Erratum:

The application of the Cauchy-Schwarz inequality in (33) on page 53 is incorrect. This breaks the proof of Lemma 26 and thus of most results in this paper. We thank Minki Hhan for discovering this problem.

Counterexample. In particular, the following is a simple counterexample to the central Theorems 4 and 12:

Consider the state $|0\mapsto 0\rangle^{\mathfrak{g}}$. (Where $\mathfrak{g}$ is $\mathfrak{f}$ or $\mathfrak{p}$, depending on whether we are in the context of Theorem 4 or 12.) We have that

$$\begin{aligned}|0\mapsto 0\rangle^{\mathfrak{g}} &= \sum_{y\in R}|0\mapsto y\rangle^{\mathfrak{g}} \;-\sum_{y\in R\setminus\{0\}}|0\mapsto y\rangle^{\mathfrak{g}} = \sqrt{N}|\varnothing\rangle^{\mathfrak{g}} \;-\sum_{y\in R\setminus\{0\}}|0\mapsto y\rangle^{\mathfrak{g}}\\ &\in \operatorname{span}\big\{|f\rangle^{\mathfrak{g}} : 0\notin \operatorname{im} f,\ |\operatorname{dom} f|\le 1\big\}_f.\end{aligned}$$

Consider a system with registers XYH in the state $\psi := |0\rangle_X|0\rangle_Y|0\mapsto 0\rangle^{\mathfrak{g}}$. Then the previous equation implies

$$\psi \in \operatorname{span}\big\{|x\rangle_X|0\rangle_Y|f\rangle^{\mathfrak{g}} : 0\notin \operatorname{im} f,\ |\operatorname{dom} f|\le 1\big\}_{xf}.$$

By Theorem 4 or 12, with $A_{x,e}, B_{x,e} := \{f : 0\notin \operatorname{im} f\}$ and with $c := \frac{1}{N}$ (for Theorem 4) or $c := \frac{1}{N-2}$ (for Theorem 12), this implies that

$$\begin{aligned}U_{query}\psi &\overset{4\sqrt{c}}{\approx} \operatorname{span}\big\{|x\rangle_X|f(x)\rangle_Y|f\rangle^{\mathfrak{g}} : x\in\operatorname{dom} f,\ 0\notin\operatorname{im} f,\ |\operatorname{dom} f|\le 2\big\}_{xf}\\ &\subseteq \operatorname{span}\big\{|x\rangle_X|y\rangle_Y|f\rangle^{\mathfrak{g}} : y\neq 0\big\}_{xyf} =: I'.\end{aligned}$$

On the other hand, $U_{query}\psi = |0\rangle_X|0\rangle_Y|0\mapsto 0\rangle^{\mathfrak{g}}$ by definition of ψ, U_{query}, and $|0\mapsto 0\rangle^{\mathfrak{g}}$. Thus $U_{query}\psi$ is orthogonal to I', in direct contradiction to $U_{query}\psi \overset{4\sqrt{c}}{\approx} I'$. (At least when $4\sqrt{c}<1$ which is the case if $N\ge 19$.)

Unaffected parts. The only unaffected part of the paper is Section 3.1 in which we give a different view of Zhandry's compressed oracle technique. This view may be useful in its own right for understanding Zhandry's compressed oracle technique. However, it does not give us a proof technique for random permutations.

Future of this eprint. If we find a way to fix the problems described above, we will update this eprint. Otherwise we will leave it in place including this erratum for future reference.

Compressed Permutation Oracles

And the Collision-Resistance of Sponge/SHA3

Dominique Unruh
University of Tartu

February 22, 2021

THIS IS A DRAFT

Do not distribute

(Git revision: `iacr-eprint-v1-0-g950e407-dirty`)

Information about mistakes and typos are appreciated (unruh@ut.ee).

Abstract

We generalize Zhandry's compressed oracle technique to invertible random permutations. (That is, to a quantum random oracle where the adversary has access to a random permutation and its inverse.) This enables security proofs with lazy sampling, i.e., where oracle outputs are chosen only when needed.

As an application of our technique, we show the collision-resistance of the sponge construction based on invertible permutations. In particular, this shows the collision-resistance of SHA3 (in the random oracle model).

1 Introduction

The random oracle [BR93] is a powerful heuristic[1] for cryptographic security proofs. It allows us to abstract from the gritty details of the definition of a hash function and to imagine it to be just a random function. We can then use powerful reasoning techniques such as lazy sampling to make security proofs simpler or, in many cases, possible in the first place. (Lazy sampling refers to the technique of choosing the outputs of the random oracle "on demand", when they are first accessed.) These techniques are useful even if we are not in the random oracle model. For example, when working with a pseudorandom function, the first step in a proof is often to replace it by a fictitious random function. Quite similar to the random oracle are random permutations (to model cryptographically-strong permutations), or ideal ciphers (a heuristic model for block ciphers, basically a key-indexed family of random permutations). In the standard model, random permutations occur in security proofs involving pseudorandom permutations (e.g., in protocols involving block ciphers). In such proofs, we often consider invertible random permutations, i.e., we give the adversary access also to the inverse of the permutation. All of this can be handled very nicely using lazy sampling.

At least, this is the situation in classical cryptography. Once quantum (or post-quantum) cryptography enters the picture, using the random oracle becomes much harder. This is because the quantum random oracle gives the adversary superposition-access to the random oracle. That is, the adversary can query the random oracle on a superposition of many different values. Then lazy sampling as in the classical case does not work any more: The adversary could query the oracle on a superposition of all inputs already in the very first query. If we were to sample the oracle at all the sampled positions, this would mean sampling the whole function in one go. But that goes against the very idea of lazy sampling. Furthermore, we cannot just measure where the oracle is queried as this would disturb the adversary state, and we need to make sure that our technique does not influence the way in which the adversary is entangled with the random oracle (in a way that the adversary can notice).

The above does not mean that the random oracle is unusable in the quantum setting.

[1] In general, this heuristic is not sound: There are contrived protocols which are secure in the random oracle model but insecure when the oracle is instantiated with *any* hash function [CGH98]. However, in practice the random oracle model has proven to be a very good heuristic. Readers who reject heuristics in security proofs may still enjoy the results in this work as a result about generic query complexity, or as a technique for security proofs involving pseudorandom permutations.

A number of techniques have been developed for handling the random oracle (history-free reductions, $2q$-wise independent functions, semi-constant distributions, small-range distributions, one-way to hiding (O2H) theorems, polynomial method, adversary method, see the related work below). However, none of these have the general applicability of the lazy sampling method, and they are often much harder to use. Then, surprisingly, Zhandry [Zha19] discovered that a variant of lazy sampling is actually possible with quantum random oracles, although it is not as simple (and as general) as in the classical case. We refer to this technique as Zhandry's "compressed oracle technique". (We give more details about it below.)

However, when talking about (invertible) random permutations, the situation is much more limited. The abovementioned tools are specific to the random function case.[2] To the best of our knowledge, no hardness results are known about invertible random permutations, not even simple query complexity results such as the hardness of searching an input with certain properties. As a consequence, we do not know anything about the post-quantum security of cryptosystems built from invertible permutations, such as the industry-standard SHA3 [NIS14].

The present work sets out to fill this gap.

Our contribution.

- We present a new variant of Zhandry's compressed oracle technique. While it is not (yet?) applicable in all situations where Zhandry's is applicable, especially when efficient simulations are required, it is arguably conceptually simpler. (See the discussion of pros/contras in Section 3.2.) The greatest advantage of the new approach is that it is not limited to random functions, but can in principle be applied to functions of any distribution.
- We instantiate our approach with invertible random permutations. That is, our framework allows us to prove query complexity and security results that involve adversaries with superposition access to a permutation h and to its inverse h^{-1}. Our approach works by keeping track of invariants throughout the adversary's execution. These invariants establish where the permutation oracle was queried and what properties the corresponding outputs satisfy. (E.g., the invariants may say "no output is 0" or "there are no collisions in the output".) This mimics the classical lazy sampling.

  We establish a number of theorems that tell us how invariants change under permutation queries (and inverse queries). The premises of these theorems are purely combinatorial (upper bounds on the size of certain sets) which means that all nontrivial quantum mechanical reasoning is hidden away.
- As an example we show that it is hard to find x, y such that $h(x\|0^c) = y\|0^c$ for an invertible permutation h ("two-sided zero search"). To the best of our

[2]Except for the O2H theorem. Some variants of the O2H theorem apply to arbitrarily distributed functions [AHU19], in particular to invertible permutations. (An invertible permutation can be modeled as a function $f : \{0,1\} \times \{0,1\}^n \to \{0,1\}^n$, uniformly sampled from the set of all functions where $f(0,\cdot)$ is a permutation and $f(1,\cdot)$ its inverse.) However, we are not aware of any work that makes use of this.

knowledge, this is the first hardness result for invertible random permutations. With our technique, the proof is easy. (Adversary success probability is $O(q^2/2^c)$ in q queries.)
Another example is that in a non-invertible random permutation, finding preimages is hard. While this is known (it follows from [Zha15a]), our bound $O(q^2/N)$ for the success probability with q queries and domain size N beats the known bound of $O(q^3/N)$.
We also analyzed both situations with respect to adversaries that make k-parallel queries, yielding tighter bounds $O(q^2k/2^c)$ and $O(q^2k/N)$, respectively.)

- We show the collision-resistance of the sponge construction [Ber+07] when instantiated with an invertible random permutation. (The sponge construction is a popular construction of cryptographic hash functions.) Classically, this is a well-known result. But in the post-quantum setting, the security of the sponge construction was known only when using random functions (or *non-invertible* permutations). However, most hash functions that use the sponge construction, including the industry standard SHA3, are based on *invertible* permutations. Thus our result is the first result to establish post-quantum security of SHA3. Our result is tight for typical parameter choices (namely, when the hash output length is not larger than the "rate" of the sponge).

Related work. *Quantum random oracles.* [Unr15; HRS16] showed that finding preimages in the random oracle is hard ([Boy+98] showed this in worst-case setting.) [Bon+11] introduced "history-free reductions" which basically amounts to replacing the random oracle by a different function right from the start. [Zha12c] showed that random oracles can be simulated using $2q$-wise independent functions. Based on this, [Unr15] introduces a technique for extracting preimages of the random oracle. [Zha12c] introduces the "semi-constant distributions" technique that allows us to program the random oracle in many random locations with a given challenge value without the adversary noticing. [Zha12a] improves upon this with the "small-range distribution" technique that allows us to simulate random oracles using random looking functions with a small range. [Zha15b] shows that random oracles are collision resistant (this is generalized by [TTU16; EU18; BES18] to the case of non-uniformly distributed functions with independently sampled outputs). 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 [Unr16]. More general methods for problems in quantum query complexity (not limited to random oracles) include the polynomial method [Bea+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.

Compressed oracles. Compressed oracles were introduced in [Zha19] and used there to show indifferentiability of the Merkle-Damgård construction, as well as security of the Fujisaki-Okamoto transform. [Chu+20] generalizes [Zha19] to Fourier transforms

over abelian groups, thus allowing random functions with a range different from $\{0,1\}^n$. Different from [Zha19], they do not have a compression/decompression algorithm but instead reason using invariants that are expressed in a basis different from the computational basis. This is reminiscent of our approach. The also introduce support for parallel queries (as do we). [Cza+20] generalizes [Zha19] to non-uniformly distributed functions, but only for the case where all outputs are independently sampled. (This is similar to what we achieve in our reformulation of Zhandry in Section 3.1, although we additionally get rid of the Fourier transform.) **[TODO: ehsan's paper?]** **[TODO: paper "zhandry with errors"?]**

Random permutations. [Zha15b] shows that random oracles are indistinguishable from (noninvertible) random permutations. This allows us to derive results for random permutations from results for random functions. (However, the results might not be tight, cf. Section 4.2.) [Zha16] shows the existence of quantum-secure pseudorandom permutations (qPRP, secure under superposition-queries of the function and its inverse) from quantum one-way functions. In particular, this implies that a random invertible permutation can be efficiently simulated.[3] However, [Zha16] does not give us any technique for analyzing schemes that *use* a qPRP; when analyzing such a scheme we typically replace the qPRP by an invertible random function in the proof, and the techniques from the present paper can be used.

Security of the sponge construction. The sponge construction was proposed by [Ber+07]. In the classical random oracle model, security of the sponge construction was shown by [Ber+08], both when the sponge is based on random functions and on invertible random permutations. They showed indifferentiability, which implies many other properties such as collision-resistance, pseudorandomness, and more. In the quantum setting, collision-resistance and the collapsing property from [Unr16] were shown in [Cza+18] in the random function case. Quantum pseudorandomness of the sponge was shown by [CHS19] but only in the case where the underlying round function is secret (the adversary cannot query it). Indifferentiability in the quantum setting was shown by [Cza+20] in the random function case. All those results immediately imply the corresponding results in the non-invertible random permutation case since random functions and permutations are indistinguishable [Zha15b]. However, for invertible random permutations, no quantum results are known.

Organization. In Section 2 we introduce relevant notational conventions.

In Section 3 we present compressed oracles for random functions. Specifically, in Section 3.1 we recap and give a new view on Zhandry's technique. In Section 3.2 we present our new compressed oracle technique. In Sections 3.3 and 3.4 we give example applications (preimage search, collision finding) and introduce the main theorems along the way. In Section 3.5 we cover extensions for handling parallel queries.

In Section 4 we extend our compressed oracle technique to handle invertible permuta-

[3] If we implement the underlying quantum one-way function using a random oracle, and we simulate that random oracle with the method from [Zha12c] or [Zha19], then we event get a simulation without computational assumptions.

tions. Specifically, in Section 4.1 we state the technique for *non-invertible* permutations. In Section 4.2 we give an example (preimage search) and introduce the main theorems along the way. In Section 4.4 we explain how inverse queries are handled and we illustrate this with an example in Section 4.5 (two-sided zero search). In Section 4.6 we give some additional convenience theorems for handling classical circuits.

In Section 5, we give the proofs of all our theorems and generalize the model along the way. See the beginning of that section for an overview. In Section 6, we show the collision resistance of the sponge construction. In Section 7, we conclude and mention open questions. In the back matter, we provide a list of theorems, a symbol and a keyword index, as well as the bibliography.

2 Preliminaries

Miscellaneous notation. λ is the empty word. $\Re(z)$ is the real part of the complex number z. $|x|$ is the absolute value of x (if x is real/complex) or the cardinality of x (if x is a set). $(a)_b$ is the *falling factorial*, defined by $(a)_b := a!/(a-b)!$, i.e., the number of injective functions from a set of size b to a set of size a. We write $\{f(x) : P(x)\}_x$ for the set of all $f(x)$ where x ranges over all values satisfying $P(x)$. (E.g., $\{x^2 : x \text{ is prime}\}_x$ is the set of squares of primes.) We use underscores to emphasize that a variable ranges over tuples, e.g., $\underline{x} \in R^k$ for k-tuples $\underline{x}$. In slight abuse of notation, we also write $\underline{0}$ for a tuple of zeroes, $f(\underline{z})$ for $\big(f(z_1), \dots, f(z_k)\big)$, $\underline{a} \oplus \underline{b}$ for $(a_1 \oplus b_1, \dots a_k \oplus b_k)$.

Total and partial functions. Throughout this work, we will extensively deal with total and partial functions to describe states, queries, and invariants. For sets D, R, let $D \to R$ be the *total functions* from D to R, and $D \hookrightarrow R$ the *total injections* (i.e., injective total functions) from D to R. Furthermore, $D \rightharpoonup R$ and $D \rightharpoonup\!\!\!\!\!\!\cdot\;\; R$ are the *partial functions* and *partial injections*, respectively. For a partial function $f : D \rightharpoonup R$, $\operatorname{dom} f \subseteq D$ is the domain (inputs on which f is defined) and $\operatorname{im} f$ is the image of f. Let $f(x := y)$ denote the updating of f. That is $f(x := y)(x) = y$, and $f(x := y)(x') = f(x')$ for $x \neq x'$.

We consider functions (total and partial) as special cases of relations between D and R (which in turn are subsets of $D \times R$). E.g., a total function $f : D \to R$ is a relation where for every $x \in D$, there is exactly one $y \in R$ such that $(x, y) \in f$. In particular, $\varnothing$ is the *empty partial function* (defined nowhere). And for two partial functions $f, g : D \rightharpoonup R$, the union $f \cup g$ is a relation and therefore *can* be a partial function itself. We say f and g are *compatible*, written $f \diamond g$, iff $f \cup g$ is a partial function. (Equivalently, f, g are compatible iff they coincide wherever both are defined.)

For k-tuples $\underline{x} \in D^k$, $\underline{y} \in R^k$, let $\underline{x} \mapsto \underline{y}$ be the relation $\big((x_1, y_2), \dots, (x_k, y_k)\big)$. In particular, if $\underline{x}$ has no duplicates, or if $\underline{x}$ has duplicates but $\underline{y}$ has duplicates in the same places, $\underline{x} \mapsto \underline{y}$ is the function that maps x_i to y_i and has domain $\underline{x}$. In the special case of 1-tuples, note that if $x \notin \operatorname{dom} f$, we have $f \cup (x \mapsto y) = f(x := y)$.

Quantum-related notation. Quantum states are elements of a (not necessarily finite-dimensional) Hilbert space $\mathcal{H}$. We usually represent quantum states with greek letters (e.g., ψ) and use ket-notation ($|x\rangle$) to refer to basis states of the computational basis unless specified otherwise, and $\langle x|$ is the adjoint of $|x\rangle$ ($\langle x| = |x\rangle^\dagger$). (I.e., $|x\rangle$ for $x \in X$ is an orthonormal basis of $\mathbb{C}^X$.) We write the inner product of ψ, ϕ as $\langle \psi, \phi \rangle$. However, the inner product of $|x\rangle$ and $|y\rangle$ is written as $\langle x|\, y\rangle$ instead of the awkward $\big\langle |x\rangle, |y\rangle \big\rangle$. (And analogously, for notation $|f\rangle^\mathsf{f}$ and similar that we will introduce later, $\langle f|\, g\rangle^\mathsf{f}$ is the inner product of $|f\rangle^\mathsf{f}$ and $|g\rangle^\mathsf{f}$.) $\|\psi\|$ is the norm of $\psi \in \mathcal{H}$, and $|||A|||$ is the corresponding operator norm of the bounded operator $A : \mathcal{H} \to \mathcal{H}'$. For $S \subseteq \mathcal{H}$, $\operatorname{span} S$ is the *(closed) span* of S, i.e., the smallest topologically closed subspace of $\mathcal{H}$ containing S. *Projector* always means orthogonal projector.

We will often need to consider the distance between a vector and a subspace: For two vectors $\psi, \psi' \in \mathcal{H}$, we write $\psi \overset{\varepsilon}{\approx} \psi'$ to denote $\|\psi - \psi'\| \leq \varepsilon$. And if S is a closed subspace of $\mathcal{H}$, then we write $\psi \overset{\varepsilon}{\approx} S$ to denote $\exists \psi' \in S.\ \psi \overset{\varepsilon}{\approx} \psi'$.[4]

Quantum oracle queries. Throughout the paper, we will frequently refer to oracle queries. Thus, we fix some variables once and for all: D always refers to the domain and R to the range of the function. (I.e., queries are always made to a function $h : D \to R$.) We will also fix once and for all the sizes of D and R as $M := |D|$ and $N := |R|$. (In particular, D, R are assumed to be finite.)

We additionally fix a set $\tilde{R}$. Let $\oplus : \tilde{R} \times R \to \tilde{R}$ be a right cancellable operation. To make notation more familiar, we additionally fix some arbitrary element $0 \in \tilde{R}$ (but do not assume any specific algebraic properties of it), and we identify $y \in R$ with $0 \oplus y \in \tilde{R}$.

Typically, $\tilde{R} := R$ is a group, and $\oplus$ its operation, e.g., modular addition or XOR. But being right cancellable is sufficient for the query operation to be unitary. This gives us some added flexibility when R does not have a natural group structure.

A reader who is not looking for generality may safely assume $R = \tilde{R} = \{0,1\}^n$ and $\oplus$ to be XOR.

A query to a fixed function $f : D \to R$ can then be implemented by the unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$. (The fact that this is unitary follows from the right cancellation property of $\oplus$.) However, we will more often be interested in queries to a function that is also stored in a quantum register: For a set $\mathsf{Func} \subseteq D \to R$ that will always be clear from the context, define the unitary $U_{query,k}$ on $\mathbb{C}^{D^k} \otimes \mathbb{C}^{\tilde{R}^k} \otimes \mathbb{C}^{\mathsf{Func}}$ by:

$$U_{query,k}|\underline{x}\rangle|\underline{y}\rangle|h\rangle = |\underline{x}\rangle|\underline{y} \oplus h(\underline{x})\rangle|h\rangle$$

For $k = 1$ this makes a single query (and $\underline{x}, \underline{y}$ are simply elements of $D, \tilde{R}$). For $k > 1$, this implements several queries to the function h in one go (parallel queries). We abbreviate $U_{query} := U_{query,1}$, i.e., $U_{query}|x\rangle|y\rangle|h\rangle = |x\rangle|y \oplus h(x)\rangle|h\rangle$ for $x \in D$, $y \in \tilde{R}$, $h \in \mathsf{Func}$.

Finally, we need the following technical definition: The projector $\mathbb{P}_r$ projects a superposition of functions to those compatible with the partial function r: For a set

[4] Or equivalently: $\|(1 - P)\psi\| \leq \varepsilon$ where P is the projector onto S.

$\mathsf{Func} \subseteq D \to R$ that is clear from the context, and for $r : D \rightarrow\!\!\!\!\!+\; R$, define

$$\mathbb{P}_r := \sum_{\substack{h \in \mathsf{Func} \\ h \heartsuit r}} |h\rangle\langle h|.$$

3 Compressed oracles

3.1 Zhandry's compressed oracles

We will now recapitulate and rephrase *Zhandry's compressed oracle technique* [Zha19]. trying to emphasize more the separation between implementation issues (encoding via "databases" etc.) and the core concepts. Then we proceed to give a different view on the technique that does not involve Fourier transforms and which is, in our opinion, conceptually simpler. This will also form the motivation for our new construction that supports permutations.

A reader who wishes to skip this part and to directly learn our new technique can safely skip ahead to Section 3.2.

In a nutshell, the compressed oracle technique is a way to simulate/implement a quantum random oracle (i.e., a uniformly random function $h : D \to R$ to which an adversary or quantum algorithm has superposition query access) in a way that has the following crucial features:

- *The adversary cannot distinguish between the original random oracle and the simulation.* This allows us to use the simulation in proofs instead of the original oracle.
- *The simulation uses an internal state that has a small representation.* This is not the case for trivial implementations of the random oracle: Those would have to pick and store the value table of the random function at the beginning. This value table would require $|D| \cdot \log|R|$ classical bits which is infeasible for typical size of the domain D. In contrast, the compressed oracle only requires roughly $q(\log|D| + \log|R|)$ qubits after q queries to the random oracle.
- *The simulation keeps track where the random oracle was already queried, and what the result of that query is.* E.g., if the adversary queries $h(x)$ (possibly in superposition between different values x), and gets $y := h(x)$, then the simulation will keep a record that a query $x \mapsto y$ was performed (or a superposition of such records). While this is trivial in the classical case, it is highly surprising that this is possible in the quantum case: Naively keeping a record of the queries would entangle the adversary's state with the state of the compressed oracle, something the adversary might detect.[5] Having this record is the arguably the main advantage

[5]For example, the adversary might initialize a register X with $\sum_x \frac{1}{\sqrt{M}}|x\rangle$, then perform a superposition query with input x. Now the compressed oracle needs to record the query $x \mapsto y$ (in superposition between different x). Now the register X is entangled with the compressed oracle's record. (Or, if the compressed oracle would measure the query input, the register X would collapse to a single value.) Now the adversary might wish to distinguish whether the compressed oracle records its queries or not. For that purpose, the adversary uncomputes the previous query. Now X would be in the original state with

of the compressed oracle technique as a proof technique. For example, it allows us to formulate invariants such as "the adversary has not yet queried an x with $h(x) = 0$".

- *The simulation is efficient.* That is, its runtime is polynomial in the number of queries performed by the adversary, and the bitlengths of the inputs and outputs of h. This is closely related to the fact that the internal state has a small representation. Previous approaches for efficiently implementing/simulating the quantum random oracle either required computational assumptions (simulation via quantum pseudorandom functions [Zha12b]) or required the simulator to know the number of queries that the adversary will perform at the outset of the simulation (simulation via $2q$-wise independent functions [Zha12d]).

Through the rest of this section, we present our reformulation of Zhandry's technique before explaining our technique.

Standard oracle. Consider the original quantum random oracle. This oracle initially classically samples a random function $h \xleftarrow{\$} (D \to R)$. And then a query to the function h is implemented by a unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus h(x)\rangle$ on the adversary's query registers X, Y.

It is easy to see that this is perfectly indistinguishable from the following construction (called the *standard oracle* in [Zha19]):[6] An additional quantum register H is initialized with $\sum_{h\in D\to R} \frac{1}{\sqrt{|D\to R|}}|h\rangle$, i.e., with the uniform superposition of all possible functions. The adversary does not get access to this register H, but instead the oracle query is changed to be the unitary $U_{query} : |x\rangle|y\rangle|h\rangle \mapsto |x\rangle|y \oplus h(x)\rangle|h\rangle$ on registers X, Y, H.

To better understand the following steps, imagine that the register H consists of many separate registers H_x ($x \in D$), each H_x storing the output $h(x)$. (That is, h is represented as a value table in H with H_x being the table entries.) Each H_x has Hilbert space $\mathbb{C}^R$.

Compressed oracle. Next, we transform the oracle into yet another representation. First, we extend the registers H_x to allow for a value $|\perp\rangle$, i.e., the Hilbert space is $\mathbb{C}^{R\cup\{\perp\}}$. This means that the register H now contains not total only functions h, but can also contain superpositions of partial functions. ($\perp$ denoting an undefined output.)

---

the original random oracle; the adversary can check whether this is the case. But if the compressed oracle keeps a record of the query $x \mapsto y$, the state X will not be in its original state but entangled with the compressed oracle. So in order to be indistinguishable, the compressed oracle needs to forget the query (i.e., erase the record $x \mapsto y$ from its state). In other words, the compressed oracle needs to not only record queries, but also "unrecord" queries in case of uncomputations. Since the compressed oracle does not know a priori whether a given query is a computation or an uncomputation (or something in between), it would seem impossible to solve this problem. The surprising fact of the compressed oracle technique is that it does solve this problem, almost as a side effect.

[6] The indistinguishability formally follows from the fact that the query commutes with a computational basis measurement of the register H, and the fact that if that computational basis measurement is performed at the beginning of the execution, then it is equivalent to uniformly (classically) sampling h.

Intuitively, $|\bot\rangle$ in some H_x will mean that the corresponding h-output is not yet determined, i.e., that any value is still possible. In particular, having $|\bot\rangle$ in all registers H_x (i.e., having the empty partial function $|\varnothing\rangle$ in H) should correspond to the initial state of the random oracle.

We make this more formal by defining an encoding/decoding operation to map between states that do not use $|\bot\rangle$ (as in the standard oracle) and states that do use $|\bot\rangle$ (compressed states).

Let Q denote the quantum Fourier transform on $\mathbb{C}^R$. We extend it to work on the register H_x by defining $Q|\bot\rangle := |\bot\rangle$. (In [Zha19] the specific case $R = \{0,1\}^n$ is considered. In this case the quantum Fourier transform is simply a Hadamard gate on each qubit in H_x. But in [Chu+20] the case for general abelian groups R is considered and other quantum Fourier transforms are used.) Let $U_\bot$ be the unitary with $U_\bot|\bot\rangle = |0\rangle$, $U_\bot|0\rangle = |\bot\rangle$, $U_\bot|y\rangle = |y\rangle$ for $y \neq 0$. Let $\mathsf{Decomp}_1 := Q \cdot U_\bot \cdot Q^\dagger$ (the *decompression operation*).

In the standard oracle, H_x has initial state $\sum_y \frac{1}{\sqrt{|R|}}|y\rangle$. If we apply $\mathsf{Decomp}_1^\dagger$ to it, we get

$$\sum_y \tfrac{1}{\sqrt{|R|}}|y\rangle \overset{Q^\dagger}{\longmapsto} |0\rangle \overset{U_\bot^\dagger}{\longmapsto} |\bot\rangle \overset{Q}{\longmapsto} |\bot\rangle.$$

Thus, by applying $\mathsf{Decomp}_1^\dagger$ to all registers H_x in the initial state of the standard oracle, we get $|\bot\rangle$ in every H_x. This leads to the following idea: Initialize all H_x with $|\bot\rangle$. And whenever we want to perform an oracle query, we decompress all H_x by applying Decomp_1 (for the initial state, this gives the initial state of the standard oracle). Then we apply U_{query} (the standard oracle). And then we compress all H_x again by applying $\mathsf{Decomp}_1^\dagger$. This will lead to exactly the same behavior as the standard oracle (since successive Decomp_1, $\mathsf{Decomp}_1^\dagger$ pairs cancel out).

In other words, we define the compressed oracle to be the oracle with the initial state $|\bot\rangle \otimes \cdots \otimes |\bot\rangle$ in register H, and that applies the following unitary to X, Y, H on each query:

$$\mathsf{CStO} := (I_X \otimes I_Y \otimes \mathsf{Decomp}^\dagger) \cdot U_{query} \cdot (I_X \otimes I_Y \otimes \mathsf{Decomp})$$
$$\text{with} \qquad \mathsf{Decomp} := \bigotimes_{x \in D} \mathsf{Decomp}_1. \quad (1)$$

Now CStO is perfectly indistinguishable from the standard oracle.

The size of the compressed oracle. So far, we have seen that the compressed oracle CStO simulates the standard oracle. But why is it useful? To see this, we will think of the register H as containing partial functions: The basis states of H are $|y_{x_1}, \dots, y_{x_N}\rangle$ for $y_{x_1} \in R \cup \{\bot\}$ where $D = \{x_1, \dots, x_N\}$. This is the value table of a partial function $f : D \rightharpoonup R$. We will identify $|y_{x_1}, \dots, y_{x_N}\rangle$ with $|f\rangle$. In particular, the initial state of CStO is then $|\bot, \dots, \bot\rangle = |\varnothing\rangle$.

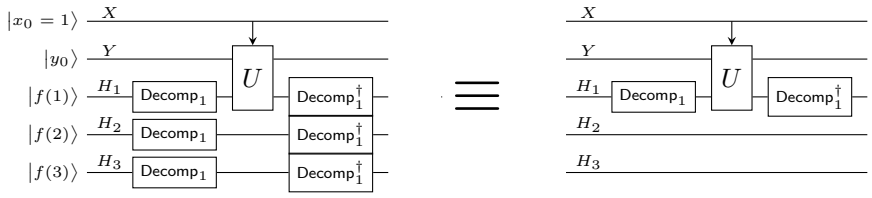


Figure 1: Operation of CStO for fixed $x_0 := 3$. U denotes the operation $|y_0\rangle|y\rangle \mapsto |y_0 \oplus y\rangle|y\rangle$.

Consider a state $|x_0\rangle|y_0\rangle|f\rangle$ before a query to the compressed oracle, with $|\mathrm{dom}\, f| \leq \ell$. (The initial state has $\ell = 0$.)

Applying CStO to this state will decompress all H_x (which does not affect $|x\rangle$) apply U_{query} (which does not affect H_x for $x \neq x_0$ for this particular state), and then compress all H_x again. This is illustrated in the left side of Figure 1 for $x_0 := 3$. On all H_x with $x \neq x_0$, Decomp_1 and $\mathsf{Decomp}_1^\dagger$ cancel out (see the right side of Figure 1). Thus, no matter what x_0, y_0, f are, the resulting state will be a superposition of f' with $f' = f$ except on x_0. In particular, $|\mathrm{dom}\, f'| \leq |\mathrm{dom}\, f| + 1 \leq \ell + 1$. Since this holds for any $|x_0, y_0, f\rangle$ with $|\mathrm{dom}\, f| \leq \ell$, this also holds for any superposition of such states. Thus we have shown[7] that any state of the compressed oracle that is a superposition of $|f\rangle$ of size $\leq \ell$ will, after a query, be a superposition of $|f\rangle$ of size $\leq \ell + 1$.

In particular, after q queries, the compressed oracle state is a superposition of partial functions of size $\leq q$. Such a partial function can be represented in approximately $q(\log|D| + \log|R|)$ bits, hence the state of the compressed oracle indeed has a much smaller representation, hence the name "compressed oracle".

And we also can see that it indeed "records" queries in some sense: if the state of the oracle contains $|f\rangle$, then every $x \in \mathrm{dom}\, f$ must have been queried. Otherwise we would have $f(x) = \perp$ as in the initial state. The converse does not hold, though, because queries can be uncomputed and thus removed from f.

