Finding Collisions in a Quantum World: Quantum Black-Box Separation of Collision-Resistance and One-Wayness

Akinori Hosoyamada[1,2] and Takashi Yamakawa[1]

[1] NTT Secure Platform Laboratories, NTT Corporation. 3-9-11, Midori-cho Musashino-shi, Tokyo 180-8585, Japan, {hosoyamada.akinori, yamakawa.takashi}@lab.ntt.co.jp

[2] Department of Information and Communication Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya-shi, Nagoya 464-8603, Japan. hosoyamada.akinori@nagoya-u.jp

Abstract. Since the celebrated work of Impagliazzo and Rudich (STOC 1989), a number of black-box impossibility results have been established. However, these works only ruled out classical black-box reductions among cryptographic primitives. Therefore it may be possible to overcome these impossibility results by using quantum reductions. To exclude such a possibility, we have to extend these impossibility results to the quantum setting.

In this paper, we initiate the study of black-box impossibility in the quantum setting. We first formalize a quantum counterpart of fully-black-box reduction following the formalization by Reingold, Trevisan and Vadhan (TCC 2004). Then we prove that there is no quantum fully-black-box reduction from collision-resistant hash functions to one-way permutations (or even trapdoor permutations). We take both of classical and quantum implementations of primitives into account. This is an extension to the quantum setting of the work of Simon (Eurocrypt 1998) who showed a similar result in the classical setting.

keywords post-quantum cryptography, one-way permutation, one-way trapdoor permutation, collision resistant hash function, fully black-box reduction, quantum reduction, impossibility

1 Introduction

1.1 Background

Black-box impossibility. Reductions among cryptographic primitives are fundamental in cryptography. For example, we know reductions from pseudorandom generators, pseudorandom functions, symmetric key encryptions, and digital signatures to one-way functions (OWF). On the other hand, there are some important cryptographic primitives including collision-resistant hash functions (CRH), key-exchanges, public key encryptions (PKE), oblivious transfers, and non-interactive zero-knowledge proofs, for which there are no known reductions to OWF. Given this situation, we want to ask if it is impossible to reduce these primitives to OWF. We remark that under the widely believed assumption that these primitives exist, OWF "imply" these primitives (i.e., these primitives are "reduced" to OWF) in a trivial sense. Therefore to make the question meaningful, we have to somehow restrict types of reductions.

For this purpose, Impagliazzo and Rudich [IR89] introduced the notion of *black-box reductions*. Roughly speaking, a black-box reduction is a reduction that uses an underlying primitive and an adversary in a black-box manner (i.e., use them just as oracles).[3] They proved that there does not exist a black-box reduction from key-exchange protocols (and especially PKE) to one-way permutations (OWP). They also observed that most existing reductions between cryptographic primitives are black-box. Thus their result can be interpreted as an evidence that we cannot construct key-exchange protocols based on OWP with commonly used techniques. After their seminal work, there have been numerous impossibility results of black-box reductions (See Section 1.4 for details).

Post-quantum and quantum cryptography. In 1994. Shor [Sho94] showed that we can efficiently compute integer factorization and discrete logarithm, whose hardness are the basis of widely used cryptographic systems, by using a quantum computer. After that, post-quantum cryptography, which treats classically computable cryptographic schemes that resist quantum attacks, has been intensively studied (e.g., [McE78,Ajt96,Reg05,JF11]). Indeed, NIST has recently started a standardization of post-quantum cryptography [NIS16]. We refer more detailed survey of post-quantum cryptography to [BL17].

As another direction to use quantum computer in cryptography, there have been study of quantum cryptography, in which even honest algorithms also use quantum computers. They include quantum key distribution [BB84], quantum encryption [ABF$^+$16,AGM18], quantum (fully) homomorphic encryption [BJ15,Mah18,Bra18], quantum copy-protection [Aar09], quantum digital signatures [GC01], quantum money [Wie83,AC12,Zha19], etc. We refer more detailed survey of quantum cryptography to [BS16].

Our motivation: black-box impossibility in a quantum world. In this paper, we consider black-box impossibility in a quantum setting where primitives and adversaries are quantum, and a reduction accesses to them quantumly.

Quantum reductions are sometimes more powerful than classical reductions. For example, Regev [Reg05] gave a quantum reduction from the learning with errors (LWE) problem to the decision version of the shortest vector problem (GapSVP) or the shortest independent vectors problem (SIVP). We note that

[3] This is an explanation for *fully-black-box reduction* using the terminology of Reingold, Trevisan, and Vadhan [RTV04]. Since we only consider fully-black-box reductions in this paper, in this introduction, we just say black-box reduction to mean fully-black-box reduction.

there are some follow-up works that give classical reduction between these problems in some parameter settings [Pei09,BLP+13], we still do not know any classical reduction that works in the same parameter setting as the quantum one by Regev. This example illustrates that quantum reductions are sometimes more powerful than classical reductions even if all problem instances (e.g., implementations of primitives, adversaries, and reduction algorithms) are classical. Therefore it may be possible to overcome black-box impossibility results shown in the classical setting by using quantum reductions.

We observe that most existing black-box impossibility results crucially rely on the fact that a reduction only classically calls underlying primitives and adversaries, and cannot be simply extended to the quantum case. Hence if we also want to rule out quantum black-box reductions, we have to give impossibility results considering quantum setting with a new technique. Especially, in this paper, we focus on the impossibility of quantum black-box reductions from CRH to OWP, which was originally shown by Simon [Sim98] in the classical setting, and revisited in some follow-up works [HR04,HHRS07,AS15]. Since both CRH and OWP are fundamental cryptographic primitives, it is a theoretically important problem to study the relation of them in the quantum setting.

1.2 Our Results

First, we formally define the notion of quantum black-box reduction based on the work by Reingold, Trevisan and Vadhan [RTV04], which gave a formal framework for the notion of black-box reductions in the classical setting. Then we prove the following theorem.

Theorem 1 (informal). *There does not exist a quantum black-box reduction from CRH to OWP.*

We note that though we do not know any candidate of OWP that resists quantum attacks, the above theorem is still meaningful since it also rules out quantum black-box reductions from CRH to OWF (since OWP is also OWF).

We also extend the result to obtain the following theorem.

Theorem 2 (informal). *There does not exist a quantum black-box reduction from CRH to trapdoor permutations (TDP).*

At high-level, we rely on the two-oracle technique introduced by Hsiao and Reyzin [HR04] to obtain the above theorems though there are many difficulties to deal with quantum reductions. See Sections 3, 4 and 5 for more details of our techniques.

Remark 1. In this paper, by quantum black-box reduction we denote reductions that have quantum superposed black-box oracle accesses to primitives. We always consider security of primitives against quantum adversaries, and do not discuss primitives that are only secure against classical adversaries. In addition, since our main goal is to show the impossibility of reductions from CRH to OWP and CRH to TDP, and when we consider primitives with interactions in the quantum setting we have some subtle issues that do not matter in the classical setting (e.g., rewinding is sometimes hard in the quantum setting [ARU14]), we treat only primitives such that both of the primitives themselves and security games are non-interactive.

1.3 Technical Overview

Here, we give a brief technical overview of our results. We focus on the proof of Theorem 1 since Theorem 2 can be proven by a natural (yet non-trivial) extension of that of Theorem 1. We remark that we omit many details and often rely on non-rigorous arguments for intuitive explanations in this subsection.

