Tight Quantum Indifferentiability of a Rate-1/3 Compression Function

Jan Czajkowski*

Weizmann Institute of Science

September 22, 2021

Abstract

We prove classical and quantum indifferentiability of a rate-1/3 compression function introduced by Shrimpton and Stam (ICALP '08). This construction was one of the first constructions based on three random functions that achieved optimal collision-resistance. We also prove that our result is tight, we define a classical and a quantum attackers that match the indifferentiability security level. Our tight indifferentiability results provide a negative result on the optimality of security of the construction by Shrimpton and Stam, security level of the strong indifferentiability notion is below that of collision-resistance.

To arrive at these results, we generalize the results of Czajkowski, Majenz, Schaffner, and Zur (arXiv '19). Our generalization allows to analyze quantum security of constructions based on multiple independent random functions, something not possible before.

*jan.czajkowski@weizmann.ac.il

Contents

1 Introduction **3**

2 Preliminaries **4**
2.1 Indifferentiability 4
2.2 Game Playing Proofs for Indifferentiability 5
2.2.1 Compressed Oracles 6
2.2.2 Punctured Oracles and Relations 7
2.2.3 One-way To Hiding Lemma 8
2.3 The Rate-1/3 Compression Function 8

3 Classical Indifferentiability of Rate-1/3 **9**

4 Bound on $\mathbb{P}$[Find] **12**
4.1 Proof of Lem. 9 15
4.1.1 Overview 15
4.1.2 Introduction 15
4.1.3 The good state 16
4.1.4 Final Bound 17
4.2 Simplification of the Bound 18
4.3 Concrete Bound for the Rate-1/3 Relation 20

5 Tight Indifferentiability of Rate-1/3 **20**
5.1 Quantum Indifferentiability Proof 21
5.2 Indifferentiability Attacks 23
5.2.1 Classical Attack 23
5.2.2 Quantum Attack 24

6 Conclusions **26**

7 Acknowledgments **26**

References **26**

A Additional Details on the Proof of Lem. 9 **29**
A.1 $|\Psi_{i-1}^{\text{Good}}\rangle$ after a query 29
A.2 Bound on $\varepsilon_{\text{step}}(j)$ 36
A.3 Bound on $\varepsilon_{\text{Find}}(i)$ 42

Symbol Index **44**

1 Introduction

In recent years there is a great effort to build a new type of a computational machine: a *quantum computer*. From the perspective of cryptography, what a quantum computer brings with itself is, among other things, a new type of adversary. Such that can run quantum algorithms like that of Shor's factoring algorithm [Sho94], Grover's search algorithm [Gro96], or Simon's algorithm [Sim97]. It has been shown that all of these algorithms can be used to attack classical cryptosystems with more (Shor's algorithm can be used to break RSA, and Simon's o break CBC MAC [Kap+16; SS16]) or less (Grover's algorithm gives a square root speedup in generic key-search attacks [LM17]) devastating effects. Such attacks are the main motivation for this work and the field of *post-quantum* cryptography.

In this work, we prove tight (quantum) security of a construction that offers optimal collision-resistance. In the interesting line of work [MT07; SS08; RS08; MP12; ABR21] the authors considered the problem of designing a $2n$–n compression functions[1] out of several primitives that map n-bits to n-bits. The objective of their work was to maintain collision-resistance of just a single primitive, that is to the level of $O(2^{n/2})$ queries. In this paper we focus on the construction from [SS08] and prove that it is (quantumly) *indifferentiable*.

Indifferentiability is a strong security notion especially suitable to hash functions and compression functions constructed out of "smaller" primitives [MRH04]. Unlike in the Random Oracle Model [BR93] (where we assume the hash function to be random), only assume the internal primitives[2] to be uniformly random functions. In recent works [Zha19; Cza+19] it has also been shown that it is possible to prove quantum indifferentiability of classical constructions. We note that in this model the adversary makes *superposition* queries to the primitives (and the construction). This Quantum-accessible Random Oracle Model has been introduced in [Bon+11].

Our results. We prove that the construction from [SS08], that we denote by Rate-1/3, is classically and quantumly indifferentiable. Moreover we show attacks with matching (quantum) query complexity, proving that our results are tight. These results imply a negative result on Rate-1/3: From the perspective of the strong notion of indifferentiability (in particular stronger than collision-resistance) security level of Rate-1/3 is no longer optimal, i.e. it is secure only up to $O(2^{n/3})$ classical and $O(2^{n/4})$ quantum queries.

The main technical result of our paper is expanding the general technique of [Cza+19] to capture situations when the adversary is interacting with multiple random functions (like in Rate-1/3). The key ingredients of our proof of quantum security are the One-way To Hiding (O2H) lemma [Unr14; AHU19] and the compressed oracle technique [Zha19; Cza+19]. To derive any concrete result using these techniques one needs a bound on the probability that after any query the (quantum) database the adversary is interacting with does not contain certain input-output pairs. Using the proof of Lemma 13 from Appendix C from [Cza+19] as a blueprint, we generalize their bound on is probability to include multiple databases. Inspecting the two proofs of Lemma 13 from[Cza+19], the main proof and the one in Appendix C, we see that there are two ways of proving this crucial bound. We decided to generalize the direct proof from the appendix as it is not immediately obvious how to include a discussion of multiple databases in the technique from [Chu+20a].

Organization. In Sec. 2, we introduce our notation and all the necessary details of

[1] That is compression functions that map $2n$-bits to n-bits.

[2] Such as compression functions that are called by the cryptographic construction.

the key concepts we use in this paper. In Sec. 3 we prove classical indifferentiability of Rate-1/3, this is our first result. The proof of classical indifferentiability also provides motivation and intuition for our main technical result. In Sec. 4 we present the general bound on the probability of Find. In Sec. 5 we use the new bound and the classical proof to prove quantum indifferentiability of Rate-1/3. Moreover we present a classical and a quantum distinguisher attacking indifferentiability of Rate-1/3. In Sec. 6 we discuss some open problems.

2 Preliminaries

By $x \leftarrow \mathsf{A}$ we denote that x is an output of a randomized algorithm A. By $x \xleftarrow{\$} \mathcal{X}$ we denote that x is sampled uniformly at random from a finite set $\mathcal{X}$. For bitstrings $x, y \in \{0,1\}^n$ the bitwise XOR is denoted by $x \oplus y$. By A^{R} and $\mathsf{A}[\mathsf{R}]$ we denote that A has oracle access to R.

We assume basic knowledge of quantum computing, if any concept that we do not explain in detail needs to be clarified, we refer the reader to [NC10] or [Wol11]. Quantum algorithms act on quantum states from finite Hilbert spaces. Quantum oracle algorithms operate by intertwining arbitrary unitary operations with unitary oracle calls (which we describe in detail in Sec. 2.2.1). By U^F we denote applying the quantum operator U to register F.

When referring to quantum registers, i.e., subsystems of the quantum state of the whole system, we use two types of notation. Let us say we have a quantum register Q holding elements of the Hilbert space $\mathcal{H} = \bigotimes_{x\in\mathcal{S}}(\mathcal{H}_x^X \otimes \mathcal{H}_x^Y)$. Our Hilbert space is a tensor product of $|\mathcal{S}|$ pairs of $\mathcal{H}_x^X \otimes \mathcal{H}_x^Y$, we assume that there is a natural order in $\mathcal{S}$ and the tensor product is applied in this order. When we want to refer to the i-th register from the left, we write Q_i. When we want to access registers holding all X-parts of Q we write Q^X, with a superscript Q_i^X we access just the X-part of Q_i. Sometimes, however, we want to access the register corresponding to a particular $x \in \mathcal{S}$, then we write $Q(x)$ and similarly the superscript marks the part of $\mathcal{H}_x^X \otimes \mathcal{H}_x^Y$ we want to specifically discuss. We use the same notation with sets of pairs.

Let us define the *Quantum Fourier Transform* (QFT), a unitary change of basis that we will make use of. For $N \in \mathbb{N}_{>0}$ and $x, \xi \in [N] = \{0, 1, \ldots, N-1\}$ the transform is defined as

$$\mathsf{QFT}_N|x\rangle := \frac{1}{\sqrt{N}} \sum_{\xi\in[N]} \omega_N^{\xi\cdot x}|\xi\rangle, \tag{1}$$

where $\omega_N := \exp(\frac{2\pi \mathrm{i}}{N})$.

2.1 Indifferentiability

Many important cryptographic functions are constructed using other primitives. Prominent examples include standardized hash functions SHA-2 [NIS15] and SHA-3 [NIS14]. In this paper we focus on constructions of compression functions. The general goal is to use an easy-to-design primitive to construct a more complicated function. In the context of compression functions, the internal primitives are fixed-length-input and fixed-length-output functions.

A common assumption that we make to prove security of cryptographic constructions, is that the internal functions are the idealized versions of the primitives. For example, we assume that the internal function is a uniformly random function. Such assumption abstracts the security flaws of the construction itself.

The question that we also need to answer, is the access privileges of the adversary. We define two interfaces: the *private* interface provides access to the construction and the *public* interface to the internal function used in the construction. The notion that captures the most realistic access model (still in the idealized assumption model) is *indifferentiability* [MRH04]. A construction is indifferentiable from a random oracle[3] if no adversary can distinguish them, even given access to the internal function. The following definition is the rephrased version of definitions from [MRH04; Cor+05], as presented in [Cza+19]. By efficient we mean algorithms that run in (quantum) time polynomial in the security parameter. Quantum queries are superpositions of classical queries, discussed in more detail in the next section.

Definition 1 (Indifferentiability [MRH04])**.** *A cryptographic (classical or quantum) system* C *is* (q, ε)-indifferentiable *from* R*, if there is an efficient (classical or quantum)* simulator S *and a negligible function* ε *such that for any efficient (classical or quantum)* distinguisher D *with binary output (0 or 1) the advantage*

$$\left|\mathbb{P}\left[b = 1 : b \leftarrow \mathsf{D}[\mathsf{C}_k^{\text{priv}\,\mathsf{C}_k^{\text{pub}}}, \mathsf{C}_k^{\text{pub}}]\right] - \mathbb{P}\left[b = 1 : b \leftarrow \mathsf{D}[\mathsf{R}_k^{\text{priv}}, \mathsf{S}^{\mathsf{R}_k^{\text{pub}}}]\right]\right| \leq \varepsilon(k)\,, \qquad (2)$$

where k *is the security parameter. The distinguisher makes at most* q *(classical or quantum) queries to* C*.*

It is important to note that if R is the random oracle (which is often the case), then both interfaces are the same. The construction C in Eq. (2) represents the *real* world and R the *ideal* world.

2.2 Game Playing Proofs for Indifferentiability

We work in the (quantum) game-playing framework. A *game* is an interactive algorithm that the adversary interacts with (plays). It is often beneficial to cast security definitions in terms of games. Especially so, because we can naturally argue about the distinguishing advantage[4] of adversaries when we present them with one of the two "similar" games.

More concretely, in the classical world Bellare and Rogaway [BR06] formalized the game-playing framework by introducing the notion of *identical-until-bad* games and showing the fundamental game-playing lemma, that provides a general bound on distinguishing advantage. Identical-until-bad games are algorithm that are syntactically identical until a flag Bad is set to "True". The fundamental game-playing lemma is as follows:

Lemma 2 (Fundamental lemma of game-playing, Lemma 2 of [BR06])**.** *Let* G *and* H *be identical-until-bad games and let* A *be an adversary that outputs a bit* b*. Then*

$$\left|\mathbb{P}\left[b = 1 : b \leftarrow \mathrm{A}^{\mathsf{H}}\right] - \mathbb{P}\left[b = 1 : b \leftarrow \mathrm{A}^{\mathsf{G}}\right]\right| \leq \mathbb{P}\left[\mathrm{Bad} = 1 : \mathrm{A}^{\mathsf{G}}\right]. \qquad (3)$$

To argue about quantum security with the use of games, several techniques have been developed. Games are modeled as quantum algorithms. A crucial element, common in game-playing proofs, however, has been added only recently. We talk about (efficient) lazy sampling, a technique introduced to quantum computing by Zhandry [Zha19]. This

[3] A random oracle is a random function with domain and range specified according to the discussed construction, the only way to access it is via oracle queries.

[4] Distinguishing advantage is the absolute value of the difference of probabilities of the adversary outputting 1 whenever interacting with one or the other game.

technique, often called the compressed random oracle technique, has been already further developed and used in multiple works [Chu+20a; Cza+19; HI19; JZM19; Chu+20b; Ros21; CEV20].

The role of Bad events can be played by punctured oracles, i.e. compressed oracles that are measured after every query. This concept has been introduced in [AHU19] and further developed in [Cza+19]. Measuring the oracle allows us to argue about the contents of the *quantum database* held by the game, similarly to the classical setup.

The fundamental quantum game-playing lemma is the One-way To Hiding (O2H) lemma introduced by Unruh in [Unr14]. In the formulation from [AHU19; Cza+19], the O2H lemma provides a bound on the distinguishing advantage for quantum adversaries interacting with punctured oracles. Putting all the elements mentioned above gives us the quantum game-playing framework. Below we provide definitions of all the parts in more detail.

2.2.1 Compressed Oracles

In our result we use a general formulation of the oracle, we use $\mathbb{Z}_N$ with addition modulo N. This formulation offers a slightly more general result and a more concise notation. Nonetheless, everything applies to $\mathbb{Z}_2^n$, this is the group (with $\oplus$ being the bitwise XOR) in which we formulate all concrete security results in Sections 3 and 5.

The first important idea behind the compressed oracles technique is purifying the random oracle. One way of formulation a random oracle is just (classically) sampling a random function and providing the adversary quantum access to it. The approach proposed in [Zha19] is to purify the (originally mixed) adversary's quantum state. The initial state of the random oracle is $\sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{\sqrt{|\mathcal{F}|}}|\mathbf{f}\rangle$, where $\mathcal{F} := \{\mathbf{f} : [N] \to [N]\}$ and $|\mathbf{f}\rangle$ holds the whole truth table of $\mathbf{f}$. In this view of the oracle, the *standard oracle* StO is the following update procedure:

$$\mathsf{StO}|x,y\rangle_A \sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{\sqrt{|\mathcal{F}|}}|\mathbf{f}\rangle_F = \sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{\sqrt{|\mathcal{F}|}}|x, y+\mathbf{f}(x)\rangle_A|\mathbf{f}\rangle_F, \tag{4}$$

where addition is done modulo N.

By performing the Quantum Fourier Transform on both the adversary's A^Y and oracle's F registers, the view of the oracle simplifies a lot, this is the brilliant observation from [Zha19]:

$$\mathsf{FO}|x,\eta\rangle_A|0^N\rangle_F = |x,\eta\rangle_A|0^N - (0,\ldots,\eta,\ldots,0)\rangle_F, \tag{5}$$

where on the right hand side register F is updated with a vector of 0's with η in the x'th row. The *Fourier oracle* is defined as the update procedure $\mathsf{FO} := \mathsf{QFT}_N^F \circ \mathsf{StO} \circ \mathsf{QFT}_N^{\dagger F}$. By using the standard basis of the oracle and the Fourier basis of the adversary's register, we get another useful, the *phase oracle*:

$$\mathsf{PhO}|x,\eta\rangle_A \sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{\sqrt{|\mathcal{F}|}}|\mathbf{f}\rangle_F = \sum_{\mathbf{f}\in\mathcal{F}} \frac{1}{\sqrt{|\mathcal{F}|}}\omega_N^{\eta\cdot\mathbf{f}(x)}|x,\eta\rangle_A|\mathbf{f}\rangle_F. \tag{6}$$

By inspecting the oracle register in the Fourier basis, we see that every query adds just one non-zero value to register F, hence after q queries there is at most q registers that are non-zero. This observation opens the possibility of *compressing* register F to contain only the non-zero entries. On an intuitive level, compressed oracles are a way to lazy sample random functions that are accessed in superposition.

The compressed Fourier oracle, that we denote by $\mathsf{CFO}_{\mathcal{Y}}$ (where the subscript denotes the uniform distribution over the set of outputs $\mathcal{Y}$). Is a procedure that maintains the database register, denoted by D. The database register that is correctly maintained is a superposition of the following states holding values in $(([N] \times \{\perp\}) \times [N])^q$ (in the formulation presented in [Cza+19]):

$$|((x_1, \eta_1), (x_2, \eta_2), \ldots, (x_s, \eta_s), (\perp, 0) \ldots, (\perp, 0)\rangle_D, \tag{7}$$

where s is the size, i.e., he number of non-empty entries, of the database, each of the first s elements in D^Y are non-zero, moreover D^X are sorted in the rising order. The symbol $\perp$ is an additional symbol signifying empty entries. A more detailed description of the compressed oracle technique can be found in [Zha19] and in the formulation we use in [Cza+19].

We model access to multiple quantum oracles at once by specifying a special quantum register I. This register holds information about the particular interface that is queried. The register encoding the interface is I and holds $a \in \{1, \ldots, k\}$. To update the oracle register, we apply CFO^{D_a} controlled on register I. Note that this setup allows the adversary to make a superposition of queries to different oracles. If however one would want to make the interface register classical, we can just perform a standard basis measurement of I. In the context of compressed oracles the multiple interfaces hold different databases. Of course if oracles are somehow related then so are the databases.

Changing the basis of the oracle register and the adversary, in a similar way to described above but to register D^Y instead of F, gives us the compressed standard oracle CStO and the compressed phase oracle CPhO. The compressed phase oracle is the one we use in our general result on punctured oracles. For our concrete security results we denote the compressed oracle for a uniform distribution over functions from $\{\mathbf{f} : \{0,1\}^n \to \{0,1\}^n\}$ by CStO_n.

2.2.2 Punctured Oracles and Relations

Punctured oracles are compressed oracles, that are measured after every query. The measurement checks if the oracle register holds (in a superposition) a database that fulfills some fixed relation.

A relation is a subset of the set of all databases.

Definition 3 (Relation R on $(D_1, \ldots, D_k)$)**.** *Let $D = (D_1, \ldots, D_k)$ be k databases each of size at most q: For each $a \in \{1, \ldots, k\}$ database D_a is a set of $(x, y) \in \mathcal{X}_a \times \mathcal{Y}_a$, where $\mathcal{X}_a$ and $\mathcal{Y}_a$ are arbitrary finite sets. A* relation R on D *is a subset*

$$R \subseteq \prod_{a=1}^{k} \left(\mathcal{X}_a \times \bigcup_{s_a \in [q+1]} (\mathcal{X}_a \times \mathcal{Y}_a)^{s_a} \right). \tag{8}$$

In this paper we focus on relation that are non-trivial only on the $\bigcup_{s_a \in [q+1]} (\mathcal{X}_a \times \mathcal{Y}_a)^{s_a}$ part.

A compressed oracle H with the database register kept in the standard basis, holds a superposition of databases of different sizes[5]. Still, any of the databases in superposition can be in R or not. The oracle punctured on relation R (defined on all of the databases maintained by H) is denoted by $\mathsf{H} \setminus R$ and defined as:

[5]In principle, this register is entangled with the adversary's register.

Definition 4 (Punctured compressed oracle $\mathsf{H} \setminus R$, Def. 9 in [Cza+19]). *Let* H *be a compressed oracle and* R *a relation on its database. The punctured compressed oracle* $\mathsf{H} \setminus R$ *is equal to* H*, except that* R *is measured after every query. By* Find *we denote the event that* R *outputs* 1 *at least once among all queries.*

We describe the algorithm that measures the relation R is more detail. We assume that membership in R is efficiently decidable. We denote the single-bit membership decision by $D \in R$, the bit is 1 if and only if database D is in R. To measure the relation we define a unitary V_R^{DJ} that XORs a bit $D \in R$ to register J; This unitary is controlled on register D, holding the whole database (possibly consisting of many databases D_a).

2.2.3 One-way To Hiding Lemma

In the quantum case, identical-until bad games are compressed oracles (or algorithms using compressed oracles) that are punctured on different relations. The role of the fundamental game-playing lemma takes the One-Way to Hiding lemma [Unr14; AHU19].