Efficient implementation. So far, we do not have an efficiently simulatable oracle because we represent the state of the compressed oracle by giving the complete value table for the partial functions f. (Each potential output is stored in a different register H_x.) However, an algorithm implementing CStO is free to store the partial functions in a more efficient way, namely as sorted lists of input/output pairs (called a *database* in [Zha19]), leading to a compact state. And an efficient circuit for the unitary CStO can be constructed by only applying Decomp_1 on those entries of the database that are involved in the present query. This then gives the oracle defined in [Zha19]. We omit the details here.

The advantage of separating definition of the efficient encoding of the compressed oracle state from the conceptual encoding as a partial function is that proofs will have to consider the concrete encoding with ordered association lists only when analyzing the runtime of the simulation and can use the mathematically simpler concept of partial

[7] Actually, we have handwavingly sketched it but a formal proof is easy and follows the same ideas.

functions everywhere else. In particular, in information-theoretical proofs, we do not need to consider the efficient encoding at all.

Getting rid of the Fourier transform. So far, we have described the compressed oracle as in Zhandry's original work (although with a different presentation). As presented originally, it would seem that the Fourier transform is an integral part of the idea of the compressed oracle.[8] We will now show that there is a different view which does not involve the Fourier transform at all. Recall the definition of $\mathsf{Decomp}_1 = Q \cdot U_\perp \cdot Q^\dagger$. Using that definition, we can compute what Decomp_1 does to various basis states:

$$|\perp\rangle \overset{Q^\dagger}{\longmapsto} |\perp\rangle \overset{U_\perp}{\longmapsto} |0\rangle \overset{Q}{\longmapsto} \sum_z Q_{z0}|z\rangle =: |*\rangle$$

$$|y\rangle \overset{Q^\dagger}{\longmapsto} \sum_z \overline{Q_{yz}}|z\rangle \overset{U_\perp}{\longmapsto} \underbrace{\sum_z \overline{Q_{yz}}|z\rangle}_{=Q^\dagger|y\rangle} + \underbrace{\overline{Q_{y0}}}_{=\langle *|\, y\rangle} \big(|\perp\rangle - |0\rangle\big) \overset{Q}{\longmapsto} |y\rangle + \langle *|\, y\rangle\big(|\perp\rangle - |*\rangle\big).$$

Note that this calculation did not use that Q is the Fourier transform, only the fact that it is unitary. And the state $|*\rangle$ is simply the first column of Q. Which, in case of the Fourier transform, is of course the uniform superposition $|*\rangle = \sum_z \frac{1}{\sqrt{N}}|z\rangle$. However, any other unitary with the same first column would lead to the same result – the definition of Decomp_1 does not actually use the Fourier transform, and it only depends on the first column of Q! In fact, the above calculation works even if the first column is not the uniform superposition. For example, if we wish to analyze random oracles that use a random function h that is not uniformly chosen, but where each $h(x)$ is independently chosen according to some distribution $\mathcal{D}$, we take a unitary Q whose first column is $|*\rangle := \sum_z \frac{1}{\sqrt{\mathcal{D}(z)}}|z\rangle$, and now Decomp_1 still maps

$$\begin{aligned} &\mathsf{Decomp}_1 : |\perp\rangle \mapsto |*\rangle \\ &\mathsf{Decomp}_1 : |y\rangle \mapsto |y\rangle + \langle *|\, y\rangle\big(|\perp\rangle - |*\rangle\big). \end{aligned} \tag{2}$$

In fact, we can just take this as the definition of Decomp_1 (relative to a given $|*\rangle$). The operators Q and $U_\perp$ are then just a technical tool to show that Decomp_1 is indeed unitary, and one possible way of implementing Decomp_1 efficiently, but they are not part of its definition.

We can still define an oracle CStO based on this new Decomp_1 in the same way as before via (1). Except now CStO will be indistinguishable from the standard oracle that has the initial state $|*\rangle \otimes \cdots \otimes |*\rangle$. Which is indistinguishable from the original random oracle if $|*\rangle$ is the uniform superposition. And if $|*\rangle = \sum_z \alpha_z|z\rangle$, then it is indistinguishable from the a random oracle where each $h(x)$ is sampled to be y with probability $|\alpha_y|^2$.[9] Everything discussed so far still applies. In particular, we still have

[8] [Zha19] considers the special case of a qubit-wise Hadamard which is the Fourier transform over the abelian group $\{0,1\}^n$. [Chu+20] generalizes this to Fourier transforms over arbitrary abelian groups.

[9] We could go even farther and use a different $|*\rangle$ for every x. This would allow us to analyze oracles where $h(x)$ is picked from different distributions for different x.

that the oracle is compressed and records queries: In the compressed state, for any x that has not been queried, H_x will be in state $|\bot\rangle$ (with the intuitive meaning that the value of $h(x)$ is not sampled yet).

Thus, by removing the Fourier transform from the picture, we have generalized the compressed oracle technique to nonuniformly distributed oracles "for free".[10] However, we stress that this approach does not yet allow us to model random permutations because a random permutation h does not have *independently* distributed $h(x)$.[11]

In our opinion, this new view of the compressed oracle has multiple advantages:

- It becomes clearer what the essence of the transformation Decomp_1 is (see also the discussion below). To assume that the Fourier transform plays a relevant role in the construction may even hinder understanding of what is really happening.
- There is no need to find a group structure on the range R of the function so that it matches the operation $\oplus$ in the definition of the oracle query unitary $|x\rangle|y\rangle \mapsto |x\rangle|y \oplus h(x)\rangle$. This may lead to less requirements in proofs.
- The technique becomes more general as we are not limited to uniformly distributed functions.

Understanding Decomp. In order to better understand what the decompression operation does, let us have another look at the definition. Also, for simplicity, let us assume

[10] [Cza+20] also generalizes Zhandry's technique to non-uniformly distributed functions. (With the condition that the outputs are independently sampled, i.e., not covering permutations.) However, their presentation still involves Fourier transforms.

[11] We had one *failed* approach how to generalize this to random permutations (and possibly other function distributions). Since we believe that this approach might be natural, we shortly describe it here and why we got stuck trying to use it:

For $S \subseteq R$, we can define Decomp_1^S to be the Decomp_1 operation for the uniform distribution on S. (I.e., Decomp_1^S is defined by (2) where $|*\rangle := \sum_{y\in S} \frac{1}{\sqrt{|S|}}|y\rangle$.) Then we can define $\mathsf{Decomp}_1^{@x}$ to apply Decomp_1^M on register H_x, where M is the set of all values that are not yet used in other registers. Formally, if $D = \{x_1, \dots, x_M\}$, $\mathsf{Decomp}_1^{@x_i}|y_1, \dots, y_M\rangle := |y_1, \dots, y_{i-1}\rangle \otimes \mathsf{Decomp}_1^{S_i}|y_i\rangle \otimes |y_{i+1}, \dots, y_M\rangle$ where $S_i := R \setminus \{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_M\}$. (Here all $y_i \in R \cup \{\bot\}$.) And then we can define a decompression for permutations as $\mathsf{Decomp}_1^{perm} := \mathsf{Decomp}_1^{@x_M} \mathsf{Decomp}_1^{@x_{M-1}} \cdots \mathsf{Decomp}_1^{@x_2} \mathsf{Decomp}_1^{@x_1}$.

It is reasonably easy to verify that $\mathsf{Decomp}_1|\bot \dots \bot\rangle = \sum_{h:D\hookrightarrow R} \frac{1}{\sqrt{|D\hookrightarrow R|}}|h\rangle$. So decompressing the initial state indeed leads to a uniform superposition of permutations (more precisely, of injections).

And it is also easy to define an oracle query in this model, namely $\mathsf{CStO}^{perm} := \mathsf{Decomp}^{perm\,\dagger} \otimes U_{query} \otimes \mathsf{Decomp}^{perm}$.

However, beyond that, things become difficult. First, the definition of Decomp_1 depends on the ordering of the domain D. If we would apply the $\mathsf{Decomp}_1^{@x_i}$ in a different order, we would get a different operator Decomp_1^{perm}. Second, it becomes very difficult to understand the behavior of CStO^{perm}. We were unable to give an explicit description of how it operates on a basis state. And it is not clear that CStO^{perm} maps a state $|y_1 \dots\rangle$ where at most ℓ of the $y_i \neq \bot$ to a superposition of states $|\tilde{y}_1 \dots\rangle$ where at most $\ell + 1$ of the $\tilde{y}_i \neq \bot$. But if this does not hold, then we do not have a compressed oracle because we have no upper bound on the size of the oracle state.

However, we do not exclude that these problems could be solved and the approach made viable. For example, it might be possible to find some operator that approximately implements CStO^{perm} and that has an easy description and that does not grow the state too much during a query. But we were unable to find such an operator.

for now that $|*\rangle$ is the uniform superposition.

$$\begin{aligned}&\mathsf{Decomp}_1 : |\bot\rangle \mapsto |*\rangle\\&\mathsf{Decomp}_1 : |y\rangle \mapsto |y\rangle + \underbrace{\langle *|\, y\rangle \big(|\bot\rangle - |*\rangle\big)}_{\text{correction term}}\end{aligned} \tag{3}$$

And thus Decomp operates as follows:

$$\mathsf{Decomp} : |y_1 y_2 y_3 \ldots\rangle \mapsto |\hat{y}_1 \hat{y}_2 \hat{y}_3 \ldots\rangle + \mathit{correction}$$

where $\hat{y} := y$ for $y \in R$ and $\hat{y} := *$ for $y = \bot$, and where *correction* is a sum of tensor products of "correction terms".

This means that in the compressed oracle state, $|\bot\rangle$ is used to denote the uniform superposition in H_x, i.e., an output that is completely undetermined so far. On the other hand, $|y\rangle$ in the compressed oracle state has a somewhat more subtle meaning. Intuitively, we might expect/want that $|y\rangle$ in the compressed oracle state means that the output is y. I.e., $|y\rangle$ in the compressed state should translate to $|y\rangle$ in the uncompressed state. In other words, the intuitively natural definition of Decomp_1 would be the definition with the "correction term" removed. Unfortunately, the resulting operation would not be unitary. So the purpose of the correction terms is to stay as close to mapping $|y\rangle$ to $|y\rangle$ as possible, while keeping the operation unitary. Note that the correction terms are small because $\langle *|\, y\rangle = 1/\sqrt{N}$.[12]

This leads to a different view of how Decomp_1 could be derived: Instead of constructing it bottom-up from Q and $U_\bot$, we could use the ansatz that Decomp is an operator defined as:

$$\begin{aligned}&\mathsf{Decomp} : |\bot \ldots \bot\rangle \mapsto |*\rangle \ldots |*\rangle\\&\mathsf{Decomp} : |y_1 y_2 y_3 \ldots\rangle \mapsto |\hat{y}_1 \hat{y}_2 \hat{y}_3 \ldots\rangle + \mathit{correction}\end{aligned} \tag{4}$$

where *correction* must be chosen in such a way that Decomp becomes unitary, such that *correction* is as small as possible, and – most importantly – that the correction terms do not make the compressed oracle state bigger (i.e., when starting with $|y_1 y_2 \ldots\rangle$ where there are at most ℓ non-$\bot$ entries, decompressing with Decomp, applying the oracle query operation U_{query}, and then recompressing with $\mathsf{Decomp}^\dagger$, we should get a superposition of states $|y'_1 y'_2 \ldots\rangle$ with at most $\ell + 1$ non-$\bot$ entries). The definitions of Decomp given above are then just one (although very natural) solution to this ansatz.

Mainly, we presented this approach to give a different view on the compressed oracle technique (that hopefully gives some intuition about what is going on). But maybe this approach is also one way to extend the compressed oracle technique to more complex cases such as oracles with non-independently chosen outputs or similar. We did not manage to do so with the random permutation case, but this view did lead us to the approach presented in the next section.

[12] However, the fact that they are small does not, unfortunately, mean that we can ignore them in calculations. They do, in many situations, add up to very relevant errors. In fact, Decomp_1 without the correction terms has operator norm 2 which means that the errors can be as big as the state itself.

3.2 The non-orthogonal view

In the preceding section, we saw that one way to interpret the compressed oracle technique is the following:

We want to simulate the "standard oracle" (whose state is the equal superposition of all functions $D \to R$). The initial state of the standard oracle is $|\varnothing\rangle^{\mathsf{f}} := \sum_{h:D\to R} \frac{1}{\sqrt{N^M}}|h\rangle$. (Recall from the preliminaries that $M := |D|$ and $N := |R|$ throughout the paper.) Other states of interest are states where some of the function outputs are fixed (e.g., $h(x_1) = y_1$) while other function outputs are still undetermined (in superposition between all possible outputs). A compact way of writing such a state (that we will use extensively in the following) is given by the following definition:

Definition 1 (Compressed function oracle states) *For $f : D \rightharpoonup R$, we define $|f\rangle^{\mathsf{f}} \in \mathbb{C}^{D\to R}$ as follows:*

$$|f\rangle^{\mathsf{f}} := \sum_{\substack{h:D\to R \\ h\diamond f}} \frac{1}{\sqrt{\big|\{h \in (D\to R) : h \diamond f\}\big|}}\,|h\rangle.$$

Since $h \diamond f$ (h compatible with f, see the preliminaries) means that $h(x) = f(x)$ where f is defined and $h(x)$ can be arbitrary elsewhere, this is exactly a state that is the superposition of all h that are fixed in some places and arbitrary in others.

The approach from the previous section was to find a basis for $\mathbb{C}^{D\to R}$ consisting of states that are close to the states of the form $|f\rangle^{\mathsf{f}}$, and then construct an operator

$$\mathsf{Decomp} : |f\rangle \mapsto |f\rangle^{\mathsf{f}} + \textit{correction} =: \widetilde{|f\rangle^{\mathsf{f}}}$$

for a suitable correction term. (This is exactly (4), but using the notation from Definition 1.) Then the state of the oracle can be represented in terms of states $|f\rangle$ (in the compressed oracle) and mapped using Decomp to states $\widetilde{|f\rangle^{\mathsf{f}}}$ of the standard oracle. Or alternatively, we can say that we represent the state of the standard oracle in the orthogonal basis $\widetilde{|f\rangle^{\mathsf{f}}}$. And the definition of this basis ensures that the size of the partial functions f grows at most by one upon each query.

Without including the correction terms, this does not work because the $|f\rangle^{\mathsf{f}}$ are not orthogonal and thus Decomp would not be unitary. However, it turns out that in many situations, we do not need to define Decomp at all! Nor do we need a basis $\widetilde{|f\rangle^{\mathsf{f}}}$! Instead, we simply describe the current state of the standard oracle as a superposition of states $|f\rangle^{\mathsf{f}}$. This representation is not unique, but to model the fact that the oracle state has a compact representation, we do not always need that this representation is unique!

In more detail, our variant of the compressed oracle technique is as follows:

- We want to solve a query problem. That is, we want to bound the probability that the adversary finds some inputs to the random oracle that satisfy a certain property. Let us call such inputs "bad". (E.g., we might want to bound the probability that an adversary finds a collision in the random oracle.)

- The oracle that we use is the standard oracle. That is, the initial state of the oracle is $|\varnothing\rangle^{\mathsf{f}}$ in some register H that is not accessible to the adversary. ($|\varnothing\rangle^{\mathsf{f}}$ is the equal superposition between all functions since $\varnothing$ is the empty partial function and $h \circledcirc \varnothing$ holds for all h.)
- When the adversary performs a query (using query registers X, Y), we model this by applying the unitary U_{query} defined in the preliminaries to X, Y, H. (Or, if we want to model k-parallel queries, we use $U_{query,k}$.) So far, nothing is "compressed".
- We state an *invariant* that is supposed to hold during the execution of the adversary with access to the oracle. I.e., we state an invariant I_j that supposedly applies to the register H after the j-th query. (And I_0 in the beginning.) The invariant I_j is a subspace expressed in terms of the states $|f\rangle^{\mathsf{f}}$. That is, it is of the form $I_j = \operatorname{span}\{|f\rangle^{\mathsf{f}} : f \in A\}$ for some set A. (For example, we could state the invariant $I_j := \operatorname{span}\{|f\rangle^{\mathsf{f}} : |\operatorname{dom} f| \leq j,\ f \text{ injective}\}$. This would mean that there is no collision in the part of the oracle that has been queried so far.)
  (In more general cases, I_j can also refer to other registers than H, e.g., $I := \operatorname{span}\{|s\rangle|f\rangle^{\mathsf{f}} : s \notin \operatorname{dom} f\}$ on registers S, H would be an invariant stating that the adversary has not yet found the value s that is stored in register S.)
- We show that if I_{j-1} holds before the j-th query, then I_j holds approximately after the j-th query. More precisely, for any state $\psi \in \mathcal{H}_{rest} \otimes I_{j-1}$ (where $\mathcal{H}_{rest}$ is the Hilbert space of all registers besides H), we have that $U_{query}\psi$ is ε_j-close to a state in $\mathcal{H}_{rest} \otimes I_j$ for some ε_j. (Closeness is with respect to $\|\cdot\|$.)
  Finding a good bound for ε_j is, of course, the tricky part of this recipe. Fortunately, we present some theorems for bounding ε_j in terms of some purely combinatorial expression (bounding the size of some sets of functions) that strongly facilitate this. See also the next section for an example.
- Now we know that the final state ψ_{final}, after q queries, is $\varepsilon := (\sum_{j=1}^{q} \varepsilon_j)$-close to a state in $I_{final} := I_q$. We then measure the final state (and in particular the adversary's output register Out). Assume that we formulated I_{final} in such a way that measuring a "bad" value is 0 for a state in I_{final}. Then we know that the probability of measuring a bad value in ψ_{final} is $\leq \varepsilon^2$.
  (For example, I_{final} might be defined as $\operatorname{span}\{|x\rangle|x'\rangle|f\rangle^{\mathsf{f}} : \neg(x \neq x' \wedge f(x) = f(x'))\}$ on registers Out, H to state that the output register does not contain a collision.)

In the following section, we will show by example how this is done in concrete cases. Note that what we have described so far is for random function oracles, not for random permutation oracles. (I.e., for the same oracles as Zhandry's compressed oracle.) We cover random permutations in Section 4.

Our "non-orthogonal view" has some advantages and disadvantages over the original compressed oracle technique:

Pro:

- *The approach is conceptually simpler.* No correction terms are needed, for example. And no special compression/decompression operation.

Contra:

- *Harder to define operations that depend on the compressed state.* Since our approach does not provide a canonical orthogonal basis in terms of compressed

- *Invariants mean what they say, not just approximately.* For example, the invariant $I := \mathrm{span}\{|f\rangle^{\mathsf{f}} : f(0) = 1\}$ is satisfied by exactly by the oracle states that are the superposition of $|h\rangle$ with $h(0) = 1$. Thus if we measure $h(0)$, we get 1 with probability 1. In contrast, if Zhandry's compressed oracle has a state satisfying the analogous invariant, measuring can result in $h(0) \neq 1$ with probability approximately $\frac{1}{N}$ which is not intuitive when the invariant is satisfied perfectly (not up to an ε-error).[13] And similarly, if on an arbitrary oracle state we measure $h(0)$ and get y, then in our setting we know that afterwards the invariant $\mathrm{span}\{|f\rangle^{\mathsf{f}} : f(0) = y\}$ holds (Theorem 4 below).
- *The approach generalizes to non-uniformly distributed oracles.* It is not restricted to functions with independently sampled outputs. In this work we mainly use it to implement invertible random permutations, but our central theorems also apply to more general cases.

states, there is no canonical way to measure the content of the compressed oracle, or to implement conditioned unitaries that depend on it. For example, it is not well-defined to say "measure the smallest x such that $f(x) \neq \perp$" since there is no unique decomposition of the state into states $|f\rangle^{\mathsf{f}}$.[14] In contrast, in Zhandry's compressed oracle such a measurement would be well-defined. (Although its behavior might not be exactly what we imagine intuitively due to the fact mentioned in the pro-column.) This may be a problem when tackling problems where a simulator performs actions depending on the content of the compressed oracle, e.g., in indifferentiability proofs (e.g., [Zha19; Cza+20]).
- *Efficient implementation unclear.* The methodology does not give rise to an obvious way of implementing the compressed oracle *efficiently*. We leave efficient implementations to future work.

3.3 Example: Zero search

To illustrate our technique, we will show that it is hard to find a zero-preimage of a random function. More precisely:

Lemma 2 (Hardness of zero search) *Let A be an oracle algorithm that performs q queries. Let h be sampled uniformly at random from $D \to R$. Then $p := \Pr[h(x) = 0 : x \leftarrow A^h()] \leq \frac{16(q+1)^2}{N}$.*

Preparations. To prove this lemma, we start with some preparations to bring the game into a form suitable for using our technique. Namely, we make sure that everything is unitary, that we use the standard oracle, and that the success of the adversary can be read off from the outcome of a measurement of some register.

[13] A state that satisfies the invariant is $|0 \mapsto 1\rangle = |0\rangle|\perp\rangle|\perp\rangle\dots$. After decompressing, we get $\big(|1\rangle + \langle *|\,0\rangle(|\perp\rangle - |*\rangle)\big) \otimes |\perp\rangle|\perp\rangle\dots$ where $|*\rangle := \sum_z \frac{1}{\sqrt{N}}|z\rangle$. The first register has a component orthogonal to $|0\rangle$ of amplitude approximately $\langle *|\,0\rangle = \frac{1}{\sqrt{N}}$, thus measuring this register yields $\neq 1$ with probability approximately $\frac{1}{N}$.

[14] We do not say there is no such measurement in general. Only that we have no canonical definition. Future research may provide such a canonical definition, hopefully also for the permutation case.

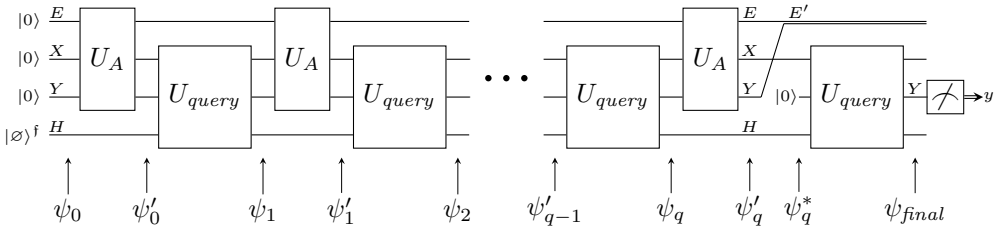


Figure 2: Circuit describing the zero search game

First, note that without loss of generality, we can assume A to be unitary. (I.e., A does not perform any measurements or other classical operations.) For simplicity, we assume that the output of A is in the register X (which is also used for oracle queries). Let U_A denote the unitary that A performs between oracle queries. U_A operates on E, X, Y where X, Y are the oracle query registers (state spaces $\mathcal{H}_X = \mathbb{C}^D$, $\mathcal{H}_Y = \mathbb{C}^R$), and E is the register holding the adversary's state (state space $\mathcal{H}_E$). Initially, E, X, Y are in state $|0\rangle$.

Second, we can reformulate the game from the lemma slightly: Instead of measuring A's output x and classically evaluating $h(x)$, we first run A^h. Then we query h in superposition on input X. (I.e., we initialize Y with $|0\rangle$ and then perform a random oracle query $|x, y\rangle \mapsto |x, y \oplus h(x)\rangle$ on the registers X, Y.) And then we measure Y. Then the adversary success probability p is the probability that this measurement returns $y = 0$.

Third, we use the standard oracle instead of the classically sampled h. That is, we initialize an extra register H with the uniform superposition of all $|h\rangle$ (for $h \in (D \to R)$), i.e., with the state $|\varnothing\rangle^{\mathfrak{f}}$. And when A queries the oracle, we apply U_{query} to X, Y, H. (And the final query that computes y is also implemented using U_{query}.)

The result of those three transformations is depicted in Figure 2.

Stating the invariants. Essentially, our invariant is that the adversary has not queried the oracle at an input x that returns 0. In other words, the state of the oracle is a superposition of states $|f\rangle^{\mathfrak{f}}$ with $0 \notin \operatorname{im} f$. (Since $\operatorname{dom} f$ contains all x's that have been queried (intuitively at least), $\operatorname{im} f$ are the corresponding outputs.) However, the invariant that we actually state is slightly different: we additionally include the fact that f has size $\leq j$ after j queries.[15] Furthermore, I_j makes no claims about the state of the registers E, X, Y. Formally:

$$I_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \operatorname{span}\big\{|f\rangle^{\mathfrak{f}} : 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \leq j\big\}_f \qquad \text{for } j \leq q.$$

The invariant after the last query (just before the measurement of y) is slightly different: After the last query, X contains the query input, and Y contains the query result in the

[15] In the present lemma, the size of f is not actually used except that the way we formulated Theorem 3 requires it. However, in other situations such as collision-resistance, or when we analyze random permutations, the size of f plays an active role in the computation of our bounds.

computational basis. (In the queries performed by the adversary, we were not able to make such a claim because there Y was not $|0\rangle$-initialized before the query.) We include this fact in the invariant. That is,

$$I_{\mathit{final}} := \mathcal{H}_E \otimes \operatorname{span}\bigl\{|x\rangle|f(x)\rangle|f\rangle^{\mathsf{f}} : x \in \operatorname{dom} f,\ 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \le q+1\bigr\}_{xf}.$$

Proving that the invariants are maintained. We will now show that after the j-th oracle query, the invariant I_j holds approximately. Specifically, we will show that ψ_j, the state after applying the j-th oracle query, satisfies

$$\psi_j \overset{j\sqrt{16/N}}{\approx} I_j \qquad \text{for} \qquad j = 1, \dots, q+1. \tag{5}$$

(Recall from the preliminaries that this means that ψ_j is close in terms of norm to the subspace I_j.)

The initial state ψ_0 is $|0\rangle|0\rangle|0\rangle|\varnothing\rangle^{\mathsf{f}}$. Since $0 \notin \operatorname{im}\varnothing$ and $|\operatorname{dom}\varnothing| = 0$, we trivially have that the initial state is in I_0 and thus $\psi_0 \overset{0}{\approx} I_0$.

Note that U_A preserves the invariant (since it operates only on E, X, Y). Thus $\psi_0' = U_A\psi_0 \overset{0}{\approx} I_0$ as well. (ψ_j' is the state before the $(j+1)$-st query.)

Assume we can show that $\psi_{j-1}' \overset{\varepsilon}{\approx} I_{j-1}$ implies $\psi_j \overset{\varepsilon+\sqrt{16/N}}{\approx} I_j$ (i.e., that the j-th invocation of U_{query} makes the invariant worse only by $\sqrt{16/N}$), then we get by induction (and using the fact that U_A preserves the invariant) that (5) holds for $j = 0, \dots, q$. (The case $j = q+1$ will need special treatment.)

We get this from the following general theorem for invariant preservation during the compressed queries:

Theorem 3 (Random function query, simple) *Fix an integer $\ell \ge 0$ (compressed function size). Fix $c \ge 0$. Let E, X, Y, H be subsystems with Hilbert spaces $\mathcal{H}_E$, $\mathcal{H}_X := \mathbb{C}^D$, $\mathcal{H}_Y := \mathbb{C}^{\tilde{R}}$, $\mathcal{H}_H := \mathbb{C}^{D\to R}$. Let $N := |R|$. Assume $N \ge 16$. Let $\{\eta_e\}_e$ be an orthogonal basis of $\mathcal{H}_E$. For each $x \in D$ and e, fix $A_{x,e} \subseteq B_{x,e} \subseteq (D \rightharpoonup R)$ (pre-/postcondition). Assume that for all x, all e, and all compatible $f, g \in A_{x,e}$ with $|\operatorname{dom} f|, |\operatorname{dom} g| \le \ell$ and $x \notin \operatorname{dom} f, \operatorname{dom} g$:*

$$\frac{\Bigl|\{z \in R : f(x := z),\ g(x := z) \notin B_{x,e}\}\Bigr|}{N} \le c. \tag{6}$$

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_H$. If

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\bigl\{\eta_e|x\rangle|y\rangle|f\rangle^{\mathsf{f}} : f \in A_{x,e},\ |\operatorname{dom} f| \le \ell\bigr\}_{\mathit{exyf}}$$

then

$$(I_E \otimes U_{\mathit{query}})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\bigl\{\eta_e|x\rangle|y\rangle|f\rangle^{\mathsf{f}} : f \in B_{x,e},\ |\operatorname{dom} f| \le \ell+1\bigr\}_{\mathit{exyf}}.$$

(The proofs of all theorems stated in the examples are deferred to Section 5.)

To use this theorem, we let $A_{x,e}, B_{x,e} := \{f : 0 \notin \operatorname{im} f\}$ (independent of x, e), $\ell := j-1$, $\psi := \psi'_{j-1}$, and η_e be an arbitrary basis. Then the conclusion of the lemma becomes "$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1}$ implies $\psi_j \overset{\varepsilon+4\sqrt{c}}{\approx} I_j$".

Thus all we need to do is to find the bound c so that (6) is satisfied. This is a simple (and purely combinatorial, i.e., non-quantum) problem. For any $f, g \in A_{x,e}$, we have that $0 \notin \operatorname{im} f, \operatorname{im} g$. The numerator in (6) is the set of value z such that $f(x := z)$ or $g(x := z)$ has 0 in its image. Obviously, the only z that can potentially satisfy this is $z = 0$. Thus the numerator is ≤ 1, and (6) is satisfied with $c := 1/N$.

Inserting this value of c, we indeed get that $\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1}$ implies $\psi_j \overset{\varepsilon+\sqrt{16/N}}{\approx} I_j$.

So we know now that $\psi_q \overset{q\sqrt{16/N}}{\approx} I_q$. We want to derive the same for ψ_{final} but unfortunately, we cannot use Theorem 3 for this because the invariant I_{final} does not fit the shape of the invariants in Theorem 3: I_{final} makes a claim not only about register H, but also about Y.

In fact, in general it would be difficult to make any conclusions about Y (such as that Y contains the output of the oracle query) because if Y is not $|0\rangle$-initialized, its state might be anything after the oracle query.

However, in the specific situation of the $(q+1)$-st oracle query, we know what Y contains because we initialize it with $|0\rangle$. (See Figure 2.) $\psi_q \overset{j\sqrt{16/N}}{\approx} I_q$, and thus also $\psi'_q \overset{j\sqrt{16/N}}{\approx} I_q$ (the state after the last U_A). But ψ^*_q, the result of initializing Y with $|0\rangle$, satisfies a stronger invariant:

$$\psi^*_q \overset{q\sqrt{16/N}}{\approx} \mathcal{H}_{E'} \otimes \mathcal{H}_X \otimes \operatorname{span}\big\{|0\rangle|f\rangle^{\mathsf{f}} : 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \leq q\big\}_f =: I^*_q. \tag{7}$$

(Since the initialization of $|0\rangle$ is an isometry that maps I_q to this invariant.) For the case where Y is $|0\rangle$-initialized, we have a variant of Theorem 3 that keeps track in the postcondition of what the result of the oracle query was (we call this "non-oblivious"):

Theorem 4 (Non-oblivious random function query, simple case) *In the situation of Theorem 3, if*

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\big\{\eta_e|x\rangle|0\rangle|f\rangle^{\mathsf{f}} : f \in A_{x,e}, |\operatorname{dom} f| \leq \ell\big\}_{\mathit{exf}}$$

then

$$(I_E \otimes U_{\mathit{query}})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\big\{\eta_e|x\rangle\big|f(x)\big\rangle|f\rangle^{\mathsf{f}} : x \in \operatorname{dom} f,\ f \in B_{x,e},\ |\operatorname{dom} f| \leq \ell+1\big\}_{\mathit{exf}}.$$

Besides the pre- and postcondition, everything in this theorem is the same as before. In particular, we can still instantiate it with $A_{x,e}, B_{x,e} := \{f : 0 \notin \operatorname{im} f\}$, $\ell := q$, $c := 1/N$. Then the assumption of the theorem is satisfied for $\psi := \psi^*_q$ and $\varepsilon := q\sqrt{16/N}$ by (7), and the conclusion is

$$\psi_{\mathit{final}} = (I_E \otimes U_{\mathit{query}})\psi^*_q \overset{q\sqrt{16/N}+4\sqrt{c}}{\approx} I_{\mathit{final}}.$$

Since $c = 1/N$, this implies (5) for $j = q+1$. Thus our invariant holds till the end.

Concluding. The last step of the circuit in Figure 2 is a measurement of Y in the computational basis. For a state in (not just close to) I_{final}, measuring Y gives $y = 0$ with probability 0. And since ψ_{final} is close to I_{final}, measuring Y in the circuit gives $y = 0$ with probability close to 0. More precisely:

Let P denote the projector corresponding to measuring 0 in Y. By (5), there is a ψ_{ideal} with $\psi_{final} \overset{(q+1)\sqrt{16/N}}{\approx} \psi_{ideal} \in I_{final}$. Then the probability of measuring $y = 0$ given ψ_{final} is

$$\begin{aligned}\|P\psi_{final}\|^2 = \big\|P(\psi_{final} - \psi_{ideal}) + \underbrace{P\psi_{ideal}}_{=0}\big\|^2 &= \big\|P(\psi_{final} - \psi_{ideal})\big\|^2 \\ &\leq \big\|\psi_{final} - \psi_{ideal}\big\|^2 \leq \big((q+1)\sqrt{16/N}\big)^2 = \frac{16(q+1)^2}{N}.\end{aligned}$$

As stated in the beginning of our analysis, the probability of measuring $y = 0$ is the advantage p of the adversary in Lemma 2. Thus we have shown Lemma 2.[16]

Of course, Lemma 2 is not a novel result. It serves mainly to illustrate our technique.