First, we recall the *two-oracle technique*, which is a technique to rule out black-box reductions among cryptographic primitives in the classical setting introduced by Hsiao and Reyzin [HR04]. Roughly speaking, they showed that a black-box reduction from a primitive $\mathcal{P}$ to another primitive $\mathcal{Q}$ does not exist if there exist oracles Φ and Ψ^{Φ} such that $\mathcal{P}$ exists and $\mathcal{Q}$ does not exist relative to these oracles. As our first contribution, we show that a similar argument carries over to the quantum setting if we appropriately define primitives and black-box reductions in the quantum setting.

For proving the separation between CRH and OWP, we consider oracles $\Phi = f$, which is a random permutation over $\{0,1\}^n$, and $\Psi^{\Phi} = \mathsf{ColFinder}^f$, which is an oracle that finds a collision of any function described by an oracle-aided quantum circuit C that accesses f as an oracle by brute-force similarly to the previous works in the classical setting [Sim98,HHRS07,AS15]. Since it is clear that CRH does not exist relative to f and $\mathsf{ColFinder}^f$, what is left is to prove that a random permutation f is hard to invert even if an adversary is given an additional oracle access to $\mathsf{ColFinder}^f$.

We first recall how this was done in the classical setting based on the proof in [AS15].[4] The underlying idea behind the proof is a very simple information theoretic fact often referred to as the "compression argument," which dates back to the work of Gennaro and Trevisan [GT00]: if we can encode a truth table of a random permutation into an encoding that can be decoded to the original truth table with high probability, then the size of the encoding should be almost as large as that of the truth table. Based on this, the strategy of the proof is to encode a truth table of f into an encoding that consists of a "partial truth table" of f that specifies values of $f(x)$ for all $x \in \{0,1\}^n \setminus G$ for an appropriately chosen subset G so that one can decode the encoding to the original truth table by recovering "forgotten values" of $f(x)$ on $x \in G$ by using the power of an adversary $\mathcal{A}$ that inverts the permutation f with oracle accesses to f and $\mathsf{ColFinder}^f$. What is non-trivial in the proof is that the decoding procedure has to simulate oracles f and $\mathsf{ColFinder}^f$ for $\mathcal{A}$ whereas the encoding only contains a partial truth table of f. To overcome this issue, they demonstrated a very

[4] Though the basic idea is similar to the proof of Simon [Sim98], we explain the description in [AS15] since this is more suitable for explaining how we extend the proof to the quantum setting.

clever way of choosing the subset G such that the simulation of oracles f and $\mathsf{ColFinder}^f$ does not require values of f on G. Especially, they showed that the larger the $\mathcal{A}$'s success probability is, the larger the subset G is, i.e., the smaller the encoding size is. By using the lower bound of the encoding size obtained by the compression argument, they upper bound $\mathcal{A}$'s success probability by a negligible function in n.

Unfortunately, their proof cannot be directly extended to the quantum setting since the choice of the subset G crucially relies on the fact that queries by $\mathcal{A}$ are classical. Indeed, $\mathcal{A}$ may query a uniform superposition of all inputs to the oracle f, in which case it is impossible to perfectly simulate the oracle f with a partial truth table. Thus, instead of directly generalizing their proof to the quantum setting, we start from another work by Nayebi et al. [NABT15], which showed that it is hard to invert a random permutation f with a quantum oracle access to f.[5] The proof strategy of their work is similar to the above, and they also rely on the compression argument, but a crucial difference is that they choose the subset G in a randomized way.[6] Specifically, they first choose a random subset $R \subset \{0.1\}^n$ of a certain size, and define G as the set of x such that (1): $\mathcal{A}$ succeeds in inverting $f(x)$ with high probability, (2): $x \in R$, and (3): query magnitudes of $\mathcal{A}$ on any element in $R \setminus \{x\}$ is sufficiently small. The condition (3) implies that $\mathcal{A}$ is still likely to succeed in inverting $f(x)$ even if the function (oracle) f is replaced with any function f' that agrees with f on $\{0,1\}^n \setminus (R \setminus \{x\})$.[7] Especially, a decoder can use the function h_y that agrees with f on $\{0,1\}^n \setminus G$ and returns y on G instead of the original oracle f when it runs $\mathcal{A}$ on an input $y \in f(G)$. Since the function h_y can be implemented by the partial truth table of f on $\{0,1\}^n \setminus G$, the decoder can simulate the oracle for $\mathcal{A}$ to correctly invert y in f for each $y \in f(G)$, which implies that the decoder can recover the original truth table of f from the partial truth table. Finally, they showed that an appropriate choice of parameters gives a lower bound of the size of G, which in turn gives an upper bound of $\mathcal{A}$'s success probability based on the compression argument.

For our purpose, we have to prove that a random permutation is hard to invert for a quantum adversary $\mathcal{A}$ even if it is given a quantum access to the additional oracle $\mathsf{ColFinder}^f$. Here, we make a simplifying assumption that the oracle $\mathsf{ColFinder}^f$ is only classically accessible since this case conveys our essential idea and can be readily generalized to the quantumly accessible case. For generalizing the proof of [NABT15] to our case, we have to find a way to simulate $\mathsf{ColFinder}^f$ by using the partial truth table of f on $\{0,1\}^n \setminus G$.

Before describing our strategy about how to simulate $\mathsf{ColFinder}^f$, here we give its more detailed definition: At the beginning of each game before $\mathcal{A}$ runs relative to $\mathsf{ColFinder}^f$, two permutations $\pi_C^{(1)}, \pi_C^{(2)} \in \mathsf{Perm}(\{0,1\}^m)$ are chosen

[5] Actually, they showed that a random permutation is hard to invert even given a classical advice string.

[6] Such a randomized encoder was also used in some works in the classical setting, e.g., [DTT10].

[7] Formally, this is proven by using the swapping lemma shown by Vazirani [Vaz98, Lem. 3.1]

uniformly at random for each circuit C ($\{0,1\}^m$ is the domain of the function C^f). On each input C, $\mathsf{ColFinder}^f$ runs the following procedures:

1. Set $w^{(1)} \leftarrow \pi_C^{(1)}(0^m)$.
2. Compute $u = C^f(w^{(1)})$ by running the circuit C relative to f on $w^{(1)}$.
3. Find the minimum t such that $C^f(\pi_C^{(2)}(t)) = u$ by running the circuit C relative to f on the input $\pi_C^{(2)}(i)$ and checking whether $C^f(\pi_C^{(2)}(i)) = u$ holds for $i = 0, 1, 2 \ldots$, sequentially. Set $w^{(2)} \leftarrow \pi_C^{(2)}(t)$.
4. Return $(w^{(1)}, w^{(2)}, u)$.