Lemma 5 (Version of Theorem 10 in [Cza+19]). *Let* R_1 *and* R_2 *be relations on the database of a quantum oracle* H*. Let* A *be an oracle algorithm making* q *quantum queries, then*

$$\begin{aligned}&\left|\mathbb{P}\left[b = 1 : b \leftarrow \mathrm{A}^{\mathsf{H}\setminus R_1}(.)\right] - \mathbb{P}\left[b = 1 : b \leftarrow \mathrm{A}^{\mathsf{H}\setminus R_1 \cup R_2}(.)\right]\right| \\ &\leq \sqrt{(q+1)\mathbb{P}[\mathrm{Find}_2 : \mathrm{A}^{\mathsf{H}\setminus R_1 \cup R_2}(.)]},\end{aligned} \tag{9}$$

where Find_2 *is the event that measuring* $R_1 \cup R_2$ *succeeds.*

The theorem above is very general, yet the complicated part is not really proving it. The more involved aspect is finding a good bound for $\mathbb{P}\left[\mathrm{Find}\right]$, exactly that is the main subject of Sec. 4.

In the quantum case we also define a notion of *almost identical* oracles.

Definition 6 (Definition 11 in [Cza+19]). *Let* H *and* G *be compressed oracles and* R_i*,* $i = 1, 2$ *relations on their databases. We call the oracles* $\mathsf{H} \setminus R_1$ *and* $\mathsf{G} \setminus R_2$ almost identical *if they are equal conditioned on the events* $\neg\mathrm{Find}_1$ *and* $\neg\mathrm{Find}_2$ *respectively, i.e. for any string* z *and any quantum algorithm* A

$$\mathbb{P}[b = 1 : b \leftarrow A^{\mathsf{H}\setminus R_1}(z) \mid \neg\mathrm{Find}_1] = \mathbb{P}[b = 1 : b \leftarrow A^{\mathsf{G}\setminus R_2}(z) \mid \neg\mathrm{Find}_2]. \tag{10}$$

We can prove the following bound on the adversary's advantage in distinguishing almost identical punctured oracles.

Lemma 7 (Lemma 12 in [Cza+19]). *If* $\mathsf{H} \setminus R_1$ *and* $\mathsf{G} \setminus R_2$ *are almost identical according to Def.6 then for any* $b \in \{0, 1\}$

$$\begin{aligned}&\left|\mathbb{P}[b = 1 : b \leftarrow A^{\mathsf{H}\setminus R_1}(z)] - \mathbb{P}[b = 1 : b \leftarrow A^{\mathsf{G}\setminus R_2}(z)]\right| \\ &\leq 2\mathbb{P}[\mathrm{Find}_1 : A^{\mathsf{H}\setminus R_1}(z)] + 2\mathbb{P}[\mathrm{Find}_2 : A^{\mathsf{G}\setminus R_2}(z)].\end{aligned} \tag{11}$$

2.3 The Rate-1/3 Compression Function

This construction (that we denote Rate-1/3) of a compression function has been first defined in [SS08]. The authors explore constructing a compression function out of three

random functions. They also prove that the constructed hash function is collision resistant. The main advantage of this construction is that it achieves optimal collision resistance of up to $O(2^{n/2})$ queries. The construction is defined as follows:

$$\textsc{Rate-1/3}_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}(x_1,x_2) := \mathbf{f}_3\left(\mathbf{f}_1(x_1) \oplus \mathbf{f}_2(x_2)\right) \oplus \mathbf{f}_1(x_1), \tag{12}$$

where $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3 : \{0,1\}^n \to \{0,1\}^n$. In Figure 1 we present the scheme of the construction.

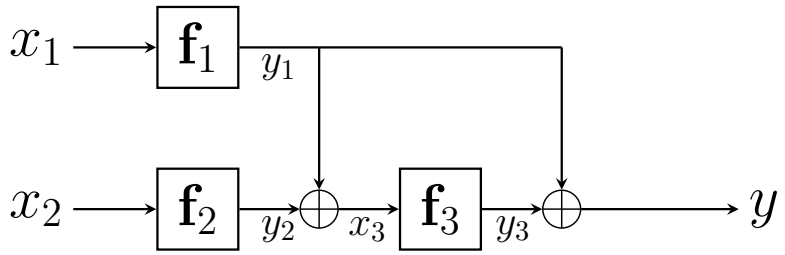

Figure 1: A schematic representation of the rate-1/3 compression function $\textsc{Rate-1/3}_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}(x_1,x_2) = y$.

3 Classical Indifferentiability of Rate-1/3

Classical indifferentiability of Rate-1/3 is the first result of our paper. We present it before all the other results to provide a better understanding of the situation we include in the general bound for the quantum distinguishing advantage.

The key central goal of [SS08] was to propose a construction that has optimal security. The notion they focused on was collision-resistance. A natural strengthening (that implies collision-resistance) of this notion is indifferentiability. We, however, show that the security of the construction, with respect to the stronger notion, is no longer optimal. Namely, the query lower bound is not $\Omega(2^{n/2})$ but rather $\Omega(2^{n/3})$.

To save space we present multiple algorithms in one. To do that we follow a convention where only the boxed algorithms perform the boxed operations. In case there are actually more than two algorithms, the color of the box also matters. If a line is not surrounded by a box, then all algorithms perform the command. We number the simulators with the number of the game it is first used in.

Theorem 8. *The compression function* $\textsc{Rate-1/3}_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$ *for uniformly random* $\mathbf{f}_1$, $\mathbf{f}_2$, *and* $\mathbf{f}_3$ *is* (q,ε)*-classically indifferentiable for* $\varepsilon = 10\frac{q^3}{2^n}$.

Proof. We carry out the proof by starting with the real world and gradually changing the adversary's interface to the ideal world. We define two simulators, the initial S_2 that just lazy samples the compression functions and S_3 that is the actual simulator. In Algorithm 1 only the boxed algorithm performs the boxed commands.

Game 1 The interface in the first game is $(\textsc{Rate-1/3}, (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3))$, where the public interface holds uniformly random $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$. The definition of the game is

$$\textbf{Game 1} := \left(b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1/3}, (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)]\right). \tag{13}$$

Algorithm 1 Classical simulators $\boxed{\mathsf{S}_2}$, $\boxed{\mathsf{S}_3}$ for RATE-1/$3_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$

procedure $\mathbf{f}_1(x_1)$
  if $x_1 \in D_1^X$ **return** the corresponding y_1
  $y_1 \xleftarrow{\$} \{0,1\}^n$
  if $\exists y_2 \in D_2^Y, x_3 \in D_3^X : y_1 = y_2 \oplus x_3$ **then** ▷ Preimage of $\mathbf{f}_3$
    $\boxed{\text{Set } \mathsf{Bad}_1 = 1}$
  Add (x_1, y_1) to D_1 and **return** y_1

procedure $\mathbf{f}_2(x_2)$
  if $x_2 \in D_2^X$ **return** the corresponding y_2
  $y_2 \xleftarrow{\$} \{0,1\}^n$
  if $\exists y_1 \in D_1^Y, x_3 \in D_3^X : y_2 = y_1 \oplus x_3$ **then** ▷ Preimage of $\mathbf{f}_3$
    $\boxed{\text{Set } \mathsf{Bad}_2 = 1}$
  Add (x_2, y_2) to D_2 and **return** y_2

procedure $\mathbf{f}_3(x_3)$
  if $x_3 \in D_3^X$ **return** the corresponding y_3
  if $\exists y_1 \in D_1^Y, y_2 \in D_2^Y : x_3 = y_1 \oplus y_2$ **then**
    $\boxed{y_3 \xleftarrow{\$} \{0,1\}^n, \text{ add } (x_3, y_3) \text{ to } D_3}$
    $\boxed{\textbf{return } y_3}$
    $\boxed{\textbf{return } \mathsf{R}(x_1, x_2) \oplus y_1}$
  else
    $y_3 \xleftarrow{\$} \{0,1\}^n$, add (x_3, y_3) to D_3
    return y_3

Game 2 In the second game we lazy sample the compression functions, the interface is $(\textsc{Rate-1}/3, \mathsf{S}_2)$, the game is defined as

$$\textbf{Game 2} := (b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1}/3, \mathsf{S}_2])\,. \tag{14}$$

This change of the interface is indistinguishable for A:

$$\left|\mathbb{P}\left[\textbf{Game 2}\right] - \mathbb{P}\left[\textbf{Game 1}\right]\right| = 0. \tag{15}$$

Game 3 The interface in the third game is $(\textsc{Rate-1}/3, \mathsf{S}_3)$, where we introduce the bad events and random oracle R, note that $\mathbf{f}_3$ is distributed uniformly at random, so adding R does not change the distribution of $\mathbf{f}_3$ at all. The new game is

$$\textbf{Game 3} := (b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1}/3, \mathsf{S}_3])\,. \tag{16}$$

The bad events we introduce happen when either $\mathbf{f}_1$ or $\mathbf{f}_2$ outputs a value that forms a preimage of $\mathbf{f}_3$. The reason why these events are significant is because if we commit to an output of $\mathbf{f}_3$ and after that a query to $\mathbf{f}_1$ or $\mathbf{f}_2$ finishes the chain of values in the construction, then we introduce a discrepancy between the construction and the random oracle. The only noticeable change for the adversary are the bad event. To calculate distinguishability we use the fundamental game-playing lemma, Lem. 2:

$$\left|\mathbb{P}\left[\textbf{Game 3}\right] - \mathbb{P}\left[\textbf{Game 2}\right]\right| \leq \mathbb{P}\left[\mathrm{Bad}_1 \vee \mathrm{Bad}_2\right] \leq 2\frac{q^3}{2^n}, \tag{17}$$

where the right hand side follows from the fact that there are at most $s_2(i) \cdot s_3(i)$ pairs (y_2, x_3) that y_1 can collide with:

$$\mathbb{P}\left[\mathrm{Bad}_1\right] \leq \sum_{i=1}^{q} \frac{s_2(i) \cdot s_3(i)}{2^n} \leq \sum_{i=1}^{q} \frac{q^2}{2^n} = \frac{q^3}{2^n}, \tag{18}$$

where in the first inequality we use the union bound. The second inequality follows from a bound on the size of D_2 and D_3, after the i-th query $s_2(i), s_3(i) \leq q$. The final bound on $\mathbb{P}\left[\mathrm{Bad}_1 \vee \mathrm{Bad}_2\right]$ comes from the union bound and applying Eq. (18) to Bad_1 and Bad_2.

Game 4 In the last game the interface is $(\mathsf{R}, \mathsf{S}_3)$, we change the private interface, i.e. the interface giving access to the construction or the random oracle. The definition of the game is

$$\textbf{Game 4} := (b = 1 : b \leftarrow \mathrm{A}[\mathsf{R}, \mathsf{S}_3])\,. \tag{19}$$

The only source of distinguishing advantage for A are the possible bad events in queries to the private interface. We know, however, that if there are no bad events then **Game 3** and **Game 4** are distributed in the same way:

$$\left|\mathbb{P}\left[\textbf{Game 4} \mid \neg\mathrm{Bad}\right] - \mathbb{P}\left[\textbf{Game 3} \mid \neg\mathrm{Bad}\right]\right| = 0. \tag{20}$$

Using the above identity we derive the final distinguishing advantage:

$$\left|\mathbb{P}\left[\textbf{Game 4}\right] - \mathbb{P}\left[\textbf{Game 3}\right]\right| \leq 4\mathbb{P}\left[\mathrm{Bad}\right] \leq 8\frac{q^3}{2^n}, \tag{21}$$

where we use Lem. 7. To get the first inequality above we consider classical algorithms in place of the quantum ones in Lem. 7 and Bad events instead of Find. The event Bad

in Eq. (21) corresponds to Bad in **Game 3**, we bound the bound from the lemma by the bigger of the two probabilities. Probability of Bad in **Game 3** is greater because there are in principle more calls to S_3—the private interface also calls the simulator. □

For the discussion of quantum databases in the next section, we would like to highlight the relation between the three databases D_1, D_2, and D_3 holding queries (and outputs) to $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ respectively. Note that there are no outputs of $\mathbf{f}_1$ or $\mathbf{f}_2$ that would equal $y_2 \oplus x_3$ or $y_1 \oplus x_3$ for any $x_3 \in D_3^X$ and any output y_2 or y_1. This means that the set of "good" databases maintained by S_3 consists of (D_1, D_2, D_3) with no "preimages" of the type we described above of $\mathbf{f}_3$. This is important to note because in our proof of indistinguishability of *quantum* games we consider superpositions of all "good" databases. The crucial observation is that by changing the contents of D_3 (by for example removing an entry) we change the set of "good" D_1 and D_2 databases.

4 Bound on $\mathbb{P}[\mathrm{Find}]$

In this section we prove a general bound on probability of Find, important in the O2H lemma. We generalize the bound from [Cza+19] by including relations defined on multiple databases. We follow the same approach so here we provide the definitions necessary to parse the main general lemma and a high-level proof, additional details of the proof can be found in Sec. A.

We state a lemma giving a bound on the probability of Find for the uniform distribution over the sets $\{\mathbf{f}_1 : \mathcal{X}_1 \to \mathcal{Y}_1\}$ and $\{\mathbf{f}_2 : \mathcal{X}_2 \to \mathcal{Y}_2\}$ and for a general relation, possibly defined on multiple databases. We also allow for R to depend on an external random oracle R. In this proof we explicitly analyze adversaries with two interfaces H_1 and H_2. We allow them to make queries to different interfaces in superposition.

The register encoding the interface is I and holds $a \in \{1, 2\}$. In what follows we assume $\mathcal{Y}_1 = [N_1]$ and $\mathcal{Y}_2 = [N_2]$. By $\bar{a}$ we denote the index other that a, namely $\bar{a} = 3 - a$. Whenever we refer to a we mean by it the interface that is queried.

In the statement of Lem. 9 we focus on two databases (and hence two independent functions) but all the results of this section almost trivially extend to any fixed number of functions.

For $a \in \{1, 2\}$ we write $\vec{x}_a \in (\mathcal{X}_a \times \{a\})^q$ to denote all the previous inputs asked by the adversary to H_a, we always consider queries x to be pairs of the query value and the interface. We mostly leave the interface part implicit. A vector with a fixed a has a fixed interface a in all tuples. (x, η, a) is the last query. Whenever $\vec{x}$ denotes queries, the vector is sorted in a rising fashion. We denote the outputs given to A by $\vec{y}_a := (y_1^a, \ldots, y_{s_a}^a)$, where $y_i^a \in \mathcal{Y}_a \times \{a\}$ are pairs of values with interface, similarly to inputs. Vector of outputs is sorted according to the corresponding inputs. The set of all queries is $\vec{x} = \vec{x}_1 \cup \vec{x}_2$, similarly for $\vec{y}$. When we use set operations[6] on vectors we mean a set consisting of entries of $\vec{x}$, note that if there are no repetitions in $\vec{x}$, then there is no ambiguity. Databases are denoted as $D_a = ((x_1^a, y_1^a), \ldots, (x_{s_a}^a, y_{s_a}^a))$.

In this section our primary subject are databases and their membership in the relation. To this end we define sets of good and bad outputs. For a relation R, the database $D =$

[6] Like the union $\cup$, intersection $\cap$, or subtraction $\setminus$.

(D_1, D_2) that contains $\vec{x}_1$ and $\vec{x}_2$ of sizes s_1 and s_2 respectively, and $x \notin D_a^X$ we have

$$\mathcal{G}^R(\vec{x}_1, \vec{x}_2) := \left\{(D_1^Y(\vec{x}_1), D_2^Y(\vec{x}_2)) \in \mathcal{Y}_1^{s_1} \times \mathcal{Y}_2^{s_2} : (D_1, D_2) \notin R\right\}, \tag{22}$$

$$\mathcal{G}_{\bar{a}}^R(\vec{x}_1, \vec{x}_2 \mid D_a) := \left\{D_{\bar{a}}^Y(\vec{x}_{\bar{a}}) \in \mathcal{Y}_{\bar{a}}^{s_{\bar{a}}} : (D_1, D_2) \notin R\right\}, \tag{23}$$

$$\mathcal{B}_a^R(x \mid D) := \{y \in \mathcal{Y}_a : (D_a \cup \{(x, y)\}, D_{\bar{a}}) \in R\}. \tag{24}$$

The bad set defined above is the subset of the codomain of the sampled function corresponding to the new value bringing D to be in R. By $\mathcal{G}_a^R(\vec{x}_1, \vec{x}_2)$ we denote the part of $\mathcal{G}^R(\vec{x}_1, \vec{x}_2)$ corresponding to $D_a^Y(\vec{x}_a)$.

Our assumptions on R are the following: The relation does not depend on the adversary's input. The size of $\mathcal{G}^R(\vec{x}_1, \vec{x}_2)$ depends only on s_1 and s_2. When addressing the size of $\mathcal{G}$ we often write $\left|\mathcal{G}^R(s_1, s_2)\right|$. Moreover $\left|\mathcal{B}_a^R(x \mid D)\right|$ is the same for all $x \notin D_a^X$.

We also define a coefficient that gives the number of outputs that bring the database to R, defined as:

$$b_a^R(s_1, s_2) := \left|\mathcal{B}_a^R(x \mid D)\right|, \tag{25}$$
$$\text{where } x \notin D, D \notin R, \text{ and } |D_a| = s_a - 1, |D_{\bar{a}}| = s_{\bar{a}},$$

as one can see from the definition (the argument of b_a^R does not include particular values in $\vec{x}_1$ and $\vec{x}_2$) above we use the assumption that $\left|\mathcal{B}_a^R(x \mid D)\right|$ is the same for all $x \notin D_a^X$. When making a query to database a we use the notation $\left|\mathcal{G}^R(s_a - 1, s_{\bar{a}})\right|$ for $\left|\mathcal{G}^R(s_1 - 1, s_2)\right|$ if $a = 1$ and $\left|\mathcal{G}^R(s_1, s_2 - 1)\right|$ if $a = 2$. Similarly we use $b_a^R(s_a + 1, s_{\bar{a}})$. An important identity that we will use later in this section is:

$$\left|\mathcal{G}^R(s_1, s_2)\right| = \left|\mathcal{G}^R(s_a - 1, s_{\bar{a}})\right| (|\mathcal{Y}_a| - b_a^R(s_1, s_2)). \tag{26}$$

To get some intuition for the above equality, let us consider a database D of size $s_a - 1 + s_{\bar{a}}$ that is not in R. According to the definition from Eq. (24), there are $b_a^R(s_1, s_2)$ outputs $y \in \mathcal{Y}_a$, such that $(D_a \cup \{(x, y)\}, D_{\bar{a}})$ for any $x \in \mathcal{X}_a$ that is not in D_a^X, is in R. As this holds for any value x, for every good database we have $|\mathcal{Y}_a| - b_a^R(s_1, s_2)$ good database with a single query added to D_a.

In general as in the good and bad sets, as well as the coefficient b, we omit the superscript R whenever the relation is clear from the context. As examples of b, consider relations on a single database, if the relation is the zero-preimage[7], then $b(s) = 1$, there is just one value $y = 0$ that causes a fresh query to be in relation; For collisions[8] we have $b(s) = s - 1$, the new y can be any of the previously queried values to make D fulfill the relation.

Two sets important in our treatment of multiple databases are $\mathcal{H}_a^{\text{ADD}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)$ and $\mathcal{H}_a^{\text{REM}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)$. To properly define them we generalize the definition of the good set conditioned on a database:

$$\mathcal{G}_{\bar{a}}(\vec{x}_a, \vec{x}_{\bar{a}} \mid \vec{y}_a) := \Big\{D_{\bar{a}}^Y(\vec{x}_{\bar{a}}) \in \mathcal{Y}_{\bar{a}}^{s_{\bar{a}}} : \\ \exists \vec{y}_a^* \in \mathcal{Y}_a^{s_a - |\vec{y}_a|}, D_a^Y(\vec{x}_a) := \vec{y}_a \cup \vec{y}_a^*, (D_1, D_2) \notin R\Big\}. \tag{27}$$

Intuitively speaking the above set is the set $\mathcal{G}_{\bar{a}}(\vec{x}_a, \vec{x}_{\bar{a}} \mid D_a)$ defined in Eq. (23) with the difference that we do not specify all values in D_a^Y. Moreover the more entries are in $\vec{x}_a$

[7] Zero-preimage is a relation consists of databases that contain $y = 0$.