3.4 Example: Collision finding

Our next example is collision finding. One new thing it illustrates is why we need invariants that depend on the environment state (i.e., what is it good for that we have the basis η_e of E in Theorems 3 and 4 and that $A_{x,e}, B_{x,e}$ can depend and e).

Lemma 5 (Collision finding) *Let A be an oracle algorithm that performs q queries. Let h be sampled uniformly at random from $D \to R$. Then $p := \Pr[h(x_1) = h(x_2) \wedge x_1 \neq x_2 : (x_1, x_2) \leftarrow A^h()] \leq \frac{16(q+1)^3}{N}$.*

Preparations. The preparations towards proving this lemma are similar as in the zero search problem (Section 3.3). Namely, we make sure that everything is unitary, and that we use the standard oracle. We assume that the adversary queries the oracle using query registers X, Y with spaces $\mathbb{C}^D$, $\mathbb{C}^{\tilde{R}}$, respectively. Oracle queries are performed using the unitary U_{query}. The adversary's behavior is modeled by a unitary U_A on E, X, Y. However, in the very last invocation of the adversary, we assume a unitary $\hat{U}_A$ instead that inputs registers E, X, Y but outputs registers E', X_1, X_2 where X_1, X_2 have state space $\mathbb{C}^D$. (X_1, X_2 contain the alleged collision x_1, x_2.) Then we initialize a register Y_1 with $|0\rangle$ (for storing the hash of x_1), perform a query on X_1, Y_1 using U_{query}, initialize a register Y_2 with $|0\rangle$ (for storing the hash of x_2), and perform a query on X_2, Y_2 using U_{query}. Finally, we measure X_1, X_2, Y_1, Y_2 in the computational basis, resulting in x_1, x_2, y_1, y_2. Then $p = \Pr[y_1 = y_2 \wedge x_1 \neq x_2]$. The resulting circuit is depicted in Figure 3.

[16] One note: We ignored the assumption $N \geq 16$ in Theorem 3 during the proof. However, Lemma 2 is trivial for $N < 16$, so this does not invalidate the proof.

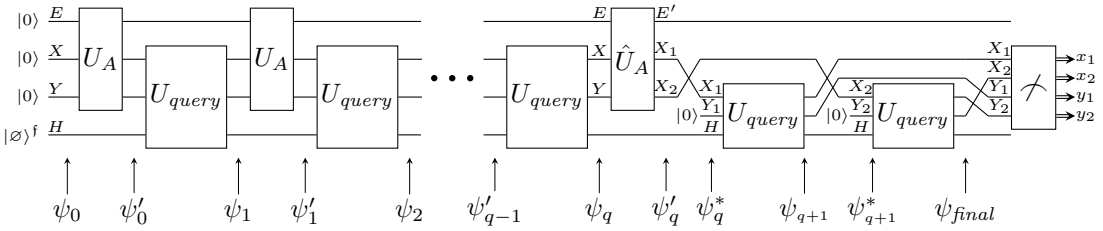


Figure 3: Circuit describing the collision finding game

Stating the invariants. The invariant that we maintain is that the adversary has found no collision so far. In other words, the oracle state is a superposition of states $|f\rangle^{\mathsf{f}}$ where f is injective. And we keep track of the size of f, which grows by at most one in each query. Formally:

$$I_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \operatorname{span}\big\{|f\rangle^{\mathsf{f}} : f \text{ injective},\ |\operatorname{dom} f| \le j\big\}_f \qquad \text{for } j \le q.$$

We state invariant I^*_q that is supposed to hold after initializing Y_1 with $|0\rangle$:

$$I^*_q := \mathcal{H}_{E'} \otimes \mathcal{H}_{X_2} \otimes \mathcal{H}_{X_1} \otimes \operatorname{span}\big\{|0\rangle_{Y_1}|f\rangle^{\mathsf{f}} : f \text{ injective},\ |\operatorname{dom} f| \le j\big\}_f.$$

After invoking U_{query} on ψ^*_q, we expect the following invariant. This invariant additionally states that X_1, Y_1 contain an input/output pair of f.

$$I_{q+1} := \mathcal{H}_{E'} \otimes \mathcal{H}_{X_2} \otimes \operatorname{span}\Big\{|x_1\rangle_{X_1}|f(x_1)\rangle_{Y_1}|f\rangle^{\mathsf{f}} : x_1 \in \operatorname{dom} f,\ f \text{ injective},\ |\operatorname{dom} f| \le q+1\Big\}_{fx_1}.$$

After initializing Y_2 with $|0\rangle$, we expect the invariant:

$$I^*_{q+1} := \mathcal{H}_{E'} \otimes \mathcal{H}_{X_2} \otimes \operatorname{span}\Big\{|x_1\rangle_{X_1}|f(x_1)\rangle_{Y_1}|0\rangle_{Y_2}|f\rangle^{\mathsf{f}} : x_1 \in \operatorname{dom} f,\ f \text{ injective},\ |\operatorname{dom} f| \le q+1\Big\}_{fx_1}.$$

And in the final state, we additionally claim that the inputs/outputs are related according to f:

$$I_{final} := \mathcal{H}_{E'} \otimes \operatorname{span}\Big\{|x_1\rangle_{X_1}|f(x_1)\rangle_{Y_1}|x_2\rangle_{X_2}|f(x_2)\rangle_{Y_2}|f\rangle^{\mathsf{f}} : x_1, x_2 \in \operatorname{dom} f,\ f \text{ injective},\ |\operatorname{dom} f| \le q+2\Big\}_{fx_1x_2}.$$

Proving that the invariants are maintained. Since $\varnothing$ is injective, $\psi_0 \in I_0$. Since U_A (or $\hat{U}_A$) does not touch register H, invariant preservation during invocations of U_A or $\hat{U}_A$ is immediate:

$$\psi_j \overset{\varepsilon}{\approx} I_j \quad \Longrightarrow \quad \psi'_j \overset{\varepsilon}{\approx} I_j \qquad \text{for} \qquad j = 1, \ldots, q.$$

(In the case $j = q$ we have a slight abuse of notation: The rhs is a state in register E', X_1, X_2, H while I_j is defined as an invariant on E, X, Y, H. But I_j only depends on H.)

Similarly, inserting a $|0\rangle$-initialized Y_1 or Y_2 register changes the invariants in the obvious way:

$$\begin{aligned} \psi'_q \overset{\varepsilon}{\approx} I_q \quad &\Longrightarrow \quad \psi^*_q \overset{\varepsilon}{\approx} I^*_q \\ \psi_{q+1} \overset{\varepsilon}{\approx} I_{q+1} \quad &\Longrightarrow \quad \psi^*_{q+1} \overset{\varepsilon}{\approx} I^*_{q+1} \end{aligned}$$

The nontrivial steps are to show that the invariants are approximately preserved by invocations of U_{query}:

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \quad \Longrightarrow \quad \psi_j = (I_E \otimes U_{query})\psi'_{j-1} \overset{\varepsilon+\delta_j}{\approx} I_{j-1} \qquad (j = 1, \dots, q) \tag{8}$$

$$\psi^*_q \overset{\varepsilon}{\approx} I^*_q \quad \Longrightarrow \quad \psi_{q+1} = (I_{E'X_2} \otimes U_{query})\psi^*_q \overset{\varepsilon+\delta_{q+1}}{\approx} I_{q+1} \tag{9}$$

$$\psi^*_{q+1} \overset{\varepsilon}{\approx} I^*_{q+1} \quad \Longrightarrow \quad \psi_{final} = (I_{E'X_1Y_1} \otimes U_{query})\psi^*_{q+1} \overset{\varepsilon+\delta_{q+2}}{\approx} I_{final} \tag{10}$$

for suitable values of δ_j.

All this together then implies:

$$\psi_{final} \overset{\varepsilon}{\approx} I_{final} \qquad \text{with} \qquad \varepsilon := \sum_{j=1}^{q+2} \delta_j. \tag{11}$$

To show (8), we use Theorem 3 with $\ell := j - 1$, $A_{x,e}, B_{\underline{x},e} := \{f : f \text{ injective}\}$, $c := (j-1)/N$. The main task is to bound the numerator in (6), i.e., we bound

$$\Big|\{z \in R : f(x := z),\ g(x := z) \text{ not injective}\}\Big|$$

for compatible injective f, g with $|\operatorname{dom} f|, |\operatorname{dom} g| \le j - 1$ and $x \notin \operatorname{dom} f, \operatorname{dom} g$. Since f is injective, the only z that make $f(x := z)$ non-injective are $z \in \operatorname{im} f$. Since $|\operatorname{im} f| \le |\operatorname{dom} f| \le j - 1$, we have that the numerator is bounded by $j - 1$. Thus (6) indeed holds with $c = (j-1)/N$.

This show (8) with $\delta_j := 4\sqrt{c} = \sqrt{16(j-1)/N}$ for $j = 1, \dots, q$.

And (9) is shown completely analogously, except that we use the non-oblivious Theorem 4 instead of Theorem 3. The resulting bound is $\delta_{q+1} := \sqrt{16q/N}$.

Finally, to show (10), we again use the non-oblivious Theorem 4. However, this time we have to consider an additional subtlety: The invariant I^*_{q+1} states that $x_1 \in \operatorname{dom} f$, where x_1 is stored in register X_1 which is not involved with the current query (the query registers are X_2 and Y_2). Furthermore, Y_1 is supposed to contain $f(x_1)$, i.e., it refers to the function f. And Y_1 is also not involved in the query. Fortunately, invariants of this form can still be handled by the theorems. In the present case, we instantiate Theorem 4 as follows:

- $\mathcal{H}_E := \mathcal{H}_{E'} \otimes \mathcal{H}_{X_1} \otimes \mathcal{H}_{Y_1}$. (The environment consists of registers E', X_1, Y_1). And $\mathcal{H}_X := \mathcal{H}_{X_2}$ and $\mathcal{H}_Y := \mathcal{H}_{Y_2}$. (The query registers are X_2, Y_2.)
- The basis η_e is instantiated as: $\eta_{e',x_1,y_1} := |e'\rangle_{E'}|x_1\rangle_{X_1}|y_1\rangle_{Y_1}$ where $|e'\rangle_{E'}$ are an arbitrary orthonormal basis of E'. (I.e., the indices e of the basis η_e are triples (e', x_1, y_1).)
- $A_{x,(e',x_1,y_1)}, B_{x,(e',x_1,y_1)} := \{f : x_1 \in \operatorname{dom} f,\ f(x_1) = y_1,\ f \text{ injective}\}$
- $\ell := q + 1$, $c := (q + 1)/N$.

With these choices, the precondition in Theorem 4 becomes:

$$\operatorname{span}\big\{|e'\rangle_{E'}|x_1\rangle_{X_1}|y_1\rangle_{Y_1}|x\rangle_{X_2}|0\rangle_{Y_2}|f\rangle^{\mathsf{f}} : f \in A_{x,(e',x_1,y_1)}, |\operatorname{dom} f| \le q+1\big\}_{e'x_1y_1xf} \;=\; I^*_{q+1}.$$

And the postcondition becomes:

$$\operatorname{span}\Big\{|e'\rangle_{E'}|x_1\rangle_{X_1}|y_1\rangle_{Y_1}|x\rangle_{X_2}|f(x)\rangle_{Y_2}|f\rangle^{\mathsf{f}} : x \in \operatorname{dom} f, f \in A_{x,(e',x_1,y_1)}, |\operatorname{dom} f| \le q+2\Big\}_{e'x_1y_1xf} \quad = \quad I_{\mathit{final}}.$$

So Theorem 4 indeed can be used to show (10) with $\delta_{q+2} := 4\sqrt{c} = \sqrt{16(q+1)/N}$. (The bound (6) is shown completely analogously as in the proofs of (8), (9).)

Hence (11) holds.

Concluding. The last step of the circuit in Figure 3 is a measurement of X_1, Y_1, X_2, Y_2 in the computational basis. For a state in (not just close to) I_{final}, measuring X_1, Y_1, X_2, Y_2 yields values x_1, y_1, x_2, y_2 such that $y_1 = f(x_1)$, $y_2 = f(x_2)$ for some injective f with $x_1, x_2 \in \operatorname{dom} f$. This implies that $(x_1 \neq x_2 \wedge y_1 = y_2)$ does not hold. Hence measuring X_1, Y_1, X_2, Y_2 in a state in I_{final} gives $(x_1 \neq x_2 \wedge y_1 = y_2)$ with probability 0. Since $\psi_{\mathit{final}} \overset{\varepsilon}{\approx} I_{\mathit{final}}$, it follows that measuring X_1, Y_1, X_2, Y_2 in ψ_{final} gives $(x_1 \neq x_2 \wedge y_1 = y_2)$ with probability $\le \varepsilon^2$. Thus $p \le \varepsilon^2$. (Recall that p was the advantage of the adversary but also the probability of measuring $(x_1 \neq x_2 \wedge y_1 = y_2)$.) And since $\varepsilon := \sum_{j=1}^{q+2} \delta_j$ and $\delta_j = \sqrt{16(j-1)/N}$ and in particular $\delta_1 = 0$, we have $\varepsilon^2 \le \frac{16(q+1)^3}{N}$. This shows Lemma 5.[17]

3.5 Parallel queries

(This section can be skipped by a reader who is not interested in parallel queries.)

In the preceding sections, we only considered adversaries that perform one oracle query at a time. In some situations, it is beneficial to consider adversaries that perform k oracle queries in parallel, and perform q rounds of such parallel queries. Of course, such an adversary cannot do anything a qk-query adversary cannot do, but the bounds achieved in the parallel-query model can be tighter.

Technically, the only difference between a single query and k-parallel queries to the standard oracle is that the adversary invokes $U_{\mathit{query},k}$ instead of U_{query}. (See the

[17] One note: We ignored the assumption $N \ge 16$ in Theorems 3 and 4 during the proof. However, Lemma 5 is trivial for $N < 16$, so this does not invalidate the proof.

preliminaries for the definition of $U_{query,k}$.) The state of the standard oracle is still the same, a superposition between all possible functions. So all we need to support k-parallel queries are variants of Theorems 3 and 4 that show how invariants evolve under parallel queries.

The two theorems below are these analogues to Theorems 3 and 4. They differ from latter theorems in the following way:

- They support k-parallel queries. To accomplish that, the query registers X, Y now contain superpositions of k-tuples (i.e., their spaces are $\mathbb{C}^{D}, \mathbb{C}^{\tilde{R}}$, respectively). And instead of values x, z, we now quantify over k-tuples $\underline{x}, \underline{z}$ everywhere.
  The biggest change is to the combinatorial condition (12): In the single query case we had a simple dichotomy between $x \notin \operatorname{dom} f$ (in which case $f(x)$ can still be updated to a new value z) and $x \notin \operatorname{dom} f$ (in which case the update does not make sense). With tuples $\underline{x}$ it becomes more difficult since part of $\underline{x}$ might be in $\operatorname{dom} f$. The correct analogue in the k-parallel query case is then $(\underline{x} \mapsto \underline{z}) \diamond f$: This means that for every entry x of $\underline{x}$, $f(x)$ either is undefined (and thus can be updated to the corresponding $\underline{z}$-entry), or it is define but already returns the right $\underline{z}$-entry.

(The parallel-query theorems are actually a generalization of the non-parallel theorems. We will later prove the non-parallel theorems as corollaries of the parallel-query theorems.)

Theorem 6 (Random function query) *Fix integers $k \geq 0$ (query parallelism), $\ell \geq 0$ (compressed function size). Fix $c \geq 0$. Let E, X, Y, H be subsystems with Hilbert spaces $\mathcal{H}_E$, $\mathbb{C}^{D^k}$, $\mathbb{C}^{\tilde{R}^k}$, $\mathcal{H} := \mathbb{C}^{D \to R}$. Let $N := |R|$. Assume $N \geq 16$. Let $\{\eta_e\}_e$ be an orthogonal basis of $\mathcal{H}_E$. For each $\underline{x} \in D^k$ and e, fix $A_{\underline{x},e}, B_{\underline{x},e} \subseteq (D \rightharpoonup R)$ (pre-/postcondition). Assume that for all $\underline{x}$, all e, and all compatible $f, g \in A_{\underline{x},e}$ with $|\operatorname{dom} f|, |\operatorname{dom} g| \leq \ell$:*

$$\frac{\Big|\{\underline{z} \in R^k : (\underline{x} \mapsto \underline{z}) \diamond f, g \text{ and } f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}} \notin B_{\underline{x},e}\}\Big|}{N^{|\underline{x} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|}} \leq c \quad [18] \tag{12}$$

where $f_{\underline{x},\underline{z}} := f \cup (\underline{x} \mapsto \underline{z})$ and $g_{\underline{x},\underline{z}} := g \cup (\underline{x} \mapsto \underline{z})$.

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_H$. If

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\big\{\eta_e|\underline{x}\rangle|\underline{y}\rangle|f\rangle^{\mathsf{f}} : f \in A_{\underline{x},e},\ |\operatorname{dom} f| \leq \ell\big\}_{e\underline{x}\underline{y}f} =: \mathbf{A}$$

then

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\big\{\eta_e|\underline{x}\rangle|\underline{y}\rangle|f\rangle^{\mathsf{f}} : f \in B_{\underline{x},e},\ |\operatorname{dom} f| \leq \ell + k\big\}_{e\underline{x}\underline{y}f} =: \mathbf{B}$$

[18] A note on interpreting this formula: If $\underline{x}$ contains duplicates that are mapped to different values in $\underline{z}$, then at a first glance it may seem that this formula may not be well-defined. (What function is $f_{\underline{x},\underline{z}}$?) However, in this case, by definition of compatibility ($\diamond$, see preliminaries), $(\underline{x} \mapsto \underline{z}) \diamond f$ cannot hold for any function f, thus values $\underline{z}$ that would lead to this problem are simply excluded by the condition $(\underline{x} \mapsto \underline{z}) \diamond f, g$.

If $\underline{x}$ contains duplicates but they map to matching values in $\underline{z}$, then $\underline{x} \mapsto \underline{z}$ is a well-defined function (see prelimiaries, again). Note that in that case, the cardinality $|\underline{x} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|$ in the denominator is a cardinality of sets, not a length of tuples. I.e., repeated elements of $\underline{x}$ are not counted twice.

Theorem 7 (Non-oblivious random function query) *In the situation of Theorem 6, if*

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\bigl\{\eta_e|\underline{x}\rangle|\underline{0}\rangle|f\rangle^{\mathsf{f}} : f \in A_{\underline{x},e},\ |\operatorname{dom} f| \le \ell\bigr\}_{e\underline{x}f}$$

then

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\bigl\{\eta_e|\underline{x}\rangle|f(\underline{x})\rangle|f\rangle^{\mathsf{f}} : \underline{x} \subseteq \operatorname{dom} f,\ f \in B_{\underline{x},e},\ |\operatorname{dom} f| \le \ell + k\bigr\}_{e\underline{x}f}.$$

3.5.1 Zero search with parallel queries

We quickly revisit the example of zero search (Section 3.3) and list the differences in the analysis in the case of parallel queries. We show:

Lemma 8 (Hardness of zero search (parallel queries)) *Let A be an oracle algorithm that performs q queries. Fix an integer $k \ge 0$ (query parallelism). Let h be sampled uniformly at random from $D \to R$. Then $p := \Pr[h(x) = 0 : x \leftarrow A^{h^k}()] \le \frac{16(q\sqrt{k}+1)^2}{N}$. (As opposed to $\le \frac{16(qk+1)^2}{N}$ as one would get as an immediate corollary from Lemma 2.)*

(Note that the adversary A gets h^k, not h. I.e., each of the q queries is a k-parallel query.)

The differences to the proof of Lemma 2 are:

- In all invariants, we replace the condition $|\operatorname{dom} f| \le j$ by $|\operatorname{dom} f| \le jk$. (And analogously $|\operatorname{dom} f| \le q-1$ by $|\operatorname{dom} f| \le (q-1)k$.) This is because in each query, the length increases by up to k.
- We use Theorem 6 instead of Theorem 3. (I.e., we use the theorem with parallel query support.) For the last query (which queries only one value), we can still use the non-oblivious Theorem 4. (Or alternatively, we could use Theorem 7 with $k := 1$.)
- When applying Theorem 6, we need to bound (12) (instead of (6)) with suitable c. Consider compatible $f, g \in A_{\underline{x},e} = \{f : 0 \notin \operatorname{dom} f\}$ with $|\operatorname{dom} f|, |\operatorname{dom} g| \le (j-1)k$. If $\underline{z} \in R^k$ is in the set in the numerator, then: It coincides with $(f \cup g)(\underline{x})$ wherever the latter is defined. And wherever $\underline{x}$ has repetitions, $\underline{z}$ has repetitions, too. (Otherwise $\underline{x} \mapsto \underline{z}$ would not be a function.) That leaves only $t := |\underline{x} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|$ entries that are not constrained to a fixed element, i.e., N^t possibilities for $\underline{z}$. Out of these unconstrained $\underline{z}$-entries one must be 0, otherwise $f \cup (\underline{x} \mapsto \underline{z})$ would be in $B_{\underline{x},e} = \{f : 0 \notin \operatorname{dom} f\}$. This means we have at most tN^{t-1} possibilities. Thus the numerator is bounded by tN^{t-1}. The denominator is N^t. Thus the fraction is bounded by $t/N \le k/N$. Thus (12) is satisfied with $c := k/N$. Thus the bound $4\sqrt{c} = \sqrt{16/N}$ must be replaced by $4\sqrt{c} = \sqrt{16k/N}$.
- In the application of Theorem 4, we keep the same value of c (nothing changed there).
- In total, we thus get $\psi_{final} \overset{\varepsilon}{\approx} I_{final}$ with $\varepsilon := q\sqrt{16k/N} + \sqrt{16/N}$. Hence $p \le \varepsilon^2 = \frac{16(q\sqrt{k}+1)^2}{N}$.

4 Compressed permutations

4.1 Adapting our approach

So far, we have only studied random functions, not random permutations. However, it turns out that it is quite simple to translate the model introduced in Section 3.2 to permutations. (Or more generally, injective functions $D \hookrightarrow R$ with $|R| \geq |D|$.)

In Section 3.2, we introduced the compressed oracle states $|f\rangle^{\mathfrak{f}}$ which are the superposition of all functions h that are compatible with the partial function f. This allows us to describe an oracle that is determined on some $x_1, \ldots, x_n$ by giving a partial function f with $\operatorname{dom} f = \{x_1, \ldots, x_n\}$. And the initial state of the oracle was then simply described by $|\varnothing\rangle^{\mathfrak{f}}$, the superposition of all $|h\rangle$.

For injective functions, we do the same, only we restrict the functions h to be injective. This leads to the following definition:

Definition 9 (Compressed permutation oracle states) *For $f : D \hookrightarrow R$, we define $|f\rangle^{\mathfrak{p}} \in \mathbb{C}^{D \hookrightarrow R}$ as follows:*

$$|f\rangle^{\mathfrak{p}} := \sum_{\substack{h : D \hookrightarrow R \\ h \heartsuit f}} \frac{1}{\sqrt{|h \in (D \hookrightarrow R) : h \heartsuit f|}} |h\rangle.$$

Note the fine differences to Definition 1: all arrows are replaced by hooked arrows, denoting injective total/partial functions.

Now the recipe given in Section 3.2 works exactly in the same way, except that we use $|f\rangle^{\mathfrak{p}}$ instead of $|f\rangle^{\mathfrak{f}}$ everywhere. Since $|\varnothing\rangle^{\mathfrak{p}}$, the initial state, is the equal superposition of all *injections*, this analyzes an oracle that gives access to a random injection. Hence we have a technique for compressed permutation oracles.

Of course, the crucial difference is that now, we need to reason about invariants expressed in terms of states $|f\rangle^{\mathfrak{p}}$, not $|f\rangle^{\mathfrak{f}}$. (And about the errors introduced to them during an oracle query U_{query}.) The behavior of oracle queries on $|f\rangle^{\mathfrak{p}}$ is more complex to analyze because $|f\rangle^{\mathfrak{p}}$ is not a tensor product. (In contrast, $|f\rangle^{\mathfrak{f}} = |f(x_1)\rangle \otimes |f(x_2)\rangle \otimes \ldots$ if we define $|\perp\rangle := \sum_z \frac{1}{\sqrt{N}}|z\rangle$.) For example, if the oracle is in state $|\varnothing\rangle^{\mathfrak{p}}$, the superposition of all injective functions, and we query it on x_0, getting y_0, then we know that querying x_1 will not give y_0. Even in the initial state, all outputs are entangled.

Fortunately, the difficulty of dealing with this can be hidden away in general-purpose theorems for reasoning about invariants, very much like the ones we introduced for the random function case (Theorem 3 etc.) The examples in the following sections show this.

However, note that what we have described so far handles permutation oracles, but not invertible permutation oracles. That is, the examples in the following two sections only give bounds on the adversary success if the adversary does not get access to the inverse of the oracle h. Query complexity results for non-invertible permutations can be easily achieved with existing results (but with worse concrete bounds). Namely, [Zha15a] shows that random functions and random permutation are indistinguishable (up to a success of $O(q^3/M)$), hence every result for random functions carries over to random

permutations with an additional error of $O(q^3/M)$.[19] For results about *invertible* random permutations/injections, see Section 4.4.

4.2 Example: Zero search

To illustrate our technique for random permutations, we will show that it is hard to find a zero-preimage of a random permutation (or injection). More precisely:

Lemma 10 (Hardness of zero search in permutations) *Assume that $N \geq M$ and $q < N/2$.*

Let A be an oracle algorithm that performs q queries. Let h be sampled uniformly at random from $D \hookrightarrow R$ (injections!). Then $p := \Pr[h(x) = 0 : x \leftarrow A^h()] \leq \frac{16(q+1)^2}{N-2q}$.

Note that by symmetry, the same bound also holds when searching for some $y \in R$ other than 0, in particular also for uniformly random y.

To the best of our knowledge, the best known bound for this is $O(q^3/N)$ (as a simple consequence of the indistinguishability of random functions and permutations [Zha15a]; in the $M = N$ case only). That is, it was known that $\Omega(\sqrt[3]{N})$ queries are needed to find a 0-preimage, we have improved this to $\Omega(\sqrt{N})$ queries (which is tight due to Grover's algorithm [Gro96]). (And we further improve on our bound in Section 4.3.1 by taking into account parallel queries.)

The proof is in large parts similar to that in the random function case (Section 3.3). We highlight mostly the differences.

Preparations. The preparation step is almost identical to the random function case. The only difference is that we replace the oracle access to the random injection h not by the standard oracle (which implements U_{query} with initial state $|\varnothing\rangle^{\mathfrak{f}}$) but by the "standard permutation oracle": This oracle has a register H with space $\mathbb{C}^{D\hookrightarrow R}$ and is initialized with the superposition $\sum_{h:D\hookrightarrow R} \frac{1}{\sqrt{|D\hookrightarrow R|}}|h\rangle$. Note that that superposition is $|\varnothing\rangle^{\mathfrak{p}}$, one of the compressed permutation states.

The circuit resulting from the preparations is still as depicted in Figure 2, only the initial state on H is $|f\rangle^{\mathfrak{p}}$ now.

Stating the invariant. Also the definition of the invariants is essentially the same and follows the same intuition. But instead of expressing them in terms of states $|f\rangle^{\mathfrak{f}}$, we express them in terms of states $|f\rangle^{\mathfrak{p}}$. Thus:

$$I_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \operatorname{span}\big\{|f\rangle^{\mathfrak{p}} : 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \leq j\big\}_f \qquad \text{for } j \leq q.$$

and

$$I_{final} := \mathcal{H}_E \otimes \operatorname{span}\big\{|x\rangle_X|f(x)\rangle_Y|f\rangle^{\mathfrak{p}} : x \in \operatorname{dom} f,\ 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \leq j\big\}_{xf}.$$

[19] However, [Zha15a] does not show the indistinguishability of random functions and random injections when domain and range have different sizes.

(Note that the use of $|\cdot\rangle^{\mathfrak{p}}$ also implicitly means that f is quantified only over $D \hookrightarrow R$, not over $D \rightarrowtail R$.)

And we also state the invariant I_q^* that is supposed to hold right before the last query (after initializing Y with $|0\rangle$):

$$I_q^* := \mathcal{H}_{E'} \otimes \mathcal{H}_X \otimes \operatorname{span}\bigl\{|0\rangle_Y |f\rangle^{\mathfrak{p}} : 0 \notin \operatorname{im} f,\ |\operatorname{dom} f| \le j\bigr\}_f$$

Proving that the invariant is maintained. This follows the lines of the random function case, but the bounds are different.

In the random function case, we showed that:

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \qquad \Longrightarrow \qquad \psi_j = (I_E \otimes U_{query})\psi'_{j-1} \overset{\varepsilon+\delta_j}{\approx} I_{j-1} \qquad (j = 1, \dots, q) \tag{13}$$

$$\psi_q^* \overset{\varepsilon}{\approx} I_q^* \qquad \Longrightarrow \qquad \psi_{final} = (I_E \otimes U_{query})\psi_q^* \overset{\varepsilon+\delta_{q+1}}{\approx} I_{final} \tag{14}$$

for $\delta_j := \sqrt{16/N}$.

Here we will show the same (for other choices of δ_j). This then implies, as before:

$$\psi_{final} \overset{\sum_{j=1}^{q+1} \delta_j}{\approx} I_{final}. \tag{15}$$

So all that remains to do is to find out for which δ_j equations (13), (14) hold.

As before, we have theorems to show invariant preservation. For (13), we use the analogue to Theorem 3 in the permutation case:

Theorem 11 (Random permutation query, simple) *Assume $|D| \le |R|$. Fix an integer $\ell \ge 0$ (compressed function size). Fix $c \ge 0$. Let E, X, Y, H be subsystems with Hilbert spaces $\mathcal{H}_E$, $\mathcal{H}_X := \mathbb{C}^D$, $\mathcal{H}_Y := \mathbb{C}^{\tilde{R}}$, $\mathcal{H}_H := \mathbb{C}^{D \hookrightarrow R}$. Let $N := |R|$. Assume $\ell \le N - 16$. Let $\{\eta_e\}_e$ be an orthogonal basis of $\mathcal{H}_E$. For each $x \in D$ and e, fix $A_{x,e} \subseteq B_{x,e} \subseteq (D \hookrightarrow R)$ (pre-/postcondition). Assume that for all x, all e, and all compatible $f, g \in A_{x,e}$ with $f \cup g$ injective and $|\operatorname{dom} f|, |\operatorname{dom} g| \le \ell$ and $x \notin \operatorname{dom} f, \operatorname{dom} g$:*

$$\frac{\Bigl|\{z \in R : z \notin \operatorname{im} f, \operatorname{im} g,\ f(x := z),\ g(x := z) \notin B_{x,e}\}\Bigr|}{N - \bigl|\operatorname{dom} f \cup \operatorname{dom} g\bigr|} \le c. \tag{16}$$

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_H$. If

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\bigl\{\eta_e |x\rangle |y\rangle |f\rangle^{\mathfrak{p}} : f \in A_{x,e},\ |\operatorname{dom} f| \le \ell\bigr\}_{exyf}$$

then

$$(I_E \otimes U_{query})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\bigl\{\eta_e |x\rangle |y\rangle |f\rangle^{\mathfrak{p}} : f \in B_{x,e},\ |\operatorname{dom} f| \le \ell + 1\bigr\}_{exyf}.$$

The main differences to Theorem 3 are that the theorem is stated in terms of $|f\rangle^{\mathfrak{p}}$, not $|f\rangle^{\mathfrak{f}}$, and that the proof obligation (16) is different (this will lead to a different c).

We thus again instantiate this theorem with $A_{x,e}, B_{x,e} := \{f : 0 \notin \operatorname{im} f\}$, $\ell := j-1$, and $\psi := \psi'_{j-1}$, and an arbitrary orthonormal basis η_e of $\mathcal{H}_E$.

We proceed to determine c. For any $f, g \in A$, we have that $0 \notin \operatorname{im} f, \operatorname{im} g$. The set in the numerator in (16) contains only values z such that $f(x := z)$ or $g(x := z)$ has 0 in its image. Obviously, the only z that can potentially satisfy this is $z = 0$. Thus the numerator is ≤ 1. We also have $|\operatorname{dom} f|, |\operatorname{dom} f| \leq \ell = j-1$, thus the denominator is $\geq N - 2(j-1)$. Thus (6) is satisfied with $c := 1/(N-2(j-1))$.

Hence the theorem implies that (13) holds with $\delta_j := \sqrt{16/(N-2(j-1))}$.