Next, we explain our strategy to simulate $\mathsf{ColFinder}^f$. We choose another random subset $R' \subset \{0,1\}^n$ of a certain size and require two additional conditions for x to be in G: (4): $x \in R'$ and (5): All oracle-aided quantum circuits C queried by $\mathcal{A}$ when it runs on input $f(x)$ is "good" w.r.t. (R', x) in the following sense.[8] We say that C is good w.r.t. (R', x) if query magnitudes of C on any element of $R' \setminus \{x\}$ is "small" when C runs on input $w^{(1)}$ or $w^{(2)}$ relative to f, where $(w^{(1)}, w^{(2)})$ is the collision found by $\mathsf{ColFinder}^f$. Intuitively, the condition (5) implies that a collision $(w^{(1)}, w^{(2)})$ found by $\mathsf{ColFinder}^f$ for any $\mathcal{A}$'s query C is not likely to change even if its oracle f is replaced with any function f' that just agrees with f on $\{0,1\}^n \setminus (R' \setminus \{x\})$. In our proof, suitable permutations $\pi_C^{(1)}$ and $\pi_C^{(2)}$ are fixed and the decoder have the truth table of them. In particular, the decoder knows the correct $w^{(1)} = \pi_C^{(1)}(0^m)$ for each C, and can compute the correct $u = C^f(w^{(1)})$ since the outputs of $C^{f'}(w^{(1)})$ is likely to be the same value as $C^f(w^{(1)})$ if f' agrees with f on $\{0,1\}^n \setminus (R' \setminus \{x\})$ due to the definition of goodness of C. Thus, in this case, the oracle $\mathsf{ColFinder}^f$ seems to be simulatable with the partial truth table of f on $\{0,1\}^n \setminus G$. However, the actual proof is not so simple since $\mathsf{ColFinder}^f$ runs C on $\pi_C^{(2)}(i)$ for (possibly exponentially) many i until it finds the minimum t such that $C^f(\pi_C^{(2)}(t)) = u$, in which case its total query magnitude on $R' \setminus \{x\}$ is not always small. [9] We overcome this issue by carefully designing the way of simulating $\mathsf{ColFinder}^f$ using the partial truth table of f. Specifically, whenever the simulation algorithm picks i, it checks if it is the "correct" one (i.e., $C^f(\pi_C^{(2)}(i)) = u$) by running C on the input $\pi_C^{(2)}(i)$ relative to f' for all possible permutations f' that are consistent with the given partial truth table of f on $\{0,1\}^n \setminus (R' \setminus \{x\})$, and judges that the index i is the correct one only if the outputs of $C^{f'}(\pi_C^{(2)}(i))$ are the same value for all possible oracles f'. (Otherwise, it judges that i is incorrect and do the same again for the next index $(i+1)$.) This procedure prevents the simulation algorithm from outputting a "wrong" collision $(w^{(1)}, \tilde{w}^{(2)})$ that is different from $(w^{(1)}, w^{(2)})$ since the actual function f is one of the candidates of f' with which the validity of the collision is checked. On the other hand, the correct collision $(w^{(1)}, w^{(2)})$ cannot be judged

[8] The definition of "good" given here corresponds to the negation of "bad" defined in the main body.

[9] Note that we consider information theoretic encoder and decoder, and we do not care whether they run efficiently.

to be a wrong one since the outputs of $C^{f'}(w^{(2)})$ are likely to be the same value for all f' due to the definition of goodness of C. In this way, we can simulate both oracles f and $\mathsf{ColFinder}^f$ by using the partial truth table of f on $\{0,1\} \setminus G$. Similarly to the proof in [NABT15], an appropriate choice of parameters enables us to upper bound $\mathcal{A}$'s success probability by a negligible function in n. This implies that OWP exists relative to oracles f and $\mathsf{ColFinder}^f$, and thus there does not exist a black-box reduction from CRH to OWP.

1.4 Related Work

Black-box impossibility. Here, we review existing works on black-box impossibility in the classical setting. We refer more details of these works to [Fis12]. Reingold, Trevisan and Vadhan [RTV04] introduced several notions of black-box reductions (later revisited by Baecher, Brzuska and Fischlin [BBF13]). We only consider *fully-black-box reductions* using their terminology.

Impagliazzo and Rudich [IR89] ruled out black-box reductions from key-exchanges to OWP by using the *relativizing technique*. In this technique, we construct an oracle O such that there exists a primitive $\mathcal{P}$ relative to O but does not exist $\mathcal{Q}$ relative to O. If such an oracle exists, then there does not exist black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$.[10] The relativizing technique can also be found in [Sim98,Rud92,Hof11] etc.

Hsiao and Reyzin [HR04] proposed an extension of the relativizing technique called the *two-oracle technique*. In this technique, we construct an oracle O_1 that gives an "ideal" implementation of a primitive $\mathcal{P}$ and another oracle O_2 that trivially breaks any implementation of a primitive $\mathcal{Q}$, and prove that the security of $\mathcal{P}$ implemented by O_1 still holds even if an adversary is given access to the oracle O_2 in addition to O_1. If we prove this, then there does not exist black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$.[11] The two-oracle technique can also be found in [DOP05,FLR$^+$10,FS12,AS15] etc.

Boneh and Venkatesan [BV98] introduced another technique to rule out black-box reductions called *meta-reduction*. In this technique, we construct a trivial inefficient adversary A against a primitive $\mathcal{P}$ and a simulator S which is computationally indistinguishable from A via oracle accesses by a polynomial-time algorithm. Then a reduction algorithm from $\mathcal{P}$ to $\mathcal{Q}$ works well even if it accesses to the simulator S instead of the adversary A. This means that we can break the security of $\mathcal{Q}$ in polynomial-time. Therefore such a reduction does not exist as long as $\mathcal{Q}$ is secure. Meta-reductions can also be found in [Cor02,Pas11,GW11] etc.

Rotem and Segev [RS18] showed a limitation of black-box impossibility by giving an example that overcomes the black-box impossibility result by Rudich [Rud88] by using a non-black-box reduction. Nonetheless, black-box impossibility

[10] In fact, they ruled out *relativizing reduction* which is a more general type of reductions than fully-black-box reduction.

[11] We note that this technique only rules out fully-black-box reduction unlike the relativizing technique.

results are still meaningful since we know very limited number of non-black-box techniques. Indeed, they left it as an open problem to overcome the black-box separation of CRH and OWP shown by Simon [Sim98].

Bitansky and Degwekar [BD19] gave a new proof for the black-box separation of CRH from OWP in the classical setting, which is conceptually different from previous ones [Sim98,HHRS07,AS15]. However, it is unclear if their proof extends to the quantum setting.

CRH from strong OWF. Holmgren and Lombardi [HL18] gave a construction of CRH based on a stronger variant of OWF which they call one-way product functions (OWPF). However, since they do not give a construction of OWPF from OWF (or OWP) even with exponential security, their result does not overcome the impossibility result by Simon [Sim98].

Impossibility of quantum reduction from OWP to NP hardness. Chia, Hallgren, and Song [CHS18] considered the problem of separating OWP from NP hardness in the quantum setting. They ruled out a special type of quantum reductions called locally random reductions under a certain complexity theoretic assumption. We note that in our work, we do not put any restriction on a type of a reduction as long as it is quantum fully-black-box, and we do not assume any unproven assumption. Also, they focus on the separation of OWP from NP hardness, and do not give a general definition of black-box reduction in the quantum setting. Thus their work is incomparable to ours.

