Dec^{**}$ on input $(e, c, d)$ works as follows:

1. If $e^*$ is defined and $e = e^*$: output $\perp$.
2. Else do:
   - Calculate all roots of the polynomial $H' - d$. Let $S$ be the set of those roots.
   - If there exists a $\hat{\delta} \in S$ such that $e = Enc_{pk}^{asy}\big(\hat{\delta}; H(\hat{\delta} \| c)\big)$:
     - compute $\hat{a} := (G \setminus \{\delta^*\})(\hat{\delta})$ and return $Dec_{\hat{a}}^{sy}(c)$.
   - Else: output $\perp$.

We emphasise that finding roots of polynomial $H' - d$ is possible in polynomial time [Ben81] and it does not involve queries to $H'$ or $H' \setminus \{\delta^*\}$.

This definition of $Dec^{**}$ is different from [TU16] not because of the use of a different O2H Theorem, but in order to fix a mistake in the proof from [TU16]. Namely, in [TU16], $Dec^{**}$ directly accesses $\delta^*$, but later when using the one-wayness of the asymmetric encryption scheme, they erroneously use that the final game (that contains $Dec^{**}$) does not access $\delta^*$ directly.

**Game 5:**

$\mathbf{let}\ H' \xleftarrow{\$} \Omega_{wise},\ a^* \xleftarrow{\$} \mathtt{KSP}^{sy},\ d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$
$\mathbf{let}\ m_0, m_1 \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}}(pk)$
$\mathbf{let}\ c^* \leftarrow Enc_{a^*}^{sy}(m_b),\ e^* \leftarrow Enc_{pk}^{asy}(\delta^*; H(\delta^*\|c^*))$
$\mathbf{let}\ b' \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}}(e^*, c^*, d^*)$

We would like to show that $\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 4] \leq \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5]$, but that is not actually true. There is one situation in which the adversary might cause $Dec^* \setminus \{\delta^*\}$ to query $(H' \setminus \{\delta^*\})(\delta^*)$ with no corresponding query being made by $Dec^{**}$. This can happen in the following event $\mathsf{BlindGuess}$: Let $\mathsf{BlindGuess}$ denote the event that $Dec^* \setminus \{\delta^*\}$ or $Dec^{**}$ is queried with input $(e, c, d)$ while $e^* = \bot$ (not challenge query has been performed yet) and $Dec_{sk}^{asy}(e) = \delta^*$.

We show that $\Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4] \leq O(q)2^{-n_1}$: If we define Game 4' to run Game 4 but stop right after the first invocation of $A_{hy}$ (i.e., before $e^*$ is defined), then $\Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4] = \Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4']$. And Game 4' never accesses $\delta^*$, except that $H'$ and $G$ are punctured at $\delta^*$. And Game 4' makes at most $O(q)$ queries to $G$ and $H'$ combined. Thus $\Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4'] \leq O(q)2^{-n_1}$ by Corollary 1.

We now show that

$$\Pr[\mathsf{Find}_{GH'} \wedge \neg\mathsf{BlindGuess} : \mathit{Game}\ 4] \leq \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5]. \tag{36}$$

For this it is sufficient to show that when $Dec^* \setminus \{\delta^*\}$ and $Dec^{**}$ are queried with the same input $(c, d, e)$, they return the same value, and that if in $Dec^* \setminus \{\delta^*\}$, $\mathsf{Find}_{GH'}$ happens but not $\mathsf{BlindGuess}$, then in $Dec^{**}$, $\mathsf{Find}_{GH'}$ happens. We distinguish the following cases (where $\delta_{true} := Dec_{sk}^{asy}(e)$):

- **Case $e = e^*$:** $Dec^* \setminus \{\delta^*\}$ and $Dec^{**}$ both return $\bot$. $Dec^* \setminus \{\delta^*\}$ performs no queries to $G \setminus \{\delta^*\}, H' \setminus \{\delta^*\}$, so $\mathsf{Find}_{GH'}$ does not occur.

- **Case $\delta_{true} = \bot$ (and not $e = e^*$):** $Dec^* \setminus \{\delta^*\}$ returns $\bot$. Since we assume perfect correctness of $Enc^{asy}$, if $e = Enc_{pk}^{asy}\big(\hat{\delta}; H(\hat{\delta}\|c)\big)$ for some $\hat{\delta}$, then $Dec_{sk}^{asy}(e) = \hat{\delta}$ and hence $\bot = \delta_{true} = Dec_{sk}^{asy}(e) = \hat{\delta} \neq \bot$. Thus there is no such $\hat{\delta}$, and $Dec^{**}$ returns $\bot$.