And the analogue for Theorem 4 the following theorem:

Theorem 12 (Non-oblivious random permutation query, simple case) *In the situation of Theorem 11, if*

$$\psi \overset{\varepsilon}{\approx} \operatorname{span}\bigl\{\eta_e|x\rangle|0\rangle|f\rangle^{\mathfrak{p}} : f \in A_{x,e}, |\operatorname{dom} f| \leq \ell\bigr\}_{exf}$$

then

$$(I_E \otimes U_{query})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \operatorname{span}\bigl\{\eta_e|x\rangle\bigl|f(x)\bigr\rangle|f\rangle^{\mathfrak{p}} : x \in \operatorname{dom} f,\ f \in B_{x,e},\ |\operatorname{dom} f| \leq \ell+1\bigr\}_{exf}.$$

This one is instantiated with $A_{x,e}, B_{x,e} := \{f : 0 \notin \operatorname{im} f\}$, $\ell := q$, and $\psi := \psi^*_q$. We get $c := 1/(N-2q)$ with the same analysis as before, and it follows that (14) holds with $\delta_{q+1} := \sqrt{16/(N-2q)}$.

Thus, we have shown for which δ_j (15) holds. It is elementary to see that

$$\sum_{j=1}^{q+1} \delta_j \leq \sqrt{\frac{16(q+1)^2}{N-2q}}.$$

Hence from (15) we get:

$$\psi_{final} \overset{\sqrt{\frac{16(q+1)^2}{N-2q}}}{\approx} I_{final}. \tag{17}$$

Hence we know how far from the invariant the final state is.

Concluding. The analysis of the last step is again identical to the random function case, except that the bound $(q+1)\sqrt{16/N}$ is replaced by the new bound $\sqrt{\frac{16(q+1)^2}{N-2q}}$. The probability of measuring $y = 0$ is the square of that bound, Lemma 10 follows.[20]

[20] One note: We ignored the assumption $\ell \leq N - 16$ in Theorem 11 during the proof. However, this assumption can only be violated when $q > N-16$. Under the assumptions of Lemma 10, when $q > N-16$, the bound $\frac{16(q+1)^2}{N-2q}$ is ≥ 1, hence the lemma is trivial in that case.

4.3 Parallel queries

(This section can be skipped by a reader who is not interested in parallel queries.)

Same as in the case of random functions (see Section 3.5), parallel queries to a compressed permutation oracle can be handled simply by replacing the query operator U_{query} by $U_{query,k}$. As in the case of random functions, we provide analogues to Theorems 11 and 12 that describe what happens to invariants in case of k-parallel queries. Also like in the case of random functions, the new theorems additionally support invariants of a more general shape. See Section 3.5 for a description of both features. The theorems are:

Theorem 13 (Random permutation query) *Assume $|D| \leq |R|$. Fix integers $k \geq 0$ (query parallelism), $\ell \geq 0$ (compressed function size). Fix $c \geq 0$. Let E, X, Y, H be subsystems with Hilbert spaces $\mathcal{H}_E$, $\mathbb{C}^{D^k}$, $\mathbb{C}^{\tilde{R}^k}$, $\mathcal{H} := \mathbb{C}^{D \hookrightarrow R}$. Let $N := |R|$. Assume $\ell \leq N - 16$. Let $\{\eta_e\}_e$ be an orthogonal basis of $\mathcal{H}_E$. For each $\underline{x} \in D^k$ and e, fix $A_{\underline{x},e}, B_{\underline{x},e} \subseteq (D \hookrightarrow R)$ (pre-/postcondition). Assume that for all $\underline{x}$, all e, and all compatible $f, g \in A_{\underline{x},e}$ with $f \cup g$ injective and $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \leq \ell$:*

$$\frac{\Big|\{\underline{z} \in R^k : (\underline{x} \mapsto \underline{z}) \diamond f, g \text{ and } f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}} \text{ injective and } f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}} \notin B_{\underline{x},e}\}\Big|}{\Big(N - \big|\mathrm{dom}\, f \cup \mathrm{dom}\, g\big|\Big)_{|\underline{x} \setminus \mathrm{dom}\, f \setminus \mathrm{dom}\, g|}} \leq c \quad [21] \tag{18}$$

where $f_{\underline{x},\underline{z}} := f \cup (\underline{x} \mapsto \underline{z})$ and $g_{\underline{x},\underline{z}} := g \cup (\underline{x} \mapsto \underline{z})$.

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathcal{H}_H$. If

$$\psi \overset{\varepsilon}{\approx} \mathrm{span}\big\{\eta_e|\underline{x}\rangle|\underline{y}\rangle|f\rangle^{\mathfrak{p}} : f \in A_{\underline{x},e},\ |\mathrm{dom}\, f| \leq \ell\big\}_{e\underline{x}\underline{y}f} =: \mathbf{A}$$

then

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \mathrm{span}\big\{\eta_e|\underline{x}\rangle|\underline{y}\rangle|f\rangle^{\mathfrak{p}} : f \in B_{\underline{x},e},\ |\mathrm{dom}\, f| \leq \ell + k\big\}_{e\underline{x}\underline{y}f} =: \mathbf{B}$$

Theorem 14 (Non-oblivious random permutation query) *In the situation of Theorem 13, if*

$$\psi \overset{\varepsilon}{\approx} \mathrm{span}\big\{\eta_e|\underline{x}\rangle|\underline{0}\rangle|f\rangle^{\mathfrak{p}} : f \in A_{\underline{x},e},\ |\mathrm{dom}\, f| \leq \ell\big\}_{e\underline{x}f}$$

then

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon+4\sqrt{c}}{\approx} \mathrm{span}\big\{\eta_e|\underline{x}\rangle|f(\underline{x})\rangle|f\rangle^{\mathfrak{p}} : \underline{x} \subseteq \mathrm{dom}\, f,\ f \in B_{\underline{x},e},\ |\mathrm{dom}\, f| \leq \ell + k\big\}_{e\underline{x}f}.$$

The main difference to the theorems from Section 3.5 (besides being stated for compressed permutation states instead of compressed function states, of course) is the combinatorial condition (18) which is somewhat more complex. (For example, the denominator involves a falling factorial $(a)_b$, see the preliminaries.)

[21] See also footnote 18 for a note on tuples $\underline{x}$ with repeated entries. And the cardinality $|\underline{x} \setminus \mathrm{dom}\, f \setminus \mathrm{dom}\, g|$ in the denominator is a cardinality of sets, not a length of tuples.

4.3.1 Zero search with parallel queries

We shortly revisit the example of zero search in permutations (Section 4.2) and list the differences in the analysis in the case of parallel queries. We show:

Lemma 15 (Hardness of zero search (permutations, parallel queries)) *Let A be an oracle algorithm that performs q queries. Fix an integer $k \geq 0$ (query parallelism). Assume $qk < N/2$. Let h be sampled uniformly at random from $D \hookrightarrow R$. Then $p := \Pr[h(x) = 0 : x \leftarrow A^{h^k}()] \leq \frac{16(q\sqrt{k}+1)^2}{N-2qk}$. (As opposed to $\leq \frac{16(qk+1)^2}{N-2qk}$ as one would get as an immediate corollary from Lemma 10.)*

(Note that the adversary A gets h^k, not h. I.e., each of the q queries is a k-parallel query.)

Note that by symmetry, the same bound also holds when searching for some $y \in R$ other than 0, in particular also for uniformly random y.

The differences to the proof of Lemma 10 are:

- In all invariants, we replace the condition $|\operatorname{dom} f| \leq j$ by $|\operatorname{dom} f| \leq jk$. (And analogously $|\operatorname{dom} f| \leq q-1$ by $|\operatorname{dom} f| \leq (q-1)k$.) This is because in each query, the length increases by up to k.
- We use Theorem 13 instead of Theorem 11. (I.e., we use the theorem with parallel query support.) For the last query (which queries only one value), we can still use the non-oblivious Theorem 12. (Or alternatively, we could use Theorem 14 with $k := 1$.)
- When applying Theorem 13, we need to bound (18) (instead of (16)) with suitable c. Consider compatible $f, g \in A_{\underline{x},e} = \{f : 0 \notin \operatorname{dom} f\}$ such that $f \cup g$ is injective and $|\operatorname{dom} f|, |\operatorname{dom} g| \leq (j-1)k$. If $\underline{z} \in R^k$ is in the set in the numerator, then:
  - It coincides with $(f \cup g)(\underline{x})$ wherever the latter is defined.
  - Wherever $\underline{x}$ has repetitions, $\underline{z}$ has repetitions, too. (Otherwise $\underline{x} \mapsto \underline{z}$ would not be a function.)
  - The remaining $t := |\underline{x} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|$ entries of $\underline{z}$ are distinct from each other and from $\operatorname{im} f \cup \operatorname{im} g$. (Otherwise $f_{\underline{x},\underline{z}}$ or $g_{\underline{x},\underline{z}}$ would not be injective.)
  - Within the remaining t entries, $\underline{z}$ has a 0. (Otherwise $f_{\underline{x},\underline{z}}$ and $g_{\underline{x},\underline{z}}$ would be in $B_{\underline{x},e} = \{f : 0 \notin \operatorname{dom} f\}$.)
  - In other words, the remaining t entries form an injection from a t element set to $U := (R \setminus \operatorname{im} f \setminus \operatorname{im} g) \ni 0$ that has 0 in its image. Let $u := |U|$. There are $t \cdot (u-1)_{t-1}$ such injections.

  Thus the numerator is bounded by $t \cdot (u-1)_{t-1}$.
  Note that $u = |R \setminus \operatorname{im} f \setminus \operatorname{im} g| = N - |\operatorname{im} f \cup \operatorname{im} g| = N - |\operatorname{dom} f \cup \operatorname{dom} g|$ where the last inequality holds because $f \cup g$ is an injection. Hence the denominator in (18) is $(u)_t$.
  Thus the lhs of (18) is bounded by

$$\frac{t \cdot (u-1)_{t-1}}{(u)_t} = t\,\frac{(u-1)!}{(u-t)!}\,\frac{(u-t)!}{u!} = \frac{t}{u} \leq \frac{k}{N-2(j-1)k}.$$

Here $t \le k$ follows from the definition of t and $|\underline{x}| = k$, and $u \ge N-2(j-1)k$ follows from the definition of u and $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \le (j-1)k$. Thus (12) is satisfied with $c := \frac{k}{N-2(j-1)k}$. Thus the bound $4\sqrt{c} = \sqrt{16/(N-2(j-1))}$ must be replaced by $4\sqrt{c} = \sqrt{16k/(N-2(j-1)k)}$.

- In the application of Theorem 4, the analysis is as in the non-parallel case. (Except that we instantiate the theorem with $\ell := qk$ instead of $\ell := q$.) We get $c = \sqrt{16/(N-2qk)}$ in that case.
- In total, we thus get $\psi_{final} \overset{\varepsilon}{\approx} I_{final}$ with $\varepsilon := \sum_{j=1}^{q} \sqrt{16k/(N-2(j-1)k)} + \sqrt{16/(N-2qk)}$. Hence $p \le \varepsilon^2 \le \frac{16(q\sqrt{k}+1)^2}{N-2qk}$.

4.4 Inverse queries

We now study permutations with inverse. That is, we consider the case where the adversary gets superposition query access to a permutation h, as well as to its inverse h^{-1}. (We only consider the case $N = M$ because then the inverse is always defined.)

As it turns out, with the setup we have so far, adding inverse queries is very simple. The main observation is that performing an inverse query is equivalent to replacing h by its inverse, performing a regular query, and then replacing h by its inverse again. This gives rise to the following definition:

Definition 16 (Invertible standard oracle) *Assume $N = M$. The* invertible standard oracle *has an internal state register H with initial state $|\varnothing\rangle^{\mathfrak{p}} \in \mathbb{C}^{D \hookrightarrow R}$ as well as query registers X, Y. The adversary has access to X, Y but not to H and can perform two different queries:*

- Regular query: *This applies U_{query} (or $U_{query,k}$ in the parallel-query variant) to X, Y, H.*
- Inversion query: *This applies the unitary* Invert *to H which is defined by* Invert : $|h\rangle \mapsto |h^{-1}\rangle$.

An adversary with access to uniformly random h, h^{-1} can thus be replaced by an adversary with access to the invertible standard oracle. (And if the original adversary performs q queries, the new adversary will perform q regular queries and $\le q+1$ inversion queries.

When analyzing an adversary that queries the invertible standard oracle, we can again use the approach of maintaining invariants expressed in terms of $|f\rangle^{\mathfrak{p}}$ states (since the initial state of the invertible standard oracle is still $|\varnothing\rangle^{\mathfrak{p}}$). We already have theorems for handling the regular queries. The only new thing we need is to know is how invariants behave under inversion queries. This is surprisingly simple:

Lemma 17 (Inversion queries)

(i) For $f \in D \overset{\perp}{\hookrightarrow} R$, $\mathrm{Invert}|f\rangle^{\mathfrak{p}} = |f^{-1}\rangle^{\mathfrak{p}}$.

(ii) For a space $\mathcal{H}$, a set $A \subseteq (D \overset{\perp}{\hookrightarrow} R)$, and an integer $\ell \ge 0$: If $\psi \overset{\varepsilon}{\approx} \mathcal{H} \otimes \mathrm{span}\{|f\rangle^{\mathfrak{p}} : f \in A,\ |\mathrm{dom}\, f| \le \ell\}_f =: I$, then $(I_{\mathcal{H}} \otimes \mathrm{Invert})\psi \overset{\varepsilon}{\approx} \mathcal{H} \otimes \mathrm{span}\{|f\rangle^{\mathfrak{p}} : f^{-1} \in A,\ |\mathrm{dom}\, f| \le \ell\}_f =: I^-$.

Lemma 17 (ii) means that if the current joint state (of the adversary and the oracle) satisfies some invariant expressed in terms of $|f\rangle^{\mathfrak{p}}$-states, then we can compute the next invariant by simply inverting all the f's. And the error ε does not increase. (I.e., inversion queries are for free.) In particular, if A is invariant under inversion, then inversion queries do not change the invariant at all.

(For readability, we have not stated Lemma 17 (ii) in the most general form possible. E.g., we do not allow the invariant to depend on the $\mathcal{H}$-register. In more complex cases where Lemma 17 (ii) is not applicable, it should be quite simple to just use Lemma 17 (i) directly.)

This is all we need to handle random permutation oracles with inverses. The next section will illustrate this. But first we give the simple proof of Lemma 17:

Proof. We first show (i). Let $F := \{h \in D \hookrightarrow R : h \diamond f\}$. Let $F^- := \{h^{-1} : h \in F\}$. Then $|F^-| = |F|$ since inversion is an involution. Since $h \diamond f$ iff $h^{-1} \diamond f^{-1}$ (at least for injective total h, injective partial f), we have that $F^- = \{h \in D \hookrightarrow R : h \diamond f^{-1}\}$. Thus

$$\begin{aligned}\mathrm{Invert}|f\rangle^{\mathfrak{p}} = \mathrm{Invert} \sum_{h\in F} \tfrac{1}{\sqrt{|F|}}|h\rangle = \sum_{h\in F} \tfrac{1}{\sqrt{|F|}}|h^{-1}\rangle \\ \overset{(*)}{=} \sum_{h\in F^-} \tfrac{1}{\sqrt{|F|}}|h\rangle = \sum_{h\in F^-} \tfrac{1}{\sqrt{|F^-|}}|h\rangle \overset{(**)}{=} |f^{-1}\rangle^{\mathfrak{p}}.\end{aligned}$$

Here $(*)$ is by index substitution $h \mapsto h^{-1}$ and definition of F^-. And $(**)$ follows from $F^- = \{h \in D \hookrightarrow R : h \diamond f^{-1}\}$ and the definition of $|\cdot\rangle^{\mathfrak{p}}$.

We now show (ii). We have

$$\begin{aligned}(I_{\mathcal{H}} \otimes \mathrm{Invert})\, I &= \mathcal{H} \otimes \mathrm{span}\{\mathrm{Invert}|f\rangle^{\mathfrak{p}} : f \in A,\ |\mathrm{dom}\, f| \le \ell\}_f \\ &\overset{(i)}{=} \mathcal{H} \otimes \mathrm{span}\{|f^{-1}\rangle^{\mathfrak{p}} : f \in A,\ |\mathrm{dom}\, f| \le \ell\}_f \\ &\overset{(*)}{=} \mathcal{H} \otimes \mathrm{span}\{|f\rangle^{\mathfrak{p}} : f^{-1} \in A,\ |\mathrm{dom}\, f^{-1}| \le \ell\}_f \overset{(**)}{=} I^-. \qquad (19)\end{aligned}$$

Here $(*)$ is by index substitution $f \mapsto f^{-1}$, and $(**)$ follows because $|\mathrm{dom}\, f^{-1}| = |\mathrm{dom}\, f|$ for injective f.

Finally, if $\psi \overset{\varepsilon}{\approx} I$, then $(I_{\mathcal{H}} \otimes \mathrm{Invert})\psi \overset{\varepsilon}{\approx} (I_{\mathcal{H}} \otimes \mathrm{Invert})\, I$ since $I_{\mathcal{H}} \otimes \mathrm{Invert}$ is unitary. With (19), (ii) follows. □

4.5 Example: Two-sided zero search

We give a simple example for reasoning about invertible random permutations. The previously used examples (zero search and preimage search) are not suitable because it is trivially possibly to solve those problems with a single query to the inverse function.

Instead, we consider the following problem: Given a permutation h (and its inverse), find values $x, y \in \{0,1\}^r$ such that $h(x\|0^c) = y\|0^c$. More precisely, we show the following:

Lemma 18 (Hardness of two-sided zero search) *Assume that* $D = R = \{0,1\}^{r+c}$ *and that* $q < 2^{c+r}/2$.

Let A *be an oracle algorithm that performs* q *queries. Let* h *be sampled uniformly at random from* $D \hookrightarrow D$ *(a random permutation). Then* $p := \Pr[h(x\|0^c) = y\|0^c : (x,y) \leftarrow A^{h,h^{-1}}()] \leq \frac{16(q+1)^2}{2^c - 2q/2^r}$.

That is, finding such x, y takes $\Omega(\sqrt{2^c})$ queries. The bound is tight by Grover's algorithm [Gro96].

Relevance of this problem. While this is, to the best of our knowledge, not a standard problem, we believe it is an interesting problem for several reasons. First, it is the simplest nontrivial problem for invertible permutations that we are aware of. Second, this problem resisted all our prior attempts (in collaboration with Alexander Belov) of solving it with existing techniques for quantum query complexity such as the adversary method [Amb02] or the polynomial method [Bea+01]. While this is admittedly a very subjective measure of difficulty, the fact that this problem can be solved quite easily with our compressed oracle technique gives evidence for the power of our technique. And third, this problem is a special case of the security of the collision-resistance of the sponge construction: if h is the round function, from x, y with $h(x\|0^c) = y\|0^c$, it is very easy to construct a collision. So solving this problem, while not sufficient, is necessary for analyzing the security of the sponge construction.

Preparations. We will call (x,y) *bad* iff both x, y end in 0^c. We say f *has no bad pair* iff there is no $x \in \operatorname{dom} f$ such that $(x, f(x))$ is a bad pair.

As in the examples before, we assume that A is unitary (represented by the unitary U_A). It uses two query registers X and Y with state spaces $\mathcal{H}_X = \mathcal{H}_Y = \mathbb{C}^{2^{r+c}}$ ($r+c$ qubits), and a register E for its state. We let A output $x \in \{0,1\}^r$ in register X (padded with c zeroes). We do not require A to output y. (It can be recomputed from $h(x\|0^c)$ anyway.) In the end, we initialize Y with $|0^{r+c}\rangle$ and perform an additional oracle query. Then we measure X and Y, resulting in values $\hat{x}, \hat{y}$. If $(\hat{x}, \hat{y})$ is bad, we say the adversary wins. Let $\hat{p}$ be the probability that the adversary wins. Then $\hat{p} \geq p$, so it will be sufficient to bound $\hat{p}$. (We do not have $\hat{p} = p$ because we have removed the requirement that the adversary outputs the correct y.) Furthermore, we replace oracle access to h and h^{-1} by the invertible standard oracle (Definition 16). Altogether, we get the circuit depicted in Figure 4. Between which queries there are Invert invocations depends on the adversary, so not all Invert invocations drawn in the figure are actually contained in the circuit.

Stating the invariant. As before, we have to state an invariant that is supposed to hold before and after every oracle query. Since the goal of the adversary is to find an x such that $(x, h(x))$ is bad, we state as an invariant that the adversary has not queried such an x so far:

$$I_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \operatorname{span}\big\{|f\rangle^{\mathfrak{p}} : |\operatorname{dom} f| \leq j,\ f \text{ has no bad pairs}\big\}_f$$

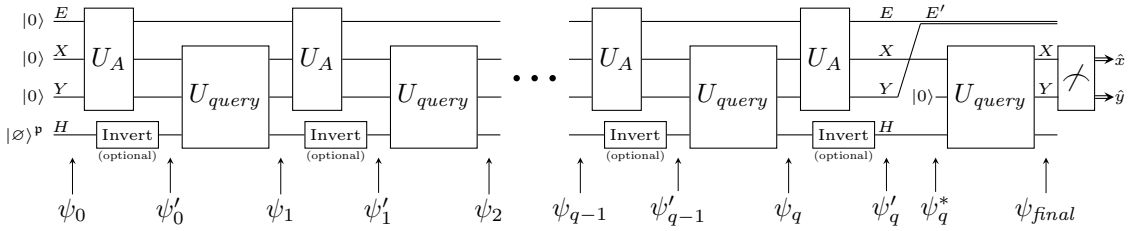


Figure 4: Circuit describing the two-sided zero search game. We drew an Invert before/after every query. Only a subset of these will actually be contained in the circuit. $\hat{p}$ is the probability that $(\hat{x}, \hat{y})$ is bad.

And after the last query (which we added to the end of the circuit), we claim that the same holds and additionally, X and Y contain the input/output of an oracle query:

$$I_{\mathit{final}} := \mathcal{H}_{E'} \otimes \operatorname{span}\big\{|x\rangle_X |f(x)\rangle_Y |f\rangle^{\mathfrak{p}} : x \in \operatorname{dom} f,\ |\operatorname{dom} f| \le j,\ f \text{ has no bad pairs}\big\}_{xf}$$

And as before, we need a variant of I_q that additionally states that we initialized Y with $|0\rangle$:

$$I_q^* := \mathcal{H}_{E'} \otimes \mathcal{H}_X \otimes \operatorname{span}\big\{|0\rangle_Y |f\rangle^{\mathfrak{p}} : |\operatorname{dom} f| \le q,\ f \text{ has no bad pairs}\big\}_{f} \tag{20}$$

Proving that the invariant is maintained. We will prove five facts:

$$\psi_0 \overset{0}{\approx} I_0 \tag{21}$$

$$\psi_j \overset{\varepsilon}{\approx} I_j \implies \psi'_j \overset{\varepsilon}{\approx} I_j \qquad \text{for } j = 0, \dots, q \tag{22}$$

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \implies \psi_j \overset{\varepsilon+\sqrt{\frac{16\cdot 2^r}{N-2(j-1)}}}{\approx} I_j \qquad \text{for } j = 1, \dots, q \tag{23}$$

$$\psi'_q \overset{\varepsilon}{\approx} I_q \implies \psi^*_q \overset{\varepsilon}{\approx} I^*_q \tag{24}$$

$$\psi^*_q \overset{\varepsilon}{\approx} I^*_q \implies \psi_{\mathit{final}} \overset{\varepsilon+\sqrt{\frac{16\cdot 2^r}{N-2q}}}{\approx} I_{\mathit{final}} \tag{25}$$

By induction, and using that $\sum_{i=1}^{q+1} \sqrt{\frac{16\cdot 2^r}{N-2q}} \le \sqrt{\frac{16(q+1)^2}{2^c - 2q/2^r}}$ (recall $N = 2^{r+c}$), this then implies

$$\psi_{\mathit{final}} \overset{\varepsilon_{\mathit{final}}}{\approx} I_{\mathit{final}} \qquad \text{with} \qquad \varepsilon_{\mathit{final}} := \sqrt{\frac{16(q+1)^2}{2^c - 2q/2^r}}. \tag{26}$$

Fact (21) follows immediately from the definitions of ψ_0 and I_j.

To show (22), first note that $\psi'_j \overset{\varepsilon}{\approx} I_j$ immediately implies $(U_A \otimes I_H)\psi'_j \overset{\varepsilon}{\approx} I_j$ because the invariant I_j refers only to register H while U_A operates only on registers other than H.

In the case that there is no inversion query Invert before the $(j+1)$-st oracle query, $\psi_j = (U_A \otimes I_H)\psi'_j$ and we have shown (22).

If there is an inversion query, $\psi_j = (I_{EXY} \otimes \text{Invert})(U_A \otimes I_H)\psi'_j$. Thus we need to show that Invert preserves the invariant.

Note that the condition in the set comprehension in the definition of I_j is invariant under inversion. (That is, f satisfies it iff f^{-1} satisfies it.) Thus, by Lemma 17 (ii) (with $\mathcal{H} := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y$, $\ell := j$, $A := \{f : f \text{ has no bad pairs}\}$, $\psi := (U_A \otimes I_H)\psi'_j$) we have that $(U_A \otimes I_H)\psi'_j \overset{\varepsilon}{\approx} I_j$ implies $\psi_j = (I_{EXY} \otimes \text{Invert})(U_A \otimes I_H)\psi'_j \overset{\varepsilon}{\approx} I_j$

This shows (22).

We now show (23). To do so, we apply Theorem 11 that we already encountered when showing the hardness of zero search in permutations. We instantiate the theorem with $\ell := j-1$ and $A_{x,e}, B_{x,e} := \{f : f \text{ has no bad pairs}\}$ and $c := \frac{2^r}{N-2(j-1)}$ and $\psi := \psi'_{j-1}$. The premise (16) in Theorem 11 then becomes:

> For all compatible f, g that have no bad pairs with $|\operatorname{dom} f|, |\operatorname{dom} g| \leq j-1$ and all $x \notin \operatorname{dom} f, \operatorname{dom} g$:
>
> $$\frac{\left|\{z \in R : z \notin \operatorname{im} f, \operatorname{im} g,\ f(x := z),\ g(x := z) \text{ have bad pairs}\}\right|}{N - \left|\operatorname{dom} f \cup \operatorname{dom} g\right|} \leq c.$$

The denominator is clearly lower bounded by $N - 2(j-1)$. To bound the numerator, consider that if f has no bad pairs, then $f(x := z)$ only has a bad pair if (x, z) is a bad pair. This can only happen if z ends in 0^c. Since $z \in \{0,1\}^{r+c}$, there are 2^r such values z. Hence the numerator is upper bounded by 2^r, and the fraction is upper bounded by $c = \frac{2^r}{N-2(j-1)}$.

Thus Theorem 11 is applicable and shows that $\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1}$ implies $\psi_j = (I_E \otimes U_{query})\psi'_{j-1} \overset{\varepsilon+4\sqrt{c}}{\approx} I_j$. Since $c = \frac{2^r}{N-2(j-1)}$, this shows (23).

We proceed to show (24). Let V denote the operation that joins E, Y in a register E' and adds a fresh $|0\rangle$-initialized register Y. (See Figure 4.) Then $\psi^*_q = V\psi'_q$. And since V is an isometry, $\psi^*_q \overset{\varepsilon}{\approx} VI_q = I^*_q$. This shows (24).

To show (25), we use the non-oblivious query theorem (Theorem 12), otherwise the analysis is completely analogous to that used to show (23).

Thus (26) holds.

Concluding. The last step of the circuit in Figure 4 is a measurement of X, Y in the computational basis. I_{final} is the span of some states of the form $|x\rangle|f(x)\rangle|f\rangle^{\mathfrak{p}}$ where f has no bad pairs. In particular, $(x, f(x))$ is not a bad pair. Thus measuring X, Y on a state in I_{final} yields $\hat{x}, \hat{y}$ that is a bad pair with probability 0. We have $\psi_{final} \overset{\varepsilon_{final}}{\approx} I_{final}$ by (26). Thus when measuring X, Y on state ψ_{final}, we get a bad pair with probability $\hat{p} \leq \varepsilon^2_{final}$. As stated in the beginning of this analysis, $p \leq \hat{p}$ (recall that p is the winning probability of A in Lemma 18), so we have proven Lemma 18.[22]

[22] One note: We ignored the assumption $\ell \leq N - 16$ in Theorem 11 during the proof. However, this assumption can only be violated when $q > N - 16$. Under the assumptions of Lemma 18, when $q > N - 16$,

4.5.1 Two-sided zero search with parallel queries

We shortly rewrite the example of two-sided zero search and list the differences in the analysis in the case of parallel queries. We show:

Lemma 19 (Hardness of two-sided zero search) *Assume that $D = R = \{0,1\}^{r+c}$ and that $qk < 2^{c+r}/2$. Fix an integer $k \geq 0$ (query parallelism).*

Let A be an oracle algorithm that performs q queries. Let h be sampled uniformly at random from $D \hookrightarrow D$ (a random permutation). Then $p := \Pr[h(x\|0^c) = y\|0^c : (x,y) \leftarrow A^{h^k,(h^{-1})^k}()] \leq \frac{16(q\sqrt{k}+1)^2}{2^c - 2qk/2^r}$. (As opposed to $\leq \frac{16(qk+1)^2}{2^c-2qk/2^r}$ as one would get as an immediate corollary from Lemma 18.)

(Note that the adversary A gets h^k and $(h^{-1})^k$, not h and h^{-1}. I.e., each of the q queries is a k-parallel query.)

The differences to the proof of Lemma 18 are:

- In all invariants, we replace the condition $|\operatorname{dom} f| \leq j$ by $|\operatorname{dom} f| \leq jk$. (And analogously $|\operatorname{dom} f| \leq q$ by $|\operatorname{dom} f| \leq qk$.) This is because in each query, the length increases by up to k.
- We use Theorem 13 instead of Theorem 11. (I.e., we use the theorem with parallel query support.) For the last query (which queries only one value), we can still use the non-oblivious Theorem 12. (Or alternatively, we could use Theorem 14 with $k := 1$.)
- When applying Theorem 13, we need to bound (18) (instead of (16)) with suitable c. Consider compatible $f, g \in A_{\underline{x},e} = \{f : f \text{ has no bad pairs}\}$ such that $f \cup g$ is injective and $|\operatorname{dom} f|, |\operatorname{dom} g| \leq (j-1)k$. If $\underline{z} \in R^k$ is in the set in the numerator, then:
  - It coincides with $(f \cup g)(\underline{x})$ wherever the latter is defined.
  - Wherever $\underline{x}$ has repetitions, $\underline{z}$ has repetitions, too. (Otherwise $\underline{x} \mapsto \underline{z}$ would not be a function and hence not compatible with f, g.)
  - The remaining $t := |\underline{x} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|$ entries of $\underline{z}$ are distinct from each other and from $\operatorname{im} f \cup \operatorname{im} g$. (Otherwise $f_{\underline{x},\underline{z}}$ or $g_{\underline{x},\underline{z}}$ would not be injective.)
  - Within the remaining t entries, $\underline{z}$ has an entry of the form $z'\|0^c$. (Otherwise $f_{\underline{x},\underline{z}}$ and $g_{\underline{x},\underline{z}}$ would be in $B_{\underline{x},e} = \{f : f \text{ has no bad pairs}\}$.)
  - In other words, the remaining t entries form an injection from a t element set to $U := (R \setminus \operatorname{im} f \setminus \operatorname{im} g)$ with at least one $z'\|0^c$ in its image. There are t possible positions for that entry, $\leq 2^r$ possibilities for the value of that entry, and $(u-1)_{t-1}$ possibilities for the remaining entries (where $u := |U|$). Thus there are $\leq t2^r \cdot (u-1)_{t-1}$ such injections.

  Thus the numerator is bounded by $t2^r \cdot (u-1)_{t-1}$.
  Note that $u = |R \setminus \operatorname{im} f \setminus \operatorname{im} g| = N - |\operatorname{im} f \cup \operatorname{im} g| = N - |\operatorname{dom} f \cup \operatorname{dom} g|$ where the last inequality holds because $f \cup g$ is an injection. Hence the denominator in (18) is $(u)_t$.

the bound $\frac{16(q+1)^2}{2^c-2q/2^r}$ is ≥ 1, hence the lemma is trivial in that case.

Thus the lhs of (18) is bounded by

$$\frac{2^r t\cdot (u-1)_{t-1}}{(u)_t} = 2^r t\,\frac{(u-1)!}{(u-t)!}\,\frac{(u-t)!}{u!} = \frac{2^r t}{u} \le \frac{2^r k}{N-2(j-1)k}.$$

Here $t \le k$ follows from the definition of t and the fact that $\underline{x}$ has length k. And $u \ge N-2(j-1)k$ follows from the definition of u and $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \le (j-1)k$. Thus (12) is satisfied with $c := \frac{2^r k}{N-2(j-1)k}$. Thus the bound $4\sqrt{c} = \sqrt{16\cdot 2^r/(N-2(j-1))}$ in (23) must be replaced by $4\sqrt{c} = \sqrt{16\cdot 2^r k/(N-2(j-1)k)}$.