Quantum Generic Attacks. Grover [Gro96] developed the famous database-search algorithm that, given black-box access to a function $f : \{0,1\}^n \rightarrow \{0,1\}$, finds an element x such that $f(x) = 1$ with $O(2^{n/2})$ quantum queries (if such x exists). Brassard, Boyer, Høyer, and Tapp developed a generalized version of the Grover search, which can be used to find a preimage of an n-bit random permutation with $O(2^{n/2})$ queries [BBHT98]. In particular, any n-bit (trapdoor) permutations can be inverted with $O(2^{n/2})$ queries. They also showed that $O(2^{n/2})$ is the tight bound for the database-search problem. Brassrad, Høyer, and Tapp [BHT98] developed a quantum collision-finding algorithm that finds a collision of a 2-to-1 function with $O(2^{n/3})$ queries. Actually their algorithm can be used to find collisions of random functions, and Zhandry [Zha15] showed that $O(2^{n/3})$ is the tight bound to find collisions of random functions in the quantum setting.

Collapsing. Ambainis, Rosmanis, and Unruh have shown that the classical-style definition of computationally binding for commitment schemes is inadequate in the quantum setting [ARU14]. Instead, Unruh introduced the notion of collapse-binding commitment, which is an extension of classical computationally-binding commitment to the quantum setting [Unr16]. He also defined the notion of collapsing hash functions, and showed that collapse-binding commitments can be constructed from collapsing hash functions. The notion of collapsing is stronger than the classical notion of collision-resistance [Unr16], i.e., collapsing hash functions are collision resistant.

Reducibility among cryptographic primitives in the quantum setting. Bennett et al. showed that bit commitments imply oblivious transfers in the quantum setting [BBCS92]. Fehr et al. showed that classical feasibility results carry over unchanged in the quantum setting [FKS$^+$13]. Dupuis et al. proved a general relation between adaptive and non-adaptive strategies in the quantum setting, and developed a secure quantum bit commitment scheme that uses an ideal 1-bit cut-and-choose primitive as a black box [DFLS16]. Song characterized sufficient conditions that classical reductions are converted into quantum reductions [Son14]. Dagdelen et al. showed that giving black-box reductions for Fiat-Shamir transformation in the QROM is presumably hard [DFG13], but later Don et al. and Liu and Zhandry constructed generic reductions for the transformation in the QROM [DFMS19,LZ19].

Quantum random oracle model with auxiliary information. Subsequent to the posting of our work online, Hhan et al. [HXY19] also used the compression technique in the quantum setting to analyze the quantum random oracle model in the presence of auxiliary information. A crucial difference between their work and this work is that they consider a setting where an adversary is given an auxiliary information which is fixed at the beginning of a security game whereas we consider a setting where an adversary can adaptively make a query to the quantum oracle $\mathsf{ColFinder}$ during the game. Thus, our results are incomparable to theirs.

2 Preliminaries

A classical algorithm is a classical Turing machine, and an efficient classical algorithm is a probabilistic efficient Turing machine. We denote the set of positive integers by $\mathbb{N}$. We write A instead of $A \otimes I$ for short, for any linear operator A. For sets X and Y, let $\mathsf{Func}(X, Y)$ denote the set of functions from X to Y, and $\mathsf{Perm}(X)$ denote the set of permutations on X. Let $\Delta(f, g)$ denote the set $\{x \in X | f(x) \neq g(x)\}$ for any functions $f, g \in \mathsf{Func}(X, Y)$. Let $\{0,1\}^*$ denote the set $\cup_{n \geq 1}\{0,1\}^*$, and by abuse of notation we let $\mathsf{Perm}(\{0,1\}^*)$ denote the set of permutations $\{P : \{0,1\}^* \to \{0,1\}^* | P(\{0,1\}^n) = \{0,1\}^n \text{ for each } n \geq 1\}$. When we say that $f : \{0,1\}^* \to \{0,1\}^*$ is a permutation, we assume that $f(\{0,1\}^n) = \{0,1\}^n$ holds for each n, and thus f is in $\mathsf{Perm}(\{0,1\}^*)$ (i.e., in this paper we do not treat permutations such that there exist $n \neq n'$ and $x \in \{0,1\}^n$ such that $f(x) \in \{0,1\}^{n'}$). We say a that a function $f : \mathbb{N} \to \mathbb{R}$ is negligible if, for any positive integer c, $f(n) \leq n^{-c}$ holds for all sufficiently large n, and we write $f(n) \leq \mathsf{negl}(n)$. Moreover, we say that f is non-negligible if, there exists a positive integer c such that $f(n) \geq n^{-c}$ for infinitely many n. Let S be a subset of $\{0,1\}^m$ and $f : S \to \{0,1\}^\ell$ be a function. We identify f with the function $f' : \{0,1\}^m \to \{0,1\}^\ell \cup \{\perp\}$ such that $f(x) = f'(x)$ for $x \in S$ and $f'(x) = \perp$ for $x \notin S$. If $S = \{0,1\}^m$, we call f a *totally defined function*, and otherwise we call f a *partially defined function*.

2.1 Quantum Algorithms

We refer basics of quantum computation to [NC10,KSVV02]. In this paper, we use the computational model of quantum circuits. Let $\mathcal{Q}$ be the standard basis of quantum circuits [KSVV02]. We assume that quantum circuits (without oracle) are constructed over the standard basis $\mathcal{Q}$, and define the size of a quantum circuit as the total number of elements in $\mathcal{Q}$ used to construct it. Let $|C|$ denote the size of each quantum circuit C. An oracle-aided quantum circuit is a quantum circuit with oracle gates. When an oracle-aided quantum circuit is implemented relative to an oracle O represented by a unitary operator U_O, the oracle gates are replaced by U_O. When there are multiple oracles, each oracle gate should specify an index of an oracle. In this paper, we assume that all oracles are stateless, that is, the behavior of the oracle is independent from a previous history and the same for all queries. For a stateless quantum oracle O, we often identify the oracle and a unitary operator that represents the oracle, and use the same notation O for both of them. Note that each classical algorithm can be regarded as a quantum algorithm. We fix an encoding $\mathcal{E}$ of (oracle-aided) quantum circuits to bit strings, and we identify $\mathcal{E}(C)$ with C. For a quantum circuit C, we will denote the event that we measure an output z when we run C on an input x and measure the final state by $C(x) = z$.

First, we define quantum algorithms. We note that we only consider classical-input-output quantum algorithms.

Definition 1 (Quantum algorithms). *A quantum algorithm $\mathcal{A}$ is a family of quantum circuits $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ that acts on a quantum system $\mathcal{H}_n = \mathcal{H}_{n,in} \otimes \mathcal{H}_{n,out} \otimes \mathcal{H}_{n,work}$ for each n. When we feed $\mathcal{A}$ with an input $x \in \{0,1\}^n$, $\mathcal{A}$ runs the circuit $\mathcal{A}_n$ on the initial state $|x\rangle|0\rangle|0\rangle$, measures the final state with the computational basis, and outputs the measurement result of the register which corresponds to $\mathcal{H}_{n,out}$. We say that $\mathcal{A}$ is an efficient quantum algorithm if it is a family of polynomial-size quantum circuits, i.e., there is a polynomial $\lambda(n)$ such that $|\mathcal{A}_n| \leq \lambda(n)$ for all sufficiently large n.*

Remark 2. Though we use a Turing machine for a computational model of classical computation, we use a quantum circuit for a computational model of quantum computation. This is just because quantum circuits are well-studied than quantum Turing machines [Yao93], and is easier to treat. We remark that we do not intend to rule out reductions with full non-uniform techniques as was done in [CLMP13].