[8] A database has a collision if it contains two entries (x_1, y_1) and (x_2, y_2) such that $y_1 = y_2$.

the more "restrictions" are on good $D_{\bar{a}}$, meaning the size of the good in principle gets smaller with $\vec{x}_a$ getting bigger. Then the two sets are defined as

$$\mathcal{H}_a^{\text{ADD}}(\vec{x}_1, \vec{x}_2, \vec{y}_a) := \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2 \mid \vec{y}_a) \setminus \mathcal{G}_{\bar{a}}(\vec{x}_a \cup \{x\}, \vec{x}_{\bar{a}} \mid \vec{y}_a), \tag{28}$$

$$\left|\mathcal{H}_a^{\text{ADD}}(s_1, s_2)\right| := \left|\mathcal{H}_a^{\text{ADD}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)\right|, \tag{29}$$

and

$$\mathcal{H}_a^{\text{REM}}(\vec{x}_1, \vec{x}_2, \vec{y}_a) := \mathcal{G}_{\bar{a}}(\vec{x}_a \setminus \{x\}, \vec{x}_{\bar{a}} \mid \vec{y}_a) \setminus \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2 \mid \vec{y}_a), \tag{30}$$

$$\left|\mathcal{H}_a^{\text{REM}}(s_1, s_2)\right| := \left|\mathcal{H}_a^{\text{REM}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)\right|. \tag{31}$$

The intuition one should have for $\mathcal{H}_a^{\text{ADD}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)$ and $\mathcal{H}_a^{\text{REM}}(\vec{x}_1, \vec{x}_2, \vec{y}_a)$ is that for the relation we discuss in this paper, they are very small sets.

The assumption that is important for when R is defined on two databases is that if good outputs of H_1 depend on inputs to H_2, we never make a query to H_2 that automatically brings D to be in R. An example of such relation is $y_1 = x_2$ (outputs of H_1 equal to any input to H_2). For these relations it is trivial to fulfill them—by just querying one of the outputs of H_1 to H_2—so the oracles have to be constructed in a way that avoids this attack. By constructing we mean adding a quantum algorithm managing queries to different interfaces. We say that such *trivial attacks* are of concern when D_a^X interacts with $D_{\bar{a}}^Y$.

Below we state a lemma bounding the probability of Find, which gives great utility to the quantum game-playing framework. The result depends only on measurements performed on the database. The basis of the database matters, as we define the relation in a particular (standard) basis. Hence, this result works exactly the same for CStO.

Lemma 9. *Let* A *be a quantum adversary interacting with a compressed punctured oracle* $\mathsf{H} \setminus R$*, with* $\mathsf{H} = \mathsf{S}(\mathsf{H}_1, \mathsf{H}_2)$*, where* S *is any quantum algorithm that ensures that the trivial attacks (important when* D_a^X *interacts with* $D_{\bar{a}}^Y$*) are avoided,* $\mathsf{H}_1 = \mathsf{CPhO}_{\mathcal{Y}_1}$ *and* $\mathsf{H}_2 = \mathsf{CPhO}_{\mathcal{Y}_2}$*. Moreover* R *is a relation following Def. 3, such that*

1. $\left|\mathcal{G}^R(\vec{x}_1, \vec{x}_2)\right|$ *from Eq.* (41) *depends only on* s_1 *and* s_2*,*
2. $\left|\mathcal{B}_a^R(x \mid D)\right|$ *from Eq.* (24) *is the same for all* $x \notin D_a^X$*.*

Then the probability of Find *is bounded by:*

$$\begin{aligned}
\mathbb{P}\left[\text{Find} : \mathsf{A}[\mathsf{H} \setminus R]\right] \leq \sum_{i=1}^{q} \Bigg(& \sum_{j=1}^{i-1} \max_{a \in \{1,2\}, s_1, s_2 \leq j-1} \Bigg(2\frac{b_a(s_1,s_2)}{N_a} \\
& + \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \\
& + \frac{b_a(s_1,s_2)}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \\
& + \frac{b_a(s_a + 1, s_{\bar{a}})}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a + 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a + 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \Bigg) \\
& + \max_{a \in \{1,2\}, s_1, s_2 \leq i-1} \left(\sqrt{\frac{N_a - b_a(s_a + 1, s_{\bar{a}})}{N_a}} \sqrt{\frac{|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \right.
\end{aligned}$$

$$+\sqrt{\frac{b_a(s_a+1,s_{\bar{a}})}{N_a}}+\frac{b_a(s_1,s_2)^{3/2}}{N_a\sqrt{N_a-b_a(s_1,s_2)}}$$
$$+\frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a-b_a(s_1,s_2))}}\mathrm{sgn}\left(\left|\mathcal{H}_a^{\mathrm{REM}}(s_1,s_2)\right|\right)\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}$$
$$+\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\mathrm{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}+\frac{\sqrt{b_a(s_1,s_2)(N_a-b_a(s_1,s_2))}}{N_a}\Bigg)\Bigg)^2, \tag{32}$$

where $a\in\{1,2\}$*,* q *is the maximal number of queries made by* A*, and* sgn *is the sign function equal* 0 *whenever the argument is* 0*.*

4.1 Proof of Lem. 9

The proof that we present follows closely the proof from Appendix C in [Cza+19], we use the same structure and reuse a lot of their results with the important change of treating relations that are defined on multiple databases.

Proof.

4.1.1 Overview

We first provide a high level overview of the proof.

To slightly simplify our main task, in Eq. (39), we start with splitting the probability of the puncturing succeeding in any query into a sum of probabilities that it succeeds in the i-th query. By succeeding we mean he puncturing measurement outputs 1.

To bound the probability that $D\in R$ is measured in the i-th query, given that it was not measured before, we introduce the *good* state $|\Psi_i^{\mathrm{Good}}\rangle$. The good state is a (more or less) artificial state that approximates the state $|\Phi_i\rangle$ of the adversary A and the oracle $\mathsf{H}\setminus R$ that she interacts with conditioned on the measurement of R always outputting 0. We introduce the good state, because it is easier to handle in explicit calculations. In Eq. (46) we show how to include the good state in the overall proof.

With this approach, the key bound that we need to evaluate is on the norm of the difference of $\mathsf{H}\setminus R|\Psi_i^{\mathrm{Good}}\rangle$ and $|\Psi_{i+1}^{\mathrm{Good}}\rangle$, as presented in Eq. (47). To calculate this bound we inspect in detail the good state after a query $\mathsf{H}\setminus R|\Psi_i^{\mathrm{Good}}\rangle$ and pinpoint the differences of this state from $|\Psi_{i+1}^{\mathrm{Good}}\rangle$. We call these differences *errors*, bounding their norm is the key task here. This analysis is presented in Sec. A.1.

Additionally, we also calculate the probability that Find happens when the joint adversary-oracle state is $|\Psi_i^{\mathrm{Good}}\rangle$. All the differences identified in the previous step and the bound on Find for the good state are formalized in Lemmas 14 and 16, which are proven in Sections A.2 and A.3 respectively.

All of the analysis outlined above works for a general relation defined for two databases (which can be almost trivially generalized to any constant number).

4.1.2 Introduction

We start the proof with a few definitions concerning compressed oracles. The measurement that we apply after every H in Def. 4 is a binary projective measurement with two elements:

$$\mathsf{J}_R:=\mathbb{1}\otimes|1\rangle_J\langle 1| \text{ and} \tag{33}$$
$$\overline{\mathsf{J}}_R:=\mathbb{1}\otimes|0\rangle_J\langle 0|, \tag{34}$$

where register J holds the (superposition of) bit $D \in R$.

In the following we focus on the punctured oracle just prior to measurement $\{\mathsf{J}_R, \overline{\mathsf{J}_R}\}$. A unitary that omits the measurement of register J in $\mathsf{H} \setminus R$ acts on registers ADJ, we define it as

$$\mathsf{H} \setminus \mathsf{V}_R := \mathsf{Queries}^\dagger \circ \mathsf{V}_R \circ \mathsf{Queries} \circ \mathsf{H}, \tag{35}$$

where the unitary Queries counts the number of $x \neq \perp$ in D (i.e. the number of non-empty registers in the quantum database) and V_R checks whether the queried values in registers D fulfill the relation R and saves the single bit answer to register J.

We proceed by rephrasing the definition of $\mathbb{P}[\text{Find} : \text{A}[\mathsf{H} \setminus R]]$:

$$\mathbb{P}[\text{Find} : \text{A}[\mathsf{H} \setminus R]] = 1 - \left\| \left(\prod_{i=q}^{1} \overline{\mathsf{J}}_R \mathsf{U}_i \mathsf{H} \setminus \mathsf{V}_R \right) |\Psi_0\rangle|0\rangle_J \right\|^2 \tag{36}$$

$$= 1 - \left\| \left(\prod_{i=q-1}^{1} \mathsf{U}_i \overline{\mathsf{J}}_R \mathsf{H} \setminus \mathsf{V}_R \right) |\Psi_0\rangle|0\rangle_J \right\|^2$$

$$+ \left\| \mathsf{U}_q \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \left(\prod_{i=q-1}^{1} \mathsf{U}_i \overline{\mathsf{J}}_R \mathsf{H} \setminus \mathsf{V}_R \right) |\Psi_0\rangle|0\rangle_J \right\|^2 = \cdots = \tag{37}$$

$$= \sum_{i=1}^{q} \left\| \mathsf{U}_i \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \underbrace{\left(\prod_{j=i-1}^{1} \mathsf{U}_j \overline{\mathsf{J}}_R \mathsf{H} \setminus \mathsf{V}_R \right) |\Psi_0\rangle|0\rangle_J}_{:=\mathsf{U}_{i-1}|\Phi_{i-1}\rangle} \right\|^2 \tag{38}$$

$$= \sum_{i=1}^{q} \| \mathsf{U}_i \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Phi_{i-1}\rangle \|^2 = \sum_{i=1}^{q} \| \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Phi_{i-1}\rangle \|^2 , \tag{39}$$

where $|\Psi_0\rangle$ is the initial state of the adversary. The definition of the "true" state is

$$|\Phi_{i-1}\rangle := \mathsf{U}_{i-1}^\dagger \left(\prod_{j=i-1}^{1} \mathsf{U}_j \overline{\mathsf{J}}_R \mathsf{H} \setminus \mathsf{V}_R \right) |\Psi_0\rangle|0\rangle_J \tag{40}$$

Above, the second and third equations follow from the fact that $\||v\rangle\|^2 = \|\mathsf{P}|v\rangle\|^2 + \|(\mathbb{1} - \mathsf{P})|v\rangle\|^2$ for all $|v\rangle$ and projectors P, the last equality follows from $\|\mathsf{U}|v\rangle\| = \||v\rangle\|$ for any unitary U.

As we already mentioned in the beginning of the proof, the quantity that we analyze now is $\|\mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Phi_{i-1}\rangle\|^2$. To this end, we introduce $|\Psi^{\text{Good}}_{i-1,R}\rangle|0\rangle_J$ for which bounding $\left\| \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Psi^{\text{Good}}_{i-1,R}\rangle|0\rangle_J \right\|^2$ is easier. The good state is essentially the state after the adversary A interacts with H but with the oracle register holding a superposition of only databases that are not in relation. Later we prove that the good state is close to the original $|\Phi_{i-1}\rangle$.

4.1.3 The good state

The state $|\Psi^{\text{Good}}_{i,R}\rangle_{AD}$ corresponds to the adversary's state just after the i-th query and before the application of U_i. The size of the database s_a depends on whether the new query x was added to, updated, or removed from the database, it equals $|\vec{x}_a \cup \{x\}|$, $|\vec{x}_a|$, or

$|\vec{x}_a \setminus \{x\}|$ respectively. After i queries s_a can range from 0 to i and the joint state of A and the oracle can be a superposition over different database sizes. By $D(\perp)$ we denote the part of the database containing empty entries. The adversary's work register is denoted by A^W and its contents by $\psi(x, \eta, \vec{x}, \vec{\eta}, w)$, where w can be any value of finite size. We denote the inner product by $\vec{\eta}_a \cdot \vec{y}_a = \sum_{i=1}^{s_a} \eta_i^a \cdot y_i^a \mod N$. In the sum below $\vec{y} = \vec{y}_1 \cup \vec{y}_2$. We define the good state as:

$$
\begin{aligned}
|\Psi_{i,R}^{\text{Good}}\rangle_{AD} := & \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w} |x,\eta,a\rangle_{A^{XYI}} |\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W} \\
& \sum_{\vec{y}\in\mathcal{G}^R(\vec{x}_1,\vec{x}_2)} \frac{1}{\sqrt{|\mathcal{G}^R(s_1,s_2)|}} \omega_N^{\vec{\eta}_1\cdot\vec{y}_1} |(x_1^1,y_1^1),\ldots,(x_{s_1}^1,y_{s_1}^1)\rangle_{D_1(\vec{x}_1)} \\
& \sum_{y_{s_1+1},\ldots,y_q\in[N]} \frac{1}{\sqrt{N^{q-s_1}}} |(\perp,y_{s_1+1}),\ldots,(\perp,y_q)\rangle_{D_1(\perp)} \\
& \omega_N^{\vec{\eta}_2\cdot\vec{y}_2} |(x_1^2,y_1^2),\ldots,(x_{s_2}^2,y_{s_2}^2)\rangle_{D_2(\vec{x}_2)} \\
& \sum_{y_{s_2+1},\ldots,y_q\in[N]} \frac{1}{\sqrt{N^{q-s_2}}} |(\perp,y_{s_2+1}),\ldots,(\perp,y_q)\rangle_{D_2(\perp)}. \qquad (41)
\end{aligned}
$$

In case we have added x to D_a, the full database D above contains (x, y_j^a). In the rest of the proof we omit the subscript R, however note that $|\Psi_i^{\text{Good}}\rangle$ does indeed depend on R.

As we already mentioned, another way to define the good state is to consider the joint state of the adversary and the non-punctured oracle H just after the i-th query. The good state is then this state after a projection of register D with $\overline{\mathsf{J}}_R$. Normalization of the projected state comes from multiplying each branch corresponding to a given size of the database by an appropriate $\sqrt{\frac{N_1^{s_1} N_2^{s_2}}{|\mathcal{G}^R(s_1,s_2)|}}$ factor. The reason why the good state is normalized is that for a fixed set of queries we can think of defining it as A interacting with the normalized database register using PhO instead of CPhO. This intuition works for every branch of the superposition (introduced by the adversary, not the superposition over different databases) separately. Now combining all branches together also gives a normalized state, because they origin from a valid interaction of a unitary adversary with CPhO.

4.1.4 Final Bound

Eq. (39) gives us

$$
\mathbb{P}[\text{Find}] \leq \sum_{i=1}^{q} \left\| \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Phi_{i-1}\rangle \right\|^2 . \qquad (42)
$$

We use the good state to bound the elements of the sum above as follows:

$$
\left\| \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Phi_{i-1}\rangle \right\| \leq \left\| |\Phi_{i-1}\rangle - |\Psi_{i-1}^{\text{Good}}\rangle \right\| + \left\| \mathsf{J}_R \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Psi_{i-1}^{\text{Good}}\rangle \right\| . \qquad (43)
$$

Next we bound the two norms in Eq. (43). First the distance of the good state from the original state interacting with the punctured oracle. We simplify the norm to be a

sum of small steps:

$$\begin{aligned}
&\left\| |\Psi_i^{\text{Good}}\rangle_{AD}|0\rangle_J - |\Phi_i\rangle_{ADJ} \right\| \\
&= \left\| |\Psi_i^{\text{Good}}\rangle_{AD}|0\rangle_J - \overline{\mathsf{J}}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{i-1}|\Phi_{i-1}\rangle_{ADJ} \right\| && (44)\\
&\leq \left\| |\Psi_i^{\text{Good}}\rangle_{AD}|0\rangle_J - \overline{\mathsf{J}}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{i-1}|\Psi_{i-1}^{\text{Good}}\rangle_{AD}|0\rangle_J \right\| \\
&+ \left\| \overline{\mathsf{J}}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{i-1}|\Psi_{i-1}^{\text{Good}}\rangle_{AD}|0\rangle_J - \overline{\mathsf{J}}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{i-1}|\Phi_{i-1}\rangle_{ADJ} \right\| && (45)\\
&\leq \varepsilon_{\text{step}}(i) + \left\| |\Psi_{i-1}^{\text{Good}}\rangle_{AD}|0\rangle_J - |\Phi_{i-1}\rangle_{ADJ} \right\| \leq \sum_{j=1}^{i} \varepsilon_{\text{step}}(j), && (46)
\end{aligned}$$

where we use the triangle inequality and recursively get rid of all queries made by A. The definition of a single step is the following Euclidean norm

$$\varepsilon_{\text{step}}(j) := \left\| |\Psi_j^{\text{Good}}\rangle_{AD}|0\rangle_J - \overline{\mathsf{J}}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{j-1}|\Psi_{j-1}^{\text{Good}}\rangle_{AD}|0\rangle_J \right\|_2 . \tag{47}$$

To calculate the bound on $\varepsilon_{\text{step}}(j)$ we first calculate how a query affects the good state. The full calculations are presented in Sec. A.1. Using these findings we prove Lem. 14 that states a bound on the norm of the difference of the good and original states.

We define the second part in Eq. (43) as

$$\varepsilon_{\text{Find}}(i) := \left\| \mathsf{J}_R\mathsf{H} \setminus \mathsf{V}_R\mathsf{U}_{i-1}|\Psi_{i-1}^{\text{Good}}\rangle \right\| . \tag{48}$$

Using the techniques developed to bound $\varepsilon_{\text{step}}(j)$, we bound $\varepsilon_{\text{Find}}(i)$ in Sec. A.3 and state the bounds in Lem. 16.

The final bound is

$$\mathbb{P}\left[\text{Find} : \text{A}[\mathsf{H} \setminus R]\right] \leq \sum_{i=1}^{q} \left(\sum_{j=1}^{i-1} \varepsilon_{\text{step}}(j) + \varepsilon_{\text{Find}}(i) \right)^2 , \tag{49}$$

with Lem. 14 and Lem. 16 we get the final bound. □

4.2 Simplification of the Bound

Whenever the outputs of two databases relate to one another the new entry in the good set is sampled in a way that D is not in R. If outputs of one oracle depend on the inputs of the other, adding a new entry gives a trivial attack, that we exclude. The only scenario that adding a new entry causes errors in the other database is when the other outputs depend on some random function of the new input (that is not accessible for the adversary). This is not the case for the relation that we discuss here, hence we can omit all errors to the other database in the ADD case. Namely $|\mathcal{H}_a^{\text{ADD}}(s_1, s_2)| = 0$.

Next, we simplify the additive terms $-\left(1 - \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\right)$. We can bound it by 0: the fewer the restrictions from D_a the more good $y^{\bar{a}}$ there are. Another term that we can simplify is $\text{sgn}\left(|\mathcal{H}_a^{\text{REM}}(s_1, s_2)|\right) = 1$.

To achieve a constant bound on the multiplicative term $\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}$ we proceed as

follows:

$$\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}=\prod_{k=1}^{s_{\bar{a}}}\left(\frac{N_{\bar{a}}-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right) \tag{50}$$

$$=\prod_{k=1}^{s_{\bar{a}}}\left(1+\frac{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right) \tag{51}$$

$$=\exp\left(\sum_{k=1}^{s_{\bar{a}}}\log\left(1+\frac{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right)\right) \tag{52}$$

$$\leq\exp\left(\sum_{k=1}^{s_{\bar{a}}}\frac{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right)\leq\exp\left(s_{\bar{a}}\frac{b_{\bar{a},\max}}{N_{\bar{a}}-b_{\bar{a},\max}}\right)\leq\exp(2)\leq 3^2, \tag{53}$$

where we use the bound $\log(1+x)\leq x$ and $s_{\bar{a}}\frac{b_{\bar{a},\max}}{N_{\bar{a}}-b_{\bar{a},\max}}\leq 2$, with $b_{\bar{a},\max}:=\max_{s_a,k\leq q}b_{\bar{a}}(s_a,k)$.