  $Dec^* \setminus \{\delta^*\}$ performs no queries to $G \setminus \{\delta^*\}, H' \setminus \{\delta^*\}$, so $\mathsf{Find}_{GH'}$ does not occur in $Dec^* \setminus \{\delta^*\}$.

- **Case $e \neq Enc_{pk}^{asy}\big(\delta_{true}; H(\delta_{true}\|c)\big)$ (and neither of the above):** $Dec^* \setminus \{\delta^*\}$ returns $\bot$. And since we assume perfect correctness, we also have $e \neq Enc_{pk}^{asy}\big(\hat{\delta}; H(\hat{\delta}\|c)\big)$ for any $\hat{\delta} \neq Dec_{sk}^{asy}(e) = \delta_{true}$. Thus $Dec^{**}$ returns $\bot$.

  And $Dec^* \setminus \{\delta^*\}$ does not query $G \setminus \{\delta^*\}, H' \setminus \{\delta^*\}$, so $\mathsf{Find}_{GH'}$ does not occur in $Dec^* \setminus \{\delta^*\}$.

- **Case $d = H'(\delta_{true})$ (and none of the above):** $Dec^* \setminus \{\delta^*\}$ returns $Dec_{G(\delta_{true})}^{sy}(c)$. And in $Dec^{**}$, since $d = H'(\delta_{true})$, we have $\delta_{true} \in S$. And since $e =$

$Enc^{asy}_{pk}\big(\delta_{true}; H(\delta_{true}\|c)\big)$ and we have perfect correctness, there cannot be another $\hat{\delta}$ with $e = Enc^{asy}_{pk}\big(\hat{\delta}; H(\hat{\delta}\|c)\big)$. Thus $Dec^{**}$ returns $Dec_{G(\delta_{true})}(c)$ as well.

$Dec^* \setminus \{\delta^*\}$ queries $(H' \setminus \{\delta^*\})(\delta_{true})$ and $(G \setminus \{\delta^*\})(\delta_{true})$, so $\mathsf{Find}_{GH'}$ occurs here iff $\delta_{true} = \delta^*$. And $Dec^{**}$ queries $(G \setminus \{\delta^*\})(\delta_{true})$, so $\mathsf{Find}_{GH'}$ occurs iff $\delta_{true} = \delta^*$ here as well. So $\mathsf{Find}_{GH'}$ occurs in $Dec^* \setminus \{\delta^*\}$ iff it occurs in $Dec^{**}$.

- **Case $\delta_{true} \neq \delta^*$ (and none of the above):** $Dec^* \setminus \{\delta^*\}$ returns $\bot$ since $d \neq H'(\delta_{true})$. Since $e = Enc^{asy}_{pk}\big(\delta_{true}; H(\delta_{true}\|c)\big)$, and using perfect correctness, we know that there can be no $\hat{\delta} \neq \delta_{true}$ satisfying $e = Enc^{asy}_{pk}\big(\hat{\delta}; H(\hat{\delta}\|c)\big)$. And since $d \neq H'(\delta_{true})$, $\delta_{true} \notin S$. Thus $Dec^{**}$ returns $\bot$.

  $Dec^* \setminus \{\delta^*\}$ only queries $H(\delta_{true}\|c)$ and $(H' \setminus \{\delta^*\})(\delta_{true})$. Thus, since $\delta_{true} \neq \delta^*$, $\mathsf{Find}_{GH'}$ does not occur in $Dec^* \setminus \{\delta^*\}$.

- **None of the above:** As in the previous case, we have that $Dec^* \setminus \{\delta^*\}$ and $Dec^{**}$ both return $\bot$.

  If $e^* \neq \bot$, then $e^* = Enc^{asy}_{pk}\big(\delta^*; H(\delta^*\|c)\big)$ by definition of $e^*$. And $e = Enc^{asy}_{pk}\big(\delta_{true}; H(\delta_{true}\|c)\big)$ (otherwise we would be in a prior case). Furthermore, $\delta_{true} = \delta^*$ (otherwise we would be in the previous case). Thus $e = e^*$. But then we would be in the first case. So we conclude that $e^* = \bot$.

  Since $e^* = \bot$ and $Dec^{asy}_{sk}(e) = \delta_{true} = \delta^*$, the event $\mathsf{BlindGuess}$ occurs in $Dec^* \setminus \{\delta^*\}$.