Next, we define oracle-aided quantum algorithms, which are quantum algorithms that can access to oracles.

Definition 2 (Oracle-aided quantum algorithms). *An oracle-aided quantum algorithm $\mathcal{A}$ is a family of oracle aided quantum circuits $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ that acts on a quantum system $\mathcal{H}_n = \mathcal{H}_{n,in} \otimes \mathcal{H}_{n,out} \otimes \mathcal{H}_{n,work}$ for each n. Let $O_1 = \{O_{1,i}\}_{i\in\mathbb{N}}, ..., O_t = \{O_{t,i}\}_{i\in\mathbb{N}}$ be families of quantum oracle gates. When we feed $\mathcal{A}$ with an input $x \in \{0,1\}^n$ relative to oracles $(O_1, ..., O_t)$, $\mathcal{A}$ runs the circuit $\mathcal{A}_n^{O_{1,n},...,O_{t,n}}$ on the initial state $|x\rangle|0\rangle|0\rangle$, measures the final state with the computational basis, and outputs the measurement result of the register which corresponds to $\mathcal{H}_{n,out}$. We note that an oracle-aided quantum circuit $\mathcal{A}_n^{O_{1,n},...,O_{t,n}}$ that makes q queries can be described by a unitary operator*

$$\mathcal{A}_n^{O_{1,n},...,O_{t,n}} = \left(\prod_{j=1}^{q(n)} (U_{j,t,n}O_{t,n} \dots U_{j,1,n}O_{1,n})\right) U_{0,n}, \tag{1}$$

where $(U_{0,n}, \{U_{j,1,n}, \dots, U_{j,t,n}\}_{j\in[q]})$ are some unitary operators.

Remark 3. We also often consider an oracle access to a quantum algorithm. This is interpreted as an oracle access to a unitary operator that represents $\mathcal{A}$.

Next, we define *randomized quantum oracles*, which are quantum oracles that flip classical random coins before algorithms start.

Definition 3 (Randomized quantum oracles). *Let R_n be a finite set for each n, and $R := \prod_{n=1}^{\infty} R_n$ (note that each element $r \in R$ is an infinite sequence $(r_1, r_2, \cdots)$). A randomized quantum oracle $O := \{O_r\}_{r\in R}$ is a family of quantum oracles such that $O_{r,n} = O_{r',n}$ if $r_n = r'_n$. When we feed $\mathcal{A}$ with an input $x \in \{0,1\}^n$ relative to O, first r_n is randomly chosen from the finite set R_n (according to some distribution), and then $\mathcal{A}$ runs the circuit $\mathcal{A}_n^{O_{r,n}}$ on the initial state $|x\rangle|0\rangle|0\rangle$. We denote $O_{r,n}$ by O_{r_n} and $\{O_{r_n}\}_{r_n\in R_n}$ by O_n, respectively, and identify O with $\{O_n\}_{n\in\mathbb{N}}$.*

Similarly, when $\mathcal{A}$ is given oracle access to multiple randomized oracles $(O_1, \dots, O_t)$, we consider that an oracle gate is randomly chosen and fixed for each of the t oracles before $\mathcal{A}$ starts. The distributions of $O_1, \dots, O_t$ can be highly dependent.

Remark 4. Later we consider the situation that a quantum algorithm $\mathcal{A}$ has access to a randomized quantum oracle O, and another quantum algorithm $\mathcal{B}$ has access to $\mathcal{A}^O$. This is interpreted as follows: Before $\mathcal{B}$ starts, $r_n \in R_n$ is chosen uniformly at random, and $\mathcal{B}$ is given an oracle access to the unitary operator that represents $\mathcal{A}_n^{O_{r_n}}$. In particular we do not change r_n while $\mathcal{B}$ is running.

Next, we define what "a quantum algorithm computes a function" means.

Definition 4 (Functions computed by quantum algorithms). *A quantum algorithm $\mathcal{A}$ computes a function $f : \{0,1\}^* \to \{0,1\}^*$ if we have $\Pr[\mathcal{A}(x) = f(x)] > 2/3$ for all $n \in \mathbb{N}$ and $x \in \{0,1\}^n$. An oracle-aided quantum algorithm $\mathcal{A}$ computes a function $f : \{0,1\}^* \to \{0,1\}^*$ relative to an oracle Γ if we have $\Pr[\mathcal{A}^\Gamma(x) = f(x)] > 2/3$ for all $n \in \mathbb{N}$ and $x \in \{0,1\}^n$.*

2.2 Technical Lemmas

This section introduces some technical lemmas for later use. First, we use the following lemma as a fact.

Lemma 1 ([ARU14], Lemma 36). $\mathsf{trD}(|\psi_1\rangle\langle\psi_1|, |\psi_2\rangle\langle\psi_2|) \leq \| |\psi_1\rangle - |\psi_2\rangle \|$ *holds for any pure states* $|\psi_1\rangle$ *and* $|\psi_2\rangle$*, where* trD *denotes the trace distance function.*

By applying the above claim, we can show the following lemma.

Lemma 2. *Let* $\Gamma = (f_1, \ldots, f_t), \Gamma' = (f'_1, \ldots, f'_t)$ *be sequences of oracles, and assume that* $\mathcal{A}$ *is given oracle access to either* Γ *or* Γ'*. Then,*

$$\left|\Pr\left[\mathcal{A}^{\Gamma}(x) = z\right] - \Pr\left[\mathcal{A}^{\Gamma'}(x) = z\right]\right| \leq \left\|\mathcal{A}_n^{\Gamma}|x, 0, 0\rangle - \mathcal{A}_n^{\Gamma'}|x, 0, 0\rangle\right\| \tag{2}$$

holds for any input $x \in \{0,1\}^n$ *and output* z*.*

Proof (of Lemma 2). Let $|\phi\rangle = \mathcal{A}_n^{\Gamma}|x, 0, 0\rangle$ and $|\phi'\rangle = \mathcal{A}_n^{\Gamma'}|x, 0, 0\rangle$. In addition, let D, D' be (classical) distributions of outputs of $\mathcal{A}^{\Gamma}$ and $\mathcal{A}^{\Gamma'}$ on input $x \in \{0,1\}^n$, respectively. Then the left hand side of eq. (2) is upper bounded by $\mathsf{TD}(D, D')$, where TD denotes the total variational distance function, and $\mathsf{TD}(D, D') \leq \mathsf{trD}(|\phi\rangle\langle\phi|, |\phi'\rangle\langle\phi'|)$ holds by the basic property of trace distance (see Theorem 9.1 in [NC10], for example). From Lemma 1, $\mathsf{trD}(|\phi\rangle\langle\phi|, |\phi'\rangle\langle\phi'|) \leq \| |\phi\rangle - |\phi'\rangle \|$ follows, and the claim holds. □

Swapping Lemma for Multiple Oracles. Next we introduce a generalized version of the *swapping lemma* [Vaz98, Lem. 3.1] for multiple oracles. The original swapping lemma formalizes our intuition that the measurement outcome of oracle-aided algorithm will not be changed so much even if the output values of the oracles are changed on a small fraction of inputs. Since this paper considers the situation that multiple oracles are available to adversaries, we extend the original lemma to a generalized one so that we can treat multiple oracles. To simplify notation, below often omit the parameter n when it is clear from context (e.g., we write just q instead of $q(n)$). Here we introduce an important notion called *query magnitude*.