- In the application of Theorem 4, the analysis is as in the non-parallel case. (Except that we instantiate the theorem with $\ell := qk$ instead of $\ell := q$.) We thus get $\sqrt{16\cdot 2^r/(N-2qk)}$ instead of $\sqrt{16\cdot 2^r/(N-2q)}$ in (25).
- In total, we thus get $\psi_{\mathit{final}} \overset{\varepsilon_{\mathit{final}}}{\approx} I_{\mathit{final}}$ with $\varepsilon_{\mathit{final}} := \sum_{j=1}^{q}\sqrt{16\cdot 2^r k/(N-2(j-1)k)} + \sqrt{16\cdot 2^r/(N-2qk)}$. Hence $p \le \varepsilon_{\mathit{final}}^2 \le \frac{16(q\sqrt{k}+1)^2}{2^c - 2qk/2^r}$.

4.6 Classical computations

As we have seen in the example in this paper, an analysis using our compressed oracle technique usually investigates a circuits that consists of two parts. First, an execution of the adversary. And then an invocation of a circuit that takes the adversary's output and checks whether it is valid. E.g., in the case of zero search (Sections 3.3 and 4.2), this check was done by evaluating the random oracle on the adversary output register X, and measuring the resulting hash in register Y (and checking whether that hash is 0). Or, in the case of collision finding (Section 3.4), we evaluated the random oracle twice, on registers X_1 and X_2. While this postprocessing is not part of the adversary's attack, we still need to track the evolution of the invariants during these queries. (E.g., that $0 \notin \mathrm{im}\, f$, or that f is injective.) And additionally we need to track the relationship between the different registers in the invariant. (E.g., that X, Y is in state $|x, f(x)\rangle$ for some x.) In the examples described so far, this is quite simple. And even in more complicated cases, things are conceptually very simple: If we show that oracle queries maintain an invariant $f \in A$ (up to some error), and we evaluate a circuit C involving oracle queries, then the invariant $f \in A$ is still satisfied afterwards. But we also have to keep track in the invariant that the output registers contain whatever the circuit computed. That is, we get an invariant roughly like this:

$$\mathrm{span}\Big\{|x\rangle|B(y)\rangle|f\rangle^{\mathfrak{p}} : f \in A\Big\}_{xf}$$

where B is whatever function the circuit C computes. (In our examples so far, B was either just $f(x)$ (see I_{final} in Section 4.2), or $f(x_1), f(x_2)$ (see I_{final} in Lemma 5).) In the examples we saw, proving this invariant was just one or two invocations of the nonoblivious query theorem (Theorem 4). For complex circuits, however, going through

the circuit step by step and tracking the change of the invariant for each gate of the circuit is very tedious and not very rewarding, especially since we essentially already know what the final invariant should be. For example, in the proof of the collision-resistance of the sponge construction (Section 6), our initial proof spent four pages attempt simply on tracking the invariants throughout a circuit that evaluates the sponge construction. To avoid such unnecessary technical proofs, we present a corollary below that allows to show the preservation of an invariant during the evaluation of circuit C that performs a classical computation in superposition.

We state the corollary only for compressed permutations, but an analogous result for compressed functions is easy to derive with almost the same proof.

Classicalish circuits. We call a quantum circuit *classicalish* if it operates on a number of quantum registers $R_1, \dots$ as well as a quantum register H with Hilbert space $\mathcal{H}_H = \mathbb{C}^{D \hookrightarrow R}$, and if each step is one of the following:

- Apply a unitary gate U to registers other than H where U is a permutation matrix.[23] (E.g., CNOT, Toffoli, swap, X.)
- Add a new register R, initialized with a fixed computational basis state $|z\rangle$.
- Add a new register Y, initialized with $|0\rangle$. Let X be any of the existing registers except H. Apply U_{query} to X, Y, H.

For a classicalish circuit C, we define the *function B_h computed by C* (for $h \in D \hookrightarrow R$) to be the unique function such that C maps $|x\rangle|h\rangle_H$ to $|B_h(x)\rangle|h\rangle$. (Strictly speaking, B_h is a function family, indexed by h.)

In our definition, B_h is indexed by a *total* function h. For a *partial* function f, define $B_f(x)$ as follows:

- If for all $h : D \hookrightarrow R$ that are compatible with f, $B_h(x)$ has the same value, then $B_f(x) := B_h(x)$ for some h compatible with f.
- Otherwise $B_f(x) := \bot$.

(In other words, $B_f(x)$ is a partial function defined whenever f contains enough information about h to compute $B_h(x)$.)

Corollary 20 (Queries in classicalish circuits) *Let E, D, H be registers with Hilbert spaces $\mathcal{H}_E$, $\mathcal{H}_D$, $\mathcal{H}_H := \mathbb{C}^{D \hookrightarrow R}$. Fix $c \geq 0$. Let C be a classicalish circuit on D, H, and let B_h be the function computed by C. Let $\underline{R}$ denote the output registers of C (not including H). Assume that C contains at most q invocations of U_{query}. Assume $\ell + q \leq N - 15$. Fix $A \subseteq (D \hookrightarrow R)$.*

Assume that for all compatible $f, g \in A$ with $f \cup g$ injective and $\operatorname{dom} f, \operatorname{dom} g \leq \ell + q - 1$:

$$\frac{\Big|\{z \in R : z \notin \operatorname{im} f, \operatorname{im} g,\ f(x := z),\ g(x := z) \notin A\}\Big|}{N - 2(\ell + q - 1)} \leq c. \tag{27}$$

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_D \otimes \mathcal{H}_H$. If

$$\psi \overset{\varepsilon}{\approx} \mathcal{H}_E \otimes \operatorname{span}\Big\{|d\rangle_D |f\rangle^{\mathfrak{p}} : f \in A,\ |\operatorname{dom} f| \leq \ell\Big\}_{df}$$

[23] We do allow these gates to have different input and output registers. For example, we can have a gate that splits a $2n$-qubit register X in two n-qubit registers Y, Z.

then

$$(I_E\otimes C)\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} \mathcal{H}_E\otimes\mathrm{span}\Big\{\big|B_f(d)\big\rangle_{\underline{R}}|f\rangle^{\mathfrak{p}} : B_f(d)\neq\bot,\ f\in A,\ |\mathrm{dom}\, f|\leq \ell+q\Big\}_{df} =: I.$$

Proof. We prove the theorem by induction over the length of C.

If C has zero gates, then B_h is the identity, and $B_f(x)\neq\bot$ for all f. Hence the first span in the theorem is a subspace of the second. And $C\psi=\psi$. Hence the theorem follows in this case.

Now assume the theorem holds for circuits of length L. Let C be a circuit of length $L+1$. Thus C consists of a circuit C' of length L followed by a single gate G. Let B'_h denote the function computed by C'. We distinguish two cases depending on G:

- *G is a unitary gate on registers other than H and G is a permutation matrix, or G adds a register initialized with $|z\rangle$.* In both cases, $G|x\rangle=|g(x)\rangle$ for some function g. (In the case of adding a register, $g(x)=(x,z)$.) Then $B_h=g\circ B'_h$ for all h. Since C has $\leq q$ invocations of U_{query}, so has C'. Thus we can apply the induction hypothesis and get:

  $$(I_E\otimes C')\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} \mathcal{H}_E\otimes\mathrm{span}\Big\{\big|B'_f(d)\big\rangle|f\rangle^{\mathfrak{p}} : B'_f(d)\neq\bot,\ f\in A,\ |\mathrm{dom}\, f|\leq \ell+q\Big\}_{df} =: I'.$$

  Since G operates on the registers other than H as $|x\rangle\mapsto|g(x)\rangle$,

  $$(I_E\otimes G)I' = \mathcal{H}_E\otimes\mathrm{span}\Big\{\big|g(B'_f(d))\big\rangle|f\rangle^{\mathfrak{p}} : B'_f(d)\neq\bot,\ f\in A,\ |\mathrm{dom}\, f|\leq \ell+q\Big\}_{df}\subseteq I.$$

  Here the $\subseteq$ uses that if $B'_f(x)\neq\bot$, then $B_f(x)=g(B'_f(x))\neq\bot$. Since $(I_E\otimes C')\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} I'$ and $(I_E\otimes G)I'\subseteq I$ and G is an isometry, $(I_E\otimes C)\psi = (I_E\otimes G)(I_E\otimes C')\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} (I_E\otimes G)I'\subseteq I$. Thus $(I_E\otimes C)\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} I$.
- *G initializes a new register Y with $|0\rangle$ and applies U_{query} to X,Y,H.* For notational ease, assume that the registers of the circuit are $\underline{R},X,Y,H$, in that order. C' has output registers $\underline{R},X,H$. Let $B'_h(x)_R$ denote the part of $B'_h(x)$ corresponding to registers $\underline{R}$, and $B'_h(x)_X$ the part corresponding to register X. Then $B'_h(x)=\big(B'_h(x)_R,B'_h(x)_X\big)$ and $B_h(x)=\big(B'_h(x)_R,B'_h(x)_X,h(B'_h(x)_X)\big)$. And C' makes at most $q-1$ invocations of U_{query}. We can then apply the induction hypothesis

(with $q-1$ instead of q) and get:

$$(I_E \otimes C')\psi \overset{\varepsilon+4(q-1)\sqrt{c}}{\approx} \mathcal{H}_E \otimes \operatorname{span}\Big\{ \big|B'_f(d)\big\rangle_{\underline{R}X} |f\rangle^{\mathfrak{p}} : \\ B'_f(d) \neq \bot,\ f \in A,\ |\operatorname{dom} f| \leq \ell + q - 1 \Big\}_{df} =: I'.$$

(Note that if we invoke the induction hypothesis with $q-1$ instead of q, the premise (27) changes. However, (27) with q replaced by $q-1$ is a weaker condition, so the premise is satisfied in the induction hypothesis.)
Let U_0 denote the isometry that maps $\psi \mapsto \psi \otimes |0\rangle_Y$. Then

$$U_0 I' = \operatorname{span}\Big\{ \big|B'_f(d)_R\big\rangle_{\underline{R}} \big|B'_f(d)_X\big\rangle_X |0\rangle_Y |f\rangle^{\mathfrak{p}} : \\ B'_f(d) \neq \bot,\ f \in A,\ |\operatorname{dom} f| \leq \ell + q - 1 \Big\}_{df} =: I_0$$

and thus $U_0(I_E \otimes C')\psi \overset{\varepsilon+4(q-1)\sqrt{c}}{\approx} I_0$. By Theorem 12 (with $\ell := \ell + q - 1$ and $A_{x,(e',r)}, B_{x,(e',r)} := \{f : (x,r) \in \operatorname{im} B'_f, f \in A\}$), this implies

$$U_{query} U_0 (I_E \otimes C')\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} \operatorname{span}\Big\{ \big|B'_f(d)_R\big\rangle_{\underline{R}} \big|B'_f(d)_X\big\rangle_X \big|h(B'_f(d)_X)\big\rangle_Y |f\rangle^{\mathfrak{p}} : \\ B'_f(d) \neq \bot,\ f(B'_f(d)_X) \neq \bot,\ f \in A,\ |\operatorname{dom} f| \leq \ell + q \Big\}_{df} =: I^*.$$

If $B'_f(d) \neq \bot$ and $f(B'_f(d)_X) \neq \bot$, then $B_f(d) = (B'_f(d)_R, B'_f(d)_X, h(B'_f(d)_X)) \neq \bot$. Thus $I^* \subseteq I$. It follows that $(I_E \otimes C)\psi = U_{query} U_0 (I_E \otimes C')\psi \overset{\varepsilon+4q\sqrt{c}}{\approx} I$. □

5 Query theorems

In the present section, we will prove the theorems introduced throughout the text that tell us how invariants evolve under oracle queries.

All those theorems are special cases of two theorems that handle queries and non-oblivious queries to "generalized compressed oracles", a generalization of both the random function and the random permutation case that we have seen before. See Section 5.2 below for more details.

The following table gives an overview of the theorems we prove, together with page numbers for the theorems and their proofs:

		random functions	random permutations	generalized oracles
simple queries	regular	Theorem 3, p. 19 proof p. 60	Theorem 11, p. 29 proof p. 60	n/a
	non-oblivious	Theorem 4, p. 20 proof p. 60	Theorem 12, p. 30 proof p. 61	
parallel queries	regular	Theorem 6, p. 25 proof p. 57	Theorem 13, p. 31 proof p. 58	Theorem 24, p. 47 proof p. 55
	non-oblivious	Theorem 7, p. 26 proof p. 58	Theorem 14, p. 31 proof p. 59	Theorem 25, p. 48 proof p. 56

Before we prove those theorems, we give some simple properties of compressed functions/permutations in the following section that will come in handy later.

5.1 Simple properties

In the following two lemmas, let D, R be finite sets (in Lemma 22 with $|D| \leq |R|$), let $M := |D|$, $N := |R|$. Let f, g denote partial functions (injections in Lemma 22) from D to R. Let r denote a partial function from D to R. Let $\ell_f := |\mathrm{dom}\, f|$, $\ell_g := |\mathrm{dom}\, g|$, $\ell_{f\text{-}g} := |\mathrm{dom}\, f \setminus \mathrm{dom}\, g|$, $\ell_{g\text{-}f} := |\mathrm{dom}\, g \setminus \mathrm{dom}\, f|$, $\ell_r := |\mathrm{dom}\, r|$, $\ell_{r\text{-}f} := |\mathrm{dom}\, r \setminus \mathrm{dom}\, f|$.

Lemma 21 (Compressed functions)

(i) $|\varnothing\rangle^{\mathfrak{f}} = \sum_{h:D\to R} \frac{1}{\sqrt{N^M}} |h\rangle$.

(ii) $\big\| |f\rangle^{\mathfrak{f}} \big\| = 1$.

(iii) $\langle f|\, g\rangle^{\mathfrak{f}} = \begin{cases} \frac{1}{\sqrt{N^{\ell_{f\text{-}g}+\ell_{g\text{-}f}}}} & (f \diamond g) \\ 0 & (f \not\diamond g) \end{cases}$

In particular, for $f \neq g$, $0 \leq \langle f|\, g\rangle^{\mathfrak{f}} \leq 1/\sqrt{N}$.

(iv) $\mathbb{P}_r |f\rangle^{\mathfrak{f}} = \frac{1}{\sqrt{N^{\ell_{r\text{-}f}}}} |f \cup r\rangle^{\mathfrak{f}}$ *if* $f \diamond r$. *And* $\mathbb{P}_r |f\rangle^{\mathfrak{f}} = 0$ *otherwise.*

Lemma 22 (Compressed permutations)

(i) $|\varnothing\rangle^{\mathfrak{p}} = \sum_{h:D\hookrightarrow R} \frac{1}{\sqrt{(N)_M}} |h\rangle$.

(ii) $\big\| |f\rangle^{\mathfrak{p}} \big\| = 1$.

(iii) $\langle f|\, g\rangle^{\mathfrak{p}} = \begin{cases} \frac{1}{\sqrt{(N-\ell_f)_{\ell_{g\text{-}f}} \cdot (N-\ell_g)_{\ell_{f\text{-}g}}}} & (f\diamond g, f\cup g \text{ inj.}) \\ 0 & (\text{otherwise}) \end{cases}$

In particular, for $f \neq g$, $0 \leq \langle f|\, g\rangle^{\mathfrak{p}} \leq 1/\sqrt{N - \max\{\ell_f, \ell_g\}}$.

(iv) $\mathbb{P}_r |f\rangle^{\mathfrak{p}} = \frac{1}{\sqrt{(N-\ell_f)_{\ell_{r\text{-}f}}}} |f \cup r\rangle^{\mathfrak{f}}$ *if* $f \diamond r$ *and* $f \cup r$ *injective. And* $\mathbb{P}_r |f\rangle^{\mathfrak{p}} = 0$ *otherwise.*

Proof of both lemmas. In addition to the notation stated before the lemmas, let $\ell_{fg} := |\mathrm{dom}\, f \cup \mathrm{dom}\, g|$ and $\ell_{fr} := |\mathrm{dom}\, f \cup \mathrm{dom}\, r|$.

Property (i) (in both lemmas) is immediate from the definition of $|\cdot\rangle^{\mathfrak{f}}$, $|\cdot\rangle^{\mathfrak{p}}$ (see Definitions 1 and 9, respectively), using the fact that $|D \to R| = N^M$ and $|D \hookrightarrow R| = (N)_M$.

Property (ii) (in both lemmas) is immediate from the definition.

For Lemma 21 (iii), first note that for any partial $h : D \rightharpoonup R$, there are $N^{M-|\mathrm{dom}\, h|}$

total functions $u : D \to R$ with $u \heartsuit f$.

Then if $f \heartsuit g$,

$$\begin{aligned}\langle f|\, g\rangle^{\mathfrak{f}} &\overset{(*)}{=} \sum_{\substack{t:D\to R, u:D\to R\\ t\heartsuit f, u\heartsuit g}} \Big(\underbrace{|\{t \in (D\to R) : t \heartsuit f\}|}_{=N^{M-\ell_f}}\Big)^{-\frac12}\Big(\underbrace{|\{u \in (D\to R) : u \heartsuit g\}|}_{=N^{M-\ell_g}}\Big)^{-\frac12} \underbrace{\langle t|\, u\rangle}_{\substack{=1 \text{ iff } t=u\\ =0 \text{ else}}} \\ &= \sum_{\substack{t:D\to R\\ t\heartsuit f\cup g}} N^{-M+\ell_f/2+\ell_g/2} = N^{M-\ell_{fg}} \cdot N^{-M+\ell_f/2+\ell_g/2} = N^{-(2\ell_{fg}-\ell_f-\ell_g)/2} \\ &\overset{(**)}{=} N^{-(\ell_{f\text{-}g}+\ell_{g\text{-}f})/2}.\end{aligned}$$

Here $(*)$ is by definition of $|\cdot\rangle^{\mathfrak{f}}$. And $(**)$ follows from the definitions of $\ell_f, \ell_g, \ell_{fg}, \ell_{f\text{-}g}, \ell_{g\text{-}f}$ and elementary reasoning about set cardinalities.

And if $f \not\heartsuit g$, then the second sum ranges over all t compatible with f and g simultaneously, but there are no such t, hence $\langle f|\, g\rangle^{\mathfrak{f}} = 0$ in that case.

The "in particular" case is immediate since for compatible $f \neq g$, at least one of $\ell_{f\text{-}g}, \ell_{g\text{-}f}$ is ≥ 1.

This shows Lemma 21 (iii).

For Lemma 22 (iii), first note that for any partial injection $h : D \rightharpoonup R$, there are $(N - |\operatorname{im} h|)_{M-|\operatorname{dom} h|} = (N - |\operatorname{dom} h|)_{M-|\operatorname{dom} h|}$ total injections $u : D \hookrightarrow R$ with $u \heartsuit h$.

Then if $f \heartsuit g$ and $f \cup g$ injective,

$$\begin{aligned}\langle f|\, g\rangle^{\mathfrak{p}} &\overset{(*)}{=} \sum_{\substack{t:D\hookrightarrow R, u:D\hookrightarrow R\\ t\heartsuit f, u\heartsuit g}} \Big(\underbrace{|\{t \in (D\hookrightarrow R) : t \heartsuit f\}|}_{=(N-\ell_f)_{M-\ell_f}}\Big)^{-\frac12}\Big(\underbrace{|\{u \in (D\hookrightarrow R) : u \heartsuit g\}|}_{=(N-\ell_g)_{M-\ell_g}}\Big)^{-\frac12} \underbrace{\langle t|\, u\rangle}_{\substack{=1 \text{ iff } t=u\\ =0 \text{ else}}} \\ &\overset{(**)}{=} \sum_{\substack{t:D\hookrightarrow R\\ t\heartsuit f\cup g}} \sqrt{\tfrac{(N-M)!}{(N-\ell_f)!}\cdot\tfrac{(N-M)!}{(N-\ell_g)!}} = (N-\ell_{fg})_{M-\ell_{fg}} \cdot \sqrt{\tfrac{(N-M)!}{(N-\ell_f)!}\cdot\tfrac{(N-M)!}{(N-\ell_g)!}} \\ &\overset{(***)}{=} \frac{1}{\sqrt{(N-\ell_f)_{\ell_{g\text{-}f}}\cdot(N-\ell_g)_{\ell_{f\text{-}g}}}}.\end{aligned}$$

Here $(*)$ is by definition of $|\cdot\rangle^{\mathfrak{p}}$. And $(**)$ uses the definition of the falling factorial $(a)_b = a!/(a-b)!$. And $(***)$ uses the definition of the falling factorial, the fact that $\ell_{fg} = \ell_f + \ell_{g\text{-}f} = \ell_g + \ell_{f\text{-}g}$, and some elementary simplification on both sides.

And if $f \not\heartsuit g$ or $f \cup g$ is not injective, then the second sum ranges over all injective total t compatible with f and g simultaneously, but there are no such t, hence $\langle f|\, g\rangle^{\mathfrak{p}} = 0$ in that case.

The "in particular" case follows since for $f \neq g$, at least one of $\ell_{f\text{-}g}, \ell_{g\text{-}f}$ is ≥ 1, and since $(a)_b \geq a$ for $b \geq 1$.

This shows Lemma 22 (iii).

For Lemma 21 (iv), recall the definition of $\mathbb{P}_r$ from the preliminaries. In the case that $f \diamond r$, we have

$$\begin{aligned}
\mathbb{P}_r|f\rangle^{\mathfrak{f}} &= \sum_{\substack{t:D\to R\\ t\heartsuit f}} \big(|\{t \in D \to R : t \heartsuit f\}|\big)^{-1/2}\, \mathbb{P}_r|t\rangle \\
&\overset{(*)}{=} \sum_{\substack{t:D\to R\\ t\heartsuit f\cup r}} \big(\underbrace{|\{t \in D \to R : t \heartsuit f\}|}_{=N^{M-\ell_f}}\big)^{-1/2}|t\rangle \\
&= N^{\ell_{fr}/2-\ell_f/2} \sum_{\substack{t:D\to R\\ t\heartsuit f\cup r}} \big(\underbrace{|\{t \in D \to R : t \heartsuit f\cup r\}|}_{=N^{M-\ell_{fr}}}\big)^{-1/2}|t\rangle \\
&= N^{\ell_{fr}/2-\ell_f/2}|f\cup r\rangle^{\mathfrak{f}} = N^{-\ell_{r\text{-}f}/2}|f\cup r\rangle^{\mathfrak{f}}.
\end{aligned}$$

Here $(*)$ uses the fact that $\mathbb{P}_r|t\rangle = |t\rangle$ if $t \diamond r$ and $= 0$ otherwise. Thus the sum gets restricted to those t with $t \diamond r$ and $t \diamond f$, hence $t \diamond f \cup r$. And the bounds under the braces were already stated in the proof of (iii).

In the case that $f \not\diamond r$, the second sum quantifies over functions t that are compatible with both f and r but there are no such t. Thus in this case, $\mathbb{P}_r|f\rangle^{\mathfrak{f}} = 0$.

This shows Lemma 21 (iv).

For Lemma 22 (iv), in the case that $f \diamond r$ and $f \cup r$ injective, we have

$$\begin{aligned}
\mathbb{P}_r|f\rangle^{\mathfrak{p}} &= \sum_{\substack{t:D\hookrightarrow R\\ t\heartsuit f}} \big(|\{t \in D \hookrightarrow R : t \heartsuit f\}|\big)^{-1/2}\, \mathbb{P}_r|t\rangle \\
&\overset{(*)}{=} \sum_{\substack{t:D\hookrightarrow R\\ t\heartsuit f\cup r}} \big(\underbrace{|\{t \in D \hookrightarrow R : t \heartsuit f\}|}_{=(N-\ell_f)_{M-\ell_f}}\big)^{-1/2}|t\rangle \\
&= \sqrt{\tfrac{(N-\ell_{fr})!}{(N-\ell_f)!}} \sum_{\substack{t:D\hookrightarrow R\\ t\heartsuit f\cup r}} \big(\underbrace{|\{t \in D \hookrightarrow R : t \heartsuit f\cup r\}|}_{=(N-\ell_{fr})_{M-\ell_{fr}}}\big)^{-1/2}|t\rangle \\
&= \sqrt{\tfrac{(N-\ell_{fr})!}{(N-\ell_f)!}} \cdot |f\cup r\rangle^{\mathfrak{f}} \overset{(**)}{=} \frac{1}{\sqrt{(N-\ell_f)_{\ell_{r\text{-}f}}}} \cdot |f\cup r\rangle^{\mathfrak{f}}.
\end{aligned}$$

Here $(*)$ uses the fact that $\mathbb{P}_r|t\rangle = |t\rangle$ if $t \diamond r$ and $= 0$ otherwise. Thus the sum gets restricted to those t with $t \diamond r$ and $t \diamond f$, hence $t \diamond f \cup r$. And the bounds under the braces were already stated in the proof of (iii). And $(**)$ uses the definition of the falling factorial, the fact that $\ell_{fr} = \ell_f + \ell_{r\text{-}f}$, and some elementary simplification.

In the case that $f \not\diamond r$ or $f \cup r$ non-injective, the second sum quantifies over injective functions t that are compatible with both f and r but there are no such t. Thus in this case, $\mathbb{P}_r|f\rangle^{\mathfrak{p}} = 0$.

This shows Lemma 22 (iv). □

5.2 Generalized case

In this section, we present and prove the theorems about compressed oracle queries and their effect on invariants in the most general setting. This setting encompasses both random functions and random permutations but is not restricted just to those two cases. This generality is bought by having harder to prove premises in the theorems. While the theorems that are specialized to random functions/permutations have a purely combinatorial premise (e.g., (6)), the premises of the general theorems involve some norms and inner products. In later sections, we derive the specialized theorems as corollaries.

First, we generalize the concept of compressed function/permutation oracles (in particular the definitions of the states $|f\rangle^{\mathfrak{f}}$ and $|f\rangle^{\mathfrak{p}}$) by stating some properties that should be satisfied by these states without prescribing a specific definition:

Definition 23 (Generalized compressed oracle) *A* generalized compressed oracle *consists of:*

- *Finite sets D and R (domain and range).*
- *A set* Func *of total functions $D \to R$. (Those correspond to the possible values of the uncompressed oracle.)*
- *A set* Valid *of partial functions $D \rightharpoonup R$. (We call functions in this set* valid*).*
- *A family of normalized states* $|f\rangle^{\mathfrak{g}} \in \mathbb{C}^{\mathsf{Func}}$ *for $f \in \mathsf{Valid}$. (We do not require that they form a basis or are orthogonal.)*

such that

(i) For $f \in \mathsf{Valid}$, and for $r : D \rightharpoonup R$: If $f \not\subset r$ or $(f \cup r) \notin \mathsf{Valid}$, then $\mathbb{P}_r|f\rangle^{\mathfrak{g}} = 0$.

(ii) For $f \in \mathsf{Valid}$, and for $r : D \rightharpoonup R$: If f, r are compatible and $(f \cup r) \in \mathsf{Valid}$, then there exists an $F \geq 0$ such that $\mathbb{P}_r|f\rangle^{\mathfrak{g}} = F|f \cup r\rangle^{\mathfrak{g}}$.

(iii) For $f, g \in \mathsf{Valid}$: If f, g are not compatible, then $\langle f|\, g\rangle^{\mathfrak{g}} = 0$.

(iv) For $f, g \in \mathsf{Valid}$: $\langle f|\, g\rangle^{\mathfrak{g}} \geq 0$.

Both the compressed function oracle and the compressed permutation oracle are examples of generalized compressed oracles. In the case of the compressed function oracle, $\mathsf{Func} := D \to R$, $\mathsf{Valid} := D \rightharpoonup R$, $\big|f\big\rangle^{\mathfrak{g}} := |f\rangle^{\mathfrak{f}}$. In the case of the compressed permutation/injection oracle, $\mathsf{Func} := D \hookrightarrow R$, $\mathsf{Valid} := D \rightharpoonup\!\!\!\!\!\!\cdot\;\; R$, $\big|f\big\rangle^{\mathfrak{g}} := |f\rangle^{\mathfrak{p}}$. The fact that these are generalized compressed oracles follows from Lemmas 21 and 22.

We can also model nonuniformly distributed oracles: For example, let $\mathcal{D}$ be a distribution over $D \to R$. Then an oracle whose state is chosen according to $\mathcal{D}$ can be modeled by the following generalized compressed oracle: $\mathsf{Func} := D \to R$ (or a suitable subset thereof), $\mathsf{Valid} := D \rightharpoonup R$ (or a suitable subset thereof), and

$$\big|f\big\rangle^{\mathfrak{g}} := \sum_{\substack{g \in \mathsf{Func} \\ g \heartsuit f}} \sqrt{\tfrac{\mathcal{D}(g)}{T_f}}\, |g\rangle \qquad \text{where} \qquad T_f := \sum_{\substack{g \in \mathsf{Func} \\ g \heartsuit f}} \mathcal{D}(g).$$

While the latter is a generalized compressed oracle, it may depend on the distribution $\mathcal{D}$ whether it is a useful one because the concrete parameters achievable in Theorem 24 below will depend on $\mathcal{D}$.

The following is the main theorem for the preservation of invariants under oracle queries in the generalized setting. Its basic structure is the same as the theorems we already saw (e.g., Theorem 13): It shows that under certain conditions, if ψ approximately satisfies an invariant, then $U\psi$ (the state after an oracle query) approximately satisfies a new invariant. The shape of these invariants in exactly as in Theorem 13 (besides being stated in terms of $|f\rangle^{\mathfrak{g}}$ instead of $|f\rangle^{\mathfrak{f}}$), so the way how this theorem is applied follows the same lines as we did with Theorem 13. However, there are a number of generalizations that we will discuss after the statement of the theorem.

Theorem 24 (Oracle query) *Fix a set $\mathbf{X}$ of tuples $\underline{x} \subseteq D$ (of possibly different lengths). Fix $c, d \geq 0$ with $d < \frac{1}{3}$. Fix $A_{\underline{x},e}, B_{\underline{x},e} \subseteq \mathsf{Valid}$ for $\underline{x} \in \mathbf{X}$ and e. Fix mutually orthogonal projectors $M_{\underline{x},e}$ on a Hilbert space $\mathcal{H}_E$ for $\underline{x} \in \mathbf{X}$ and e. For every $\underline{x} \in \mathbf{X}$, e, and $\underline{z} \subseteq R$ with $|\underline{x}| = |\underline{z}|$ (same length), fix a scaled isometry[24] $V_{\underline{x},e,\underline{z}}$ on a Hilbert space $\mathcal{H}_Y$.*

Assume that for all $\underline{x} \in \mathbf{X}$ and e, and all compatible $f, g \in A_{\underline{x},e}$,

$$\sum_{\substack{\underline{z} \\ (\underline{x}\mapsto\underline{z})\heartsuit f,g \\ f\cup(\underline{x}\mapsto\underline{z}),g\cup(\underline{x}\mapsto\underline{z})\in\mathsf{Valid}\setminus B_{\underline{x},e}}} \left\|\!\left|V_{\underline{x},e,\underline{z}}\right|\!\right\|^2 \cdot \left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}}\right\| \cdot \left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|g\rangle^{\mathfrak{g}}\right\| \cdot \frac{\left\langle f\cup(\underline{x}\mapsto\underline{z})\middle|\, g\cup(\underline{x}\mapsto\underline{z})\right\rangle^{\mathfrak{g}}}{\langle f|\, g\rangle^{\mathfrak{g}}} \leq c. \tag{28}$$

(Using the convention $0/0 := 0$, $a/0 := \infty$ for $a \neq 0$.)

Assume that for all compatible $f, g \in A_{\underline{x},e}$ with $f \neq g$: $\langle f|\, g\rangle^{\mathfrak{g}} \leq d$.

Let

$$U := \sum_{\substack{\underline{x}\in\mathbf{X},\ e,\ \underline{z} \\ |\underline{x}|=|\underline{z}|}} M_{\underline{x},e} \otimes V_{\underline{x},e,\underline{z}} \otimes \mathbb{P}_{\underline{x}\mapsto\underline{z}}$$

Fix a unit vector $\psi \in \mathcal{H}_E \otimes \mathcal{H}_Y \otimes \mathbb{C}^{\mathsf{Func}}$. If

$$\psi \overset{\varepsilon}{\approx} \sum_{\underline{x}\in\mathbf{X},\, e} \operatorname{im} M_{\underline{x},e} \otimes \mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in A_{\underline{x},e}\}_f \ =: \ \mathbf{A}$$

then

$$U\psi \overset{\varepsilon+\frac{2\sqrt{c}}{\sqrt{1-3d}}}{\approx} \sum_{\underline{x}\in\mathbf{X},\, e} \operatorname{im} M_{\underline{x},e} \otimes \mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B_{\underline{x},e}\}_f \ =: \ \mathbf{B}$$

($\sum$ is the sum of subspaces, i.e., the span of all linear combinations.)

(The theorem is shown in Section 5.2.1, page 55.) This theorem generalizes the previous theorems (Theorems 6 and 13) in the following ways:

[24] That is, an isometry multiplied with a complex scalar. The most important example are isometries and the zero-operator.