Thus we have shown in all cases that $Dec^* \setminus \{\delta^*\}$ and $Dec^{**}$ return the same value, and that if in $Dec^* \setminus \{\delta^*\}$, $\mathsf{Find}_{GH'}$ happens but not $\mathsf{BlindGuess}$, then in $Dec^{**}$, $\mathsf{Find}_{GH'}$ happens. (36) follows. Thus

$$\begin{aligned}\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 4] &\leq \Pr[\mathsf{Find}_{GH'} \wedge \neg\mathsf{BlindGuess} : \mathit{Game}\ 4] + \Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4] \\ &\overset{(36)}{\leq} \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5] + \Pr[\mathsf{BlindGuess} : \mathit{Game}\ 4] \\ &\leq \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5] + O(q)2^{-n_1}. \quad (37)\end{aligned}$$

Note that this analysis of the differences between $Dec^*$ and $Dec^{**}$ differs from the proof from [TU16]. We stress that this particular change is unrelated to the introduction of the new O2H Theorem but is a consequence of the fact that we changed the definition of $Dec^{**}$ above to avoid a mistake in [TU16]. However, this change is made easier by the new O2H Theorem because we only need to show that $\mathsf{Find}_{GH'}$ in one game implies $\mathsf{Find}_{GH'}$ in another game, instead of having to show that all oracle calls are in sync. Nevertheless, we conjecture that a similar change could be made to repair the proof in [TU16].

Note that $Dec^{**}$ does not use the secret key of the asymmetric encryption scheme to decrypt the ciphertext. This will allow us below to make use of the one-way / IND-CPA security of $\Pi^{asy}$. (This is only possible if the secret key is never used.)

The next step is to replace the random coins $H(\delta^*\|c^*)$ of the asymmetric encryption scheme by truly random coins from $\mathtt{COIN}^{asy}$. It will facilitate the use of the O2H Theorem

below if we do not just change $H(\delta^*\|c^*)$ but all outputs of the form $H(\delta^*\|\cdot)$. This is easier because $c^*$ is not known at the beginning of the game. If we were to change only $H(\delta^*\|c^*)$, we would need an adaptive version of the O2H Theorem [Unr14] that is more complicated and has a somewhat worse bound. Changing all of $H(\delta^*\|\cdot)$ is possible in our setting since our O2H Theorem allows us to change the oracle not just at a single location (in Theorem 1, $S$ can be an arbitrary large set).

**Game 5b:**

**let** $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $R \xleftarrow{\$} \Omega_R$,
**let** $m_0, m_1 \leftarrow A_{hy}^{H(\delta^*:=R),G\setminus\{\delta^*\},H'\setminus\{\delta^*\},\widetilde{Dec}^{**}}(pk)$
**let** $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; R(c^*))$
**let** $b' \leftarrow A_{hy}^{H(\delta^*:=R),G\setminus\{\delta^*\},H'\setminus\{\delta^*\},\widetilde{Dec}^{**}}(e^*, c^*, d^*)$

Here $\Omega_R$ is the set of all functions $\{0,1\}^* \to \mathtt{COIN}^{asy}$. And $H(\delta^* := R)$ refers to the function $H$, except that $H(\delta^* := R)$ returns $R(c)$ on input $\delta^*\|c$ for all $c$. And $\widetilde{Dec}^{**}$ is $Dec^{**}$, except that all occurrences of $H$ are replaced by $H(\delta^* := R)$.

Since $H$ is uniformly random, replacing it everywhere by $H(\delta^* := R)$ (for fresh uniformly random $R$) does not change its distribution. And replacing invocations $H(\delta^*\|c^*)$ by $R(c^*)$ does not change the game either because $H(\delta^*\|c^*)$ gives that output anyway. Thus

$$\Pr[1 \leftarrow \mathit{Game}\ 5] = \Pr[1 \leftarrow \mathit{Game}\ 5b]. \tag{38}$$

**Game 6:**

**let** $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $R \xleftarrow{\$} \Omega_R$,
**let** $m_0, m_1 \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}}(pk)$
**let** $c^* \leftarrow Enc_{a^*}^{sy}(m_b)$, $e^* \leftarrow Enc_{pk}^{asy}(\delta^*; R(c^*))$
**let** $b' \leftarrow A_{hy}^{H,G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}}(e^*, c^*, d^*)$

We use the O2H Theorem 1 to obtain an upper bound for $|\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5b]} - \sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]}|$. (In [TU16], the adaptive O2H Theorem from [Unr14] was used instead. We removed the need for an adaptive O2H Theorem by simply changing $H$ at multitude of inputs instead of only at $\delta\|c^*$.)