Query Magnitude. Let $\Gamma = (f_1, \ldots, f_g)$ be a sequence of quantum oracles, where each f_i is a fixed oracle and not randomized. Let $\mathcal{A}$ be a q-query oracle-aided quantum algorithm relative to the oracle Γ.

Fix an input x, and let $|\phi_j^{f_i}\rangle$ be the quantum state of $\mathcal{A}^{\Gamma}$ on input $x \in \{0,1\}^n$ just before the j-th query to f_i. Without loss of generality, we consider that the unitary operator O_{f_i} acts on the first $(m_i(n) + \ell_i(n))$-qubits of the quantum system. (Here we assume that f_i is a function from $\{0,1\}^{m_i(n)}$ to $\{0,1\}^{\ell_i(n)}$.) Then $|\phi_j^{f_i}\rangle = \sum_{z \in \{0,1\}^{m_i(n)}} \alpha_z |z\rangle \otimes |\psi_z\rangle$ holds for some complex numbers α_z and quantum states $|\psi_z\rangle$. If we measure the first $m_i(n)$ qubits of the state $|\phi_j^{f_i}\rangle$ with the computational basis, we obtain z with probability $|\alpha_z|^2$. Intuitively, this probability corresponds the "probability" that z is sent to f_i as the j-th quantum query by $\mathcal{A}$.

Definition 5 (Query magnitude to f_i).

1. *The* query magnitude of the j-th quantum query of $\mathcal{A}$ to f_i at z on input $x \in \{0,1\}^n$ *is defined by*
$$\mu_{z,j}^{\mathcal{A},f_i}(x) := |\alpha_z|^2. \tag{3}$$
2. *The* (total) query magnitude of $\mathcal{A}$ to f_i at z on input $x \in \{0,1\}^n$ *is defined by*
$$\mu_{z}^{\mathcal{A},f_i}(x) := \sum_j \mu_{z,j}^{\mathcal{A},f_i}(x). \tag{4}$$

The following lemma can be proven in the same way as the original swapping lemma [Vaz98, Lem. 3.1], using the hybrid argument introduced by Bennet et al. [BBBV97].

Lemma 3 (Swapping lemma with multiple oracles). *Let $\Gamma = (f_1, \dots, f_t)$, $\Gamma' = (f'_1, \dots, f'_t)$ be sequences of oracles, where each f_i and f'_i are fixed oracles and not randomized. Assume that $\mathcal{A}$ is given oracle access to either Γ or Γ'. Then*

$$\left\| \mathcal{A}_n^{\Gamma} |x,0,0\rangle - \mathcal{A}_n^{\Gamma'} |x,0,0\rangle \right\| \le 2 \sum_{1 \le i \le t} \sqrt{q(n) \sum_{z \in \Delta(f_i, f'_i)} \mu_z^{\mathcal{A},f_i}(x)} \tag{5}$$

holds for all $x \in \{0,1\}^n$.

Proof. In this proof we write q instead of $q(n)$, for simplicity. For $1 \le k \le q$ and $1 \le \ell \le t$, let $\Gamma_{(k,\ell)}$ be an intermediate oracle between Γ and Γ': When we run an oracle-aided quantum algorithm $\mathcal{A}$ relative to $\Gamma_{(k,\ell)}$, first $\mathcal{A}$ queries to Γ until the k-th query to $f_{\ell-1}$ (or the $(k-1)$-th query to f_t if $\ell = 1$), and then $\mathcal{A}$ queries to Γ' from the k-th query to f_ℓ until the last query to f_t. Then, the corresponding unitary operator $\mathcal{A}_n^{\Gamma_{(k,\ell)}}$ is described as

$$\begin{aligned} \mathcal{A}_n^{\Gamma_{(k,\ell)}} = & \left(\prod_{j=k+1}^{q} \left(U_{j,t,n} O_{f'_t,n} \dots U_{j,1,n} O_{f'_1,n} \right) \right) \\ & \cdot U_{k,t,n} O_{f'_t,n} \cdots O_{f'_\ell,n} U_{k,\ell-1,n} O_{f_{\ell-1},n} \cdots U_{k,1,n} O_{f_1,n} \\ & \cdot \left(\prod_{j=1}^{k-1} \left(U_{j,t,n} O_{f_t,n} \dots U_{j,1,n} O_{f_1,n} \right) \right) U_{0,n}. \end{aligned} \tag{6}$$

Let $|\phi_{(i,j)}^{(k,\ell)}\rangle$ be the quantum state of $\mathcal{A}$ just before the i-th query to f_j or f'_j, when we run $\mathcal{A}$ relative to $\Gamma_{(k,\ell)}$ on input $x \in \{0,1\}^n$. By $|\phi_{(q+1,1)}^{(k,\ell)}\rangle$ we denote the final quantum state of $\mathcal{A}$ when we run $\mathcal{A}$ relative to $\Gamma_{(k,\ell)}$ on input $x \in \{0,1\}^n$. Let $\Gamma_{(q+1,1)}$ denote Γ. Below we regard that $f_{t+1} = f_1$, $f'_{t+1} = f'_1$, and $(k, t+1) =$

$(k+1, 1)$, for simplicity. Then, since unitary operators preserve norms of vectors, we have that

$$\begin{aligned} \left\| \mathcal{A}_n^{\Gamma} |x,0,0\rangle - \mathcal{A}_n^{\Gamma'} |x,0,0\rangle \right\| &= \left\| |\phi^{(q+1,1)}_{(q+1,1)}\rangle - |\phi^{(1,1)}_{(q+1,1)}\rangle \right\| \\ &\leq \sum_{1\leq \ell \leq t} \sum_{1 \leq k \leq q} \left\| |\phi^{(k,\ell+1)}_{(q+1,1)}\rangle - |\phi^{(k,\ell)}_{(q+1,1)}\rangle \right\| \end{aligned} \tag{7}$$

and

$$\left\| |\phi^{(k,\ell+1)}_{(q+1,1)}\rangle - |\phi^{(k,\ell)}_{(q+1,1)}\rangle \right\| = \left\| O_{f_\ell} |\phi^{(k,\ell)}_{(k,\ell)}\rangle - O_{f'_\ell} |\phi^{(k,\ell)}_{(k,\ell)}\rangle \right\| \tag{8}$$

hold. Let $\Pi_{\Delta(f_\ell, f'_\ell)}$ be the projector onto the space spanned by the vectors that correspond to elements of $\Delta(f_\ell, f'_\ell)$. Then we have

$$\begin{aligned} \left\| O_{f_\ell} |\phi^{(k,\ell)}_{(k,\ell)}\rangle - O_{f'_\ell} |\phi^{(k,\ell)}_{(k,\ell)}\rangle \right\| &= \left\| (O_{f_\ell} - O_{f'_\ell}) \Pi_{\Delta(f_\ell,f'_\ell)} |\phi^{(k,\ell)}_{(k,\ell)}\rangle \right\| \\ &\leq 2 \cdot \left\| \Pi_{\Delta(f_\ell,f'_\ell)} |\phi^{(k,\ell)}_{(k,\ell)}\rangle \right\| = 2 \sqrt{\sum_{z \in \Delta(f_\ell, f'_\ell)} \mu^{\mathcal{A}, f_\ell}_{z,k}(x)}. \end{aligned} \tag{9}$$