For bounding the part with $|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|$ we use the following derivation:

$$\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}$$

$$=\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\prod_{k=1}^{s_{\bar{a}}}\left(\frac{N_{\bar{a}}-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right)-1} \tag{54}$$

$$=\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\prod_{k=1}^{s_{\bar{a}}}\left(1+\frac{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)}{N_{\bar{a}}-b_{\bar{a}}(s_a,k)}\right)-1} \tag{55}$$

$$\leq\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\prod_{k=1}^{s_{\bar{a}}}\left(\underbrace{1+\frac{\max_{k\leq s_{\bar{a}}}\{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)\}}{N_{\bar{a}}-b_{\bar{a}}(s_a,s_{\bar{a}})}}_{\leq\exp\left(\frac{\max_{k\leq s_{\bar{a}}}\{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)\}}{N_{\bar{a}}-b_{\bar{a}}(s_a,s_{\bar{a}})}\right)}\right)-1} \tag{56}$$

$$\leq\sqrt{2\frac{2q\max_{k\leq s_{\bar{a}}}\left(b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)\right)}{N_a}}, \tag{57}$$

where the last inequality comes from bounding $e^x-1\leq 2x$ (valid for $0\leq x\leq 1$) and assuming that $\frac{N_a-b_a(s_1,s_2)}{N_{\bar{a}}-b_{\bar{a}}(s_1,s_2)}\leq 2$.

In the final expression we make the following assumptions: $\left|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)\right|=0$, $\frac{s_{\bar{a}}b_{\bar{a},\max}}{N_{\bar{a}}-b_{\bar{a},\max}}\leq 2$, $\frac{\max_{k\leq s_{\bar{a}}}\{b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)\}}{N_{\bar{a}}-b_{\bar{a}}(s_a,s_{\bar{a}})}\leq 1$, $\frac{N_a-b_a(s_1,s_2)}{N_{\bar{a}}-b_{\bar{a}}(s_1,s_2)}\leq 2$, and that b_a is a monotonously increasing function: $b_a(s_1,s_2)\leq b_a(s_a+1,s_{\bar{a}})$ and $b_a(s_1,s_2)\leq b_a(q,q)$. Given these assumptions and some straight forward simplifications we arrive at

$$\mathbb{P}\left[\text{Find}:\text{A}[\text{H}\setminus R]\right]$$

$$\leq\sum_{i=1}^{q}\left(\sum_{j=1}^{i-1}\max_{a\in\{1,2\},s_1,s_2\leq j-1}\left(9\frac{b_a(s_a+1,s_{\bar{a}})}{\sqrt{N_a(N_a-b_a(q,q))}}\right)\right.$$

$$\left.+\max_{a\in\{1,2\},s_1,s_2\leq i-1}\left(2\sqrt{\frac{b_a(s_a+1,s_{\bar{a}})}{N_a}}+3\frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a-b_a(q,q))}}\right.\right.$$

$$+\frac{b_a(s_1,s_2)^{3/2}}{N_a\sqrt{N_a-b_a(q,q)}}+\sqrt{2\frac{2q\cdot\max_{k\leq s_{\bar{a}}}\left(b_{\bar{a}}(s_a,k)-b_{\bar{a}}(s_a-1,k)\right)}{N_a}}\Bigg)\Bigg)^2. \tag{58}$$

4.3 Concrete Bound for the Rate-1/3 Relation

In the indifferentiability proof of the Rate-1/3 construction defined in Sec. 2.3 we lazy sample three functions. The generalization of Lem. 9 to $\mathsf{H}=(\mathsf{H}_1,\mathsf{H}_2,\mathsf{H}_3)$ can be done in a straight forward way, note that we do not make use of the fact that $a\in\{1,2\}$ in any place of the proof. We define the relation

$$R_{\text{Rate-1/3}}:=\{(D_1,D_2,D_3)\in\mathcal{D}_3:\exists y_1\in D_1^Y,y_2\in D_2^Y,x_3\in D_3^X,y_1=y_2\oplus x_3\}, \tag{59}$$

where[9] $\mathcal{D}_3:=\left(\bigcup_{s\in[q+1]}\left(\{0,1\}^n\times\{0,1\}^n\right)^s\right)^3$ and $\oplus$ is the bitwise XOR. We state a lemma giving a bound on the probability of Find for the triple of compressed oracles $(\mathsf{CStO}_n,\mathsf{CStO}_n,\mathsf{CStO}_n)$. The coefficients are $b_1(s_1,s_2,s_3)\leq s_2\cdot s_3$, $b_2(s_1,s_2,s_3)\leq s_1\cdot s_3$, and $b_3(s_1,s_2,s_3)=0$. The b_1 function is such, because for each output of $\mathbf{f}_1$, there are at most $s_2\cdot s_3$ sums $y_2\oplus x_3$ of outputs of $\mathbf{f}_2$ and inputs of $\mathbf{f}_3$ that can bring D to be in R. Similarly for b_2. Outputs of $\mathbf{f}_3$ do not cause D to be in relation.

We prove the corollary by using the bound from Lem. 9 simplified as in Eq. (58) with function b defined above.

Corollary 10. *For any quantum adversary* A *interacting with a punctured oracle* $(\mathsf{CStO}_n,\mathsf{CStO}_n,\mathsf{CStO}_n)\setminus R_{\text{Rate-1/3}}$*, where* $R_{\text{Rate-1/3}}$ *is defined in Eq.* (59)*, the probability of* Find *is bounded by:*

$$\begin{aligned}&\mathbb{P}\left[\text{Find}:\mathsf{A}[(\mathsf{CStO}_n,\mathsf{CStO}_n,\mathsf{CStO}_n)\setminus R_{\text{Rate-1/3}}]\right]\\&\leq 36\frac{q^3}{2^n}+84\frac{q^5}{2^n\sqrt{2^n-q^2}}+70\frac{q^7}{2^n(2^n-q^2)},\end{aligned} \tag{60}$$

where q *is the maximal number of queries made by* A.

For $q\in O\left(2^{n/4}\right)$ the bound above is just $O\left(q^3/2^n\right)$.

The above bound is essentially the classical bound on finding input-output pairs of $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ such that $y_1\oplus y_2=x_3$. The quantum bound on finding such a triple can be found by applying the above bound on $\mathbb{P}\left[\text{Find}\right]$ and Thm. 5. Both the classical and quantum bounds are tight, we prove it in Sec. 5.2.

5 Tight Indifferentiability of Rate-1/3

This section contains our main results, we prove quantum indifferentiability of Rate-1/3. Additionally we prove that our classical and quantum indifferentiability results are tight. On the one hand we show positive results, establishing that Rate-1/3 is secure with respect to a very strong notion. On the other hand we bound the indifferentiability security guarantees away from the optimal—in terms of collision resistance—level of the lower bound of $\Omega(2^{n/2})$. This means that Rate-1/3 is not as good of a compression function as we might have thought. From a more technical perspective, our work paves the way for

[9]The result also holds for $\mathcal{D}_3:=\bigcup_{s\in[q+1]}(\mathcal{X}_1\times\mathcal{Y})^s\times\bigcup_{s\in[q+1]}(\mathcal{X}_2\times\mathcal{Y})^s\times\bigcup_{s\in[q+1]}(\mathcal{Y}\times\mathcal{Y})^s$ with arbitrary finite Abelian group $\mathcal{Y}$.

similar results (i.e., proving quantum security) for other constructions that have already been proven classically indifferentiable to an optimal level [ABR21].

The proof of quantum indifferentiability is set in the quantum-game playing framework. It is structured in an almost identical way as the classical proof but uses different ingredients for the main statements. We also use quantum analogues for some of the objects the adversary interacts with.

The key object in our proof are punctured oracles, introduced in this context in [Cza+19] and of which we give a recap in Sec. 2.2. They are subroutines of the games, played by the adversary, that capture lazy-sampling and checks for Bad events. Similarly to the classical framework from [BR06], whenever Bad is set to 1, the adversary can identify the game she is playing.

Including punctured oracles especially makes sense due to Thm. 5, which is the quantum counterpart of the fundamental game-playing lemma (Lem. 2). The second distinguishability bound that we use is shown in Lem. 7. This is a relatively simple statement, that is true for games that are almost identical (Def. 6).

We note that when discussing more than two oracles that are punctured with relations that depend on all of them, we use the punctured oracle notation only on those that are directly influenced by the puncturing. The distinguishability bound, however, can only be calculated by considering all of the oracles.

5.1 Quantum Indifferentiability Proof

Quantum indifferentiability can proved in a very similar manner to the classical case, presented in Sec. 3.

Theorem 11. *The compression function* $\textsc{Rate-1/3}_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$ *for uniformly random* $\mathbf{f}_1$, $\mathbf{f}_2$, *and* $\mathbf{f}_3$ *and for* $q \in O\left(2^{n/4}\right)$ *is* (q,ε)*-quantumly indifferentiable for* $\varepsilon = \sqrt{190(q+1)\frac{q^3}{2^n}} + 760\frac{q^3}{2^n}$.

Proof. The proof of quantum indifferentiability mirrors the classical proof. Again we define two simulators, the initial S_2 that just lazy samples the compression functions and S_3 that is the actual simulator.

Algorithm 2 Quantum simulators $\boxed{\mathsf{S}_2}$, $\boxed{\mathsf{S}_3}$ for $\textsc{Rate-1/3}_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$

---

procedure $\mathbf{f}_1(x_1)$
  Apply $\boxed{\mathsf{CStO}_n}$, $\boxed{\mathsf{CStO}_n \setminus R_{\textsc{Rate-1/3}}}$ ▷ $R_{\textsc{Rate-1/3}}$ defined in (59)

procedure $\mathbf{f}_2(x_2)$
  Apply $\boxed{\mathsf{CStO}_n}$, $\boxed{\mathsf{CStO}_n \setminus R_{\textsc{Rate-1/3}}}$

procedure $\mathbf{f}_3(x_3)$
  if $\exists y_1 \in D_1^Y, y_2 \in D_2^Y : x_3 = y_1 \oplus y_2$ **then**
    $\boxed{\text{Apply } \mathsf{CStO}_n}$
    $\boxed{\textbf{return } \mathsf{R}(x_1,x_2) \oplus y_1}$
  else
    Apply CStO_n

Game 1 The interface in the first game is $(\textsc{Rate-1/3}, (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3))$, where $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$ are

uniformly random functions. The definition of the game is

$$\textbf{Game 1} := (b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1/3}, (\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3)]) \,. \tag{61}$$

Game 2 In the second step we lazy sample the compression functions, the interface is $(\textsc{Rate-1/3}, \mathsf{S}_2)$. The new game is

$$\textbf{Game 2} := (b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1/3}, \mathsf{S}_2]) \,. \tag{62}$$

By the fact that the compressed oracles are indistinguishable from random oracles (see e.g. Theorem 7 in [Cza+19]), this change of the interface is indistinguishable for A:

$$\left|\mathbb{P}\left[\textbf{Game 2}\right] - \mathbb{P}\left[\textbf{Game 1}\right]\right| = 0. \tag{63}$$

Game 3 The interface in the third game is $(\textsc{Rate-1/3}, \mathsf{S}_3)$, where we introduce the punctured oracle and R, introducing the random oracle does not change the distribution of the outputs of $\mathbf{f}_3$ so this change does not add to the distinguishability advantage. The new game is defined as

$$\textbf{Game 3} := (b = 1 : b \leftarrow \mathrm{A}[\textsc{Rate-1/3}, \mathsf{S}_3]) \,. \tag{64}$$

We puncture on the same events as in the classical proof, relation $R_{\textsc{Rate-1/3}}$is defined in Eq. (59). The only noticeable change for the adversary is the punctured oracle. The distinguishing advantage can be bounded by the O2H lemma, Thm. 5:

$$\left|\mathbb{P}\left[\textbf{Game 3}\right] - \mathbb{P}\left[\textbf{Game 2}\right]\right| \leq \sqrt{(q+1)\mathbb{P}\left[\mathrm{Find}\right]} \leq \sqrt{190(q+1)\frac{q^3}{2^n}}, \tag{65}$$

where the bound on $\mathbb{P}\left[\mathrm{Find}\right]$ comes from Corollary 10.

Game 4 In the last step of this proof the interface is $(\mathsf{R}, \mathsf{S}_3)$, we change the private interface, the definition of the game is

$$\textbf{Game 3} := (b = 1 : b \leftarrow \mathrm{A}[\mathsf{R}, \mathsf{S}_3]) \,. \tag{66}$$

Similar to the classical case we have:

$$\left|\mathbb{P}\left[\textbf{Game 4} \mid \neg\mathrm{Find}\right] - \mathbb{P}\left[\textbf{Game 3} \mid \neg\mathrm{Find}\right]\right| = 0. \tag{67}$$

Using the above identity we derive the final distinguishing advantage:

$$\left|\mathbb{P}\left[\textbf{Game 4}\right] - \mathbb{P}\left[\textbf{Game 3}\right]\right| \leq 4\mathbb{P}\left[\mathrm{Find}\right] \leq 760\frac{q^3}{2^n}, \tag{68}$$

where we use Lem. 7, Find is the event of finding the relation in **Game 3**, we bound the bound from the lemma by the bigger of the two probabilities.

The last game includes the random oracle in the private interface, which concludes the proof. □

5.2 Indifferentiability Attacks

In the last paragraphs of this section we present the attacks on the indifferentiability of RATE-1/3. Their complexity matches the security bounds from the previous paragraphs, proving their tightness. Comparing them with the results of [SS08], the attacks show that the construction does not maintain its efficiency when moving from collision resistance to indifferentiability In the recent [ABR21], the authors discuss an optimally collision-resistant construction that (with a small modification) also is indifferentiable from a random oracle with the same distinguishing advantage. Interestingly, they also present an attack that proves tightness of their result. Inspired by their approach we present similar proofs for tightness of our indifferentiability bounds.

For an exponential distinguisher we assume the simulator makes at most polynomially many queries to R per distinguisher query.

5.2.1 Classical Attack

We describe a classical distinguisher $\mathrm{D_c}$ that wins the indifferentiability game with constant probability after making $3 \cdot 2^{n/3} + 1$ queries. We say that there is a collision in a list of input-output pairs if there are two distinct pairs with the second entries equal each other. The classical distinguisher works as follows:

1. Query $\mathbf{f}_3$ with $2^{n/3}$ uniformly random distinct values and save all the input-output pairs in list $\mathcal{L}_3$.
2. Query $\mathbf{f}_1, \mathbf{f}_2$ with $2^{n/3}$ uniformly random distinct values each and save all the input-output pairs in lists $\mathcal{L}_1, \mathcal{L}_2$ respectively.
3. If there is any collision in $\mathcal{L}_1$ or $\mathcal{L}_2$, then output 1.
4. If there is no triple $(y_1, y_2, x_3) \in \mathcal{L}_1^Y \times \mathcal{L}_2^Y \times \mathcal{L}_3^X$ such that $y_1 \oplus y_2 = x_3$, then output 1.
5. Say that $((x_1, y_1), (x_2, y_2), (x_3, y_3)) \in \mathcal{L}_1 \times \mathcal{L}_2 \times \mathcal{L}_3$ is such that $y_1 \oplus y_2 = x_3$. If $\mathsf{R}(x_1, x_2) \neq y_1 \oplus y_3$ (where R is the random oracle in the ideal world and the construction in the real world), then output 1.
6. If at any point there were any inconsistencies in $\mathbf{f}_1$, $\mathbf{f}_2$, or $\mathbf{f}_3$, then output 1, otherwise output 0.

Theorem 12. *The distinguisher* $\mathrm{D_c}$ *described above achieves constant distinguishing advantage in the classical indifferentiability game of* $\textsc{Rate-1}/3_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$ *for any simulator after making* $3 \cdot 2^{n/3} + 1$ *classical queries.*

Proof. If $\mathbf{f}_1$, $\mathbf{f}_2$ are uniformly random, then with high probability all y_1 and y_2 are distinct. So in the real world $\mathrm{D_c}$ does not output 1 in Point 3 with high probability. This can be seen using the standard collision-finding bound (Appendix A.4 in [KL14]):

$$\mathbb{P}\left[\text{At least 1 collision in } q \text{ samples from } \{0,1\}^n\right] \leq \frac{q^2}{2^{n+1}}, \tag{69}$$

where the probability is over the q uniformly random samples. Note that for $q = 2^{n/3}$ the probability of seeing any collisions is negligible.

In what follows we assume that there are no collisions in $\mathcal{L}_1$ and $\mathcal{L}_2$. We write Bad to denote the event that there is a triple $(y_1, y_2, x_3) \in \mathcal{L}_1^Y \times \mathcal{L}_2^Y \times \mathcal{L}_3^X$ such that $y_1 \oplus y_2 = x_3$. The probability that there is at least one $x_3 = y_1 \oplus y_2$ can be calculated by going through $\mathcal{L}_3^X$ one by one and checking if the equation holds for any y_1 and y_2. To calculate the probability of Bad given no collisions in $\mathcal{L}_1$ and $\mathcal{L}_2$—i.e. the probability that in the real world $\mathsf{D_c}$ does not output 1 in Point 4—we use the proof of Lemma 4 from [ABR21] for $k = 1$ and a fixed a. Formally we have:

$$\begin{aligned}
&\mathbb{P}\left[\text{Bad} \mid \text{No collisions in } \mathcal{L}_1, \mathcal{L}_2\right] \\
&= \sum_{x_3 \in \mathcal{L}_3^X} \mathbb{P}\left[y_1 \oplus y_2 = x_3 : y_1 \in \mathcal{L}_1, y_2 \in \mathcal{L}_2 \mid \text{No collisions in } \mathcal{L}_1, \mathcal{L}_2\right] \qquad (70) \\
&= \sum_{x_3 \in \mathcal{L}_3^X} \frac{(q!)^2(2^n-1)!}{((q-1)!)^2 2^n!} = \frac{q^3}{2^n} = 1. \qquad (71)
\end{aligned}$$

The equality above works under the assumption that there are no collisions in $\mathcal{L}_1^Y$ and $\mathcal{L}_2^Y$, the final bound on $\mathbb{P}\left[\text{Bad}\right]$ reads:

$$\begin{aligned}
&\mathbb{P}\left[\text{Bad}\right] = \mathbb{P}\left[\text{Bad} \wedge \text{No collisions in } \mathcal{L}_1, \mathcal{L}_2\right] \\
&+ \mathbb{P}\left[\text{Bad} \wedge \text{At least 1 collision in } \mathcal{L}_1 \text{ or } \mathcal{L}_2\right] \qquad (72) \\
&\geq \mathbb{P}\left[\text{No collisions in } \mathcal{L}_1, \mathcal{L}_2\right] \geq \left(1 - \frac{q^2}{2^{n+1}}\right)^2. \qquad (73)
\end{aligned}$$

Given the bound above, in Point 5 in the real world the distinguisher sees a triple of input-output pairs such that $y_1 \oplus y_2 = x_3$ with overwhelming probability[10]. In the real world equality $\mathsf{R}(x_1, x_2) = \textsc{Rate-1}/3_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}(x_1, x_2) = y_1 \oplus y_3$ holds and $\mathsf{D_c}$ does not output anything at that point.

On the other hand, the probability that in the ideal world $\mathsf{D_c}$ does not output 1 in Point 5 is negligible. The simulator has already committed to the values of y_3 and the queries to $\mathbf{f}_1$ and $\mathbf{f}_2$ are uniformly random (and hence unpredictable, guessing them has negligible chance of success). Moreover, the simulator makes $\mathbf{p}(n)2^{n/3}$ queries, where $\mathbf{p}$ is some polynomial. The probability that any of the (x_1, x_2) pairs yields $y_3 \oplus y_1$, for any y_3 and any y_1 corresponding to x_1, when queried to R is upper-bounded by $\mathbf{p}(n)2^{n/3}\frac{2^{n/3}}{2^n}$, which is negligible. This concludes our proof, any simulator that provides consistent answers will fail the check in Point 5 and any inconsistent answers are caught by the last point of $\mathsf{D_c}$. □

5.2.2 Quantum Attack

We describe a quantum distinguisher $\mathsf{D_q}$ that wins the indifferentiability game with constant probability after making $3 \cdot 2^{n/4} + 1$ quantum queries:

1. Make $2^{n/4}$ uniformly random distinct classical queries to $\mathbf{f}_2$ and $\mathbf{f}_3$ and save all the input-output pairs in list $\mathcal{L}_2, \mathcal{L}_3$ respectively.