The only difference between Game 5b and Game 6 is that Game 6 uses $H(\delta^* := R)$ instead of $H$. We can therefore apply Theorem 1 to replace $H(\delta^* := R)$ by $H$. Specifically, in Theorem 1, let $G$ be $H$, let $H$ be $H(\delta^* := R)$, let $S := \{\delta^*\|\cdot\}$ (the set of all bitstrings starting with $\delta^*$), let $z := \delta^*$, and let $A^H(\delta^*)$ be the algorithm that simulates Game 6 and outputs 1 iff $\mathsf{Find}_{GH'}$ happens. Then $P_{\text{right}} = \Pr[1 \leftarrow A^H(\delta^*)] = \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]$ and $P_{\text{left}} = \Pr[1 \leftarrow A^{H(\delta^*:=R)}(\delta^*)] = \Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5b]$. $A$ makes up to $O(qq_{dec})$-queries to $H$ (including the ones made indirectly via the simulated $Dec^{**}$).

Thus by Theorem 1, $|\sqrt{P_{\text{left}}} - \sqrt{P_{\text{right}}}| \leq \sqrt{O(qq_{Dec})P_{\text{find}}}$ where $P_{\text{find}} = \Pr[\mathsf{Find}_H : A^{H\setminus\{\delta^*\|\cdot\}}(\delta^*)] = \Pr[\mathsf{Find}_H : \mathit{Game}\ 7]$.[6] (We write $\mathsf{Find}_H$ instead of $\mathsf{Find}$ here to distinguish it from the event $\mathsf{Find}_{GH'}$ introduced above.) Here Game 7 is as defined below, the result from replacing $H$ by the punctured oracle $H \setminus \{\delta^*\| \cdot\}$ in Game 6. Thus

$$\left|\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]} - \sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5b]}\right| \leq \sqrt{O(qq_{dec})\Pr[\mathsf{Find}_H : \mathit{Game}\ 7]}. \tag{39}$$

**Game 7:**

**let** $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$, $R \xleftarrow{\$} \Omega_R$,
**let** $m_0, m_1 \leftarrow A_{hy}^{H\setminus\{\delta^*\|\cdot\},G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}}(pk)$
**let** $c^* \leftarrow Enc^{sy}_{a^*}(m_b)$, $e^* \leftarrow Enc^{asy}_{pk}(\delta^*; R(c^*))$
**let** $b' \leftarrow A_{hy}^{H\setminus\{\delta^*\|\cdot\},G\setminus\{\delta^*\},H'\setminus\{\delta^*\},Dec^{**}\setminus\{\delta^*\|\cdot\}}(e^*, c^*, d^*)$

Here $Dec^{**} \setminus \{\delta^*\|\cdot\}$ denotes $Dec^{**}$ with all invocations of $H$ replaced by $H \setminus \{\delta^*\|\cdot\}$.

In this game, we see the advantage of the new formalization of the O2H Theorem in terms of a $\mathsf{Find}$-event. In [TU16], this game is much more complicated: In [TU16], the Game 3 introduces a wrapper around the game that stops the game at a random oracle query to $G$ or $H'$. And then in Game 7, there is a wrapper around that wrapper that stops at a random query to $H$. While this does not introduce any actual problems in the proof, we believe that it strongly simplifies the presentation not to have to deal with nested wrappers.

Summarizing the inequalities we have derived so far, we get:

$$\begin{aligned}
&|\Pr[1 \leftarrow \mathit{Game}\ 0] - \tfrac{1}{2}| \\
&\overset{(31,32,33,34)}{\leq} \ell(n) + \mathsf{negl}(\mathsf{n})^{sy} + \sqrt{O(q)\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 3]} \\
&\overset{(35,37)}{\leq} \ell(n) + \mathsf{negl}(\mathsf{n})^{sy} + \sqrt{O(q)\big(\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5] + O(q)2^{-n_1}\big)} \\
&\leq \ell(n) + \mathsf{negl}(\mathsf{n})^{sy} + \sqrt{O(q)}\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5]} + O(q)2^{-n_1/2} \\
&\overset{(38,39)}{\leq} \ell(n) + \mathsf{negl}(\mathsf{n})^{sy} + \sqrt{O(q)}\Big(\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]} + \sqrt{O(qq_{dec})\Pr[\mathsf{Find}_H : \mathit{Game}\ 7]}\Big) + O(q)2^{-n_1/2} \\
&= \ell(n) + \mathsf{negl}(\mathsf{n})^{sy} + \sqrt{O(q)}\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]} + O(qq_{dec}^{1/2})\sqrt{\Pr[\mathsf{Find}_H : \mathit{Game}\ 7]} + O(q)2^{-n_1/2}
\end{aligned} \tag{40}$$

Note: in the inequality $\overset{(38,39)}{\leq}$, we see the advantage of using the form of Theorem 1 that bounds the difference of square-roots of probabilities (between $\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]}$

[6] Here, we get the factor $O(qq_{dec})$ which plays a dominant role in the final bound. This factor is mostly due to the classical queries performed by $Dec^{**} \setminus \{\delta^*\}$, the number of quantum queries is $O(q)$, i.e., much smaller. It is an interesting question whether there is a strengthening of the O2H Theorem that distinguishes between classical and quantum queries and thus leads to a better bound here.

and $\sqrt{\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5b]}$). If we had used the original form bounding the difference between probabilities (between $\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6]$ and $\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 5b]$), $O(q)\Pr[\mathsf{Find}_H : \mathit{Game}\ 7]$ would be under a fourth root instead of a square root after $\overset{(38,39)}{\leq}$. We expect that a similar benefit occurs whenever the O2H Theorem is nested (i.e., the O2H Theorem is used to analyze the guessing game $P_{\text{find}}$ resulting from another application of the O2H Theorem).