From inequalities (7), (8), and (9), it follows that

$$\begin{aligned} \left\| \mathcal{A}_n^{\Gamma} |x,0,0\rangle - \mathcal{A}_n^{\Gamma'} |x,0,0\rangle \right\| &\leq 2 \sum_{1\leq \ell \leq t} \sum_{1 \leq k \leq q} \sqrt{\sum_{z \in \Delta(f_\ell, f'_\ell)} \mu^{\mathcal{A}, f_\ell}_{z,k}(x)} \\ &\leq 2 \sum_{1\leq \ell \leq t} \sqrt{q \sum_{1 \leq k \leq q} \sum_{z \in \Delta(f_\ell, f'_\ell)} \mu^{\mathcal{A}, f_\ell}_{z,k}(x)} \\ &= 2 \sum_{1\leq \ell \leq t} \sqrt{q \sum_{z \in \Delta(f_\ell, f'_\ell)} \mu^{\mathcal{A}, f_\ell}_{z}(x)}, \end{aligned} \tag{10}$$

where we used the concavity of the square root function for the second inequality. □

3 Quantum Primitives and Black-Box Quantum Reductions

Here, we define quantum primitives, which is a quantum counterpart of a primitive, in addition to the notion of fully-black-box reduction in quantum regime (see Def. 2.1 and Def. 2.3 in [RTV04] for classical definitions). Note that we consider reductions that have quantum superposed black-box oracle accesses to primitives. We always consider security of primitives against quantum adversaries, and do not discuss primitives that are only secure against classical adversaries. When we consider primitives with interactions in the quantum setting we have some subtle issues that do not matter in the classical setting (e.g., rewinding is sometimes hard in the quantum setting [ARU14]). Thus we treat only primitives such that both of the primitives themselves and security games are non-interactive.

Definition 6 (Quantum primitives). *A quantum primitive* $\mathcal{P}$ *is a pair* $\langle F_{\mathcal{P}}, R_{\mathcal{P}} \rangle$*, where* $F_{\mathcal{P}}$ *is a set of quantum algorithms* $\mathcal{I}$*, and* $R_{\mathcal{P}}$ *is a relation over pairs* $\langle \mathcal{I}, \mathcal{A} \rangle$ *of quantum algorithms* $\mathcal{I} \in F_{\mathcal{P}}$ *and* $\mathcal{A}$*. A quantum algorithm* $\mathcal{I}$ *implements* $\mathcal{P}$ *or is an implementation of* $\mathcal{P}$ *if* $\mathcal{I} \in F_{\mathcal{P}}$*. If* $\mathcal{I} \in F_{\mathcal{P}}$ *is efficient, then* $\mathcal{I}$ *is an efficient implementation of* $\mathcal{P}$*. A quantum algorithm* $\mathcal{A}$ $\mathcal{P}$*-breaks* $\mathcal{I} \in F_{\mathcal{P}}$ *if* $\langle \mathcal{I}, \mathcal{A} \rangle \in R_{\mathcal{P}}$*. A secure implementation of* $\mathcal{P}$ *is an implementation* $\mathcal{I}$ *of* $\mathcal{P}$ *such that no efficient quantum algorithm* $\mathcal{P}$*-breaks* $\mathcal{I}$*. The primitive* $\mathcal{P}$ *quantumly exists if there exists an efficient and secure implementation of* $\mathcal{P}$*.*

Definition 7 (Quantum primitives relative to oracle). *Let* $\mathcal{P} = \langle F_{\mathcal{P}}, R_{\mathcal{P}} \rangle$ *be a quantum primitive, and* $\Gamma = (O_1, \ldots, O_t)$ *be a family of (possibly randomized) quantum oracles. An oracle-aided quantum algorithm* $\mathcal{I}$ *implements* $\mathcal{P}$ *relative to* Γ *or is an implementation of* $\mathcal{P}$ *relative to* Γ *if* $\mathcal{I}^{\Gamma} \in F_{\mathcal{P}}$*. If* $\mathcal{I}^{\Gamma} \in F_{\mathcal{P}}$ *is efficient, then* $\mathcal{I}$ *is an efficient implementation of* $\mathcal{P}$ *relative to* Γ*. A quantum algorithm* $\mathcal{A}$ $\mathcal{P}$*-breaks* $\mathcal{I} \in F_{\mathcal{P}}$ *relative to* Γ *if* $\langle \mathcal{I}^{\Gamma}, \mathcal{A}^{\Gamma} \rangle \in R_{\mathcal{P}}$*. A secure implementation of* $\mathcal{P}$ *is an implementation* $\mathcal{I}$ *of* $\mathcal{P}$ *relative to* Γ *such that no efficient quantum algorithm* $\mathcal{P}$*-breaks* $\mathcal{I}$ *relative to* Γ*. The primitive* $\mathcal{P}$ *quantumly exists relative to* Γ *if there exists an efficient and secure implementation of* $\mathcal{P}$ *relative to* Γ*.*

Remark 5. In the above definition, $\mathcal{I}^{\Gamma}$ and $\mathcal{A}^{\Gamma}$ are considered to be quantum algorithms (rather than oracle-aided quantum algorithms) once an oracle Γ is fixed so that $\mathcal{I}^{\Gamma} \in F_{\mathcal{P}}$ and $\langle \mathcal{I}^{\Gamma}, \mathcal{A}^{\Gamma} \rangle \in R_{\mathcal{P}}$ are well-defined. This is possible since we assume that an oracle Γ is stateless. (If Γ is randomized, we regard the randomness of Γ as a part of the randomness of the quantum algorithms $\mathcal{I}^{\Gamma}$ and $\mathcal{A}^{\Gamma}$. See also Remark 4.)

Next we define quantum fully-black-box reductions, which is a quantum counterpart of fully-black-box reductions [RTV04, Def. 2.3].

Definition 8 (Quantum fully-black-box reductions). *A pair* (G, S) *of efficient oracle-aided quantum algorithms is a quantum fully-black-box reduction from a quantum primitive* $\mathcal{P} = \langle F_{\mathcal{P}}, R_{\mathcal{P}} \rangle$ *to a quantum primitive* $\mathcal{Q} = \langle F_{\mathcal{Q}}, R_{\mathcal{Q}} \rangle$ *if the following two conditions are satisfied:*

1. *For every implementation* $\mathcal{I} \in F_{\mathcal{Q}}$*, we have* $G^{\mathcal{I}} \in F_{\mathcal{P}}$*.*
2. *For every implementation* $\mathcal{I} \in F_{\mathcal{Q}}$ *and every quantum algorithm* $\mathcal{A}$*, if* $\mathcal{A}$ $\mathcal{P}$*-breaks* $G^{\mathcal{I}}$*, then* $S^{\mathcal{A},\mathcal{I}}$ $\mathcal{Q}$*-breaks* $\mathcal{I}$*.*

Hsiao and Reyzin showed that if there exists an oracle (family) that separates primitives $\mathcal{P}$ and $\mathcal{Q}$, then there is no fully-black-box reduction from $\mathcal{P}$ to $\mathcal{Q}$ [HR04, Prop. 1]. The following lemma guarantees that a similar claim holds in the quantum setting. Although we need no arguments which is specific to the quantum setting, we give a proof for completeness.