2. If there is any collision in $\mathcal{L}_2$, then output 1.

[10] By overwhelming we mean $1 -$ negligible.

3. Run amplitude amplification algorithm [BH97] (this is a generalization of Grover's algorithm [Gro96]), making $2^{n/4}$ quantum queries to $\mathbf{f}_1$, to find a preimage of $y_2 \oplus x_3$ under $\mathbf{f}_1$ for any $(y_2, x_3) \in \mathcal{L}_2^Y \times \mathcal{L}_3^X$. If the preimage search does not succeed, then output 1.

4. Say that the preimage search output (x_1, y_1) and $((x_2, y_2), (x_3, y_3)) \in \mathcal{L}_1 \times \mathcal{L}_2 \times \mathcal{L}_3$ are such that $y_2 \oplus x_3 = y_1$. Then if $\mathsf{R}(x_1, x_2) \neq y_1 \oplus y_3$ (where and R is the random oracle in the ideal world and the construction in the real world), then output 1.

5. If at any point there were any inconsistencies in $\mathbf{f}_1$, $\mathbf{f}_2$, or $\mathbf{f}_3$, then output 1, otherwise output 0.

Theorem 13. *The distinguisher* $\mathsf{D_q}$ *described above achieves constant distinguishing advantage in the quantum indifferentiability game of* $\textsc{Rate-1}/3_{\mathbf{f}_1,\mathbf{f}_2,\mathbf{f}_3}$ *for any simulator after making* $2 \cdot 2^{n/4} + 1$ *classical queries and* $2^{n/4}$ *quantum queries.*

Proof. The above algorithm is a successful distinguisher because in the real world the preimage search succeeds with high probability. If it does succeed in the ideal world though, we still check if the committed value matches the random oracle.

Let us start by evaluating the probability of outputting 1 in Point 2 if $\mathsf{D_q}$ is interacting with the real world. Using the bound from Eq. (69) with $q = 2^{n/4}$ we get the following bound on the probability that all y_2 are distinct:

$$\mathbb{P}\left[\text{No collisions in } \mathcal{L}_2\right] \geq 1 - \frac{(2^{n/4}-1)2^{n/4}}{2^{n+1}}. \tag{74}$$

To analyze the probability of outputting 1 in Point 3 of $\mathsf{D_q}$, we calculate the number of targets for the preimage search. We have $2^{n/4}$ values of y_2 and $2^{n/4}$ values x_3. Given the assumption of distinctness of all values in $\mathcal{L}_2^Y$ we can use the bound on the probability of collisions in the set $\{y_2 \oplus x_3 : y_2 \in \mathcal{L}_2^Y, x_3 \in \mathcal{L}_3^X\}$ that we denote as $\mathcal{L}_2^Y \oplus \mathcal{L}_3^X$:

$$\begin{aligned}&\mathbb{P}\left[\text{There is a collision in } \mathcal{L}_2^Y \oplus \mathcal{L}_3^X \mid \text{No collisions in } \mathcal{L}_2\right] \\ &\leq \frac{(q!)^2 2^n (2^n-2)!}{(q-2)! \cdot 2! \cdot 2^n!} = \frac{(q-1)q}{2(2^n-1)},\end{aligned} \tag{75}$$

the above bound is the statement of Lemma 3 for $k = 2$ in [SS08]. Taking the above two bounds into account, the probability that $\mathsf{D_q}$ outputs 1 in the real world is negligible.

Under the assumption that there are no collisions in $\mathcal{L}_2^Y$ and in $\mathcal{L}_2^Y \oplus \mathcal{L}_3^X$, the number of possible values we want to find a preimage of is $2^{n/4} \cdot 2^{n/4} = 2^{n/2}$. In the real world, every y_1 is distributed uniformly at random, so the probability of any output of $\mathbf{f}_1$ being a preimage of $\mathcal{L}_2^Y \oplus \mathcal{L}_3^X$ is $\frac{2^{n/2}}{2^n}$. The number of queries required to achieve constant probability of finding a preimage is the square root of the inverse of the probability that a given function output is marked (so in our case equals $y_2 \oplus x_3$). In our setting $\mathsf{D_q}$ has to do $\frac{1}{\sqrt{2^{n/2}/2^n}} = 2^{n/4}$ queries to $\mathbf{f}_1$. For more details on the amplitude amplification algorithm we also direct to [Wol11].

Up to Point 4, in the real world $\mathsf{D_q}$ does not output 1 with constant probability (coming from the success probability of amplitude amplification algorithm).

In the ideal world and after all the queries to $\mathbf{f}_2$ and $\mathbf{f}_3$, the distinguisher outputs 1 with overwhelming probability. The simulator commits to values y_3 and y_2. Moreover, by inspecting the amplitude amplification algorithm more carefully, we see that $\mathsf{D_q}$ gets

a uniformly random solution to the search problem, hence the simulator can predict x_1 in the earlier stages with only negligible probability. The probability that the quantum simulator finds x_1 that yields $\mathsf{R}(x_1, x_2) = y_2 \oplus x_3 \oplus y_3$, where we set $y_1 = y_2 \oplus x_3$, for some fixed $(x_2, y_2) \in \mathcal{L}_2$ and any $(x_3, y_3) \in \mathcal{L}_3$ is upper-bounded by $(\mathbf{p}(n)2^{n/4})^2 \frac{2^{n/4}}{2^n}$. We arrive at this bound by using Theorem 1 in [HRS16] and setting $\lambda = \frac{2^{n/4}}{2^n}$, the probability that R outputs $y_2 \oplus x_3 \oplus y_3$ for some $(x_3, y_3) \in \mathcal{L}_3$, and the number of queries the simulator can make is $q = \mathbf{p}(n)2^{n/4}$, where $\mathbf{p}$ is a polynomial. Hence, no matter in what way the simulator modifies the distribution of $\mathbf{f}_1$, she has a negligible probability of providing an answer to Point 4 that fulfills the $\mathsf{R}(x_1, x_2) \neq y_1 \oplus y_3$ check. □

6 Conclusions

We expand the quantum game-playing technique from [Cza+19] to include multiple functions. This is an important development as many constructions do not just reuse a single internal function. Our bound on $\mathbb{P}\left[\text{Find}\right]$ is general enough to allow a simple evaluation of the distinguishing advantage for a plethora of constructions.

Following the work of [ABR21] we show a distinguisher that tightens the security bounds, proving that the construction of Shrimpton and Stam does not yield an optimally secure compression function in terms of the notion of indifferentiability. For future work we leave the interesting question of proving optimal *quantum* collision-resistance of the construction. Another open problem is bringing the results of [ABR21] to the post-quantum world, namely proving quantum indifferentiability (and collision-resistance) of their ABR+ construction.

The open questions that we leave, especially these concerning indifferentiability, will be considerably easier to answer now, that we have a versatile tool to prove quantum security of construction involving many independent functions.

7 Acknowledgments

Majority of this work was done when the author was a PhD candidate at the University of Amsterdam, where he was supported by an NWO VIDI grant (Project No. 639.022.519). The author thanks Christian Schaffner for constructive feedback on important parts of the manuscript.

References

[AHU19] Andris Ambainis, Mike Hamburg, and Dominique Unruh. "Quantum Security Proofs Using Semi-classical Oracles". In: *CRYPTO 2019*. 2019, pp. 269–295. DOI: 10.1007/978-3-030-26951-7_10. URL: https://eprint.iacr.org/2018/904 (cit. on pp. 3, 6, 8).

[ABR21] Elena Andreeva, Rishiraj Bhattacharyya, and Arnab Roy. "Compactness of Hashing Modes and Efficiency Beyond Merkle Tree". In: *Advances in Cryptology – EUROCRYPT 2021*. Springer International Publishing, 2021, pp. 92–123. DOI: 10.1007/978-3-030-77886-6_4 (cit. on pp. 3, 21, 23, 24, 26).

[BR93] Mihir Bellare and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols". In: *Proceedings of the 1st ACM conference on Computer and communications security*. ACM. 1993, pp. 62–73. DOI: 10.1145/168588.168596 (cit. on p. 3).

[BR06] Mihir Bellare and Phillip Rogaway. "The Security of Triple Encryption and a Framework for Code-Based Game-Playing Proofs". In: *EUROCRYPT 2006*. https://eprint.iacr.org/2004/331. Springer Berlin Heidelberg, 2006, pp. 409–426. DOI: 10.1007/11761679_25 (cit. on pp. 5, 21).

[Bon+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random Oracles in a Quantum World". In: *Advances in Cryptology – ASIACRYPT 2011*. LNCS 7073. 2011, pp. 41–69. DOI: 10.1007/978-3-642-25385-0_3 (cit. on p. 3).

[BH97] G. Brassard and P. Hoyer. "An exact quantum polynomial-time algorithm for Simon's problem". In: *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems*. 1997, pp. 12–23. DOI: 10.1109/ISTCS.1997.595153 (cit. on p. 25).

[CEV20] Céline Chevalier, Ehsan Ebrahimi, and Quoc Huy Vu. "On the Security Notions for Encryption in a Quantum World". In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 237. URL: https://eprint.iacr.org/2020/237 (cit. on p. 6).

[Chu+20a] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. "On the Compressed-Oracle Technique, and Post-Quantum Security of Proofs of Sequential Work". In: *arXiv preprint arXiv:2010.11658* (2020) (cit. on pp. 3, 6).

[Chu+20b] Kai-Min Chung, Siyao Guo, Qipeng Liu, and Luowen Qian. "Tight Quantum Time-Space Tradeoffs for Function Inversion". In: *Electron. Colloquium Comput. Complex.* 27 (2020), p. 90. URL: https://arxiv.org/pdf/2006.05650.pdf (cit. on p. 6).

[Cor+05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. "Merkle-Damgård Revisited: How to Construct a Hash Function". In: *Advances in Cryptology – CRYPTO 2005*. Springer Berlin Heidelberg, 2005, pp. 430–448. DOI: 10.1007/11535218_26 (cit. on p. 5).

[Cza+19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. "Quantum Lazy Sampling and Game-Playing Proofs for Quantum Indifferentiability". In: *CoRR* abs/1904.11477 (2019). arXiv: 1904.11477. URL: http://arxiv.org/abs/1904.11477 (cit. on pp. 3, 5, 6, 7, 8, 12, 15, 21, 22, 26, 37).

[Gro96] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM. 1996, pp. 212–219 (cit. on pp. 3, 25).

[HI19] Akinori Hosoyamada and Tetsu Iwata. "4-Round Luby-Rackoff Construction is a qPRP". In: *ASIACRYPT 2019*. 2019, pp. 145–174. DOI: 10.1007/978-3-030-34578-5_6. URL: https://doi.org/10.1007/978-3-030-34578-5%5C_6 (cit. on p. 6).

[HRS16] Andreas Hülsing, Joost Rijneveld, and Fang Song. "Mitigating Multi-Target Attacks in Hash-based Signatures". In: *Public Key Cryptography – PKC 2016*. Vol. 9614. Springer-Verlag Berlin Heidelberg, 2016, pp. 387–416. DOI: 10.1007/978-3-662-49384-7_15. URL: https://eprint.iacr.org/2015/1256 (cit. on p. 26).

[JZM19] Haodong Jiang, Zhenfeng Zhang, and Zhi Ma. "Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model". Cryptology ePrint Archive, Report 2019/134. https://eprint.iacr.org/2019/134. 2019 (cit. on p. 6).

[Kap+16] Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and Marıéa Naya-Plasencia. "Breaking symmetric cryptosystems using quantum period finding". In: *Annual Cryptology Conference*. Springer. 2016, pp. 207–237 (cit. on p. 3).

[KL14] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC Cryptography and Network Security Series. Taylor & Francis, 2014 (cit. on p. 23).

[LM17] Gregor Leander and Alexander May. "Grover Meets Simon – Quantumly Attacking the FX-construction". In: *ASIACRYPT 2017*. Springer International Publishing, 2017, pp. 161–178. DOI: 10.1007/978-3-319-70697-9_6 (cit. on p. 3).

[MT07] Ueli M. Maurer and Stefano Tessaro. "Domain Extension of Public Random Functions: Beyond the Birthday Barrier". In: *CRYPTO 2007*. Springer, 2007, pp. 187–204. DOI: 10.1007/978-3-540-74143-5_11 (cit. on p. 3).

[MRH04] Ueli Maurer, Renato Renner, and Clemens Holenstein. "Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology". In: *Theory of Cryptography*. Springer Berlin Heidelberg, 2004, pp. 21–39. DOI: 10.1007/978-3-540-24638-1_2 (cit. on pp. 3, 5).

[MP12] Bart Mennink and Bart Preneel. "Hash Functions Based on Three Permutations: A Generic Security Analysis". In: *Advances in Cryptology – CRYPTO 2012*. Springer Berlin Heidelberg, 2012, pp. 330–347. ISBN: 978-3-642-32009-5. DOI: 10.1007/978-3-642-32009-5_20 (cit. on p. 3).

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10th anniversary. Cambridge: Cambridge University Press, 2010. ISBN: 978-1107002173 (cit. on p. 4).

[NIS14] NIST. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Draft FIPS 202. 2014. URL: http://csrc.nist.gov/publications/drafts/fips-202/fips_202_draft.pdf (cit. on p. 4).

[NIS15] NIST. *Secure Hash Standard (SHS)*. Draft FIPS 180-4. 2015. DOI: 10.6028/NIST.FIPS.180-4 (cit. on p. 4).

[RS08] Phillip Rogaway and John Steinberger. "Constructing Cryptographic Hash Functions from Fixed-Key Blockciphers". In: *Advances in Cryptology – CRYPTO 2008*. Springer Berlin Heidelberg, 2008, pp. 433–450. ISBN: 978-3-540-85174-5. DOI: 10.1007/978-3-540-85174-5_24 (cit. on p. 3).

[Ros21] Ansis Rosmanis. "Tight bounds for inverting permutations via compressed oracle arguments". In: *arXiv preprint arXiv:2103.08975* (2021). URL: https://arxiv.org/abs/2103.08975 (cit. on p. 6).

[SS16] Thomas Santoli and Christian Schaffner. "Using Simon's algorithm to attack symmetric-key cryptographic primitives". In: *arXiv preprint arXiv:1603.07856* (2016) (cit. on p. 3).

[Sho94] Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700. URL: https://doi.org/10.1109/SFCS.1994.365700 (cit. on p. 3).

[SS08] Thomas Shrimpton and Martijn Stam. "Building a Collision-Resistant Compression Function from Non-compressing Primitives". In: *Automata, Languages and Programming*. 2008, pp. 643–654. DOI: 10.1007/978-3-540-70583-3_52. URL: https://eprint.iacr.org/2007/409 (cit. on pp. 3, 8, 9, 23, 25).

[Sim97] Daniel R Simon. "On the power of quantum computation". In: *SIAM journal on computing* 26.5 (1997), pp. 1474–1483. DOI: 10.1137/S0097539796298637 (cit. on p. 3).

[Unr14] Dominique Unruh. "Revocable Quantum Timed-Release Encryption". In: *EUROCRYPT 2014*. Springer Berlin Heidelberg, 2014, pp. 129–146. DOI: 10.1007/978-3-642-55220-5_8 (cit. on pp. 3, 6, 8).

[Wol11] Ronald de Wolf. "Quantum computing: Lecture notes". In: *University of Amsterdam* (2011). URL: https://arxiv.org/abs/1907.09415 (cit. on pp. 4, 25).

[Zha19] Mark Zhandry. "How to Record Quantum Queries, and Applications to Quantum Indifferentiability". In: *CRYPTO 2019*. Springer International Publishing, 2019, pp. 239–268. ISBN: 978-3-030-26951-7. DOI: 10.1007/978-3-030-26951-7_9 (cit. on pp. 3, 5, 6, 7).

A Additional Details on the Proof of Lem. 9

In the following paragraphs we provide all the necessary details to prove Lem. 9.

A.1 $|\Psi_{i-1}^{\text{Good}}\rangle$ after a query

To prove the main technical lemmas of this section we analyze how a single query to the oracle affects the good state. We provide a detailed expression of the state after the query.

To prove Lem. 14 we analyze how far apart the state $|\Psi_{i-1}^{\text{Good}}\rangle$ is after a query from $|\Psi_{i}^{\text{Good}}\rangle$. To achieve this goal we inspect the state $\mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Psi_{i-1}^{\text{Good}}\rangle_{AD} |0\rangle_J$. We distinguish different modes of operation: ADD when the queried x is added to D, UPD when x was already in D and is not removed from the database, REM when we remove x from D, and NOT where there is no change in the database because register A^Y is in state $|0\rangle$. These modes correspond to different branches of superposition in $\mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Psi_{i-1}^{\text{Good}}\rangle_{AD} |0\rangle_J$. We write

$$\mathsf{U}_{i-1} |\Psi_{i-1}^{\text{Good}}\rangle_{AD} = |\xi_{i-1}(\text{ADD})\rangle + |\xi_{i-1}(\text{UPD})\rangle + |\xi_{i-1}(\text{REM})\rangle + |\xi_{i-1}(\text{NOT})\rangle \tag{76}$$

and analyze the action of $\mathsf{H} \setminus \mathsf{V}_R$ on the above states separately.

For $|\xi_{i-1}(\text{NOT})\rangle$ there is no change to the state. For $|\xi_{i-1}(\text{UPD})\rangle$ and $|\xi_{i-1}(\text{REM})\rangle$, we treat the updated x as the last one in D_a, this does not have to be true but it simplifies

notation. Note that we want the corresponding y_{s_a} to depend on previous queries to H_a. This assumption is without loss of generality as there is no fixed order for $\sum_{\vec{y}_a}$ in Eq. (41). The empty register is moved to the back of D, we do not write it out for simplicity but still consider it done.

After querying $|\Psi_{i-1}^{\text{Good}}\rangle|0\rangle_J$ we encounter states multiplied by $|0\rangle_J$ that do not appear in the definition of the good state and those multiplied by $|1\rangle_J$. We call these vectors *errors*. We mark the former errors by a superscript Bad and the latter with Find, note that indeed all branches of superposition that have $|1\rangle_J$ increase $\mathbb{P}\left[\text{Find}\right]$.

In general, a query to H_a can cause errors in D_a and $D_{\bar{a}}$. The former results from, e.g., adding a new entry to D_a; We sample a uniform entry and some values bring D_a to be in relation. The latter errors occur when the set of good outputs in $D_{\bar{a}}$ changes after we, e.g., add a new entry to D_a. The rule we follow is that $y^a_{s_a}$ is the last value sampled. The second rule is that all values can be sampled one by one, query by query. These rules imply that we can sample $\vec{y}_a$ first, then $\vec{y}_{\bar{a}}$, then $y^a_{s_a}$. This reasoning, however does not apply to relations that depend on inputs, so whenever contents of D^X (a or $\bar{a}$) changes we need to make up for it by changing the set we sample $\vec{y}_{\bar{a}}$ from.