We are left to bound the success probability in Games 6 and 7 (i.e., the probability of $\mathsf{Find}_{GH'}$ and $\mathsf{Find}_H$, respectively). Since in both games $R(c^*)$ is uniformly random, and only used as the randomness for $\mathit{Enc}^{asy}_{pk}(\delta^*; R(c^*))$, we can use the one-wayness of $\mathit{Enc}^{asy}_{pk}$ to show that $\delta^*$ is hidden. If $\delta^*$ would not occur anywhere else in Games 6 and 7 (as was assumed in the proof from [TU16]), it would then be simple to prove that the probabily of guessing $\delta^*$ is small.

In our setting, however, we have an additional complication: $\delta^*$ is used in Games 6 and 7 also as part of the puncturing of the oracles. For example, to simulate $G \setminus \delta^*$, we need to know $\delta^*$. So, hypothetically, it might be possible that access to $G \setminus \delta^*$ might leak some information about $\delta^*$.

We give here two different proofs that this is not the case, one simpler one assuming IND-CPA security of $\mathit{Enc}_{asy}$, and one slightly more complicated one assuming only one-time security of $\mathit{Enc}_{asy}$. The proof is almost identical for Games 6 and 7, so we only give the proof for Game 7.

**Proof using IND-CPA.** Since $R(c^*)$ is uniformly random, and only used once as the randomness for $\mathit{Enc}^{asy}_{pk}(\delta^*; R(c^*))$, IND-CPA security of $\mathit{Enc}^{asy}$ implies that $\mathit{Enc}^{asy}_{pk}(\delta^*; R(c^*))$ is indistinguishable from $\mathit{Enc}^{asy}_{pk}(0)$. More precisely, $|\Pr[\mathsf{Find}_H : \mathit{Game}\ 7] - \Pr[\mathsf{Find}_H : \mathit{Game}\ 8]| \leq \mathsf{negl}(\mathsf{n})^{\mathsf{asy}}$ where Game 8 results from Game 7 by replacing $\mathit{Enc}^{asy}_{pk}(\delta^*; R(c^*))$ by $\mathit{Enc}^{asy}_{pk}(0)$:

**Game 8:**

**let** $H' \xleftarrow{\$} \Omega_{wise}$, $a^* \xleftarrow{\$} \mathtt{KSP}^{sy}$, $d^* \xleftarrow{\$} \mathtt{MSP}^{asy}$,
**let** $m_0, m_1 \leftarrow A^{H\setminus\{\delta^*\|\cdot\},G\setminus\{\delta^*\},H'\setminus\{\delta^*\},\mathit{Dec}^{**}}_{hy}(pk)$
**let** $c^* \leftarrow \mathit{Enc}^{sy}_{a^*}(m_b)$, $e^* \leftarrow \mathit{Enc}^{asy}_{pk}(0)$
**let** $b' \leftarrow A^{H\setminus\{\delta^*\|\cdot\},G\setminus\{\delta^*\},H'\setminus\{\delta^*\},\mathit{Dec}^{**}\setminus\{\delta^*\|\cdot\}}_{hy}(e^*, c^*, d^*)$

Note that an oracle query to $G \setminus \{\delta^*\}$ is equivalent to querying $G$ and $\mathcal{O}^{SC}_{\{\delta^*\}}$ consecutively. Analogously for $H \setminus \{\delta^*\|\cdot\}$ and $G \setminus \{\delta^*\}$. (In the case of $H \setminus \{\delta^*\|\cdot\}$, $\mathcal{O}^{SC}_{\{\delta^*\}}$ is applied only to the first part of the query input.) Thus, we can define an adversary $B^{\mathcal{O}^{SC}_{\{\delta^*\}}}()$ that picks the functions $H, G, H'$ itself, and then executes Game 8 using $\mathcal{O}^{SC}_{\{\delta^*\}}$ to simulate $G \setminus \{\delta^*\}$, $H \setminus \{\delta^*\|\cdot\}$, and $G \setminus \{\delta^*\}$.