- It is not limited to the random function or random permutation case but applies to generalized compressed oracles. (Using $|f\rangle^{\mathfrak{g}}$ instead of $|f\rangle^{\mathfrak{f}}$ or $|f\rangle^{\mathfrak{p}}$.)
- It generalizes how oracle queries are formalized: Before, there was a distinguished register X that contained the oracle input $\underline{x}$ in the computational basis. Now the oracle input $\underline{x}$ can be encoded in an arbitrary way in the state of the environment register E. (The register E here encompasses both the query input register X and the environment register E from Theorems 6 and 13.) How $\underline{x}$ is derived from E is described by a projective measurement $M_{\underline{x},e}$. (Thus $M_{\underline{x},e}$ takes the role of the environment basis η_e in Theorems 6 and 13, and the role of measuring $\underline{x}$.)
- The output of the oracle is also handled in a more general fashion: Instead of having a register Y that encodes a tuple of outputs $\underline{y}$ in the computational basis, we allow the output register Y to have an arbitrary form. The operation of the oracle on Y in dependence of the output $\underline{z}$ is then specified by an isometry $V_{\underline{x},e,\underline{z}}$. For example, instead of XORing the result onto Y, the oracle could perform a phase shift that depends on $\underline{z}$. Also $V_{\underline{x},e,\underline{z}}$ can depend on $\underline{x}, e$, so the way the oracle operates can depend on other parts of the state. (And $V_{\underline{x},e,\underline{z}}$ is actually only required to be a scaled isometry. This might allow for oracles which have a non-zero probability of returning, by setting $V_{\underline{x},e,\underline{z}} := 0$ for some $\underline{x}, e, \underline{z}$.)
- There is no fixed length of query inputs $\underline{x}$ any more. Thus, the oracle can be queried in superposition between different amounts of parallelism. $\mathbf{X} = D^k$ would mean k-parallelism. $\mathbf{X} = D^*$ would mean unbounded parallelism. Anything in between is possible, too.
- There is no built-in requirement in the invariants that $\operatorname{dom} f \leq \ell$ for some integer ℓ. It is, of course, still possible (and in many situation necessary) to track the size of f in the invariants, but this needs to be included manually in the sets $A_{\underline{x},e}$ and $B_{\underline{x},e}$. This means that we have more flexibility, e.g., the size of ℓ might depend on e or on some outputs of f, etc.

While many of these generalizations are probably needed only in rare circumstances, we decided to keep them in the theorem, and refer to the specialized theorems for random functions/permutations for easier to read theorems.

Of course, analogous to the random function and permutation cases, we also need a theorem for non-oblivious queries (compare with Theorems 7 and 14). In this theorem, there is nothing conceptually new. The only generalization to mention is that instead of requiring Y to be initialized with $|0\rangle$, we more generally allow Y to be initialized with any state $\phi_{Y,\underline{x},e}$ that can depend on the query input $\underline{x}$ and the state of the environment E (via e).

Theorem 25 (Non-oblivious oracle query) *In the situation of Theorem 24, fix additionally some $\psi_{Y,\underline{x},e} \in \mathcal{H}_Y$.*

If

$$\psi \overset{\varepsilon}{\approx} \sum_{\underline{x}\in\mathbf{X},\, e} \operatorname{im} M_{\underline{x},e} \otimes \operatorname{span}\big\{\psi_{Y,\underline{x},e} \otimes |f\rangle^{\mathfrak{g}} : f \in A_{\underline{x},e}\big\}_f \;=:\; \mathbf{A}$$

then

$$U\psi \overset{\varepsilon+\frac{2\sqrt{c}}{\sqrt{1-3d}}}{\approx} \sum_{\underline{x}\in\mathbf{X},\, e} \operatorname{im} M_{\underline{x},e} \otimes \operatorname{span}\left\{V_{\underline{x},e,\underline{z}}\psi_{Y,\underline{x},e} \otimes |f\rangle^{\mathfrak{g}} : f \in B_{\underline{x},e} \wedge f(\underline{x}) = \underline{z}\right\}_{\underline{z}f} =: \mathbf{B}$$

(The theorem is shown in Section 5.2.1, page 56.)

5.2.1 Proofs in the generalized case

We now proceed to prove Theorems 24 and 25. In order to do so, we first show two lemmas that are limited special cases of those theorems. Namely, we are considering the case where the input $\underline{x}$ to the oracle is fixed. (In particular, no superposition between different inputs.) While very restricted, this case nontheless captures all the core difficulties in the proof. The general case is then a simple corollary. (Using the fact that all $|\underline{x}\rangle$ are orthogonal.) We also express the invariant preservation slightly differently (the norm of the projection onto the orthogonal complement of the invariant is small) which is technically a bit simpler when used in the proof of the general case.

The lemma that is a limited special case of Theorem 24 is the following:

Lemma 26 (Oracle query (fixed input)) *Let $A, B \subseteq \mathsf{Valid}$ (pre-/postcondition). Fix $c, d \geq 0$ with $d < \frac{1}{3}$. Fix a tuple $\underline{x} \subseteq D$. For every list $\underline{z} \subseteq R$ with $|\underline{z}| = |\underline{x}|$ (same length), fix a scaled isometry $V_{\underline{z}}$ on Hilbert space $\mathcal{H}_Y$. Assume that for all compatible $f, g \in A$:*

$$\sum_{\substack{\underline{z} \\ (\underline{x}\mapsto\underline{z})\diamond f,g \\ f\cup(\underline{x}\mapsto\underline{z}),g\cup(\underline{x}\mapsto\underline{z})\in\mathsf{Valid}\setminus B}} \left\|\!\left|V_{\underline{z}}\right|\!\right\|^2 \cdot \left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}}\right\| \cdot \left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|g\rangle^{\mathfrak{g}}\right\| \cdot \frac{\left\langle f\cup(\underline{x}\mapsto\underline{z})\middle|\, g\cup(\underline{x}\mapsto\underline{z})\right\rangle^{\mathfrak{g}}}{\langle f|\, g\rangle^{\mathfrak{g}}} \leq c. \quad (29)$$

(Using the convention $0/0 := 0$, $a/0 := \infty$ for $a \neq 0$.)

Assume further that for all compatible $f, g \in A$ with $f \neq g$: $\langle f|\, g\rangle^{\mathfrak{g}} \leq d$.

Let $U := \sum_{\underline{z}} V_{\underline{z}} \otimes \mathbb{P}_{\underline{x}\mapsto\underline{z}}$. Then for $\psi \in \left(\mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in A\}_f\right)$, we have

$$\left\|(1 - P_B)U\psi\right\| \leq \tfrac{2\sqrt{c}}{\sqrt{1-3d}} \cdot \|\psi\|$$

where P_B is the projector onto $\mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B\}_f$.

Proof. Since $\psi \in \left(\mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in A\}_f\right)$, there are scalars $\alpha_{yf} \in \mathbb{C}$ such that

$$\psi = \sum_{f\in A} \gamma_f \otimes |f\rangle^{\mathfrak{g}} \qquad \text{where} \qquad \gamma_f := \sum_y \alpha_{yf}|y\rangle \quad (30)$$

Here $|y\rangle$ are an arbitrary orthonormal basis of $\mathcal{H}_Y$. Let W_j for $j = 0, \ldots, 3$ be the quadrants of the complex plane. More precisely, $W_j := \left\{re^{ti\pi/2}.\ t \in [j, j+1), r > 0\right\}$. Note that the W_j form a partition of $\mathbb{C} \setminus \{0\}$. Furthermore, for $a, b \in W_j$, $\Re(\overline{a}b) \geq 0$.

We can then decompose ψ and γ_f as follows:

$$\gamma_{fj} := \sum_{\substack{y \\ \alpha_{yf} \in W_j}} \alpha_{yf}|y\rangle, \quad \psi_j := \sum_{f \in A} \gamma_{fj} \otimes |f\rangle^{\mathfrak{g}}. \qquad \text{Then } \gamma_f = \sum_{j=0}^{3} \gamma_{fj}, \quad \psi = \sum_{j=0}^{3} \psi_j.$$

The purpose of this decomposition is the following: Inner products involving the summands of a single ψ_j will have non-negative real parts. This will avoid negative factors in inequalities later on (in Claim 3) which would make it impossible to use monotonicity arguments.

For $j = 0, \ldots, 3$, define:

$$\phi^{good} := \sum_{\substack{f \in A,\ \underline{z} \\ (\underline{x} \mapsto \underline{z}) \heartsuit f \\ f \cup (\underline{x} \mapsto \underline{z}) \in B}} V_{\underline{z}} \gamma_f \otimes \mathbb{P}_{\underline{x} \mapsto \underline{z}} |f\rangle^{\mathfrak{g}}.$$

$$\phi^{bad} := \sum_{j=0}^{3} \phi_j^{bad}, \qquad \phi_j^{bad} := \sum_{\substack{f \in A,\ \underline{z} \\ (\underline{x} \mapsto \underline{z}) \heartsuit f \\ f \cup (\underline{x} \mapsto \underline{z}) \in \mathsf{Valid} \setminus B}} V_{\underline{z}} \gamma_{fj} \otimes \mathbb{P}_{\underline{x} \mapsto \underline{z}} |f\rangle^{\mathfrak{g}}.$$

$$\phi^{nope} := \sum_{\substack{f \in A,\ \underline{z} \\ (\underline{x} \mapsto \underline{z}) \not\heartsuit f \\ \text{or } f \cup (\underline{x} \mapsto \underline{z}) \notin \mathsf{Valid}}} V_{\underline{z}} \gamma_f \otimes \mathbb{P}_{\underline{x} \mapsto \underline{z}} |f\rangle^{\mathfrak{g}}.$$

Intuitively, ϕ^{good} is the part of the final state (after the query U) that satisfies the predicate B, ϕ^{bad} the part that does not satisfy B, and ϕ^{nope} will turn out to be 0. Note that

$$\phi^{good} + \phi^{bad} + \phi^{nope} = \sum_{f \in A,\ \underline{z}} V_{\underline{z}} \gamma_f \otimes \mathbb{P}_{\underline{x} \mapsto \underline{z}} |f\rangle^{\mathfrak{g}} = \sum_{\underline{z}} \left(V_{\underline{z}} \otimes \mathbb{P}_{\underline{x} \mapsto \underline{z}} \right) \sum_{f \in A} \gamma_f \otimes |f\rangle^{\mathfrak{g}} = U\psi. \tag{31}$$

Here the first equality uses that $\sum_j \gamma_{fj} = \gamma_j$ (see above), and that each $f \in A, \underline{z}$ satisfies the condition under exactly one of the three sums. And the last inequality follows from (30) and the definition of U in the statement of the lemma.

The following claim shows that ϕ^{good} is indeed good (namely, satisfies the postcondition B):

Claim 1 $\phi^{good} \in \mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B\}_f$.

Proof of claim. Fix $f \in A$ and $\underline{z}$ such that $(\underline{x} \mapsto \underline{z})$ is compatible with f and $f \cup (\underline{x} \mapsto \underline{z}) \in B$.

Since $B \subseteq \mathsf{Valid}$ (by assumption of the lemma), $f \cup (\underline{x} \mapsto \underline{z}) \in \mathsf{Valid}$ and thus by Definition 23 (ii), $\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} = F|f \cup (\underline{x} \mapsto \underline{z})\rangle^{\mathfrak{g}}$ for some $F \in \mathbb{C}$.

Since $f \cup (\underline{x} \mapsto \underline{z}) \in B$, $F|f \cup (\underline{x} \mapsto \underline{z})\rangle^{\mathfrak{g}} \in \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B\}_f$ and hence $V_{\underline{z}}\gamma_{fj} \otimes \mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} \in \mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B\}_f$.

Thus all the summands in the definition of ϕ^{good} are in $\mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B\}_f$. The claim follows. ◇

Claim 2 $\phi^{nope} = 0$.

Proof of claim. By Definition 23 (i), $\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} = 0$ when $(\underline{x} \mapsto \underline{z}) \not\circledcirc f$ or $f \cup (\underline{x} \mapsto \underline{z}) \notin \mathsf{Valid}$. Thus all the summands in the definition of ϕ^{nope} are 0. ◇

The following claim is the core of this lemma: It bounds the size of the "bad" parts of the final state.

Claim 3 $\|\phi_j^{bad}\| \leq \sqrt{c} \cdot \|\psi_j\|$.

Proof of claim. We have

$$\|\phi_j^{bad}\|^2 = \Re\big(\|\phi_j^{bad}\|^2\big) = \Re\Big(\sum_{f,g,\underline{z},\underline{z}'\,(*)} \langle V_{\underline{z}}\gamma_{fj}, V_{\underline{z}'}\gamma_{gj}\rangle \cdot \Big\langle \mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}}, \mathbb{P}_{\underline{x}\mapsto\underline{z}'}|g\rangle^{\mathfrak{g}}\Big\rangle\Big).$$

Here $(*)$ means that the sum ranges over all $f, g, \underline{z}, \underline{z}'$ with $f, g \in A$, $(\underline{x} \mapsto \underline{z}) \circledcirc f$, $(\underline{x} \mapsto \underline{z}') \circledcirc g$, $f \cup (\underline{x} \mapsto \underline{z}), g \cup (\underline{x} \mapsto \underline{z}') \in \mathsf{Valid} \setminus B$.

By Definition 23 (ii), and since $A \subseteq \mathsf{Valid}$, we have $\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} = F_{f\underline{z}}|f \cup (\underline{x} \mapsto \underline{z})\rangle^{\mathfrak{g}}$ for some $F_{f\underline{z}} \geq 0$ for $f, \underline{z}$ satisfying $(*)$. And analogously $\mathbb{P}_{\underline{x}\mapsto\underline{z}'}|g\rangle^{\mathfrak{g}} = F_{g\underline{z}'}|g \cup (\underline{x} \mapsto \underline{z}')\rangle^{\mathfrak{g}}$ for some $F_{g\underline{z}'} \geq 0$. Thus we can continue our calculation as:

$$\cdots = \Re\Big(\sum_{f,g,\underline{z},\underline{z}'\,(*)} \langle V_{\underline{z}}\gamma_{fj}, V_{\underline{z}'}\gamma_{gj}\rangle \cdot F_{f\underline{z}}F_{g\underline{z}'}\langle f \cup (\underline{x} \mapsto \underline{z})|\, g \cup (\underline{x} \mapsto \underline{z}')\rangle^{\mathfrak{g}}\Big)$$

If $\underline{z} \neq \underline{z}'$, then $f \cup (\underline{x} \mapsto \underline{z})$ and $g \cup (\underline{x} \mapsto \underline{z}')$ are not compatible. Thus by Definition 23 (iii), $\langle f \cup (\underline{x} \mapsto \underline{z})|\, g \cup (\underline{x} \mapsto \underline{z}')\rangle^{\mathfrak{g}} = 0$. Thus all summands with $\underline{z} \neq \underline{z}'$ vanish, thus:

$$\cdots = \Re\Big(\sum_{f,g,\underline{z},(**)} \langle V_{\underline{z}}\gamma_{fj}, V_{\underline{z}}\gamma_{gj}\rangle \cdot F_{f\underline{z}}F_{g\underline{z}}\langle f \cup (\underline{x} \mapsto \underline{z})|\, g \cup (\underline{x} \mapsto \underline{z})\rangle^{\mathfrak{g}}\Big)$$

Here $(**)$ means that the sum ranges over all $f, g, \underline{z}$ with $f, g \in A$, $(\underline{x} \mapsto \underline{z}) \circledcirc f, g$, $f \cup (\underline{x} \mapsto \underline{z}), g \cup (\underline{x} \mapsto \underline{z}) \in \mathsf{Valid} \setminus B$. (The only change is that we removed $\underline{z}'$ from the indices of the sum and replaced all occurrences by $\underline{z}$.)

Since $V_{\underline{z}}$ is a scaled isometry by assumptions of the lemma, $\langle V_{\underline{z}}\gamma_{fj}, V_{\underline{z}}\gamma_{gj}\rangle = v_{\underline{z}}\langle\gamma_{fj},\gamma_{gj}\rangle$ where $v_{\underline{z}} := \|\!|V_{\underline{z}}|\!\|^2$. Furthermore, when $f, g, \underline{z}$ satisfy $(**)$, for $G_{fg\underline{z}} := \langle f \cup (\underline{x}\mapsto\underline{z}) |\, g\cup(\underline{x}\mapsto\underline{z})\rangle^{\mathfrak{g}}/\langle f|\, g\rangle^{\mathfrak{g}}$ we have $G_{fg\underline{z}} \geq 0$ by Definition 23 (iv). (We set $G_{fg\underline{z}} := 0$ in the 0/0 case, and $:= \infty$ when only the denominator is 0. The case $G_{fg\underline{z}} = \infty$ can only occur when $v_{\underline{z}}F_{f\underline{z}}F_{g\underline{z}} = 0$ since otherwise (29) would not hold for finite c.) Then:

$$\cdots = \Re\Big(\sum_{f,g,\underline{z},(**)} \langle\gamma_{fj},\gamma_{gj}\rangle \cdot \underbrace{v_{\underline{z}}F_{f\underline{z}}F_{g\underline{z}}G_{fg\underline{z}}\langle f|\, g\rangle^{\mathfrak{g}}}_{\geq 0}\Big)$$

We can reorder the sum:

$$\cdots = \sum_{f,g\in A} \underbrace{\Big(\sum\nolimits_{\underline{z},(***)} v_{\underline{z}}F_{f\underline{z}}F_{g\underline{z}}G_{fg\underline{z}}\Big)}_{\leq c} \cdot \underbrace{\Re\big(\langle\gamma_{fj},\gamma_{gj}\rangle\big)}_{\geq 0} \cdot \underbrace{\langle f|\, g\rangle^{\mathfrak{g}}}_{\geq 0}$$

Here $(***)$ means the sum ranges over all $\underline{z}$ with $(\underline{x}\mapsto\underline{z}) \diamond f, g$, $f\cup(\underline{x}\mapsto\underline{z}), g\cup(\underline{x}\mapsto\underline{z}) \in \mathsf{Valid}\setminus B$.

We were able to pull the $\Re$ into the sum because $\langle\gamma_{fj},\gamma_{gj}\rangle$ was only multiplied with reals.

In the previous sum, we have $\langle f|\, g\rangle^{\mathfrak{g}} \geq 0$ by Definition 23 (iv).

And $\Re\big(\langle\gamma_{fj},\gamma_{gj}\rangle\big) \geq 0$ follows since $\langle\gamma_{fj},\gamma_{gj}\rangle = \sum_{\alpha_{yf},\alpha_{yg}\in W_j}{}^{y} \overline{\alpha_{yf}}\alpha_{yg}$, and $\overline{\alpha_{yf}}\alpha_{yg}$ has a non-negative real part (as is the case for any two elements of the same W_j).

And the fact that the sum over $\underline{z}$ is bounded by c is by assumption (29). (And using that $f, g \in A$ and the definitions of $v_{\underline{z}}, F_{f\underline{z}}, F_{g\underline{z}}, G_{fg\underline{z}}$. Note also that $\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}}\| = \big\|F_{f\underline{z}}|f\cup(\underline{x}\mapsto\underline{z})\rangle^{\mathfrak{g}}\big\| = |F_{f\underline{z}}| = F_{f\underline{z}}$.)

Thus:

$$\cdots \leq c\ \Re\Big(\sum_{f,g\in A}\langle\gamma_{fj},\gamma_{gj}\rangle\cdot\langle f|\, g\rangle^{\mathfrak{g}}\Big) = c\ \Re\big(\|\psi_j\|^2\big) = c\ \|\psi_j\|^2.$$

Thus we showed

$$\|\phi_j^{bad}\|^2 \leq c\ \|\psi_j\|^2.$$

Taking the square root on both sides proves the claim. ◇

Unfortunately, we cannot directly use Claim 3 to get a bound on $\|\phi^{bad}\|$ in terms of $\|\psi\|$, only in terms of $\sum\|\psi_j\|$. Since in general, $\sum\|\psi_j\|$ can be much larger than $\|\sum\psi_j\|$ (e.g., when $\sum\psi_j = 0$), that bound could be very loose. Fortunately the following claim shows that, in the present setting, this is not the case:

Claim 4 $\sum_j\|\psi_j\| \leq \frac{2}{\sqrt{1-3d}}\cdot\|\psi\|$.

Proof of claim. For $j = 0, \dots, 3$ we have

$$\|\psi_j\|^2 = \Re\big(\|\psi_j\|^2\big) = \Re\Big(\sum_{f,g\in A} \langle \gamma_{fj}, \gamma_{gj}\rangle \cdot \underbrace{\langle f|\, g\rangle^{\mathfrak{g}}}_{\geq 0}\Big) = \sum_{f,g\in A} \underbrace{\Re\big(\langle \gamma_{fj}, \gamma_{gj}\rangle\big)}_{\geq 0} \cdot \underbrace{\langle f|\, g\rangle^{\mathfrak{g}}}_{\geq 0}$$

Here the fact that $\langle f|\, g\rangle^{\mathfrak{g}} \geq 0$ is by Definition 23 (iv) (using $f, g \in A \subseteq \mathsf{Valid}$). And $\Re\big(\langle \gamma_{fj}, \gamma_{gj}\rangle\big) \geq 0$ by definition of γ_{fj} (this fact was already show in the proof of Claim 3).

We can omit all summands with $f \neq g$ and only make the sum smaller:

$$\cdots \geq \sum_{f\in A} \Re\big(\langle \gamma_{fj}, \gamma_{fj}\rangle\big) \cdot \langle f|\, f\rangle^{\mathfrak{g}} = \sum_{f\in A} \|\gamma_{fj}\|^2. \tag{32}$$

For all $j \neq k$, we have

$$\big|\langle \psi_j, \psi_k\rangle\big| = \Big|\sum_{f,g\in A} \langle \gamma_{fj}, \gamma_{gk}\rangle \cdot \underbrace{\langle f|\, g\rangle^{\mathfrak{g}}}_{\geq 0}\Big| \leq \sum_{f,g\in A} \underbrace{\big|\langle \gamma_{fj}, \gamma_{gk}\rangle\big|}_{=0 \text{ if } f=g} \cdot \underbrace{\langle f|\, g\rangle^{\mathfrak{g}}}_{\leq d \text{ if } f\neq g}$$

Here $\langle f|\, g\rangle \geq 0$ is by Definition 23 (iv) (using also that $f, g \in A \subseteq \mathsf{Valid}$). And $\langle f|\, g\rangle^{\mathfrak{g}} \leq d$ for $f \neq g$ holds by assumption of the lemma when $f \diamond g$ and by Definition 23 (iii) when $f \not\diamond g$. And $\langle \gamma_{fj}, \gamma_{gk}\rangle = 0$ for $f = g$ holds because by definition of γ_{fj}, we have $\langle \gamma_{fj}, \gamma_{fk}\rangle = \sum_{\alpha_{yf}\in W_j, \alpha_{yf}\in W_k}{}^{y} \overline{\alpha_{yf}}\alpha_{yg} = 0$ (the last equality follows since W_j and W_k are disjoint and thus we have an empty sum).

We can thus continue the computation:

$$\begin{aligned}\dots &\leq d \cdot \sum_{f,g\in A} \big|\langle \gamma_{fj}, \gamma_{gk}\rangle\big| \leq d \cdot \sum_{f,g\in A} \|\gamma_{fj}\| \cdot \|\gamma_{gk}\| \\ &\overset{\text{CSI}}{\leq} d \cdot \sqrt{\sum_{f\in A} \|\gamma_{fj}\|^2 \cdot \sum_{g\in A} \|\gamma_{gk}\|^2} \overset{(32)}{\leq} d \cdot \|\psi_j\| \cdot \|\psi_k\|. \end{aligned} \tag{33}$$

Here CSI refers to the Cauchy-Schwarz inequality.

Finally, we compute:

$$\begin{aligned}\|\psi\|^2 = \Big\|\sum_j \psi_j\Big\|^2 &= \sum_j \|\psi_j\|^2 + \sum_{j\neq k} \langle \psi_j, \psi_k\rangle \geq \sum_j \|\psi_j\|^2 - \sum_{j\neq k} \big|\langle \psi_j, \psi_k\rangle\big| \\ &\overset{(33)}{\geq} \sum_j \|\psi_j\|^2 - d \cdot \sum_{j\neq k} \|\psi_j\| \cdot \|\psi_k\| = \tfrac{1+d}{4}\Big(\sum_j \|\psi_j\|^2\Big)\Big(\sum_j 1^2\Big) - d \cdot \Big(\sum_j \|\psi_j\|\Big)^2 \\ &\overset{\text{CSI}}{\geq} \tfrac{1+d}{4}\Big(\sum_j \|\psi_j\| \cdot 1\Big)^2 - d \cdot \Big(\sum_j \|\psi_j\|\Big)^2 = \tfrac{1-3d}{4} \cdot \Big(\sum_j \|\psi_j\|\Big)^2. \end{aligned}$$

Since $d < \frac{1}{3}$ by assumption of the lemma, $\left(\sum_j \|\psi_j\|\right)^2 \leq \frac{4}{1-3d}\|\psi\|^2$. The claim follows. ◇

We now have

$$\|\phi^{bad}\| = \Big\|\sum_j \phi_j^{bad}\Big\| \leq \sum_j \|\phi_j^{bad}\| \overset{\text{Claim 3}}{\leq} \sqrt{c}\cdot\sum_j \|\psi_j\| \overset{\text{Claim 4}}{\leq} \tfrac{2\sqrt{c}}{\sqrt{1-3d}}\cdot\|\psi\|. \tag{34}$$

And thus

$$\big\|(1-P_B)U\psi\big\| \overset{(31)}{=} \big\|(1-P_B)(\phi^{good}+\phi^{bad}+\phi^{nope})\big\| \overset{(*)}{=} \big\|(1-P_B)\phi^{bad}\big\| \overset{(**)}{\leq} \big\|\phi^{bad}\big\| \overset{(34)}{\leq} \tfrac{2\sqrt{c}}{\sqrt{1-3d}}\cdot\|\psi\|.$$

Here $(*)$ uses Claim 1 to show that $(1-P_B)\phi^{good} = 0$ and Claim 2 to show that $\phi^{nope} = 0$. And $(**)$ uses that $1-P_B$ has operator norm ≤ 1 since P_B is a projector.

The last inequality shows the lemma. □

And the lemma that is a limited special case of the non-oblivious Theorem 24 is the following:

Lemma 27 (Non-oblivious oracle query (fixed input)) *In the situation of Lemma 26, assume additionally that $\psi = \psi_Y \otimes \psi_H$ for some ψ_Y, ψ_H. Then we have*

$$\big\|(1-P'_B)U\psi\big\| \leq \tfrac{2\sqrt{c}}{\sqrt{1-3d}}\cdot\|\psi\|$$

where P'_B is the projector onto $\operatorname{span}\big\{V_{\underline{z}}\psi_Y \otimes |f\rangle^{\mathfrak{g}} : f \in B \wedge f(\underline{x}) = \underline{z}\big\}_{\underline{z}f}$.

This lemma is a simple corollary of the previous proof.

Proof. Since we assume all assumptions from Lemma 26, everything shown in the proof of that lemma still applies. Furthermore, note that in that proof, when we defined γ_f in (30), we used an orthonormal basis $|y\rangle$ of $\mathcal{H}_Y$ that was chosen arbitrarily. We can therefore instead chose it such that ψ_Y and $|0\rangle$ are colinear. Everything in the proof still holds with that choice of basis.

We strengthen Claim 1 as follows:

Claim 1 $\phi^{good} \in \operatorname{span}\big\{V_{\underline{z}}\psi_Y \otimes |f\rangle^{\mathfrak{g}} : f \in B \wedge f(\underline{x}) = \underline{z}\big\}_{\underline{z}f}$.

Proof of claim. Fix $f \in A$ and $\underline{z}$ such that $(\underline{x} \mapsto \underline{z})$ is compatible with f and $f' := f \cup (\underline{x} \mapsto \underline{z}) \in B$.

Since $B \subseteq \mathsf{Valid}$ (by assumption of the lemma), $f \cup (\underline{x} \mapsto \underline{z}) \in \mathsf{Valid}$ and thus by Definition 23 (ii), $\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} = F|f'\rangle$ for some $F \in \mathbb{C}$.

Since $f' \in B$ and $f'(\underline{x}) = \underline{z}$, we have $V_{\underline{z}}\gamma_{fj} \otimes \mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{g}} = V_{\underline{z}}\gamma_{fj} \otimes F|f'\rangle^{\mathfrak{g}} \in \operatorname{span}\big\{V_{\underline{z}}\psi_Y \otimes |f\rangle^{\mathfrak{g}} : f \in B \wedge f(\underline{x}) = \underline{z}\big\}_{\underline{z}f}$.

Thus all the summands in the definition of ϕ^{good} are in $\operatorname{span}\{V_{\underline{z}}\psi_Y \otimes |f\rangle^{\mathfrak{g}} : f \in B \wedge f(\underline{x}) = \underline{z}\}_{\underline{z}f}$. ◇

We can now redo the very last calculation from the proof of Lemma 26 for P'_B instead of P_B:

$$\big\|(1-P'_B)U\psi\big\| \overset{(31)}{=} \big\|(1-P'_B)(\phi^{good}+\phi^{bad}+\phi^{nope})\big\| \overset{(*)}{=} \big\|(1-P'_B)\phi^{bad}\big\| \overset{(**)}{\leq} \big\|\phi^{bad}\big\| \overset{(34)}{\leq} \tfrac{2\sqrt{c}}{\sqrt{1-3d}}\cdot\|\psi\|.$$

Here $(*)$ uses Claim 1 to show that $(1-P'_B)\phi^{good}=0$ and Claim 2 to show that $\phi^{nope}=0$. And $(**)$ uses that $1-P'_B$ has operator norm ≤ 1 since P_B is a projector. □

Proofs of the main theorems. We can now derive Theorems 24 and 25 from the preceding lemmas:

Proof of Theorem 24. Without loss of generality, we can assume that all $M_{\underline{x},e}$ are rank-1. (Otherwise, we split them into more projectors.) Then $M_{\underline{x},e}$ is the projector onto some normalized $\eta_{\underline{x},e} \in \mathcal{H}_E$.

Since $\psi \overset{\varepsilon}{\approx} \mathbf{A}$ by assumption, we have that $\psi \overset{\varepsilon}{\approx} P_{\mathbf{A}}\psi$ where $P_{\mathbf{A}}$ is the projector on $\mathbf{A}$.

Let $P_{B,\underline{x},e}$ be the projector on $\mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{g}} : f \in B_{\underline{x},e}\}_f$. Let $U_{\underline{x},e} := \sum_{\underline{z}} V_{\underline{x},e,\underline{z}} \otimes \mathbb{P}_{\underline{x}\mapsto\underline{z}}$.

Let $\psi_{\underline{x},e} \in \mathcal{H}_Y \otimes \mathbb{C}^{\mathsf{Func}}$ be such that $(M_{\underline{x},e}\otimes I\otimes I)P_{\mathbf{A}}\psi = \eta_{\underline{x},e}\otimes\psi_{\underline{x},e}$. Let $\psi_{rest} := \big((I-\sum_{\underline{x},e} M_{\underline{x},e})\otimes I\otimes I\big)P_{\mathbf{A}}\psi$.

Since the $M_{\underline{x},e}$ are mutually orthogonal, we have that the $\eta_{\underline{x},e}\otimes\psi_{\underline{x},e}$ and ψ_{rest} are all mutually orthogonal. And $P_{\mathbf{A}}\psi = \psi_{rest} + \sum_{\underline{x},e}\eta_{\underline{x},e}\otimes\psi_{\underline{x},e}$.

Since $P_{\mathbf{A}}\psi \in \mathbf{A}$, we have $\eta_{\underline{x},e}\otimes\psi_{\underline{x},e} = (M_{\underline{x},e}\otimes I\otimes I)P_{\mathbf{A}}\psi \in \operatorname{im} M_{\underline{x},e}\otimes\mathcal{H}_Y\otimes\operatorname{span}\{|f\rangle^{\mathfrak{g}} : f\in A_{\underline{x},e}\}_f$. Then $\psi_{\underline{x},e}\in\mathcal{H}_Y\otimes\operatorname{span}\{|f\rangle^{\mathfrak{g}} : f\in A_{\underline{x},e}\}_f$.