Adding a new entry to a database results in setting the register corresponding to (x,a) to $\sum_{y^a_{s_a+1}\in[N_a]}\frac{1}{\sqrt{N_a}}\omega_{N_a}^{\eta y^a_{s_a+1}}|x,y^a_{s_a+1}\rangle$, just as expected from a phase oracle for the uniform distribution. As we mentioned earlier, there are errors in two databases, D_a and $D_{\bar{a}}$. First we go over the errors in D_a and leave $D_{\bar{a}}$ unchanged. In the equality that follows we single out all the branches of superposition that are not parts of $|\xi_i(\text{ADD})\rangle$:

$$
\begin{aligned}
\mathsf{H}|\xi_{i-1}(\text{ADD})\rangle = &\sum_{x,\eta,a,\vec{x},\vec{\eta},w}\alpha_{x,\eta,a,\vec{x},\vec{\eta},w}|x,\eta,a\rangle_{A^{XYI}}|\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W}\\
&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_1,\vec{x}_2)}\frac{1}{\sqrt{|\mathcal{G}_a(s_1,s_2)|}}\omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a}|(x_1^a,y_1^a),\ldots,(x^a_{s_a},y^a_{s_a})\rangle_{D_a(\vec{x}_a)}\\
&\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)}\frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\left(\frac{1}{\sqrt{N_a}}\sum_{y^a_{s_a+1}\notin\mathcal{B}_a(x|D(\vec{x}))}\omega_{N_a}^{\eta y^a_{s_a+1}}|x,y^a_{s_a+1}\rangle_{D_a(x)}\right.\\
&\left.+\frac{1}{\sqrt{N_a}}\sum_{y^a_{s_a+1}\in\mathcal{B}_a(x|D(\vec{x}))}\omega_{N_a}^{\eta y^a_{s_a+1}}|x,y^a_{s_a+1}\rangle_{D_a(x)}\right)\\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}}|(x_1^{\bar{a}},y_1^{\bar{a}}),\ldots,(x^{\bar{a}}_{s_{\bar{a}}},y^{\bar{a}}_{s_{\bar{a}}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})}\\
&\sum_{y^a_{s_a+2},\ldots,y^a_q\in[N_a]}\frac{1}{\sqrt{N_a^{q-s_a-1}}}|(\perp,y^a_{s_a+2}),\ldots,(\perp,y^a_q)\rangle_{D_a(\perp)}\\
&\sum_{y^{\bar{a}}_{s_{\bar{a}}+1},\ldots,y^{\bar{a}}_q\in[N_{\bar{a}}]}\frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}}|(\perp,y^{\bar{a}}_{s_{\bar{a}}+1}),\ldots,(\perp,y^{\bar{a}}_q)\rangle_{D_{\bar{a}}(\perp)}.
\end{aligned}
\tag{77}
$$

Errors that are left to be analyzed come from $D_{\bar{a}}$, let us present the split in the sum over $\vec{y}_a$ that we will use in the ADD case:

$$
\begin{aligned}
&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_1,\vec{x}_2)}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)}\sum_{y^a_{s_a+1}\notin\mathcal{B}_a(x|D(\vec{x}))}\\
&=\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_1,\vec{x}_2)}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_a\cup\{x\},\vec{x}_{\bar{a}}|\vec{y}_a)}\sum_{y^a_{s_a+1}\notin\mathcal{B}_a(x|D(\vec{x}))}\\
&+\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_1,\vec{x}_2)}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)\setminus\mathcal{G}_{\bar{a}}(\vec{x}_a\cup\{x\},\vec{x}_{\bar{a}}|\vec{y}_a)}\sum_{y^a_{s_a+1}\notin\mathcal{B}_a(x|D(\vec{x}))}.
\end{aligned}
\tag{78}
$$

Next we include the full impact of V_R. Two things happen in the expression below. First we split the sum over $\vec{y}_{\bar{a}}$ in the first element in the parentheses, secondly we rewrite the normalization factors to simplify the analysis later on. We underline parts of the state that are important later on. With red color we denote the errors. After applying $\mathsf{Queries}^\dagger \circ \mathsf{V}_R \circ \mathsf{Queries}$ the state is:

$$
\begin{aligned}
&\mathsf{ADD} : \mathsf{H} \setminus \mathsf{V}_R |\xi_{i-1}(\mathsf{ADD})\rangle|0\rangle_J \\
&= \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w} |x,\eta,a\rangle_{A^{XYI}} |\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W} \\
&\sum_{\vec{y}_a \in \mathcal{G}_a(\vec{x}_1,\vec{x}_2)} \frac{1}{\sqrt{|\mathcal{G}_a(s_1,s_2)|}} \omega_{N_a}^{\vec{\eta}_a \cdot \vec{y}_a} |(x_1^a, y_1^a), \ldots, (x_{s_a}^a, y_{s_a}^a)\rangle_{D_a(\vec{x}_a)} \\
&\Bigg(\sqrt{\frac{N_a - b_a(s_a+1, s_{\bar{a}})}{N_a}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \\
&\underbrace{\sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_a \cup \{x\}, \vec{x}_{\bar{a}} | \vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_a+1, s_{\bar{a}})|}}}_{\text{(i) } |\Psi_i^{\text{Good}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \\
&\underbrace{\sum_{y_{s_a+1}^a \notin \mathcal{B}_a(x|D(\vec{x}))} \frac{1}{\sqrt{N_a - b_a(s_a+1, s_{\bar{a}})}} \omega_{N_a}^{\eta y_{s_a+1}^a} |x, y_{s_a+1}^a\rangle_{D_a(x)} |0\rangle_J}_{\text{(ii) } |\Psi_i^{\text{Good}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \\
&+ \underbrace{\sqrt{\frac{N_a - b_a(s_a+1, s_{\bar{a}})}{N_a}} \sqrt{\frac{|\mathcal{H}_a^{\mathsf{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \sum_{\vec{y}_{\bar{a}} \in \mathcal{H}_a^{\mathsf{ADD}}(\vec{x}_1,\vec{x}_2,\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{H}_a^{\mathsf{ADD}}(s_1,s_2)|}}}_{\text{(i) } |\Psi_{i,1}^{\text{Find}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \\
&\underbrace{\sum_{y_{s_a+1}^a \notin \mathcal{B}_a(x|D(\vec{x}))} \frac{1}{\sqrt{N_a - b_a(s_a+1, s_{\bar{a}})}} \omega_{N_a}^{\eta y_{s_a+1}^a} |x, y_{s_a+1}^a\rangle_{D_a(x)} |1\rangle_J}_{\text{(ii) } |\Psi_{i,1}^{\text{Find}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \\
&+ \underbrace{\sqrt{\frac{b_a(s_a+1, s_{\bar{a}})}{N_a}} \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}}_{\text{(i) } |\Psi_{i,2}^{\text{Find}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \\
&\underbrace{\sum_{y_{s_a+1}^a \in \mathcal{B}_a(x|D(\vec{x}))} \frac{1}{\sqrt{b_a(s_a+1, s_{\bar{a}})}} \omega_{N_a}^{\eta y_{s_a+1}^a} |x, y_{s_a+1}^a\rangle_{D_a(x)} |1\rangle_J}_{\text{(ii) } |\Psi_{i,2}^{\text{Find}}(\mathsf{ADD}, a, s_1, s_2)\rangle} \Bigg) \\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}} \cdot \vec{y}_{\bar{a}}} |(x_1^{\bar{a}}, y_1^{\bar{a}}), \ldots, (x_{s_{\bar{a}}}^{\bar{a}}, y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})} \\
&\sum_{y_{s_a+2}^a, \ldots, y_q^a \in [N_a]} \frac{1}{\sqrt{N_a^{q-s_a-1}}} |(\perp, y_{s_a+2}^a), \ldots, (\perp, y_q^a)\rangle_{D_a(\perp)} \\
&\sum_{y_{s_{\bar{a}}+1}^{\bar{a}}, \ldots, y_q^{\bar{a}} \in [N_{\bar{a}}]} \frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}} |(\perp, y_{s_{\bar{a}}+1}^{\bar{a}}), \ldots, (\perp, y_q^{\bar{a}})\rangle_{D_{\bar{a}}(\perp)},
\end{aligned}
\tag{79}
$$

where the appropriate position of register J is after D. The size of the domain of $\vec{y}_{\bar{a}}$ is

denoted by

$$|\mathcal{G}_{\bar{a}}(s_a+1, s_{\bar{a}})| := |\mathcal{G}_{\bar{a}}(\vec{x}_a \cup \{x\}, \vec{x}_{\bar{a}} \mid \vec{y}_a)|\,, \tag{80}$$

which uses our assumption that the size of the good set does not depend on the actual values stored in D, just their number. By $|\Psi_i^{\text{Good}}(\text{ADD}; a, s_1, s_2)\rangle$, $|\Psi_{i,1}^{\text{Find}}(\text{ADD}; a, s_1, s_2)\rangle$, and $|\Psi_{i,2}^{\text{Find}}(\text{ADD}; a, s_1, s_2)\rangle$ we mean states equal to the above state but with just the underlined part in the parentheses. We used color in Eq. (79) to indicate the parts that we consider errors. By adding arguments to states we mean that these values are fixed. We add a as the argument to specify the queried interface and s_1 and s_2 to specify the sizes of the databases. The formal definition of states with a, s_1, or s_2 specified is the underlined branch of the superposition projected to register A^I containing a and databases with s_1 and s_2 inputs not equal $\perp$. Above we use $|\mathcal{H}_a^{\text{ADD}}(s_1, s_2)|$ defined in Eq. (28). In Eq. (28) we use the fact that the cardinality of $\mathcal{G}$ depends only on s_1 and s_2, conditioning on $\vec{y}_a$ does not influence the cardinality either, so we omit it in the arguments of $|\mathcal{H}_a^{\text{ADD}}(s_1, s_2)|$.

For the state $|\Psi_{i,2}^{\text{Find}}(\text{ADD}; a, s_1, s_2)\rangle$ we omit entirely the analysis of the other database. That is because the register $D(x)$ is the one responsible for D being in relation and there is no need to analyze $D_{\bar{a}}$.

When we update or remove from the database we start by presenting the non-punctured oracle to make clear the source of errors when discussing the punctured oracle. The counting procedure Queries acts by just analyzing D^X. The only point where we operate in the Fourier basis is when we update the number of non-empty entries in the database. Namely, we apply the Quantum Fourier Transform to register $D^Y(x)$, where x in the queried value, decrease the number of non-empty if the register holds 0, apply the transform again. Below we are a bit sloppy with notation of $\vec{\eta}$, and $\vec{\eta}_a$ does not contain $\eta_{s_a}^a$:

$$\begin{aligned}
&\mathsf{H}\left(|\xi_{i-1}(\text{UPD})\rangle + |\xi_{i-1}(\text{REM})\rangle\right)\\
&= \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w} |x,\eta,a\rangle_{A^{XYI}} |\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W}\\
&\sum_{\vec{y}_a \in \mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})} \frac{1}{\sqrt{|\mathcal{G}_a(s_a-1, s_{\bar{a}})|}} \omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a} |(x_1^a, y_1^a), \ldots, (x_{s_a-1}^a, y_{s_a-1}^a)\rangle_{D_a(\vec{x}_a\setminus\{x\})}\\
&\sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}\\
&\Bigg(\sum_{y_{s_a}^a \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a - b_a(s_1,s_2)}} \omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a} |x, y_{s_a}^a\rangle_{D_a(x)}\\
&- \frac{1}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \sum_{y_{s_a}^a \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \in [N_a]} \frac{1}{\sqrt{N_a}} |x, y_{s_a}^{a\prime}\rangle_{D(x)}\\
&+ \frac{1}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \sum_{y_{s_a}^a \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y_{s_a}^{a\prime}\rangle_{D(x)} \Bigg)\\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}} |(x_1^{\bar{a}}, y_1^{\bar{a}}), \ldots, (x_{s_{\bar{a}}}^{\bar{a}}, y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})}\\
&\sum_{y_{s_a+1}^a, \ldots, y_q^a \in [N_a]} \frac{1}{\sqrt{N_a^{q-s_a}}} |(\perp, y_{s_a+1}^a), \ldots, (\perp, y_q^a)\rangle_{D_a(\perp)}\\
&\sum_{y_{s_{\bar{a}}+1}^{\bar{a}}, \ldots, y_q^{\bar{a}} \in [N_{\bar{a}}]} \frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}} |(\perp, y_{s_{\bar{a}}+1}^{\bar{a}}), \ldots, (\perp, y_q^{\bar{a}})\rangle_{D_{\bar{a}}(\perp)},
\end{aligned} \tag{81}$$

The states that we add to the first element in the parentheses come from performing the Fourier transform on a state that is not of the form $\mathsf{QFT}_{N_a}|\eta\rangle$. Note that this discrepancy is the result of considering the good state. Whether we are in the branch UPD or REM depends on whether $\eta = -\eta_s$ or not.

Similarly to the case of ADD, we first present the state with the error parts of D_a exposed.

$$
\begin{aligned}
&\mathsf{H}|\xi_{i-1}(\mathrm{UPD})\rangle = \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w}|x,\eta,a\rangle_{A^{XYI}}|\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W}\\
&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})} \frac{1}{\sqrt{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}}\omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a}|(x_1^a,y_1^a),\ldots,(x_{s_a-1}^a,y_{s_a-1}^a)\rangle_{D_a(\vec{x}_a\setminus\{x\})}\\
&\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\\
&\Bigg(\sum_{y_{s_a}^a\notin\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a-b_a(s_1,s_2)}}\omega_N^{(\eta_{s_a}^a+\eta)y_{s_a}^a}|x,y_{s_a}^a\rangle_{D_a(x)}\\
&+\frac{1}{N_a\sqrt{N_a-b_a(s_1,s_2)}}\sum_{y_{s_a}^a\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a}\sum_{y_{s_a}^{a\prime}\notin\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}|x,y_s^{a\prime}\rangle_{D(x)}\\
&+\frac{1}{N_a\sqrt{N_a-b_a(s_1,s_2)}}\sum_{y_{s_a}^a\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a}\sum_{y_{s_a}^{a\prime}\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}|x,y_{s_a}^{a\prime}\rangle_{D(x)}\\
&-\frac{1}{N_a\sqrt{N_a-b_a(s_1,s_2)}}\sum_{y_{s_a}^a\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a}\sum_{y_{s_a}^{a\prime}\in[N_a]}|\perp,y_{s_a}^{a\prime}\rangle_{D_a(x)}\Bigg)\\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}}|(x_1^{\bar{a}},y_1^{\bar{a}}),\ldots,(x_{s_{\bar{a}}}^{\bar{a}},y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})}\\
&\sum_{y_{s_a+1}^a,\ldots,y_q^a\in[N_a]}\frac{1}{\sqrt{N_a^{q-s_a}}}|(\perp,y_{s_a+1}^a),\ldots,(\perp,y_q^a)\rangle_{D_a(\perp)}\\
&\sum_{y_{s_{\bar{a}}+1}^{\bar{a}},\ldots,y_q^{\bar{a}}\in[N_{\bar{a}}]}\frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}}|(\perp,y_{s_{\bar{a}}+1}^{\bar{a}}),\ldots,(\perp,y_q^{\bar{a}})\rangle_{D_{\bar{a}}(\perp)},
\end{aligned}
\tag{82}
$$

where we have used the fact that $\eta_{s_a}^a+\eta\neq 0$ which implies $\sum_{y_{s_a}^a\notin\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a} = -\sum_{y_{s_a}^a\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a}$.

Splitting the sums in the case of removing an entry works as follows:

$$
\begin{aligned}
&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}\setminus\{x\})}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}|\vec{y}_a)} = \sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}\setminus\{x\})}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}\setminus\{x\}|\vec{y}_a)}\\
&-\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}\setminus\{x\})}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}\setminus\{x\}|\vec{y}_a)\setminus\mathcal{G}_{\bar{a}}(\vec{x}|\vec{y}_a)}.
\end{aligned}
\tag{83}
$$

Let us present in detail the effect of $\mathsf{Queries}^\dagger\circ\mathsf{V}_R\circ\mathsf{Queries}$ on the branch of updated databases:

$$
\begin{aligned}
&\mathrm{UPD}:\mathsf{H}\setminus\mathsf{V}_R|\xi_{i-1}(\mathrm{UPD})\rangle|0\rangle_J\\
&=\sum_{x,\eta,a,\vec{x},\vec{\eta},w}\alpha_{x,\eta,a,\vec{x},\vec{\eta},w}|x,\eta,a\rangle_{A^{XYI}}|\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W}\\
&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})}\frac{1}{\sqrt{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}}\omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a}|(x_1^a,y_1^a),\ldots,(x_{s_a-1}^a,y_{s_a-1}^a)\rangle_{D_a(\vec{x}_a\setminus\{x\})}
\end{aligned}
$$

$$
\Big(|\Psi_i^{\text{Good}}(\text{UPD}; a, s_1, s_2)\rangle|0\rangle_J
$$
$$
+ |\Psi_{i,1}^{\text{Bad}}(\text{UPD}; a, s_1, s_2)\rangle|0\rangle_J
$$
$$
+ |\Psi_{i,1}^{\text{Find}}(\text{UPD}; a, s_1, s_2)\rangle|1\rangle_J
$$
$$
- |\Psi_{i,2}^{\text{Bad}}(\text{UPD}; a, s_1, s_2)\rangle|0\rangle_J + |\Psi_{i,2}^{\text{Find}}(\text{UPD}; a, s_1, s_2)\rangle|1\rangle_J\Big)
$$
$$
\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}}|(x_1^{\bar{a}}, y_1^{\bar{a}}), \ldots, (x_{s_{\bar{a}}}^{\bar{a}}, y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})}
$$
$$
\sum_{y_{s_a+1}^a, \ldots, y_q^a \in [N_a]} \frac{1}{\sqrt{N_a^{q-s_a}}}|(\perp, y_{s_a+1}^a), \ldots, (\perp, y_q^a)\rangle_{D_a(\perp)}
$$
$$
\sum_{y_{s_{\bar{a}}+1}^{\bar{a}}, \ldots, y_q^{\bar{a}} \in [N_{\bar{a}}]} \frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}}|(\perp, y_{s_{\bar{a}}+1}^{\bar{a}}), \ldots, (\perp, y_q^{\bar{a}})\rangle_{D_{\bar{a}}(\perp)}, \tag{84}
$$

where the states with superscripts Good, Bad, and Find denote the states that are defined as the state from Eq. (84) with expressions from Equations (85), (86), (87), (88), or (89) below put in the correct spot, without any other element from the parentheses. Note that the states in the equation above come from splitting the sum over $\vec{y}_{\bar{a}}$ into the parts of Eq. (82) from the corresponding lines. Below we define in detail all the states from Eq. (84).

The good state gives the following expression in the parentheses in Eq. (84):

$$
|\Psi_i^{\text{Good}}(\text{UPD}; a, s_1, s_2)\rangle : \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}
$$
$$
\sum_{y_{s_a}^a \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a - b_a(s_1, s_2)}} \omega_N^{(\eta_{s_a}^a + \eta) y_{s_a}^a} |x, y_{s_a}^a\rangle_{D_a(x)}. \tag{85}
$$

Bad states are those with $|0\rangle_J$ that are not good:

$$
|\Psi_{i,1}^{\text{Bad}}(\text{UPD}; a, s_1, s_2)\rangle : \frac{1}{N_a} \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}} \sum_{y_{s_a}^a \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}
$$
$$
\omega_{N_a}^{(\eta_{s_a}^a + \eta) y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a - b_a(s_1, s_2)}} |x, y_{s_a}^{a\prime}\rangle_{D(x)}, \tag{86}
$$
$$
|\Psi_{i,2}^{\text{Bad}}(\text{UPD}; a, s_1, s_2)\rangle : \frac{1}{\sqrt{N_a(N_a - b_a(s_1, s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}
$$
$$
\sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_a \setminus \{x\}, \vec{x}_{\bar{a}}|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}}
$$
$$
\sum_{y_{s_a}^a \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \omega_{N_a}^{(\eta_{s_a}^a + \eta) y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y_{s_a}^{a\prime}\rangle_{D_a(x)}. \tag{87}
$$

The states with the superscript Find are states for which $D \in R$:

$$
|\Psi_{i,1}^{\text{Find}}(\text{UPD}; a, s_1, s_2)\rangle : \sqrt{\frac{b_a(s_1, s_2)}{N_a^2(N_a - b_a(s_1, s_2))}} \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}
$$
$$
\sum_{y_{s_a}^a \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \omega_{N_a}^{(\eta_{s_a}^a + \eta) y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{b_a(s_1, s_2)}} |x, y_{s_a}^{a\prime}\rangle_{D(x)}, \tag{88}
$$

$$|\Psi_{i,2}^{\text{Find}}(\text{UPD}; a, s_1, s_2)\rangle : \frac{1}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \sqrt{\frac{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \sum_{\vec{y}_{\bar{a}} \in \mathcal{H}_1^{\text{REM}}(\vec{x}_1,\vec{x}_2,\vec{y}_a)}$$
$$\frac{1}{\sqrt{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}} \sum_{y_{s_a}^a \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \omega_{N_a}^{(\eta_{s_a}^a+\eta)y_{s_a}^a} \sum_{y_{s_a}^{a\prime} \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y_{s_a}^{a\prime}\rangle_{D_a(x)}, \tag{89}$$

where $|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|$ is defined in Eq. (30).