(Note that in Game 8, $\delta^*$ is never used directly, only in the definition of the punctured oracles. Otherwise, it would not be possible to construct $B$ in this way (without additional input $\delta^*$). In particular, we need that $\mathit{Dec}^{**}$ does not access $\delta^*$ directly. This is why we had to change the definition of $\mathit{Dec}^{**}$ from the definition used in [TU16].)

Then we have

$$\Pr[\mathsf{Find} : B^{\mathcal{O}^{SC}_{\{\delta^*\}}}()] = \Pr[\mathsf{Find}_{GH'} \vee \mathsf{Find}_H : \mathit{Game}\ 8] \geq \Pr[\mathsf{Find}_H : \mathit{Game}\ 8]. \tag{41}$$

And $B$ makes $O(qq_{dec})$ queries to $\mathcal{O}^{SC}_{\{\delta^*\}}$. (Note that $Dec^{**}$ makes one oracle query for every $\hat{\delta} \in S$, and there are at most $O(q)$ values in $S$ since $H' - d$ is a polynomial of degree $O(q)$.)

By Corollary 1, we then have $\Pr[\mathsf{Find} : B^{\mathcal{O}^{SC}_{\{\delta^*\}}}()] \in O(qq_{dec})2^{-n_1}$ and thus $\Pr[\mathsf{Find}_H : \mathit{Game}\ 7] \overset{(41)}{\leq} \mathsf{negl}(n)^{\mathsf{asy}} + O(qq_{dec})2^{-n_1}$.

Analogously, we get $\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6] \leq \mathsf{negl}(n)^{\mathsf{asy}} + O(qq_{dec})2^{-n_1}$.

Plugging this into (40), we get

$$\begin{aligned}|\Pr[1 \leftarrow \mathit{Game}\ 0] - \tfrac{1}{2}| &\leq \ell(n) + \mathsf{negl}(n)^{sy} + \sqrt{O(q)}\sqrt{\mathsf{negl}(n)^{\mathsf{asy}} + O(qq_{dec})2^{-n_1}} \\ &\quad + O(qq_{dec}^{1/2})\sqrt{\mathsf{negl}(n)^{\mathsf{asy}} + O(qq_{dec})2^{-n_1}} + O(q)2^{-n_1/2} \\ &\leq \ell(n) + \mathsf{negl}(n)^{sy} + O(qq_{dec}^{1/2})\sqrt{\mathsf{negl}(n)^{\mathsf{asy}}} + O(q^{3/2}q_{dec})2^{-n_1/2}.\end{aligned}$$

**Proof using one-wayness.** Note that an oracle query to $G \setminus \{\delta^*\}$ is equivalent to querying $G$ and $\mathcal{O}^{SC}_{\{\delta^*\}}$ consecutively. Analogously for $H \setminus \{\delta^* \| \cdot\}$ and $G \setminus \{\delta^*\}$. (In the case of $H \setminus \{\delta^* \| \cdot\}$, $\mathcal{O}^{SC}_{\{\delta^*\}}$ is applied only to the first part of the query input.)

Thus we can define an adversary $\hat{A}^{\mathcal{O}^{SC}_{\{\delta^*\}}}(e^*)$ that simulates Game 7, using $\mathcal{O}^{SC}_{\{\delta^*\}}$ to simulate the queries to the punctured oracles, and using $e^*$ as the ciphertext $Enc^{asy}_{pk}(\delta^*; R(c^*))$. Then

$$\Pr[\mathsf{Find} : \hat{A}^{\mathcal{O}^{SC}_{\{\delta^*\}}}(e^*)] = \Pr[\mathsf{Find}_{GH'} \vee \mathsf{Find}_H : \mathit{Game}\ 7] \geq \Pr[\mathsf{Find}_H : \mathit{Game}\ 7] \tag{42}$$

where $e^* := Enc^{asy}_{pk}(\delta^*; R(c^*))$ and $\delta^*$ uniform. And $\hat{A}$ makes $O(qq_{dec})$ queries. (Note that $Dec^{**}$ makes one oracle query for every $\hat{\delta} \in S$, and there are at most $O(q)$ values in $S$ since $H' - d$ is a polynomial of degree $O(q)$.)

And by Theorem 2,

$$\Pr[\mathsf{Find} : \hat{A}^{\mathcal{O}^{SC}_{\{\delta^*\}}}(e^*)] \leq O(qq_{dec}) \Pr[\delta^* = B(e^*)] \tag{43}$$

where $B$ is the adversary that stops $\hat{A}$ at a random query (see Theorem 2). Note that the runtime of $B$ is approximately the same as that of $\hat{A}$.