Then by Lemma 26 (with $A := A_{\underline{x},e}$, $B := B_{\underline{x},e}$, $V_{\underline{z}} := V_{\underline{z},e}$, $\psi := \psi_{\underline{x},e}$, $P_B := P_{B,\underline{x},e}$), we have

$$\big\|(1-P_{B,\underline{x},e})U_{\underline{x},e}\psi_{\underline{x},e}\big\| \leq L\,\|\psi_{\underline{x},e}\| \qquad \text{for} \qquad L := \tfrac{2\sqrt{c}}{\sqrt{1-3d}}. \tag{35}$$

Let $P_{\mathbf{B}}$ be the projector onto $\mathbf{B}$. Then

$$\begin{aligned}
\big\|(1-P_{\mathbf{B}})UP_{\mathbf{A}}\psi\big\|^2 &= \Big\|\sum_{\underline{x},e}(1-P_{\mathbf{B}})U(\eta_{\underline{x},e}\otimes\psi_{\underline{x},e}) + (1-P_{\mathbf{B}})U\psi_{rest}\Big\|^2 \\
&\leq \sum_{\underline{x},e}\Big\|(1-P_{\mathbf{B}})U(\eta_{\underline{x},e}\otimes\psi_{\underline{x},e})\Big\|^2 \\
&\overset{(*)}{=} \sum_{\underline{x},e}\big\|(1-P_{\mathbf{B}})(I\otimes U_{\underline{x},e})(\eta_{\underline{x},e}\otimes\psi_{\underline{x},e})\big\|^2 \\
&\overset{(**)}{=} \sum_{\underline{x},e}\big\|\big(1-(I\otimes P_{B,\underline{x},e})\big)(I\otimes U_{\underline{x},e})(\eta_{\underline{x},e}\otimes\psi_{\underline{x},e})\big\|^2
\end{aligned}$$

$$= \sum_{\underline{x},e} \left\| \eta_{\underline{x},e} \otimes (1 - P_{B,\underline{x},e}) U_{\underline{x},e} \psi_{\underline{x},e} \right\|^2 = \sum_{\underline{x},e} \left\| (1 - P_{B,\underline{x},e}) U_{\underline{x},e} \psi_{\underline{x},e} \right\|^2$$

$$\overset{(35)}{\leq} L^2 \sum_{\underline{x},e} \|\psi_{\underline{x},e}\|^2 \leq L^2 \Big(\|\psi_{rest}\|^2 + \sum_{\underline{x},e} \|\eta_{\underline{x},e} \otimes \psi_{\underline{x},e}\|^2 \Big)$$

$$\overset{(***)}{=} L^2 \Big\| \psi_{rest} + \sum_{\underline{x},e} \eta_{\underline{x},e} \otimes \psi_{\underline{x},e} \Big\|^2 = L^2 \|P_{\mathbf{A}} \psi\|^2 \leq L^2 \|\psi\|^2 = L^2.$$

Here $(*)$ is by the definition of U and $U_{\underline{x},e}$, and by the fact that $\eta_{\underline{x},e}$ is in the image of the projector $M_{\underline{x},e}$. And $(**)$ follows since $\eta_{\underline{x},e}$ is in the image of $M_{\underline{x},e}$ and thus $P_{\mathbf{B}}$ and $I \otimes P_{B,\underline{x},e}$ are equal on vectors of the form $\eta_{\underline{x},e} \otimes \cdots$. And $(***)$ follows since the $\psi_{\underline{x},e}$ and ψ_{rest} are all mutually orthogonal.

Thus $\|(1 - P_{\mathbf{B}}) U P_{\mathbf{A}} \psi\| \leq L$. Hence

$$U\psi \overset{\varepsilon}{\approx} U P_{\mathbf{A}} \psi \overset{L}{\approx} P_{\mathbf{B}} U P_{\mathbf{A}} \psi \in \mathbf{B}$$

(The $\overset{\varepsilon}{\approx}$ follows since $\psi \overset{\varepsilon}{\approx} P_{\mathbf{A}} \psi$ and U is unitary.)

Thus $U\psi \overset{\varepsilon+L}{\approx} \mathbf{B}$. By definition of L this is the conclusion of the theorem. □

Proof of Theorem 25. The proof is identical to the proof of Theorem 24, except for the following additions and differences:

- In the previous proof, we chose $\psi_{x,e}$ such that $(M_{\underline{x},e} \otimes I \otimes I)\psi = \eta_{\underline{x},e} \otimes \psi_{\underline{x},e}$. Combined with the strengthened assumption $\psi \overset{\varepsilon}{\approx} \mathbf{A}$ of the current theorem, this means $\psi_{\underline{x},e} \in \operatorname{span}\{\psi_{Y,\underline{x},e} \otimes |f\rangle^{\mathfrak{g}}\}_f$. This implies that $\psi_{\underline{x},e} = \psi_{Y,\underline{x},e} \otimes \psi_{H,\underline{x},e}$ for some $\psi_{H,\underline{x},e} \in \mathcal{H}$.
- The definition of $P_{B,\underline{x},e}$ is changed to $P_{B,\underline{x},e} := \operatorname{span}\big\{V_{\underline{x},e,\underline{z}} \psi_{Y,\underline{x},e} \otimes |f\rangle^{\mathfrak{g}} : f \in B_{\underline{x},e} \wedge f(\underline{x}) = \underline{z}\big\}$.
- Instead of Lemma 26, we use Lemma 27 (with $A := A_{\underline{x},e}$, $B := B_{\underline{x},e}$, $V_{\underline{z}} := V_{\underline{z},e}$, $\psi := P_{\mathbf{A}} \psi_{\underline{x},e}$, $P'_B := P_{B,\underline{x},e}$, $\psi_Y := \psi_{Y,\underline{x},e}$, $\psi_H := \psi_{H,\underline{x},e}$). □

5.3 Random functions

We now show Theorems 6 and 7 as corollaries of the general Theorems 24 and 25.

Basically, the proofs boil down to instantiating the generalized compressed oracle (Definition 23) with $|f\rangle^{\mathfrak{f}}$ instead of $|f\rangle^{\mathfrak{g}}$, and then heavily simplifying the preconditions (using the basic properties from Lemma 21).

Theorem 6 (Random function query).

Proof of Theorem 6. The compressed function oracle is a special case of the generalized compressed oracle with $\mathsf{Func} := D \to R$ and $\mathsf{Valid} := D \rightharpoonup R$ and $|f\rangle^{\mathfrak{g}} := |f\rangle^{\mathfrak{f}}$. Thus we show this theorem by instantiating Theorem 24. In that theorem, let $\mathbf{X} := D^k$, $A_{\underline{x},e} := A_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \le \ell\}$, $B_{\underline{x},e} := B_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \le \ell + k\}$, $\mathcal{H}_E := \mathcal{H}_E \otimes \mathbb{C}^{D^k}$, $M_{\underline{x},e} := \eta_e \eta_e^\dagger \otimes |\underline{x}\rangle\langle\underline{x}|$, $V_{\underline{x},e,\underline{z}} := U_{\oplus \underline{z}}$ which is defined by $U_{\oplus \underline{z}}|\underline{y}\rangle := |\underline{y} \oplus \underline{z}\rangle$, $c := c$, $d := \frac{1}{4}$.

To apply Theorem 24, we need to show two bounds. First, for all $f \neq g$, $\langle f|\, g\rangle^{\mathfrak{f}} \le d = \frac{1}{4}$. This follows from $N \ge 16$ by Lemma 21 (iii).

Second, we need to show the bound from (28). Instantiated in our setting, this bound becomes: For all $\underline{x}, e$ and compatible $f, g \in A_{\underline{x},e}$ with $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \le \ell$:

$$\sum_{\underline{z} \in D^k\,(*)} \underbrace{\left\|\left\|U_{\oplus \underline{z}}\right\|\right\|^2}_{=1} \cdot \underbrace{\left\|\left|\mathbb{P}_{\underline{x} \mapsto \underline{z}}|f\rangle^{\mathfrak{f}}\right\|\right|}_{=N^{-\ell_{\underline{x}\text{-}f}/2}} \cdot \underbrace{\left\|\left|\mathbb{P}_{\underline{x} \mapsto \underline{z}}|g\rangle^{\mathfrak{f}}\right\|\right|}_{=N^{-\ell_{\underline{x}\text{-}g}/2}} \cdot \underbrace{\frac{\left\langle f \cup (\underline{x} \mapsto \underline{z})\middle|\, g \cup (\underline{x} \mapsto \underline{z})\right\rangle^{\mathfrak{f}}}{\langle f|\, g\rangle^{\mathfrak{f}}}}_{=N^{-\ell_{f\underline{x}\text{-}g\underline{x}}/2 - \ell_{g\underline{x}\text{-}f\underline{x}}/2 + \ell_{f\text{-}g}/2 + \ell_{g\text{-}f}/2}} \le c \qquad (36)$$

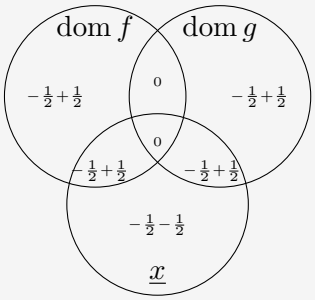
where $(*)$ means that the sum ranges over $\underline{z}$ satisfying: $(\underline{x} \mapsto \underline{z}) \heartsuit f, g$ and $f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}} \notin B_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \le \ell + k\}$.

The bounds under (36) come from Lemma 21 (iv) and (iii). In those bounds, $\ell_{\underline{x}\text{-}f} := \big|\underline{x} \setminus \mathrm{dom}\, f\big|$, $\ell_{f\underline{x}\text{-}g\underline{x}} := \big|(\mathrm{dom}\, f \cup \underline{x}) \setminus (\mathrm{dom}\, g \cup \underline{x})\big|$, $\ell_{f\text{-}g} := \big|\mathrm{dom}\, f \setminus \mathrm{dom}\, g\big|$, etc. (All $|\cdot|$ are cardinalities of sets here, not of multisets/tuples. I.e., $\underline{x}$ is interpreted as a set here.)

We have

$$-\ell_{\underline{x}\text{-}f}/2 - \ell_{\underline{x}\text{-}g}/2 - \ell_{f\underline{x}\text{-}g\underline{x}}/2 - \ell_{g\underline{x}\text{-}f\underline{x}}/2 + \ell_{f\text{-}g}/2 + \ell_{g\text{-}f}/2 = -\ell_{\underline{x}\text{-}fg}. \qquad (37)$$

This is seen most easily by checking that each region in the following Venn diagram contributes the indicated amount to the lhs.



The only non-zero region is $\underline{x} \setminus \mathrm{dom}\, f \setminus \mathrm{dom}\, g$ which contributes 1 to the lhs. Hence the lhs is equal to $\ell_{\underline{x}\text{-}fg}$. This shows (37).

From (37), it follows that the summand in (36) equals $N^{-\ell_{\underline{x}\text{-}fg}}$. Thus (36) can be restated as $Z/N^{-\ell_{\underline{x}\text{-}fg}} \le c$ where Z is the number of $\underline{z}$ that satisfy $(*)$. Since $|\mathrm{dom}\, f| \le \ell$ by assumption, and $|\underline{x}| \le k$, we always have $f \cup (\underline{x} \mapsto \underline{z}) \in (f : |\mathrm{dom}\, f| \le \ell + k)$. Hence $f \cup (\underline{x} \mapsto \underline{z}) \notin B_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \le \ell + k\}$ iff $f \cup (\underline{x} \mapsto \underline{z}) \notin B_{\underline{x},e}$. Analogously for

$g \cup (\underline{x} \mapsto \underline{z})$. Thus the set of $\underline{z}$ that satisfy $(*)$ is the set in the numerator in (12). Hence (12) is equivalent to $Z/N^{-\ell_{\underline{x}\text{-}fg}} \leq c$ which in turn is equivalent to (36). Since we assumed (12), (36) holds and Theorem 24 is applicable.

Furthermore, note that for our choices of $\mathbf{X}, M_{\underline{x},e}, V_{\underline{x},e,\underline{z}}$, the operator U defined in Theorem 24 is equal to $I_E \otimes U_{query,k}$. And $\mathbf{A}, \mathbf{B}$ defined in Theorem 24 are the same as $\mathbf{A}, \mathbf{B}$ defined in the present theorem.

Thus Theorem 24 shows that

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon+\frac{2\sqrt{c}}{\sqrt{1-3d}}}{\approx} \mathbf{B}.$$

Since $d = \frac{1}{4}$, we have $\varepsilon + \frac{2\sqrt{c}}{\sqrt{1-3d}} = \varepsilon + 4\sqrt{c}$. The theorem follows. □

Theorem 7 (Non-oblivious random function query).

Proof of Theorem 7. Analogous to Theorem 6, except using Theorem 25 instead of Theorem 24. □

5.4 Random permutations

We now show Theorems 6 and 7 as corollaries of the general Theorems 24 and 25.

The proofs boil down to instantiating the generalized compressed oracle (Definition 23) with $|f\rangle^{\mathfrak{p}}$ instead of $|f\rangle^{\mathfrak{g}}$, and then heavily simplifying the preconditions (using the basic properties from Lemma 22).

Theorem 13 (Random permutation query).

Proof of Theorem 13. The compressed permutation oracle is a special case of the generalized compressed oracle with $\mathsf{Func} := D \hookrightarrow R$ and $\mathsf{Valid} := D \overset{!}{\hookrightarrow} R$ and $|f\rangle^{\mathfrak{g}} := |f\rangle^{\mathfrak{p}}$. Thus we show this theorem by instantiating Theorem 24. In that theorem, let $\mathbf{X} := D^k$, $A_{\underline{x},e} := A_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \leq \ell\}$, $B_{\underline{x},e} := B_{\underline{x},e} \cap \{f : |\mathrm{dom}\, f| \leq \ell + k\}$, $\mathcal{H}_E := \mathcal{H}_E \otimes \mathbb{C}^{D^k}$, $M_{\underline{x},e} := \eta_e \eta_e^\dagger \otimes |\underline{x}\rangle\langle\underline{x}|$, $V_{\underline{x},e,\underline{z}} := U_{\oplus\underline{z}}$ which is defined by $U_{\oplus\underline{z}}|\underline{y}\rangle := |\underline{y} \oplus \underline{z}\rangle$, $c := c$, $d := \frac{1}{4}$.

To apply Theorem 24, we need to show two bounds. First, for all $f \neq g$, $\langle f|\, g\rangle^{\mathfrak{f}} \leq d = \frac{1}{4}$. This follows from $\ell \leq N - 16$ by Lemma 22 (iii).

Second, we need to show the bound from (28). Instantiated in our setting, this bound becomes: For all $\underline{x}, e$ and compatible $f, g \in A_{\underline{x},e}$ with $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \leq \ell$:

$$\sum_{\underline{z}\in D^k (*)} \underbrace{\left\|\left\|U_{\oplus\underline{z}}\right\|\right\|^2}_{=1} \cdot \underbrace{\left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|f\rangle^{\mathfrak{f}}\right\|}_{=\frac{1}{\sqrt{(N-\ell_f)\ell_{\underline{x}\text{-}f}}}} \cdot \underbrace{\left\|\mathbb{P}_{\underline{x}\mapsto\underline{z}}|g\rangle^{\mathfrak{f}}\right\|}_{=\frac{1}{\sqrt{(N-\ell_g)\ell_{\underline{x}\text{-}g}}}} \cdot \underbrace{\frac{\left\langle f \cup (\underline{x}\mapsto\underline{z})\right|\, g \cup (\underline{x}\mapsto\underline{z})\rangle^{\mathfrak{f}}}{\langle f|\, g\rangle^{\mathfrak{f}}}}_{=\sqrt{\frac{(N-\ell_f)\ell_{g\text{-}f}\cdot(N-\ell_g)\ell_{f\text{-}g}}{(N-\ell_{f_{\underline{x}}})\ell_{g_{\underline{x}}\text{-}f_{\underline{x}}}\cdot(N-\ell_{g_{\underline{x}}})\ell_{f_{\underline{x}}\text{-}g_{\underline{x}}}}}} \leq c \quad (38)$$

where $(*)$ means that the sum ranges over $\underline{z}$ satisfying: $(\underline{x} \mapsto \underline{z}) \heartsuit f, g$, and $f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}}$ injective and $f_{\underline{x},\underline{z}}, g_{\underline{x},\underline{z}} \notin B_{\underline{x},e} \cap \{f : |\operatorname{dom} f| \leq \ell + k\}$.

The bounds under (38) come from Lemma 22 (iv) and (iii). In those bounds, $\ell_{f\underline{x}} := \big|\operatorname{dom} f \cup \underline{x}\big|$, $\ell_{\underline{x}\text{-}f} := \big|\underline{x} \setminus \operatorname{dom} f\big|$, $\ell_{f\underline{x}\text{-}g\underline{x}} := \big|(\operatorname{dom} f \cup \underline{x}) \setminus (\operatorname{dom} g \cup \underline{x})\big|$, $\ell_{f\text{-}g} := \big|\operatorname{dom} f \setminus \operatorname{dom} g\big|$, etc. (All $|\cdot|$ are cardinalities of sets here, not of multisets/tuples. I.e., $\underline{x}$ is interpreted as a set here.)

Unfolding the definition of the falling factorial $(a)_b = a!/(a-b)!$, the summand in (38) becomes

$$\sqrt{\frac{(N-\ell_f-\ell_{\underline{x}\text{-}f})!}{\cancel{(N-\ell_f)!}} \frac{(N-\ell_g-\ell_{\underline{x}\text{-}g})!}{\cancel{(N-\ell_g)!}} \frac{\cancel{(N-\ell_f)!}}{(N-\ell_f-\ell_{g\text{-}f})!} \frac{\cancel{(N-\ell_g)!}}{(N-\ell_g-\ell_{f\text{-}g})!} \frac{(N-\ell_{f\underline{x}}-\ell_{g\underline{x}\text{-}f\underline{x}})!}{(N-\ell_{f\underline{x}})!} \frac{(N-\ell_{g\underline{x}}-\ell_{f\underline{x}\text{-}g\underline{x}})!}{(N-\ell_{g\underline{x}})!}}$$

(Note that all those factorials have nonnegative arguments since $|\operatorname{dom} f \cup \operatorname{dom} g \cup \underline{x}| \leq |D| \leq N$.) Since $|S \cup T| = |S| + |T \setminus S|$ for sets S, T, we can rewrite this to:

$$\sqrt{\frac{\cancel{(N-\ell_{f\underline{x}})!}}{(N-\ell_{fg})!} \frac{\cancel{(N-\ell_{g\underline{x}})!}}{(N-\ell_{fg})!} \frac{(N-\ell_{fg\underline{x}})!}{\cancel{(N-\ell_{f\underline{x}})!}} \frac{(N-\ell_{fg\underline{x}})!}{\cancel{(N-\ell_{g\underline{x}})!}}} = \frac{(N-\ell_{fg\underline{x}})!}{(N-\ell_{fg})!} = \frac{(N-\ell_{fg}-\ell_{\underline{x}\text{-}fg})!}{(N-\ell_{fg})!} = \frac{1}{(N-\ell_{fg})_{\ell_{\underline{x}\text{-}fg}}}.$$

Thus the summand in (38) is $1/(N-\ell_{fg})_{\ell_{\underline{x}\text{-}fg}}$. Thus (38) can be restated as $Z/(N-\ell_{fg})_{\ell_{\underline{x}\text{-}fg}} \leq c$ where Z is the number of $\underline{z}$ that satisfy $(*)$. Since $|\operatorname{dom} f| \leq \ell$ by assumption, and $|\underline{x}| \leq k$, we always have $f \cup (\underline{x} \mapsto \underline{z}) \in \{f : |\operatorname{dom} f| \leq \ell + k\}$. Hence $f \cup (\underline{x} \mapsto \underline{z}) \notin B_{\underline{x},e} \cap \{f : |\operatorname{dom} f| \leq \ell + k\}$ iff $f \cup (\underline{x} \mapsto \underline{z}) \notin B_{\underline{x},e}$. Analogously for $g \cup (\underline{x} \mapsto \underline{z})$. Thus the set of $\underline{z}$ that satisfy $(*)$ is the set in the numerator in (18). Hence (18) is equivalent to $Z/(N-\ell_{fg})_{\ell_{\underline{x}\text{-}fg}} \leq c$ which in turn is equivalent to (38). Since we assumed (18), (38) holds and Theorem 24 is applicable.

Furthermore, note that for our choices of $\mathbf{X}, M_{\underline{x},e}, V_{\underline{x},e,\underline{z}}$, the operator U defined in Theorem 24 is equal to $I_E \otimes U_{query,k}$. And $\mathbf{A}, \mathbf{B}$ defined in Theorem 24 are the same as $\mathbf{A}, \mathbf{B}$ defined in the present theorem.

Thus Theorem 24 shows that

$$(I_E \otimes U_{query,k})\psi \overset{\varepsilon + \frac{2\sqrt{c}}{\sqrt{1-3d}}}{\approx} \mathbf{B}.$$

Since $d = \frac{1}{4}$, we have $\varepsilon + \frac{2\sqrt{c}}{\sqrt{1-3d}} = \varepsilon + 4\sqrt{c}$. The theorem follows. □

Theorem 14 (Non-oblivious random permutation query).

Proof of Theorem 14. Analogous to Theorem 13, except using Theorem 25 instead of Theorem 24. □

5.5 Single query case

We now show the theorems for the single query case (i.e., the non-parallel case) for random functions and permutations. These are straightforward corollaries of the corresponding theorems for parallel queries with $k = 1$.

Theorem 3 (Random function query, simple).

Proof of Theorem 3. This theorem is obtained as a special case of Theorem 6 with $k := 1$ (and thus we write x instead of $\underline{x}$). Then (after applying elementary simplifications) we get the present theorem, except that the bound (12) becomes:

> Assume that for all x, all e, and all compatible $f, g \in A_{x,e}$ with $|\operatorname{dom} f|, |\operatorname{dom} g| \leq \ell$:
>
> $$\frac{\Big|\{z \in R : (x \mapsto z) \circledcirc f, g \text{ and } f_{xz}, g_{xz} \notin B_{x,e}\}\Big|}{N^{|\{x\} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|}} \leq c$$

where $f_{xz} := f \cup (x \mapsto z)$, $g_{xz} := g \cup (x \mapsto z)$.
We then distinguish two cases:

- If $x \in \operatorname{dom} f \cup \operatorname{dom} g$: Then for z with $(x \mapsto z) \circledcirc f, g$, we have $f \cup (x \mapsto z) = f \in A_{x,e} \subseteq B_{x,e}$ or $g \cup (x \mapsto z) = g \in A_{x,e} \subseteq B_{x,e}$. (Recall that $A_{x,e} \subseteq B_{x,e}$ by assumption of the present theorem.) But this contradicts the last condition in the numerator. Hence the numerator is 0 and the inequality is trivially satisfied.
- If $x \notin \operatorname{dom} f \cup \operatorname{dom} g$: Then $(x \mapsto z) \circledcirc f, g$ holds trivially, and $f \cup (x \mapsto z) = f(x := z)$, $g \cup (x \mapsto z) = g(x := z)$, so the numerator is equal to the numerator in (6). And $|\{x\} \setminus \operatorname{dom} f \setminus \operatorname{dom} g| = |\{x\}| = 1$, so the denominator becomes N. Thus in this case, the inequality is identical to (6).

Thus the present theorem is a special case of Theorem 6. □

Theorem 4 (Non-oblivious random function query, simple case).

Proof of Theorem 4. Analogous to Theorem 3, except using Theorem 7 instead of Theorem 6. □

Theorem 11 (Random permutation query, simple).

Proof of Theorem 11. This theorem is obtained as a special case of Theorem 13 with $k := 1$ (and thus we write x instead of $\underline{x}$). Then (after applying elementary simplifications) we get the present theorem, except that the bound (18) becomes:

> Assume that for all x, all e, and all compatible $f, g \in A_{x,e}$ with $f \cup g$ injective and $|\operatorname{dom} f|, |\operatorname{dom} g| \leq \ell$:
>
> $$\frac{\Big|\{z \in R : (x \mapsto z) \circledcirc f, g \text{ and } f_{xz}, g_{xz} \text{ inj. and } f_{xz}, g_{xz} \notin B_{x,e}\}\Big|}{\Big(N - \big|\operatorname{dom} f \cup \operatorname{dom} g\big|\Big)_{|\{x\} \setminus \operatorname{dom} f \setminus \operatorname{dom} g|}} \leq c$$

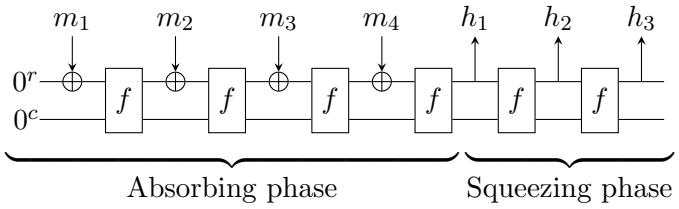


Figure 5: The sponge construction with a four block input $m_1 \| m_2 \| m_3 \| m_4$ and a three block output $h_1 \| h_2 \| h_3$.

where $f_{xz} := f \cup (x \mapsto z)$, $g_{xz} := g \cup (x \mapsto z)$.
We then distinguish two cases:

- If $x \in \operatorname{dom} f \cup \operatorname{dom} g$: Then for z with $(x \mapsto z) \diamond f, g$, we have $f \cup (x \mapsto z) = f \in A_{x,e} \subseteq B_{x,e}$ or $g \cup (x \mapsto z) = g \in A_{x,e} \subseteq B_{x,e}$. (Recall that $A_{x,e} \subseteq B_{x,e}$ by assumption of the present theorem.) But this contradicts the last condition in the numerator. Hence the numerator is 0 and the inequality is trivially satisfied.

- If $x \notin \operatorname{dom} f \cup \operatorname{dom} g$: Then $(x \mapsto z) \diamond f, g$ holds trivially, and f_{xz}, g_{xz} are injective iff $z \notin \operatorname{dom} f \cup \operatorname{dom} g$, and $f \cup (x \mapsto z) = f(x := z)$, $g \cup (x \mapsto z) = g(x := z)$, so the numerator is equal to the numerator in (16). And $|\{x\} \setminus \operatorname{dom} f \setminus \operatorname{dom} g| = |\{x\}| = 1$, so the denominator becomes $N - |\operatorname{dom} f \cup \operatorname{dom} g|$. Thus in this case, the inequality is identical to (16).

Thus the present theorem is a special case of Theorem 13. □

Theorem 12 (Non-oblivious random permutation query, simple case).

Proof of Theorem 12. Analogous to Theorem 11, except using Theorem 14 instead of Theorem 13. □

6 Collision-resistance of sponges

We will now give a quick overview over the sponge construction and then prove its collision-resistance (when based on invertible permutations).

6.1 The sponge construction

In this section, we review the *sponge construction* introduced by [Ber+07]. The sponge construction is parametrized by three integers: r (the *rate*), c (the *capacity*), and d (the *output length*). The sponge will hash messages m consisting of r-bit blocks and return a d-bit hash. The state of the sponge consists of $r + c$ bits. The sponge is further parametrized by a *round function* $f : \{0,1\}^{r+c} \to \{0,1\}^{r+c}$. Given a message m consisting of (padded) message blocks $m_1, m_2, \ldots$, the message is first absorbed into

the state as depicted in the *absorbing phase* in Figure 5. Then an output $h_1 h_2 \ldots$ is extracted as depicted in the *squeezing phase*. Finally, the hash consists of the first d bits of $h_1 h_2 \ldots$. (We do not make the padding function explicit, it suffices for our purposes that it is injective.)

More formally, for a round function f and an input $m = m_1 \ldots m_n$ with $m_i \in \{0,1\}^r$, we define the state $\mathcal{S}_f(m) \in \{0,1\}^{r+c}$ of the sponge by: $\mathcal{S}_f(\lambda) := (0,0)$ and $\mathcal{S}_f(m_1 \ldots m_n) := f\big(\mathcal{S}_f(m_1 \ldots m_{n-1}) \oplus (m_n, 0)\big)$. (We write (x,y) instead of $x \| y$ for r-bit/c-bit strings x, y for better readability.) Then the i-th output block h_i consists of the first r bits of $\mathcal{S}_f(m \| (0,0)^{i-1})$, and the final hash $\mathcal{S}_f^{hash}(m)$ consists of the first d bits of $h_1 h_2 \ldots$. (In particular, if $d \leq r$, as is the case for SHA3, then $\mathcal{S}_f^{hash}(m)$ is the first d bits of $\mathcal{S}_f(m)$.)

Depending on the scheme instantiating the sponge construction, f is an arbitrary function (e.g., in Gluon [Ber+12]) or a permutation (e.g., in SHA3 [NIS14], Quark [Aum+10], Photon [GPP11], and Spongent [Bog+13]). In the post-quantum setting, the security of the sponge has been analyzed in the case that f is a random function [Cza+18; Cza+20]. This automatically implies security when f is a non-invertible random permutation, i.e., if the adversary can query f but not f^{-1}, since a non-invertible random permutation is indistinguishable from a random function [Zha15a]. However, in the case that f is an invertible permutation (e.g., in SHA3), no post-quantum security results about the sponge construction were known prior to this work. (This is different from the classical case, where the sponge construction is known to be indifferentiable from a random oracle, and thus collision-resistant, pseudorandom, and more [Ber+08].)

In the following sections, we will show that the sponge construction is collision-resistant for invertible random permutations.

6.2 Invariant for collision-resistance

We now proceed to develop the invariant used to describe the fact that the adversary has not yet found a collision in the sponge. At this point, the analysis is purely combinatorial, the post-quantum aspect comes in the next section.

A reader who is interested only in the post-quantum aspects can skip this section and read only the last paragraph which summarizes everything that is needed for the next section.

Throughout this section, we assume that $d \leq r$, i.e., the output length is no larger than the rate. (The case $d > r$ is covered implicitly, see the comment after Theorem 28 in the next section.)

As in the previous section, $\mathcal{S}_f(m)$ is the state of the sponge on input m. If f is a partial function, we let $\mathcal{S}_f(m) := \perp$ if any of the f-invocations of the definition of $\mathcal{S}_f(m)$ was undefined.

We first translate the collision problem into a graph-theoretic problem: Given a partial injective function $f : \{0,1\}^{r+c} \hookrightarrow \{0,1\}^{r+c}$, we define the following directed *sponge graph*: Its nodes are bitstrings (a,b) with $a \in \{0,1\}^r$, $b \in \{0,1\}^c$. We call $(0,0)$ the *root*. The graph has three kinds of edges:

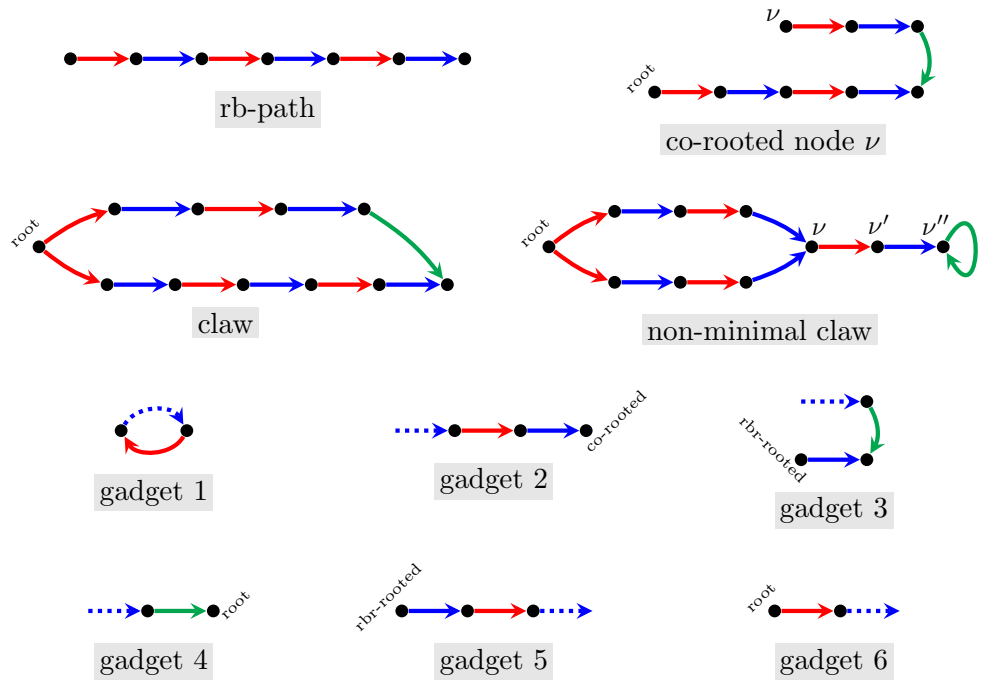


Figure 6: Various subgraphs related to sponges. This figure requires color.

- *Blue edges:* For any a, b, if $f(a, b) \neq \perp$, then there is a blue edge $(a, b) \to f(a, b)$.
- *Red edges:* For any a, a', b, there is a red edge $(a, b) \to (a', b)$.
- *Gree edges:* For any a, a', b, b', there is a green edge $(a, b) \to (a', b')$ if the first d bits of a and a' are equal.

An *rb-path* is a (possibly empty) path that starts with a red edge, alternates red and blue edges, and ends with a blue edge (see Figure 6). An *rbr-path* is a path that starts with a red edge, alternates red and blue edges, and ends with a red edge (i.e., an rb-path with one additional red edge). A *rooted rb/rbr-path* is an rb/rbr-path starting at the root. An *rb/rbr-rooted node* is a node at the end of a rooted rb/rbr-path. A *co-rooted node* is a node ν such that there is an rb-path starting at ν whose end-node is connected by a green edge to an rb-rooted node (see Figure 6). Note that in all those definitions (and in the definition of a claw below), we do not assume that all edges are distinct. E.g., an rb-path can contain the same edge several times.