The branch of superposition corresponding to removing x from D with just the errors in D_a exposed is

$$\begin{aligned}
&\text{REM} : \mathsf{H}|\xi_{i-1}(\text{REM})\rangle \\
&= \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w} |x,\eta,a\rangle_{A^{XYI}} |\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W} \\
&\sum_{\vec{y}_a \in \mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})} \frac{1}{\sqrt{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}} \omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a} |(x_1^a,y_1^a),\ldots,(x_{s_a-1}^a,y_{s_a-1}^a)\rangle_{D_a(\vec{x}_a\setminus\{x\})} \\
&\sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \left(\sqrt{\frac{N_a - b_a(s_1,s_2)}{N_a}} \sum_{y_{s_a}^a \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y_{s_a}^a\rangle_{D(x)} \right. \\
&+ \frac{b_a(s_1,s_2)}{N_a} \sum_{y_{s_a}^a \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a - b_a(s_1,s_2)}} |x, y_{s_a}^a\rangle_{D_a(x)} \\
&\left. + \frac{\sqrt{b_a(s_1,s_2)(N_a - b_a(s_1,s_2))}}{N_a} \sum_{y_{s_a}^a \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{b_a(s_1,s_2)}} |x, y_{s_a}^a\rangle_{D_a(x)} \right) \\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}} |(x_1^{\bar{a}},y_1^{\bar{a}}),\ldots,(x_{s_{\bar{a}}}^{\bar{a}},y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})} \\
&\sum_{y_{s_a+1}^a,\ldots,y_q^a \in [N_a]} \frac{1}{\sqrt{N_a^{q-s_a}}} |(\perp, y_{s_a+1}^a),\ldots,(\perp,y_q^a)\rangle_{D_a(\perp)} \\
&\sum_{y_{s_{\bar{a}}+1}^{\bar{a}},\ldots,y_q^{\bar{a}} \in [N_{\bar{a}}]} \frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}} |(\perp, y_{s_{\bar{a}}+1}^{\bar{a}}),\ldots,(\perp,y_q^{\bar{a}})\rangle_{D_{\bar{a}}(\perp)},
\end{aligned} \tag{90}$$

where we simplified the formula from Eq. (81) using the fact $\eta = -\eta_{s_a}^a$. Splitting the sums in the case of removing an entry works as shown in Eq. (83).

Now we present the full state, after checking for $D \in R$:

$$\begin{aligned}
&\text{REM} : \mathsf{H} \setminus \mathsf{V}_R|\xi_{i-1}(\text{REM})\rangle|0\rangle_J \\
&= \sum_{x,\eta,a,\vec{x},\vec{\eta},w} \alpha_{x,\eta,a,\vec{x},\vec{\eta},w} |x,\eta,a\rangle_{A^{XYI}} |\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W} \\
&\sum_{\vec{y}_a \in \mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})} \frac{1}{\sqrt{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}} \omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a} |(x_1^a,y_1^a),\ldots,(x_{s_a-1}^a,y_{s_a-1}^a)\rangle_{D_a(\vec{x}_a\setminus\{x\})} \\
&\left(\sqrt{\frac{N_a - b_a(s_1,s_2)}{N_a}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \right. \\
&\underbrace{\left. \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}}|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}} \sum_{y_{s_a}^a \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y_{s_a}^a\rangle_{D(x)} |0\rangle_J \right)}_{|\Psi_i^{\text{Good}}(\text{REM},a,s_1,s_2)\rangle}
\end{aligned}$$

$$
\begin{aligned}
&\Bigg(\underbrace{-\sqrt{\frac{N_a - b_a(s_1,s_2)}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\mathrm{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \sum_{\vec{y}_{\bar{a}} \in \mathcal{H}_a^{\mathrm{REM}}(\vec{x}_1,\vec{x}_2,\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{H}_a^{\mathrm{REM}}(s_1,s_2)|}}}_{\text{(i) } |\Psi_{i,1}^{\mathrm{Find}}(\mathrm{REM},a,s_1,s_2)\rangle} \\
&\underbrace{\sum_{y^a_{s_a} \in [N_a]} \frac{1}{\sqrt{N_a}} |\perp, y^a_{s_a}\rangle_{D(x)} |1\rangle_J}_{\text{(ii) } |\Psi_{i,1}^{\mathrm{Find}}(\mathrm{REM},a,s_1,s_2)\rangle} \\
&+ \underbrace{\frac{b_a(s_1,s_2)}{N_a} \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}}_{\text{(i) } |\Psi_i^{\mathrm{Bad}}(\mathrm{REM};a,s_1,s_2)\rangle} \\
&\underbrace{\sum_{y^a_{s_a} \notin \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{N_a - b_a(s_1,s_2)}} |x, y^a_{s_a}\rangle_{D_a(x)} |0\rangle_J}_{\text{(ii) } |\Psi_i^{\mathrm{Bad}}(\mathrm{REM};a,s_1,s_2)\rangle} \\
&+ \underbrace{\frac{\sqrt{b_a(s_1,s_2)(N_a - b_a(s_1,s_2))}}{N_a} \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} \frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}}_{\text{(i) } |\Psi_{i,2}^{\mathrm{Find}}(\mathrm{REM},a,s_1,s_2)\rangle} \\
&\Bigg(\underbrace{\sum_{y^a_{s_a} \in \mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))} \frac{1}{\sqrt{b_a(s_1,s_2)}} |x, y^a_{s_a}\rangle_{D_a(x)} |1\rangle_J}_{\text{(ii) } |\Psi_{i,2}^{\mathrm{Find}}(\mathrm{REM},a,s_1,s_2)\rangle} \Bigg) \\
&\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}} \cdot \vec{y}_{\bar{a}}} |(x_1^{\bar{a}}, y_1^{\bar{a}}), \ldots, (x_{s_{\bar{a}}}^{\bar{a}}, y_{s_{\bar{a}}}^{\bar{a}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})} \\
&\sum_{y^a_{s_a+1}, \ldots, y^a_q \in [N_a]} \frac{1}{\sqrt{N_a^{q-s_a}}} |(\perp, y^a_{s_a+1}), \ldots, (\perp, y^a_q)\rangle_{D_a(\perp)} \\
&\sum_{y^{\bar{a}}_{s_{\bar{a}}+1}, \ldots, y^{\bar{a}}_q \in [N_{\bar{a}}]} \frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}} |(\perp, y^{\bar{a}}_{s_{\bar{a}}+1}), \ldots, (\perp, y^{\bar{a}}_q)\rangle_{D_{\bar{a}}(\perp)},
\end{aligned} \tag{91}
$$

where $\mathcal{H}_a^{\mathrm{REM}}(\vec{x}_1, \vec{x}_2, \vec{y}_a) = \mathcal{G}_{\bar{a}}(\vec{x}_a \setminus \{x\}, \vec{x}_{\bar{a}} \mid \vec{y}_a) \setminus \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2 \mid \vec{y}_a)$.

A.2 Bound on $\varepsilon_{\mathrm{step}}(j)$

Lemma 14. *For states defined in the preceding sections we have*

$$
\begin{aligned}
&\left\| |\Psi_i^{\mathrm{Good}}\rangle_{AD} |0\rangle_J - |\Phi_i\rangle_{ADJ} \right\| \leq \sum_{j=1}^{i} \varepsilon_{\mathrm{step}}(j) \\
&\leq \sum_{j=1}^{i} \max_{a \in \{1,2\}, s_1, s_2 \leq j-1} \Bigg(2\frac{b_a(s_1,s_2)}{N_a} + \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \\
&+ \frac{b_a(s_1,s_2)}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \\
&+ \frac{b_a(s_a + 1, s_{\bar{a}})}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a + 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a + 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \Bigg).
\end{aligned} \tag{92}
$$

Proof. We prove the statement by recursion, the derivation is shown in Eq. (46). The step function $\varepsilon_{\text{step}}(j)$ defined in Eq. (47). For $i = 0$ the statement is true, as $|\Psi_0^{\text{Good}}\rangle|0\rangle_J = |\Phi_0\rangle = |\Psi_0\rangle|0\rangle_J$.

Eqs. (79), (84), and (91) give us exact expressions for $|\Psi_{j-1}^{\text{Good}}\rangle$ after a query. Following [Cza+19], we distinguish two types of errors compared to $|\Psi_j^{\text{Good}}\rangle|0\rangle_J$: an additive error of adding a small-weight state to the original one and a multiplicative error where one branch of the superposition is multiplied by some factor.

The additive error includes all states of small-weight states multiplied by $|0\rangle_J$ with the superscript Bad. The multiplicative error, on the other hand, includes the branches of the superposition where we add or remove an entry from the database.

In the following, we first deal with the additive errors and after that with multiplicative. We define $|\psi_j^{\times}\rangle_{ADJ}$ as the state $\bar{\mathsf{J}}_R\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{j-1}|\Psi_{j-1}^{\text{Good}}\rangle|0\rangle_J$ with all branches classified as the additive error excluded. By "classified as the additive error" we mean states with superscript Bad and highlighted in red in Equations (79, 84, 91). The state is defined as

$$\begin{aligned}
|\psi_j^{\times}\rangle_{ADJ} := \Bigg(& \sum_{a,s_1,s_2} |\Psi_j^{\text{Good}}(\text{NOT}; a, s_1, s_2)\rangle \\
& + \sqrt{\frac{N_a - b_a(s_a+1, s_{\bar{a}})}{N_a}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}} |\Psi_j^{\text{Good}}(\text{ADD}; a, s_1, s_2)\rangle \\
& + |\Psi_j^{\text{Good}}(\text{UPD}; a, s_1, s_2)\rangle \\
& + \sqrt{\frac{N_a - b_a(s_1, s_2)}{N_a}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}} |\Psi_j^{\text{Good}}(\text{REM}; a, s_1, s_2)\rangle \Bigg) |0\rangle_J,
\end{aligned} \tag{93}$$

where the states above correspond to branches of superposition where we do nothing (NOT, for $\eta = 0$), add an entry, update the database, and remove an entry from D. Bounding the difference of the states is done as follows

$$\begin{aligned}
& \left\| |\Psi_j^{\text{Good}}\rangle|0\rangle_J - \bar{\mathsf{J}}_R\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{j-1}|\Psi_{j-1}^{\text{Good}}\rangle|0\rangle_J \right\| \\
& \leq \left\| |\Psi_j^{\text{Good}}\rangle|0\rangle_J - |\psi_j^{\times}\rangle_{ADJ} \right\| + \left\| |\psi_j^{\times}\rangle_{ADJ} - \bar{\mathsf{J}}_R\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{j-1}|\Psi_{j-1}^{\text{Good}}\rangle|0\rangle_J \right\|.
\end{aligned} \tag{94}$$

The second term above is the norm of all states amplifying the additive error—we call them the bad states.

The additive error $\||\psi_j^{\times}\rangle_{ADJ} - \bar{\mathsf{J}}_R\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{j-1}|\Psi_{j-1}^{\text{Good}}\rangle|0\rangle_J\|$ is bound by first splitting the three cases underlined above:

$$\left\| |\Psi_j^{\text{Bad}}\rangle \right\| \leq \left\| |\Psi_{j,1}^{\text{Bad}}(\text{UPD})\rangle \right\| + \left\| |\Psi_{j,2}^{\text{Bad}}(\text{UPD})\rangle \right\| + \left\| |\Psi_j^{\text{Bad}}(\text{REM})\rangle \right\|, \tag{95}$$

where $|\Psi_j^{\text{Bad}}\rangle$ is the sum of all three bad states, the bound follows from the triangle inequality.

Calculating all of the three norms above is done by first focusing on a particular interface that is queried and by focusing on particular sizes of databases:

$$\left\| |\Psi_j^{\text{Bad}}\rangle \right\| = \sqrt{\sum_a \sum_{s_1,s_2=0}^{j} |\beta(a, s_1, s_2)|^2 \left\| |\Psi_j^{\text{Bad}}(a, s_1, s_2)\rangle \right\|^2}, \tag{96}$$

where $\beta(a, s_1, s_2)$ is the amplitude of the good state projected to states with the specified parameters: For a projector P_{a,s_1,s_2} to adversaries that query interface a and databases of sizes s_1 and s_2 we have $\beta(a, s_1, s_2) := \mathsf{P}_{a,s_1,s_2}|\Psi_j^{\text{Good}}\rangle$ and $|\Psi_j^{\text{Bad}}(a, s_1, s_2)\rangle := \mathsf{P}_{a,s_1,s_2}|\Psi_j^{\text{Bad}}\rangle$.

1 Additive errors The states classified as additive errors are marked by the Bad superscript. We start with the case UPD of updating the database from Eq. (84). The hard part here is analyzing the factor $\sum_{y^a_{s_a}\in\mathcal{B}(x|D(\vec{x}\setminus\{x\}))}\omega_N^{(\eta^a_{s_a}+\eta)y^a_{s_a}}$; This is a a complex number that depends on $\eta^a_{s_a}$, so it enters the norm in a non-trivial way. To ease notation, we change the value we sum over to $y^a_{s_a}\in[b_a(s_1,s_2)]$ and change $y^a_{s_a}$ in the expression to $\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))(y^a_{s_a})$, by which we denote the $y^a_{s_a}$-th element of $\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))$. The order in the bad set is just the rising order, note that $\mathcal{Y}_a=[N_a]$.

Given the change of variables we use the triangle inequality to focus on the norm of a state with a single phase factor $\omega_N^{(\eta^a_{s_a}+\eta)\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))(y^a_{s_a})}$, instead of the whole sum:

$$\begin{aligned}&\left\||\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2)\rangle\right\|\\&\leq\sum_{y^a_{s_a}\in[b_a(s_1,s_2)]}\left\||\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))(y^a_{s_a}))\rangle\right\|,\end{aligned}\tag{97}$$

where we omit the index of the UPD errors because the techniques here work in almost the same way for both states. The input $D(\vec{x}\setminus\{x\})$ should not be treated as an actual argument of the state, we still consider the superposition over different inputs, we just mean that in the state $|\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2)\rangle$ we change the variable $y^a_{s_a}$. In what follows we denote the state on the right hand side of the above equation by $|\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle$.

Now we focus on the state with a fixed $\mathcal{B}'(y^a_{s_a})$, we bound the norm of this state.

Claim 15. *For all* $y^a_{s_a}\in[b_a(s_1,s_2)]$

$$\left\||\Psi_{j,1}^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))(y^a_{s_a}))\rangle\right\|\leq\frac{1}{N_a}\text{ and}\tag{98}$$

$$\begin{aligned}&\left\||\Psi_{j,2}^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))(y^a_{s_a}))\rangle\right\|\\&\leq\sqrt{\frac{1}{N_a(N_a-b_a(s_1,s_2))}}\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}.\end{aligned}\tag{99}$$

Proof. First we show that taking the good state in the UPD branch and modifying the sum (and its normalization factor) over $y^a_{s_a}$, to $\sum_{y^a_{s_a}\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}$ or $\sum_{y^a_{s_a}\in[N_a]}$ yields states with norm bounded by 1. Given that we can show that the norm of $|\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}_a(x\mid D(\vec{x}\setminus\{x\}))(y^a_{s_a}))\rangle$ multiplied by the corresponding right hand side of Eq. (98) and (99) equals to the norm of the state with the sum modified to $\sum_{y^a_{s_a}\in[N_a]}$.

The two states are:

$$\begin{aligned}&\sum_{x,\eta,a,\vec{x},\vec{\eta},w}\alpha_{x,\eta,a,\vec{x},\vec{\eta},w}|x,\eta,a\rangle_{A^{XYI}}|\psi(x,\eta,a,\vec{x},\vec{\eta},w)\rangle_{A^W}\\&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})}\frac{1}{\sqrt{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}}\omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a}|(x^a_1,y^a_1),\dots,(x^a_{s_a-1},y^a_{s_a-1})\rangle_{D_a(\vec{x}_a\setminus\{x\})}\\&\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)}\frac{1}{\sqrt{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}}|(x^{\bar{a}}_1,y^{\bar{a}}_1),\dots,(x^{\bar{a}}_{s_{\bar{a}}},y^{\bar{a}}_{s_{\bar{a}}})\rangle_{D_{\bar{a}}(\vec{x}_{\bar{a}})}\\&\sum_{y^a_{s_a+1},\dots,y^a_q\in[N_a]}\frac{1}{\sqrt{N_a^{q-s_a}}}|(\perp,y^a_{s_a+1}),\dots,(\perp,y^a_q)\rangle_{D_a(\perp)}\end{aligned}$$

$$\sum_{y^{\bar{a}}_{s_{\bar{a}}+1},\ldots,y^{\bar{a}}_{q}\in[N_{\bar{a}}]}\frac{1}{\sqrt{N_{\bar{a}}^{q-s_{\bar{a}}}}}|(\perp,y^{\bar{a}}_{s_{\bar{a}}+1}),\ldots,(\perp,y^{\bar{a}}_{q})\rangle_{D_{\bar{a}}(\perp)}$$
$$\otimes\begin{cases}\sum_{y^a_{s_a}\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}\frac{1}{\sqrt{b_a(s_1,s_2)}}\omega_N^{(\eta^a_{s_a}+\eta)y^a_{s_a}}|x,y^a_{s_a}\rangle_{D_a(x)}=:|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle,\\ \sum_{y^a_{s_a}\in[N_a]}\frac{1}{\sqrt{N_a}}\omega_{N_a}^{(\eta^a_{s_a}+\eta)y^a_{s_a}}|x,y^a_{s_a}\rangle_{D_a(x)}=:|\widetilde{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle.\end{cases}\tag{100}$$

The first one, $|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle$ is the one that we use in the last step of the proof, as described in the previous paragraph. The second one will be used to show that the norm of $|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle$ is bounded by 1.

Also note that $\left\||\widetilde{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|=\left\||\Psi_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|$. There are two arguments that prove it: The weight associated with the fact that the two states correspond to the $(\text{UPD};a,s_1,s_2)$ branch of the superposition is the same. They both can be generated by first considering an adversary-oracle state interacting with a non-punctured oracle and then performing some projection to D. The second argument is that the database register is normalized. Hence the overall weight is the same.

Having in mind that $\sum_{y^a_{s_a}\in\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}=\sum_{y^a_{s_a}\in[N_a]}-\sum_{y^a_{s_a}\notin\mathcal{B}_a(x|D(\vec{x}\setminus\{x\}))}$ we see that

$$\begin{aligned}&b_a(s_1,s_2)\left\||\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|^2\\&=N_a\left\||\widetilde{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|^2\\&-(N_a-b_a(s_1,s_2))\left\||\Psi_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|^2\leq b_a(s_1,s_2),\end{aligned}\tag{101}$$

hence $\left\||\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle\right\|^2\leq 1$.

Now that we know that $|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle$ is sub-normalized we show that

$$\left\||\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle\right\|\leq\frac{1}{\sqrt{b_a(s_1,s_2)}}.\tag{102}$$

To prove this bound, consider measuring register $D_a(x)$ of $|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2)\rangle$ in the computational basis. The probability of getting any outcome $y^a_{s_a}$ is necessarily $\frac{1}{b_a(s_1,s_2)}$, as the outputs of the oracle are uniformly random. The post-measurement state, for an outcome $y^a_{s_a}$, is $\sqrt{b_a(s_1,s_2)}\cdot$
$|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle$. Naturally, norm of this post-measurement state is at most 1.