Then from the one-wayness of the asymmetric encryption scheme, we have $\Pr[\delta^* = B(e^*)] \overset{(42,43)}{\leq} \mathsf{negl}(n)^{\mathsf{asy}}$. Hence $\Pr[\mathsf{Find}_H : \mathit{Game}\ 7] \leq O(qq_{dec})\,\mathsf{negl}(n)^{\mathsf{asy}}$.

Analogously, $\Pr[\mathsf{Find}_{GH'} : \mathit{Game}\ 6] \leq O(qq_{dec})\,\mathsf{negl}(n)^{\mathsf{asy}}$.

Plugging this into (40), we get

$$\begin{aligned}|\Pr[1 \leftarrow \mathit{Game}\ 0] - \tfrac{1}{2}| &\leq \ell(n) + \mathsf{negl}(n)^{sy} + \sqrt{O(q)}\sqrt{O(qq_{dec})\,\mathsf{negl}(n)^{\mathsf{asy}}} \\ &\quad + O(qq_{dec}^{1/2})\sqrt{O(qq_{dec})\,\mathsf{negl}(n)^{\mathsf{asy}}} + O(q)2^{-n_1/2} \\ &\leq \ell(n) + \mathsf{negl}(n)^{sy} + O(q^{3/2}q_{dec})\sqrt{\mathsf{negl}(n)^{\mathsf{asy}}}.\end{aligned}$$

(In the last inequality, to drop the last summand, we used that $\mathsf{negl}(n)^{\mathsf{asy}} \geq 2^{-n_1}$ since the plaintext is $n_1$ bit long and thus can be guessed with probability at least $2^{-n_1}$.) $\square$

## References


[Amb02] Andris Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, June 2002. URL: http://dx.doi.org/10.1006/jcss.2002.1826, doi:10.1006/jcss.2002.1826.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *55th FOCS*, pages 474–483. IEEE Computer Society Press, October 2014. doi:10.1109/FOCS.2014.57.

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, July 2001. URL: http://doi.acm.org/10.1145/502090.502097, doi:10.1145/502090.502097.

[BBHT98] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998. doi:10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

[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, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 41–69. Springer, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_3.

[BEF18] Dan Boneh, Saba Eskandarian, and Ben Fisch. Post-quantum EPID group signatures from symmetric primitives. Cryptology ePrint Archive, Report 2018/261, 2018. https://eprint.iacr.org/2018/261.

[Ben81] Michael Ben-Or. Probabilistic algorithms in finite fields. In *22nd FOCS*, pages 394–398. IEEE Computer Society Press, October 1981. doi:10.1109/SFCS.1981.37.

[BES18] Marko Balogh, Edward Eaton, and Fang Song. Quantum collision-finding in non-uniform random functions. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 467–486. Springer, Heidelberg, 2018. doi:10.1007/978-3-319-79063-3_22.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. doi:10.1145/168588.168596.

[BR95] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *EUROCRYPT'94*, volume 950 of *LNCS*, pages 92–111. Springer, Heidelberg, May 1995. doi:10.1007/BFb0053428.

[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 Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 17*, pages 1825–1842. ACM Press, October / November 2017. doi:10.1145/3133956.3133997.

[CHR+18] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. SOFIA: $\mathcal{MQ}$-based signatures in the QROM. In Michel Abdalla and Ricardo Dahab, editors, *PKC 2018, Part II*, volume 10770 of *LNCS*, pages 3–33. Springer, Heidelberg, March 2018. doi:10.1007/978-3-319-76581-5_1.

[DRS18] David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 419–440. Springer, Heidelberg, 2018. doi:10.1007/978-3-319-79063-3_20.

[Eat17] Edward Eaton. Leighton-Micali hash-based signatures in the quantum random-oracle model. In Carlisle Adams and Jan Camenisch, editors, *SAC 2017*, volume 10719 of *LNCS*, pages 263–280. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-72565-9_13.

[EU18] E. Ehsan Ebrahimi and Dominique Unruh. Quantum collision-resistance of non-uniformly distributed functions: upper and lower bounds. *Quantum Information & Computation*, 18(15&16):1332–1349, 2018. URL: http://www.rintonpress.com/xxqic18/qic-18-1516/1332-1349.pdf.

[FO13] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, January 2013. doi:10.1007/s00145-011-9114-1.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

[GPS17] Steven D. Galbraith, Christophe Petit, and Javier Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70694-8_1.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In Yael Kalai and Leonid Reyzin,

editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017. doi:10.1007/978-3-319-70500-2_12.

[HKSU18] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. Generic authenticated key exchange in the quantum random oracle model. Cryptology ePrint Archive, Report 2018/928, 2018. https://eprint.iacr.org/2018/928.