We make a two observations:

- For any message m with $\mathcal{S}_f(m) \neq \perp$, there is a corresponding rooted rb-path where the i-th red edge is $(a, b) \to (a \oplus m_i, b)$, and the blue edges are defined by f. No two messages m have the same corresponding rooted rb-path.
- If m, m' form a *collision relative to* f, i.e., if $m \neq m'$ and $\mathcal{S}_f^{hash}(m) = \mathcal{S}_f^{hash}(m') \neq \perp$, then the corresponding rooted rb-paths are different, and the end-nodes of the corresponding rooted rb-paths are connected by a green edge (the green edge means that the first d bits of the state are equal, thus $\mathcal{S}_f^{hash}(m) = \mathcal{S}_f^{hash}(m')$).
  Thus, if m, m' form a collision relative to f, then the graph has a *claw*, which we

define as: Two different rooted rb-paths whose end points are connected by a green edge. (Two examples of claws are shown in Figure 6.)
We also say "f has a claw/has no claw" to mean that the sponge graph corresponding to f has a claw/has no claw.

- A *minimal claw* is a claw so that no proper subset of the vertices forms a claw as well. In a minimal claw, the two rooted rb-paths do not have the same last blue edge. (Compare with the "non-miminal claw" in Figure 6.) We see this as follows: If in a claw C, the two paths have the same last blue edge $\nu' \to \nu''$, then the last red edge $\nu \to \nu'$ would be also the same (since f is injective). Then the there would be two different rooted rb-paths to ν, and there is a green edge $\nu \to \nu$ (by definition of green edges). Thus we have a smaller claw, namely C with ν, ν', ν'' removed. Hence C is not minimal.

In our proof of collision-resistance, we will therefore show that the adversary cannot produce a claw.

Fix a partial injective f with $|\operatorname{dom} f| \leq \ell$ such that the sponge graph G for f has no claw. Fix some node $x \notin \operatorname{dom} f$, and consider a node $z \notin \operatorname{im} f$ such that the sponge graph G' for $f(x := z)$ will have a claw. How can I choose z? Let me count the ways. G' has one additional blue edge $x \to z$. If G' has a claw (but G does not), then it has a minimal-claw containing $x \to z$. Then z must be a co-rooted node in G'. Thus one of the following cases occurs:

- There is an edge $z \to x$. (See gadget 1 in Figure 6, the new edge $x \to z$ is dashed.) Since there are only 2^r red edges going to x, there are at most 2^r possibilities for z.
- There is a path $z \to \nu' \to \nu''$ where ν'' is co-rooted, and the edges are red and blue, and $\nu' \neq x$. (See gadget 2 in Figure 6.) Here we can assume $\nu' \neq x$ because otherwise we would be in the previous case. Thus $\nu' \to \nu''$ is an edge in G. Since there are only ℓ blue edges in G, there are at most ℓ possibilities for ν'. And since there are only 2^r red edges going to any given node ν', there are at most $2^r\ell$ possibilities for z.
- There is a path $z \to \nu' \leftarrow \nu''$ where ν'' is rbr-rooted and the edges are green and blue. (See gadget 3 in Figure 6.) Since we assumed that $x \to z$ is part of a *minimal* claw, we have that the blue edge is not $x \to z$. (See the discussion after the definition of minimal claws.) Thus that edge is part of G. Since there are only ℓ blue edges in G, there are at most ℓ possibilities for ν'. And since there are only 2^{r+c-d} green edges going to any given node ν', there are at most $2^{r+c-d}\ell$ possibilities for z.
- There is a green edge from z to the root. (See gadget 4 in Figure 6.) Since there are only 2^{r+c-d} green edges going to the root, there are at most 2^{r+c-d} possibilities for z.

Thus for every $f : \{0,1\}^{r+c} \hookrightarrow \{0,1\}^{r+c}$ with $|\operatorname{dom} f| \leq \ell$ that has no claw, and every $x \notin \operatorname{dom} f$,

$$\Big|\{z \notin \operatorname{im} f : f(x := z) \text{ has a claw}\}\Big| \leq 2^{r+c-d}(\ell+1) + 2^r(\ell+1). \tag{39}$$

This tells us how hard it is to get a claw using an f-query. Next we analyze the how hard

it is to get a claw using an f^{-1}-query.

Fix a partial injective f with $|\mathrm{dom}\, f| \leq \ell$ such that the sponge graph G for f has no claw. Fix some node $x \notin \mathrm{im}\, f$. We will bound the number of $z \notin \mathrm{dom}\, f$ such that the sponge graph G' for $f(z := x)$ has a claw. G' has one additional blue edge $z \to x$. If G' has a claw (but G does not), then z must be an rbr-rooted node in G. (If there are several occurrences of $z \to x$ in the claw, we consider one closest to the root.) Thus one of the following cases occurs:

- There is a path $\nu'' \to \nu' \to z$ in G where ν'' is rbr-rooted, and the edges are blue and red. (See gadget 5 in Figure 6.) Since there are only ℓ blue edges in G, there are at most ℓ possibilities for ν'. And since there are only 2^r red edges leaving any given node ν', there are at most $2^r\ell$ possibilities for z.
- There is a red edge from the root to z. (See gadget 6 in Figure 6.) Since there are only 2^r red edges leaving the root, there are at most 2^r possibilities for z.

Thus for every $f : \{0,1\}^{r+c} \hookrightarrow\!\!\!\!\!\!+\;\; \{0,1\}^{r+c}$ with $|\mathrm{dom}\, f| \leq \ell$ that has no claw, and every $x \notin \mathrm{im}\, f$,

$$\Big|\{z \notin \mathrm{dom}\, f :\ f(z := x) \text{ has a claw}\}\Big| \leq 2^r(\ell + 1).$$

We say f has a *co-claw* iff f^{-1} has a claw. Then the preceding bound immediately implies (by substituting f by f^{-1}): For every injective $f : \{0,1\}^{r+c} \hookrightarrow\!\!\!\!\!\!+\;\; \{0,1\}^{r+c}$ with $|\mathrm{dom}\, f| \leq \ell$ that has no co-claw, and every $x \notin \mathrm{dom}\, f$,

$$\Big|\{z \notin \mathrm{im}\, f :\ f(x := z) \text{ has a co-claw}\}\Big| \leq 2^r(\ell + 1). \tag{40}$$

(Not having a co-claw is the invariant whose preservation we show when the oracle is inverted using Invert.)

For the remainder of the analysis (next section), we only need to remember: If m, m' form a collision relative to f (i.e., their sponge-hashes are equal), then f has a claw. And f has a co-claw iff f^{-1} has a claw. And bounds (39) and (40). The definition of claw, and the graph-theoretic notions above were only needed for the derivation of these facts.

6.3 Proof of collision-resistance

We are now ready to show the collision-resistance of the sponge construction:

Theorem 28 (Collision-resistance of sponges) *Let c, r, d be the parameters of the sponge construction. Assume $d \leq r$. Let $f : \{0,1\}^{r+c} \hookrightarrow \{0,1\}^{r+c}$ be uniformly random. Let A be a quantum algorithm. Assume each of the messages returned by A contains $\leq b$ blocks of r bits (not necessarily both of the same length, and the empty message is allowed). Assume that A makes q_+ queries to f and q_- queries to f^{-1}. (And let $q_{tot} := q_+ + q_- + 2b$.) Assume $q_{tot} \leq 2^{r+c} - 15$.*

Then

$$\begin{aligned} p &:= \Pr\left[m \neq m' \wedge \mathcal{S}_f^{hash}(m) = \mathcal{S}_f^{hash}(m') : (m, m') \leftarrow A^{f, f^{-1}}()\right] \\ &\leq \frac{16 q_{tot}}{1 - q_{tot} 2^{-r-c+1}} \left((q_+ + 2b)\sqrt{2^{-d} + 2^{-c}} + q_- \sqrt{2^{-c}} \right)^2 \\ &\overset{(*)}{\leq} \frac{34 q_{tot}^3}{2^{\min\{c,d\}}}. \end{aligned}$$

where $(*)$ *only holds under the additional condition* $q_{tot} \leq 2^{r+c}/34$.

In other words, to find a collision, we need $\Omega\big(\min\{2^{d/3}, 2^{c/3}\}\big)$ queries.

Note that if $d \leq r$ (as is the case, e.g., with SHA3), this bound is essentially tight: If $d \leq c$, we can find a collision by running the BBHT algorithm [Boy+98] on $\mathcal{S}_f^{hash}$. This takes $\Omega(2^{d/3})$ queries. And if $c \leq d$, we can find an inner collision (i.e., $m \neq m'$ such that the last c bits of $\mathcal{S}_f(m), \mathcal{S}_f(m')$ are equal) using BBHT in $\Omega(2^{c/3})$ queries. And from an inner collision, we can trivially construct a collision by extending m, m' by one block each.

If $d > r$, the theorem as stated does not apply. However, since a collision with output length d implies a collision with output length r, the theorem bounds the success probability in the $d > r$ case by $\frac{34 q_{tot}^3}{2^{\min\{c,r\}}}$. However, this bound may not be tight.

We now proceed to proving Theorem 28.

Preparations. As in the various examples in this paper, we first transform the game from Theorem 28 into a suitable form:

We replace A by a unitary adversary, represented by a unitary U_A acting on a state register E, and on query registers X, Y (consisting of $r + c$ qubits each). We replace oracle access to f and f^{-1} by access to the invertible standard oracle (Definition 16). That is, queries to f are replaced by applying U_{query} on X, Y, H. And queries to f^{-1} are replaced by $U_{inverse} := (I_X \otimes I_Y \otimes \text{Invert}) \cdot U_{query} \cdot (I_X \otimes I_Y \otimes \text{Invert})$. The last invocation of the adversary uses a different unitary $\hat{U}_A$ instead that stores the adversary's output m, m' in registers M, M'. (I.e., M, M' both have an orthonormal basis $|m\rangle$ where m ranges over messages of $\leq b$ blocks.) The result is depicted in Figure 7 in the first half ("adversary invocation") of the circuit.

Next, we apply $C_{2sponge}$ to M, M', H. $C_{2sponge}$ is a quantum circuit that computes the sponge hashes of M, M'. That is,

$$C_{2sponge}|m\rangle_M|m'\rangle_{M'}|h\rangle_H = |m\rangle_M|m'\rangle_{M'}|\mathcal{S}_h^{hash}(m)\rangle_S|\mathcal{S}_h^{hash}(m')\rangle_S|z\rangle \tag{41}$$

for some z (that may depend on m, m', h). We do not specify a concrete circuit, but it is easy to see that $C_{2sponge}$ can be implemented using at most $2b$ queries to U_{query} (since both m, m' are at most b blocks long). Furthermore, $C_{2sponge}$ can be implemented as a classicalish circuit (as defined in Section 4.6; basically this means that the circuit only uses CNOT, Toffoli, X and swap gates and fresh auxiliary qubits).

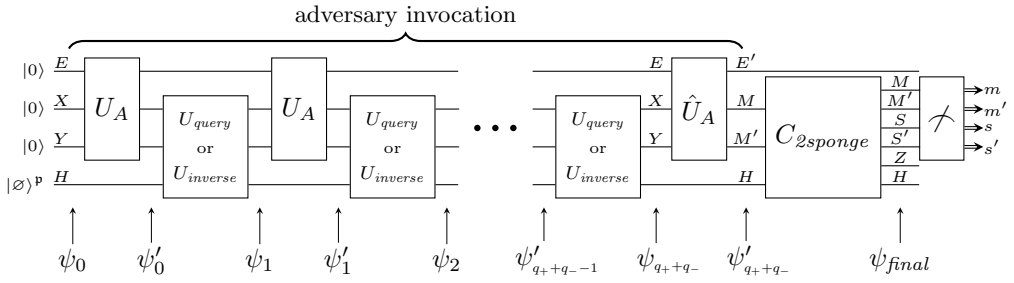


Figure 7: Circuit used in the collision-resistance proof for the sponge construction.

Finally, we measure registers M, M', S, S' in the computational basis, giving m, m', s, s'. Then m, m' are the outputs of the adversary, and s, s' are the sponge hashes of m, m'. Thus the adversary's success probability p is:

$$p = \Pr[m \neq m' \wedge s = s']. \tag{42}$$

Maintaining the invariant. Let

$$I_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \operatorname{span}\{|f\rangle^{\mathfrak{p}} : f \text{ has no claw, } |\operatorname{dom} f| \leq j\}_f.$$

Then by definition, ψ_0 satisfies $\psi_0 \in I_0$, i.e., $\psi_0 \overset{0}{\approx} I_0$. (See Figure 7 for the definition of ψ_j, ψ'_j, etc.) And since U_A does not operate on H, we have:

$$\psi_j \overset{\varepsilon}{\approx} I_j \implies \psi'_j \overset{\varepsilon}{\approx} I_j \qquad \text{for } j = 0, \ldots, q_+ + q_- - 1.$$

For the j-th query (i.e., the invariant preservation between ψ'_{j-1} and ψ_j, with $j = 1, \ldots, q_+ + q_-$), we distinguish two cases:

- If the j-query is U_{query}, then we have

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \implies \psi_j = (I_E \otimes U_{query})\psi'_{j-1} \overset{\varepsilon+\delta_+}{\approx} I_j \qquad \text{for } \delta_+ := 4\sqrt{\frac{2^{r+c-d}q_{tot} + 2^r q_{tot}}{2^{r+c} - 2q_{tot}}}.$$

  This follows by Theorem 11 (with $\ell := j-1$, $A_{x,e}, B_{x,e} := \{f : f \text{ has no claw}\}$, $c := \frac{2^{r+c-d}j + 2^r j}{2^{r+c} - 2(j-1)}$) In this theorem, condition (16) becomes

$$\frac{\left|\{z : z \notin \operatorname{im} f, z \notin \operatorname{im} g, f(x := z), g(x := z) \text{ have a claw}\}\right|}{2^{r+c} - \left|\operatorname{dom} f \cup \operatorname{dom} g\right|} \leq \frac{2^{r+c-d}j + 2^r j}{2^{r+c} - 2(j-1)}$$

  for injective f, g with $|\operatorname{dom} f|, |\operatorname{dom} g| \leq j-1$ and $x \notin \operatorname{dom} f, \operatorname{dom} g$. This follows directly from (39) and $|\operatorname{dom} f|, |\operatorname{dom} g| \leq j-1$. And we have $4\sqrt{c} \leq \delta_+$.

- If the j-th query is $U_{inverse}$, by Lemma 17 (ii) we have

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \implies (I_{EXY} \otimes \mathrm{Invert})\psi'_{j-1} \overset{\varepsilon}{\approx} I^-_{j-1}$$

where

$$I^-_j := \mathcal{H}_E \otimes \mathcal{H}_X \otimes \mathcal{H}_Y \otimes \mathrm{span}\{|f\rangle^{\mathfrak{p}} : f \text{ has no co-claw, } |\mathrm{dom}\, f| \le j\}_f.$$

And furthermore:

$$(I_{EXY} \otimes \mathrm{Invert})\psi'_{j-1} \overset{\varepsilon}{\approx} I^-_{j-1} \implies (I_E \otimes U_{query})(I_{EXY} \otimes \mathrm{Invert})\psi'_{j-1} \overset{\varepsilon+\delta_-}{\approx} I^-_j$$
$$\text{for } \delta_- := 4\sqrt{\frac{2^r q_{tot}}{2^{r+c} - 2q_{tot}}}.$$

This follows by Theorem 11 (with $\ell := j-1$, $A_{x,e}, B_{x,e} := \{f : f^{-1} \text{ has no co-claw}\}$, $c := \frac{2^r j}{2^{r+c}-2(j-1)}$). In this theorem, condition (16) becomes

$$\frac{\Big|\big\{z : z \notin \mathrm{im}\, f, \mathrm{im}\, g, f(x := z), g(x := z) \text{ have a co-claw}\big\}\Big|}{2^{r+c} - \big|\mathrm{dom}\, f \cup \mathrm{dom}\, g\big|} \le \frac{2^r j}{2^{r+c} - 2(j-1)}$$

for injective f, g with $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \le j-1$ and $x \notin \mathrm{dom}\, f, \mathrm{dom}\, g$. This follows directly from (40) and $|\mathrm{dom}\, f|, |\mathrm{dom}\, g| \le j-1$. And we have $4\sqrt{c} \le \delta_-$.
And furthermore, by Lemma 17 (ii) again, we have

$$(I_E \otimes U_{query})(I_{EXY} \otimes \mathrm{Invert})\psi'_{j-1} \overset{\varepsilon}{\approx} I^-_j \implies \psi_j = (I_E \otimes U_{inverse})\psi'_{j-1} \overset{\varepsilon}{\approx} I_j$$

So altogether:

$$\psi'_{j-1} \overset{\varepsilon}{\approx} I_{j-1} \implies \psi_j \overset{\varepsilon+\delta_-}{\approx} I_j \qquad \text{for} \qquad j = 1, \ldots, q_+ + q_-.$$

Combining the implications we got so far, we get:

$$\psi_{q_+ + q_-} \overset{q_+\delta_+ + q_-\delta_-}{\approx} I_{q_+ + q_-}.$$

Since $\hat{U}_A$ does not operate on H, this implies

$$\begin{aligned}\psi'_{q_+ + q_-} &\overset{q_+\delta_+ + q_-\delta_-}{\approx} I'_{q_+ + q_-} \\ &:= \mathcal{H}_{E'} \otimes \mathcal{H}_M \otimes \mathcal{H}_{M'} \otimes \mathrm{span}\{|f\rangle^{\mathfrak{p}} : f \text{ has no claw, } |\mathrm{dom}\, f| \le q_+ + q_-\}_f \\ &= \mathcal{H}_{E'} \otimes \mathrm{span}\{|m\rangle_M |m'\rangle_{M'} |f\rangle^{\mathfrak{p}} : f \text{ has no claw, } |\mathrm{dom}\, f| \le q_+ + q_-\}_{mm'f} \quad (43)\end{aligned}$$

Finally, we apply Corollary 20 for the application of $C_{2sponge}$ in the end. By Corollary 20 (with $C := I_{E'} \otimes C_{2sponge}$, $q := 2b$, $\ell := q_+ + q_-$, $c := \frac{2^{r+c-d}q_{tot} + 2^r q_{tot}}{2^{r+c} - 2(q_{tot}-1)}$, $A := \{f : f \text{ has no claw}\}$, $D := MM'$, $\underline{R} := MM'SS'Z$), (43) implies

$$\psi_{final} = (I_{E'} \otimes C_{2sponge})\psi'_{q_+ + q_-} \overset{(q_+ + 2b)\delta_+ + q_-\delta_-}{\approx} \mathcal{H}_{E'} \otimes \mathrm{span}\Big\{|B_f(m,m')\rangle_{MM'SS'Z} |f\rangle^{\mathfrak{p}} :$$
$$B_f(m,m') \neq \perp,\ f \text{ has no claw, } |\mathrm{dom}\, f| \le q_{tot}\Big\}_{mm'f} =: I_{final}$$

where B_f is the function computed by $C_{2sponge}$ (cf. Section 4.6). For the application of Corollary 20, we need to show that (27) holds, i.e., for f, g with no claws and $\operatorname{dom} f, \operatorname{dom} g \leq q_{tot} - 1$,

$$\frac{\left|\left\{z : z \notin \operatorname{im} f, z \notin \operatorname{im} g, f(x := z), g(x := z) \text{ have a claw}\right\}\right|}{2^{r+c} - 2(q_{tot} - 1)} \leq \frac{2^{r+c-d} q_{tot} + 2^r q_{tot}}{2^{r+c} - 2(q_{tot} - 1)}$$

This follows directly from (39) and $|\operatorname{dom} f|, |\operatorname{dom} g| \leq q_{tot} - 1$. And we have $4\sqrt{c} \leq \delta_+$.

Concluding. The last step of the circuit in Figure 7 is a measurement of M, M', S, S' in the computational basis. Consider a state $\phi \in I_{final}$ (not just close to this subspace). Then, when measuring registers M, M', S, S', we get m, m', s, s' such that there exists an $f, \hat{m}, \hat{m}'$ so that f has no claw, and $mm'ss'$ are the first four components of $B_f(\hat{m}, \hat{m}') \neq \bot$. Since B_h is the function computed by $C_{2sponge}$, by definition of $C_{2sponge}$ (see (41)), we have that $B_h(\hat{m}\hat{m}') = (\hat{m}, \hat{m}', \mathcal{S}_h^{hash}(m), \mathcal{S}_h^{hash}(m'), z)$. Thus $s = \mathcal{S}_f^{hash}(m) \neq \bot$, $s' = \mathcal{S}_f^{hash}(m') \neq \bot$. Since f has no claw, m, m' do not form a collision relative to f (see the last paragraph of Section 6.2), i.e., $\neg\big(m \neq m' \wedge s = s'\big)$.

Summarizing, when measuring M, M', S, S' on a state $\phi \in I_{final}$, we get $m \neq m' \wedge s = s'$ with probability 0.

Since $\psi_{final} \overset{(q_+ + 2b)\delta_+ + q_-\delta_-}{\approx} I_{final}$, measuring M, M', S, S' in the circuit gives $m \neq m' \wedge s = s'$ with probability $\leq \big((q_+ + 2b)\delta_+ + q_-\delta_-\big)^2$. By (42), this probability is p. Thus

$$\begin{aligned} p \leq \big((q_+ + 2b)\delta_+ + q_-\delta_-\big)^2 &= \frac{16 q_{tot}}{1 - q_{tot} 2^{-r-c+1}} \Big((q_+ + 2b)\sqrt{2^{-d} + 2^{-c}} + q_-\sqrt{2^{-c}}\Big)^2 \\ &\overset{(*)}{\leq} \frac{34 q_{tot}^3}{2^{\min\{c,d\}}}. \end{aligned}$$

where $(*)$ only holds when $q_{tot} \leq 2^{r+c}/34$. This shows Theorem 28.

7 Conclusion & open questions

We extended Zhandry's compressed oracle technique to invertible permutations. This gives us the first proof technique for quantum hardness results for invertible random permutations. We have analyzed the collision-resistance of the sponge construction using this technique. (And as a consequence, we have the first post-quantum security proof for SHA3.) However, the technique has still some limitations: it is not clear how to *efficiently* simulate a random permutation, nor how to measure what oracle positions have been queried or how to otherwise perform actions that depend on the oracle queries performed so far. In particular, we do not know how to make indifferentiability proofs (or similar proofs involving simulators that "look inside" the random oracle). Overcoming these hurdles is an interesting open problem.

Our technique also allows us to reason about ideal ciphers (since an ideal cipher can be seen as a family of independent invertible random permutations). Interesting future

work would be to make this explicit, e.g., by applying the technique to the analysis of schemes involving block ciphers in the ideal cipher model.

Concerning the sponge construction, open questions are properties such as the collapsing property, pseudorandomness, (second-)preimage resistance (this follows from collision-resistance, but maybe a direct proof gives better parameters), the fact that the sponge construction is a secure MAC, and of course the indifferentiability of the sponge construction. (All in the setting with an invertible round function.)

Acknowledgments. We thank Aleksander Belov for valuable discussions on the two-sided zero search problem. This work was supported by the ERC consolidator grant CerQuS, by the Estonian Research Council grant PRG946, and by the Estonian Centre of Exellence in IT (EXCITE) funded by ERDF.

List of Theorems

Symbol index

$U_{query,k}$	Unitary: Querying the oracle (k-parallel)	7, 9
$\operatorname{dom} f$	Domain of (partial) function f	6
$\operatorname{im} f$	Image of (partial) function f	6
$D \rightharpoonup R$	Partial functions from D to R	6
$D \rightharpoonup\!\!\!\!\cdot\; R$	Partial injective functions from D to R	6
$D \hookrightarrow R$	Injective functions from D to R	6
$D \to R$	Functions from D to R	6
D	Domain of the random oracle	7
0	"Neutral element" used to implement quantum queries	7
Decomp_1	Decompression operation in Zhandry's compressed oracle, one register	10
Invert	Inverts the random permutation oracle	33
$\tilde{R}$	Range of the output register for oracle queries	
M	Size of the random oracle domain ($M := \lvert D\rvert$)	7
N	Size of the random oracle range ($N := \lvert R\rvert$)	7
R	Range of the random oracle	7

Keyword Index

References

[AHU19] A. Ambainis, M. Hamburg, and D. Unruh. "Quantum Security Proofs Using Semi-classical Oracles". In: *Crypto 2019*. Springer, 2019, pp. 269–295.

[Amb02] A. Ambainis. "Quantum Lower Bounds by Quantum Arguments". In: *J. Comput. Syst. Sci.* 64.4 (June 2002), pp. 750–767. ISSN: 0022-0000. DOI: 10.1006/jcss.2002.1826.

[ARU14] A. Ambainis, A. Rosmanis, and D. Unruh. "Quantum Attacks on Classical Proof Systems: The Hardness of Quantum Rewinding". In: *55th FOCS*. Philadelphia, PA, USA: IEEE Computer Society Press, 2014, pp. 474–483. DOI: 10.1109/FOCS.2014.57.

[Aum+10] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia. "Quark: A Lightweight Hash". In: *CHES 2010*. Vol. 6225. LNCS. Springer, 2010, pp. 1–15. ISBN: 978-3-642-15030-2. DOI: 10.1007/978-3-642-15031-9_1.

[Bea+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. "Quantum Lower Bounds by Polynomials". In: *J. ACM* 48.4 (July 2001), pp. 778–797. ISSN: 0004-5411. DOI: 10.1145/502090.502097.

[Ber+07] G. Bertoni, J. Daemen, M. Peeters, and G. van Assche. *Sponge functions*. Ecrypt Hash Workshop, http://sponge.noekeon.org/SpongeFunctions.pdf. May 2007.

[Ber+08] G. Bertoni, J. Daemen, M. Peeters, and G. van Assche. "On the Indifferentiability of the Sponge Construction". In: *Eurocrypt 2008*. Vol. 4965. LNCS. Berlin, Heidelberg: Springer, 2008, pp. 181–197. ISBN: 978-3-540-78966-6 978-3-540-78967-3. DOI: 10.1007/978-3-540-78967-3_11.

[Ber+12] T. P. Berger, J. D'Hayer, K. Marquet, M. Minier, and G. Thomas. "The GLUON Family: A Lightweight Hash Function Family Based on FCSRs". In: *Africacrypt 2012*. Ed. by A. Mitrokotsa and S. Vaudenay. Berlin, Heidelberg: Springer, 2012, pp. 306–323. ISBN: 978-3-642-31410-0. DOI: 10.1007/978-3-642-31410-0_19.

[BES18] M. Balogh, E. Eaton, and F. Song. "Quantum Collision-Finding in Non-uniform Random Functions". In: *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*. Ed. by T. Lange and R. Steinwandt. Springer, Heidelberg, Germany, 2018, pp. 467–486. DOI: 10.1007/978-3-319-79063-3_22.

[Bog+13] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede. "SPONGENT: The Design Space of Lightweight Cryptographic Hashing". In: *IEEE Transactions on Computers* 62.10 (2013), pp. 2041–2053. ISSN: 0018-9340. DOI: 10.1109/TC.2012.196.

[Bon+11] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. "Random Oracles in a Quantum World". In: *ASIACRYPT 2011*. Ed. by D. H. Lee and X. Wang. Vol. 7073. LNCS. Seoul, South Korea: Springer, Heidelberg, Germany, 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0_3.

[Boy+98] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. "Tight Bounds on Quantum Searching". In: *Fortschritte der Physik* 46.4-5 (1998). Eprint is arXiv:quant-ph/9605034, pp. 493–505. ISSN: 1521-3978. DOI: 10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P.

[BR93] M. Bellare and P. Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *CCS '93*. ACM, 1993, pp. 62–73.

[CGH98] R. Canetti, O. Goldreich, and S. Halevi. "The Random Oracle Methodology, Revisited". In: *STOC 1998*. ACM, 1998, pp. 209–218.

[CHS19] J. Czajkowski, A. Hülsing, and C. Schaffner. "Quantum Indistinguishability of Random Sponges". In: *Advances in Cryptology – CRYPTO 2019*. Ed. by A. Boldyreva and D. Micciancio. Cham: Springer International Publishing, 2019, pp. 296–325. ISBN: 978-3-030-26951-7.

[Chu+20] K.-M. Chung, S. Fehr, Y. Huang, and T. Liao. *On the Compressed-Oracle Technique, and Post-Quantum Security of Proofs of Sequential Work*. arXiv:2010.11658 [quant-ph]. 2020.

[Cza+18] J. Czajkowski, L. Groot Bruinderink, A. Hülsing, C. Schaffner, and D. Unruh. "Post-quantum security of the sponge construction". In: *PQCrypto 2018*. Vol. 10786. LNCS. Springer, 2018, pp. 185–204.

[Cza+20] J. Czajkowski, C. Majenz, C. Schaffner, and S. Zur. *Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability*. arXiv:1904.11477 [quant-ph]. 2020.

[EU18] E. E. Ebrahimi and D. Unruh. "Quantum collision-resistance of non-uniformly distributed functions: upper and lower bounds". In: *Quantum Information & Computation* 18.15&16 (2018), pp. 1332–1349. DOI: 10.26421/QIC18.15-16.

[GPP11] J. Guo, T. Peyrin, and A. Poschmann. "The PHOTON Family of Lightweight Hash Functions". In: *Crypto 2011*. Springer, 2011, pp. 222–239. ISBN: 978-3-642-22792-9. DOI: 10.1007/978-3-642-22792-9_13.

[Gro96] L. K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: *STOC 1006*. 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[HRS16] A. Hülsing, J. Rijneveld, and F. Song. "Mitigating Multi-target Attacks in Hash-Based Signatures". In: *PKC 2016 , Part I*. Ed. by C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang. Vol. 9614. LNCS. Taipei, Taiwan: Springer, Heidelberg, Germany, 2016, pp. 387–416. DOI: 10.1007/978-3-662-49384-7_15.

[NIS14] NIST. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Draft FIPS 202. Available at http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf. 2014.

[TTU16] E. E. Targhi, G. N. Tabia, and D. Unruh. "Quantum Collision-Resistance of Non-uniformly Distributed Functions". In: *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*. Ed. by T. Takagi. Springer, Heidelberg, Germany, 2016, pp. 79–85. DOI: 10.1007/978-3-319-29360-8_6.

[Unr15] D. Unruh. "Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model". In: *EUROCRYPT 2015, Part II*. Ed. by E. Oswald and M. Fischlin. Vol. 9057. LNCS. Sofia, Bulgaria: Springer, Heidelberg, Germany, 2015, pp. 755–784. DOI: 10.1007/978-3-662-46803-6_25.

[Unr16] D. Unruh. “Computationally Binding Quantum Commitments”. In: *EUROCRYPT 2016, Part II*. Ed. by M. Fischlin and J.-S. Coron. Vol. 9666. LNCS. Vienna, Austria: Springer, Heidelberg, Germany, 2016, pp. 497–527. DOI: 10.1007/978-3-662-49896-5_18.

[Zha12a] M. Zhandry. “How to Construct Quantum Random Functions”. In: *53rd FOCS*. New Brunswick, NJ, USA: IEEE Computer Society Press, 2012, pp. 679–687. DOI: 10.1109/FOCS.2012.37.

[Zha12b] M. Zhandry. “How to Construct Quantum Random Functions”. In: *FOCS 2013*. IEEE, 2012, pp. 679–687. DOI: 10.1109/FOCS.2012.37.

[Zha12c] M. Zhandry. “Secure Identity-Based Encryption in the Quantum Random Oracle Model”. In: *CRYPTO 2012*. Ed. by R. Safavi-Naini and R. Canetti. Vol. 7417. LNCS. Santa Barbara, CA, USA: Springer, Heidelberg, Germany, 2012, pp. 758–775. DOI: 10.1007/978-3-642-32009-5_44.

[Zha12d] M. Zhandry. “Secure Identity-Based Encryption in the Quantum Random Oracle Model”. In: *Crypto 2012*. Vol. 7417. LNCS. Springer, 2012, pp. 758–775. ISBN: 978-3-642-32008-8. DOI: 10.1007/978-3-642-32009-5_44.

[Zha15a] M. Zhandry. “A note on the quantum collision and set equality problems”. In: *Quantum Inf. Comput.* 15.7&8 (2015), pp. 557–567. DOI: 10.26421/QIC15.7-8.

[Zha15b] M. Zhandry. “A Note on the Quantum Collision and Set Equality Problems”. In: *Quantum Information and Computation* 15.7&8 (2015).

[Zha16] M. Zhandry. *A Note on Quantum-Secure PRPs*. Cryptology ePrint Archive, Report 2016/1076. http://eprint.iacr.org/2016/1076. 2016.

[Zha19] M. Zhandry. “How to Record Quantum Queries, and Applications to Quantum Indifferentiability”. In: *Crypto 2019*. Vol. 11693. LNCS. Springer, 2019, pp. 239–268. DOI: 10.1007/978-3-030-26951-7_9.