Now we can use the state $|\overline{\Psi}_j^{\text{Good}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle$ to analyze the norm of $|\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle$. First let us inspect the norm squared of the bad state:

$$\begin{aligned}&\left\||\Psi_j^{\text{Bad}}(\text{UPD};a,s_1,s_2,\mathcal{B}'(y^a_{s_a}))\rangle\right\|^2=\sum_{x,\eta,a,\vec{x},\vec{\eta}',\vec{\eta},w',w}\sum_{\eta^{a\prime}_{s_a},\eta^a_{s_a}}\bar{\alpha}_{x,\eta,a,\vec{x},\vec{\eta}',\eta^{a\prime}_{s_a},w'}\\&\alpha_{x,\eta,a,\vec{x},\vec{\eta},\eta^a_{s_a},w}\langle\psi(x,\eta,a,\vec{x},\vec{\eta}',\eta^{a\prime}_{s_a},w')|\psi(x,\eta,a,\vec{x},\vec{\eta},\eta^a_{s_a},w)\rangle\\&\sum_{\vec{y}_a\in\mathcal{G}_a(\vec{x}_a\setminus\{x\},\vec{x}_{\bar{a}})}\frac{1}{|\mathcal{G}_a(s_a-1,s_{\bar{a}})|}\omega_{N_a}^{\vec{\eta}_a\cdot\vec{y}_a}\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)}\frac{1}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}\omega_{N_{\bar{a}}}^{\vec{\eta}_{\bar{a}}\cdot\vec{y}_{\bar{a}}}\\&\frac{1}{N_a^2(N_a-b_a(s_1,s_2))}\bar{\omega}_{N_a}^{(\eta^{a\prime}_{s_a}+\eta)\mathcal{B}'(y^a_{s_a})}\omega_{N_a}^{(\eta^a_{s_a}+\eta)\mathcal{B}'(y^a_{s_a})}\gamma^2\underbrace{\sum_{y^{a\prime}_{s_a}\in[\nu]}}_{=\nu},\end{aligned}\tag{103}$$

where $\nu = N_a - b_a(s_1, s_2)$ and $\gamma = 1$ for $|\Psi^{\text{Bad}}_{j,1}(\text{UPD}; a, s_1, s_2, \mathcal{B}'(y^a_{s_a}))\rangle$ and $\nu = N_a$ and $\gamma = \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}$ for $|\Psi^{\text{Bad}}_{j,2}(\text{UPD}; a, s_1, s_2, \mathcal{B}'(y^a_{s_a}))\rangle$ (in the second case the sum goes over $y^{a\prime}_{s_a} \notin \mathcal{B}_a(x \mid D(\vec{x} \setminus \{x\}))$ instead of $y^{a\prime}_{s_a} \in [\nu]$). It is easy to notice, that the only difference between Eq. (103) and norm squared of $|\overline{\Psi}^{\text{Good}}_j(\text{UPD}; a, s_1, s_2, \mathcal{B}'(y^a_{s_a}))\rangle$ lies in the factor $\frac{\nu\gamma^2}{N_a^2(N_a - b_a(s_1,s_2))}$. This factor in the modified good state equals $\frac{1}{b_a(s_1,s_2)}$. This observation implies that

$$\begin{aligned}
&\left\| |\Psi^{\text{Bad}}_j(\text{UPD}; a, s_1, s_2, \mathcal{B}'(y^a_{s_a}))\rangle \right\| \\
&= \sqrt{\frac{b(s) \cdot \nu\gamma^2}{N_a^2(N_a - b_a(s_1, s_2))}} \left\| |\overline{\Psi}^{\text{Good}}_j(\text{UPD}; s, \mathcal{B}'(y^a_{s_a}))\rangle \right\|. \qquad (104)
\end{aligned}$$

Together with the bound on the norm in the left hand side this proves the claimed bounds. □

Claim 15, together with the bound from Eq. (97) gives us:

$$\left\| |\Psi^{\text{Bad}}_{j,1}(\text{UPD}; a, s_1, s_2)\rangle \right\| \leq \frac{b_a(s_1, s_2)}{N_a}, \qquad (105)$$

$$\left\| |\Psi^{\text{Bad}}_{j,2}(\text{UPD}; a, s_1, s_2)\rangle \right\| \leq \frac{b_a(s_1, s_2)}{\sqrt{N_a(N_a - b_a(s_1, s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}}. \qquad (106)$$

The bounds from Eq. (105) in Eq. (96) give us the bound on the additive error in the UPD branch. The additive error for the REM branch ($|\Psi^{\text{Bad}}_j(\text{REM})\rangle$ in Eq. (91)) is much easier to calculate: As register $D(x)$ is normalized and all the rest of the state is the same as $|\Psi^{\text{Good}}_j(\text{REM})\rangle$, the only error comes from the factor $\frac{b_a(s_1,s_2)}{N_a}$. To calculate the norm of the state we can follow the analysis of Eq. (103). Finally we get:

$$\left\| |\Psi^{\text{Bad}}_{j,1}(\text{UPD})\rangle \right\| \leq \max_{a,s_1,s_2} \left(\frac{b_a(s_1, s_2)}{N_a} \right), \qquad (107)$$

$$\left\| |\Psi^{\text{Bad}}_{j,2}(\text{UPD})\rangle \right\| \leq \max_{a,s_1,s_2} \left(\frac{b_a(s_1, s_2)}{\sqrt{N_a(N_a - b_a(s_1, s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1, s_2)|}} \right), \qquad (108)$$

$$\left\| |\Psi^{\text{Bad}}_j(\text{REM})\rangle \right\| \leq \max_{a,s_1,s_2} \left(\frac{b_a(s_1, s_2)}{N_a} \right), \qquad (109)$$

where $a \in \{1, 2\}$ and $s_1, s_2 \leq j - 1$.

2 Multiplicative errors The multiplicative error is a factor that multiplies a part of the state $|\psi^{\times}_j\rangle_{ADJ}$. Similarly as before we need to take care of the fact that the joint state of the adversary and the oracle is a sum over databases of different sizes and queries to different interfaces:

$$|\psi^{\times}_j\rangle = \sum_{a,s_1,s_2} |\psi^{\times}_j(a, s_1, s_2)\rangle, \qquad (110)$$

where the states $|\psi^{\times}_j(a, s_1, s_2)\rangle$ are orthogonal. The terms are also orthogonal in $|\Psi^{\text{Good}}_j\rangle = \sum_{a,s_1,s_2} |\Psi^{\text{Good}}_j(a, s_1, s_2)\rangle$.

There are two sources of multiplicative errors, ADD from Eq. (79) and REM from Eq. (91), we split the two sources with the triangle inequality. We deal with both in the same way, just the final bound is different.

Let us write down the two parts, one affected by the error and the second not:

$$|\Psi_j^{\text{Good}}\rangle_{AD}|0\rangle_J = \sum_{a,s_1,s_2} \alpha(a,s_1,s_2)|\varphi_1(a,s_1,s_2)\rangle + \beta(a,s_1,s_2)|\varphi_2(a,s_1,s_2)\rangle, \tag{111}$$

$$\begin{aligned}|\psi_j^{\times}\rangle_{ADJ} = &\sum_{a,s_1,s_2} \alpha(a,s_1,s_2)|\varphi_1(a,s_1,s_2)\rangle \\ &+ (1+g)\sqrt{1-e}\beta(a,s_1,s_2)|\varphi_2(a,s_1,s_2)\rangle,\end{aligned} \tag{112}$$

where $(1+g)\sqrt{1-e}$ is the multiplicative error, in the case ADD the error is $g = \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1$ and $e = \frac{b_a(s_a+1,s_{\bar{a}})}{N_a}$. In the case REM the error is $g = \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1$ and $e = \frac{b_a(s_1,s_2)}{N_a}$. We know that
$\sum_{a,s_1,s_2} |\alpha(a,s_1,s_2)|^2 + |\beta(a,s_1,s_2)|^2 \leq 1$, because we excluded a single branch of the superposition, for ADD and REM. This inequality implies
$\sum_{a,s_1,s_2} |\beta(a,s_1,s_2)|^2 \leq 1$. We continue with the bound

$$\begin{aligned}&\left\| |\psi_j^{\times}\rangle_{ADJ} - |\Psi_j^{\text{Good}}\rangle_{AD}|0\rangle_J \right\| \\ &= \left\| \sum_{a,s_1,s_2} \left(1-(1+g)\sqrt{1-e}\right)\beta(a,s_1,s_2)|\varphi_2(a,s_1,s_2)\rangle \right\| \qquad (113) \\ &= \sqrt{\sum_{a,s_1,s_2} \left(1-(1+g)\sqrt{1-e}\right)^2 |\beta(a,s_1,s_2)|^2} \leq \max_{a,s_1,s_2} \left(1-(1+g)\sqrt{1-e}\right) \\ &\leq \max_{a,s_1,s_2} \left((1+g)e - g\right). \qquad (114)\end{aligned}$$

Maximization is done over $a \in \{1,2\}$ and $s_1, s_2 \leq j-1$.

Bound on one step From Equations (94), (107), and (114) (for the two sources of error) the bound on the single step is

$$\begin{aligned}\varepsilon_{\text{step}}(j) \leq &\max_{a\in\{1,2\},s_1,s_2\leq j-1} \left(2\frac{b_a(s_1,s_2)}{N_a} + \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a-b_a(s_1,s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \right. \\ &+ \frac{b_a(s_1,s_2)}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \\ &\left. + \frac{b_a(s_a+1,s_{\bar{a}})}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \right)\end{aligned} \tag{115}$$

and the final bound is

$$\begin{aligned}&\left\| |\Psi_i^{\text{Good}}\rangle_{AD}|0\rangle_J - |\Phi_i\rangle_{ADJ} \right\| \\ &\leq \sum_{j=1}^{i} \max_{a\in\{1,2\},s_1,s_2\leq j-1} \left(2\frac{b_a(s_1,s_2)}{N_a} + \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a-b_a(s_1,s_2))}} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \right. \\ &+ \frac{b_a(s_1,s_2)}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \\ &\left. + \frac{b_a(s_a+1,s_{\bar{a}})}{N_a} \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - \left(\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a+1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} - 1 \right) \right).\end{aligned} \tag{116}$$

□

A.3 Bound on $\varepsilon_{\text{Find}}(i)$

To bound the norm of $\left\|\mathsf{J}_R\mathsf{U}_i\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{i-1}|\Psi_{i-1}^{\text{Good}}\rangle\right\|$ we have to bound the norm of all states from Sec. A.1 that have the superscript Find (they contain $|1\rangle_J$).

Lemma 16. *For states defined in preceding sections we have*

$$\begin{aligned}
&\left\|\mathsf{J}_R\mathsf{U}_i\mathsf{H}\setminus\mathsf{V}_R\mathsf{U}_{i-1}|\Psi_{i-1}^{\text{Good}}\rangle\right\| = \varepsilon_{\text{Find}}(i)\\
&\leq \max_{a\in\{1,2\},s_1,s_2\leq i-1}\Bigg(\sqrt{\frac{N_a-b_a(s_a+1,s_{\bar{a}})}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\\
&+\sqrt{\frac{b_a(s_a+1,s_{\bar{a}})}{N_a}}+\frac{b_a(s_1,s_2)^{3/2}}{N_a\sqrt{N_a-b_a(s_1,s_2)}}\\
&+\frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a-b_a(s_1,s_2))}}\text{sgn}\left(\left|\mathcal{H}_a^{\text{REM}}(s_1,s_2)\right|\right)\sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a-1,s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}\\
&+\sqrt{\frac{N_a-b_a(s_1,s_2)}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}+\frac{\sqrt{b_a(s_1,s_2)(N_a-b_a(s_1,s_2))}}{N_a}\Bigg). \qquad (117)
\end{aligned}$$

Proof. For all Find states we start bounding the norm by splitting the norm by a and sizes of the databases, like in Eq. (96). Let us now go through the three important modes of operation, i.e. adding, updating, or removing from the database.

1 The ADD **case** For the state $|\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle$ we first analyze its norm squared; Forgetting for now the factors multiplying the whole state, the situation is similar to Eq. (101) but instead of focusing on the sum over $y^a_{s_a+1}$ we show that the norm squared of $|\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle$ is a sum of norms of states that differ in the sum over $\vec{y}_{\bar{a}}$. The states on the right hand side have the sum over $\vec{y}_{\bar{a}}$ split according to

$$\sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)\setminus\mathcal{G}_{\bar{a}}(\vec{x}_a\cup\{x\},\vec{x}_{\bar{a}}|\vec{y}_a)} = \sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2|\vec{y}_a)} - \sum_{\vec{y}_{\bar{a}}\in\mathcal{G}_{\bar{a}}(\vec{x}_a\cup\{x\},\vec{x}_{\bar{a}}|\vec{y}_a)}. \qquad (118)$$

Note that both states constructed with sums over sets $\mathcal{G}_{\bar{a}}(\vec{x}_1,\vec{x}_2\mid\vec{y}_a)$ and $\mathcal{G}_{\bar{a}}(\vec{x}_a\cup\{x\},\vec{x}_{\bar{a}}\mid\vec{y}_a)$ have unit norm. The former state has norm equal to $|\Psi_{j-1}^{\text{Good}}(\text{ADD})\rangle$, the norm of this state does not change when replacing register $D(x)$ with an empty entry of the database. The latter state is just the good state before the application of the adversary's unitary: $\mathsf{U}_j^\dagger|\Psi_j^{\text{Good}}(\text{ADD})\rangle$. This analysis follows the same reasoning as presented in the proof of Claim 15. Given that $|\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle$ without the additional factor $\sqrt{\frac{N_a-b_a(s_a+1,s_{\bar{a}})}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}$ has bounded norm we have the following bound:

$$\left\||\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle\right\| \leq \max_{a,s_1,s_2}\sqrt{\frac{N_a-b_a(s_a+1,s_{\bar{a}})}{N_a}}\sqrt{\frac{|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}. \qquad (119)$$

The second state has norm bounded in the following way:

$$\left\||\Psi_{j,2}^{\text{Find}}(\text{ADD})\rangle\right\| \leq \max_{a,s_1,s_2}\sqrt{\frac{b_a(s_a+1,s_{\bar{a}})}{N_a}}. \qquad (120)$$

This bound holds , because except for the factor in front of the state and register $D_a(x)$ the state is just a good state (one from just before the query we analyze in Eq. (79)). Moreover register $D(x)$ is normalized (given the fact that η is explicit in the adversary's register).

2 The UPD **case** In this case we have

$$\left\| |\Psi_{j,1}^{\text{Find}}(\text{UPD})\rangle \right\| \leq \max_{a,s_1,s_2} \frac{b_a(s_1,s_2)^{3/2}}{N_a\sqrt{N_a - b_a(s_1,s_2)}}, \tag{121}$$

where we follow the same reasoning as in the proof of Lem. 14 and Claim 15. For $|\Psi_{j,2}^{\text{Find}}(\text{UPD})\rangle$ we consider the norm square and see that we deal with a difference of two norms with different sets for $\vec{y}_{\bar{a}}$, similarly to $|\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle$, but the split is done as follows:

$$\sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_a \setminus \{x\}, \vec{x}_{\bar{a}} | \vec{y}_a) \setminus \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2 | \vec{y}_a)} = \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_a \setminus \{x\}, \vec{x}_{\bar{a}} | \vec{y}_a)} - \sum_{\vec{y}_{\bar{a}} \in \mathcal{G}_{\bar{a}}(\vec{x}_1, \vec{x}_2 | \vec{y}_a)} . \tag{122}$$

The first state is just $|\Psi_{j,2}^{\text{Bad}}(\text{UPD})\rangle$. The second state is more problematic to deal with , so we just lower bound its norm by 0. We know the bound on $|\Psi_{j,2}^{\text{Bad}}(\text{UPD})\rangle$ so by just taking care of the additional factors we get the bound:

$$\begin{aligned}
&\left\| |\Psi_{j,2}^{\text{Find}}(\text{UPD})\rangle \right\| \\
&\leq \max_{a,s_1,s_2} \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \text{sgn}\left(\left| \mathcal{H}_a^{\text{REM}}(s_1,s_2) \right| \right) \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}},
\end{aligned} \tag{123}$$

where the sign function ensures that if there is no error in $D_{\bar{a}}$ the norm of the state is 0.

3 The REM **case** Finally we have

$$\left\| |\Psi_{j,1}^{\text{Find}}(\text{REM})\rangle \right\| \leq \max_{a,s_1,s_2} \sqrt{\frac{N_a - b_a(s_1,s_2)}{N_a}} \sqrt{\frac{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}}, \tag{124}$$

that can be derived in the same way as the bound on norm of $|\Psi_{j,1}^{\text{Find}}(\text{ADD})\rangle$. For the second state we have

$$\left\| |\Psi_{j,2}^{\text{Find}}(\text{REM})\rangle \right\| \leq \max_{a,s_1,s_2} \frac{\sqrt{b_a(s_1,s_2)(N_a - b_a(s_1,s_2))}}{N_a} \tag{125}$$

and to get it we follow the same reasoning as for $|\Psi_{j,2}^{\text{Find}}(\text{ADD})\rangle$.

We use these bounds and the triangle inequality to bound the second term in Eq. (43):

$$\begin{aligned}
&\left\| \mathsf{J}_R \mathsf{U}_i \mathsf{H} \setminus \mathsf{V}_R \mathsf{U}_{i-1} |\Psi_{i-1}^{\text{Good}}\rangle \right\| \\
&\leq \max_{a \in \{1,2\}, s_1, s_2 \leq i-1} \left(\sqrt{\frac{N_a - b_a(s_a+1, s_{\bar{a}})}{N_a}} \sqrt{\frac{|\mathcal{H}_a^{\text{ADD}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \right. \\
&+ \sqrt{\frac{b_a(s_a+1, s_{\bar{a}})}{N_a}} + \frac{b_a(s_1,s_2)^{3/2}}{N_a\sqrt{N_a - b_a(s_1,s_2)}} \\
&+ \frac{b_a(s_1,s_2)}{\sqrt{N_a(N_a - b_a(s_1,s_2))}} \text{sgn}\left(\left| \mathcal{H}_a^{\text{REM}}(s_1,s_2) \right| \right) \sqrt{\frac{|\mathcal{G}_{\bar{a}}(s_a - 1, s_{\bar{a}})|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} \\
&\left. + \sqrt{\frac{N_a - b_a(s_1,s_2)}{N_a}} \sqrt{\frac{|\mathcal{H}_a^{\text{REM}}(s_1,s_2)|}{|\mathcal{G}_{\bar{a}}(s_1,s_2)|}} + \frac{\sqrt{b_a(s_1,s_2)(N_a - b_a(s_1,s_2))}}{N_a} \right) .
\end{aligned} \tag{126}$$

□

Symbol Index

$x \xleftarrow{\$} \mathcal{X}$	x chosen uniformly from set $\mathcal{X}$	4
$[N]$	The N-element set $[N] := \{0, 1, \ldots, N-1\}$	
$x \leftarrow \mathrm{A}$	x is assigned the output of algorithm A	4
$\lvert x \rvert$	Cardinality of a set x / length of a string x/ absolute value	
A, B	An adversary, a classical or quantum algorithm	8
Bad	A "bad" event in a game.	5
$\mathcal{B}(1 \mid D)$	The bad set of for relation R	13
$\mathsf{CFO}_{\mathcal{Y}}$	Compressed Fourier Oracle	7
$\mathsf{CPhO}_{\mathcal{Y}}$	Compressed Phase Oracle	7
$\mathsf{CStO}_{\mathcal{Y}}$	Compressed Standard Oracle	7
D	The distinguisher	5
$\mathcal{D}$	The set of databases	20
$\mathbf{f}$	A function $\mathbf{f} : \{0,1\}^n \to \{0,1\}^n$ or $\mathbf{f} : [N] \to [N]$.	
Find	Event of measurement of the relation R returning 1	8
FO	Fourier Oracle, $\mathsf{QFT}_N^{YF} \circ \mathsf{StO} \circ \mathsf{QFT}_N^{\dagger YF}$	6
$\mathcal{F}$	The set of functions $\{\mathbf{f} : [N] \to [N]\}$	6
$\lvert \Psi_q^{\mathrm{Good}} \rangle$	The state with database $D \notin R$.	17
$\mathcal{G}(s)$	The good set of for relation R	13
$\lvert \psi \rangle$	A quantum state, a normalized vector in a Hilbert space	
$\lVert \lvert \psi \rangle \rVert$	Norm of a quantum state	
$O(n)$	Complexity class "big O"	
PhO	Phase Oracle, $\mathsf{QFT}_N^{Y} \circ \mathsf{StO} \circ \mathsf{QFT}_N^{\dagger Y}$	6
J_R	Projector on relation R.	15
QFT_N	The Quantum Fourier Transform	4
Rate-1/3	The Rate 1/3 Construction	9
R	Random Oracle	
S	Classical and quantum simulators.	9, 21
StO	Standard Oracle	6
V_R	The unitary outputting $D \in R$.	16
$\oplus$	Bitwise XOR	