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. Mitigating multi-target attacks in hash-based signatures. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *PKC 2016, Part I*, volume 9614 of *LNCS*, pages 387–416. Springer, Heidelberg, March 2016. doi:10.1007/978-3-662-49384-7_15.

[JZC+18] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 96–125. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96878-0_4.

[JZM] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. unpublished manuscript, first revision of [JZM19], personal communication.

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. Cryptology ePrint Archive, Report 2019/052, 2019. https://eprint.iacr.org/2019/052.

[LM95] Frank T. Leighton and Silvio Micali. Large provably fast and secure digital signature schemes based on secure hash functions. US Patent 5,432,852, 1995.

[NC00] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, first edition, 2000.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 520–551. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_17.

[SY17] Fang Song and Aaram Yun. Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 283–309. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63715-0_10.

[TTU16] Ehsan Ebrahimi Targhi, Gelo Noel Tabia, and Dominique Unruh. Quantum collision-resistance of non-uniformly distributed functions. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 79–85. Springer, Heidelberg, 2016. doi:10.1007/978-3-319-29360-8_6.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Quantum security of the Fujisaki-Okamoto and OAEP transforms. In *TCC 2016-B*, volume 9986 of *LNCS*, pages 192–216. Springer, 2016. doi:10.1007/978-3-662-53644-5_8.

[Unr14] Dominique Unruh. Quantum position verification in the random oracle model. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 1–18. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_1.

[Unr15a] Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 755–784. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_25.

[Unr15b] Dominique Unruh. Revocable quantum timed-release encryption. *Journal of the ACM*, 62(6):49:1–76, 2015. Preprint on IACR ePrint 2013/606.

[Unr17] Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 65–95. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70694-8_3.

[YAJ+17] Youngho Yoo, Reza Azarderakhsh, Amir Jalali, David Jao, and Vladimir Soukharev. A post-quantum digital signature scheme based on supersingular isogenies. In Aggelos Kiayias, editor, *FC 2017*, volume 10322 of *LNCS*, pages 163–181. Springer, Heidelberg, April 2017.

[Zal99] Christof Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746–2751, Oct 1999. URL: https://arxiv.org/abs/quant-ph/9711070, doi:10.1103/PhysRevA.60.2746.

[Zha12a] Mark Zhandry. How to construct quantum random functions. In *53rd FOCS*, pages 679–687. IEEE Computer Society Press, October 2012. doi:10.1109/FOCS.2012.37.

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 758–775. Springer, Heidelberg, August 2012. doi:10.1007/978-3-642-32009-5_44.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7&8), 2015.

# Symbol index

| | | |
|---|---|---|
| $\mathcal{O}_I^{SC}$ | Semi-classical oracle for set $I$ | |
| $\mathsf{Find}$ | Semi-classical $\mathcal{O}_S^{SC}$ returns 1 | |
| $\Delta(X, Y)$ | Statistical distance between distributions/random variables $X$ and $Y$ | |
| $\mathrm{flip}_i(l)$ | Flips $i$-th bit of $l$ | |
| $F(\rho_1, \rho_2)$ | Fidelity between $\rho_1$ and $\rho_2$ | 16 |
| $\mathrm{TD}(\rho_1, \rho_2)$ | Trace distance between $\rho_1$ and $\rho_2$. | 16 |
| $\mathrm{tr}\,\rho$ | Trace of $\rho$ | |
| $B(\rho_1, \rho_2)$ | Bures distance between $\rho_1$ and $\rho_2$ | 16 |
| $\lvert\Psi\rangle$ | Refers to a quantum state (or, for $x \in M$, $\lvert x\rangle$ refers to a basis vector of $\mathbb{C}^M$) | |
| $\mathrm{tr}\,A\rho$ | Partial trace of $\rho$, removing register $A$ | |
| $\langle\Psi\rvert$ | Adjoint of $\lvert\Psi\rangle$, i.e., $\langle\Psi\rvert^\dagger$ | |
| $\mathbb{C}$ | Complex numbers | |
| $\mathcal{E}$ | A quantum operation (superoperator) | |
| $\mathcal{D}$ | A distribution | |
| $x \xleftarrow{\$} M$ | $x$ picked uniformly from the set $M$ | |
| $H \setminus I$ | Oracle $H$, punctured at $I$ | |
| $\lvert x\rvert$ | Absolute value of $x$ / cardinality of set $x$ | |
| $x \leftarrow A$ | $x$ assigned output of algorithm $A$ / picked according to distribution $A$ | |
| $\mathsf{Guess}$ | Query to fully-quantum oracle is in $S$ | |
| $\lVert x\rVert$ | Norm of $x$ | |
| $\mathrm{Exp}_z[y]$ | Expectation of $y$, taken over the randomness of $z$ | |