Lemma 4 (Two oracle technique). *There exists no quantum fully-black-box reduction from* $\mathcal{P}$ *to* $\mathcal{Q}$ *if there exist families of quantum oracles* Γ_1 *and* $\Gamma_2 = \{\Psi_{\lambda}^{\Phi}\}_{\Phi \in \Gamma^1, \lambda \in \Lambda}$*, where* Λ *is a non-empty set, and the following two conditions hold.*

1. Existence of $\mathcal{Q}$. *There exists an efficient oracle-aided quantum algorithm $\mathcal{J}_0$ that satisfies the following conditions:*

1. *$\mathcal{J}_0^{\Phi} \in F_{\mathcal{Q}}$ holds for any $\Phi \in \Gamma_1$.*
2. *For any efficient oracle-aided algorithm $\mathcal{B}$ and any $\lambda \in \Lambda$, there exists $\Phi \in \Gamma_1$ such that $\mathcal{B}^{\Phi,\Psi_\lambda^\Phi}$ does not $\mathcal{Q}$-break $\mathcal{J}_0^{\Phi}$.*

2. Non-Existence of $\mathcal{P}$. *For any efficient oracle-aided quantum algorithm $\mathcal{I}$ such that $\mathcal{I}^{\Phi} \in F_{\mathcal{P}}$ holds for any $\Phi \in \Gamma_1$, there exists an efficient oracle-aided quantum algorithm $\mathcal{A}_{\mathcal{I}}$ and $\lambda \in \Lambda$ such that $\mathcal{A}_{\mathcal{I}}^{\Psi_\lambda^\Phi}$ $\mathcal{P}$-breaks $\mathcal{I}^{\Phi}$ for any $\Phi \in \Gamma_1$.*

Proof. Suppose that there exists a quantum fully-black-box reduction (G, S) from $\mathcal{P} = \langle F_{\mathcal{P}}, R_{\mathcal{P}} \rangle$ to $\mathcal{Q} = \langle F_{\mathcal{Q}}, R_{\mathcal{Q}} \rangle$. Then, by the first property of quantum fully-black-box reduction and the first condition of Lemma 4, $G^{\mathcal{J}_0^{\Phi}} \in F_{\mathcal{P}}$ holds for any $\Phi \in \Gamma_1$. Thus, if we set $\mathcal{I}_0 := G^{\mathcal{J}_0}$, from the second condition of Lemma 4, it follows that there exists an efficient oracle-aided quantum algorithm $\mathcal{A}_{\mathcal{I}_0}$ and $\lambda \in \Lambda$ such that $\mathcal{A}_{\mathcal{I}_0}^{\Psi_\lambda^\Phi}$ $\mathcal{P}$-breaks ${\mathcal{I}_0}^{\Phi}$ for any $\Phi \in \Gamma_1$. Therefore, from the second property of quantum fully-black-box reduction, it follows that $S^{\mathcal{A}_{\mathcal{I}_0}^{\Psi_\lambda^\Phi}, \mathcal{J}_0^{\Phi}}$ $\mathcal{Q}$-breaks $\mathcal{J}_0^{\Phi}$ for any $\Phi \in \Gamma_1$. Since G, $\mathcal{A}_{\mathcal{I}_0}$, and $\mathcal{J}_0$ are all efficient, there exists an efficient oracle-aided quantum algorithm $\mathcal{B}$ such that $\mathcal{B}^{\Phi,\Psi_\lambda^\Phi} = S^{\mathcal{A}_{\mathcal{I}_0}^{\Psi_\lambda^\Phi}, \mathcal{J}_0^{\Phi}}$. Now we have that there exists an efficient oracle-aided algorithm $\mathcal{B}$ and $\lambda \in \Lambda$ such that $\mathcal{B}^{\Phi,\Psi_\lambda^\Phi}$ $\mathcal{Q}$-breaks $\mathcal{J}_0^{\Phi}$ for any $\Phi \in \Gamma_1$. However, it contradicts the second part of the first condition of Lemma 4, which completes the proof. □

Remark 6. Remember that each fixed (resp., randomized) quantum oracle O is an infinite family of unitary gates $\{O_n\}_{n\in\mathbb{N}}$ (resp., $O = \{O_n\}_{n\in\mathbb{N}}$ and $O_n = \{O_{r_n}\}_{r_n \in R_n}$, where R_n is the set of random coins), where O_n is used when an oracle-aided algorithm runs relative to O on an input in $\{0,1\}^n$. For example, (the quantum oracle of) a permutation $f \in \mathsf{Perm}(\{0,1\}^*)$ is represented as a family $\{f_n\}_{n\in\mathbb{N}}$, where $f_n = f|_{\{0,1\}^n}$. We implicitly assume that $\Psi_{\lambda,n}^{\Phi}$ depends only on Φ_n and is independent of Φ_m for $m \neq n$.

Later, to prove impossibility of quantum fully-black-box reductions from collision resistant hash functions to one-way permutations, we will apply this lemma with the condition that Λ is the set of all polynomials in n, $\Gamma_1 = \mathsf{Perm}(\{0,1\}^*)$, and $\Gamma_2 = \{\mathsf{ColFinder}_\lambda^f\}_{f\in\Gamma_1,\lambda\in\Lambda}$. Here, $\mathsf{ColFinder}_\lambda^f$ is a randomized oracle that takes, as inputs, oracle-aided quantum circuits that computes functions, and returns collision of the functions. The number $\lambda(n)$ denotes the maximum size of circuits that $\mathsf{ColFinder}_{\lambda,n}^f$ takes as inputs for each $n \in \mathbb{N}$.

3.1 Concrete Primitives

This section defines concrete quantum primitives. Namely, we define one-way permutations, trapdoor permutations, and collision-resistant hash functions.

We define two quantum counterparts for each classical primitives. One is the *classical-computable* primitive that can be implemented on classical computers, and the other is the *quantum-computable* primitive that can be implemented on quantum computers but may not be implemented on classical computers. Here we note that, in this paper, all adversaries are quantum algorithms for both of classical-computable and quantum-computable primitives.

Definition 9 (One-way permutation). *Quantum-computable (resp., classical-computable) quantum-secure one-way permutation* $\mathsf{QC\text{-}qOWP}$ *(resp.,* $\mathsf{CC\text{-}qOWP}$*) is a quantum primitive defined as follows: Implementation of* $\mathsf{QC\text{-}qOWP}$ *(resp.,* $\mathsf{CC\text{-}qOWP}$*) is an efficient quantum (resp., classical) algorithm* Eval *that computes a function* $f : \{0,1\}^* \to \{0,1\}^*$ *such that* $f_n := f|_{\{0,1\}^n}$ *is a permutation over* $\{0,1\}^n$*. For an implementation* $\mathcal{I}$ *of* $\mathsf{QC\text{-}qOWP}$ *(resp.,* $\mathsf{CC\text{-}qOWP}$*) that computes* f *and a quantum algorithm* $\mathcal{A}$*, we say that* $\mathcal{A}$ $\mathsf{QC\text{-}qOWP}$*-breaks* $\mathcal{I}$ *(resp.,* $\mathsf{CC\text{-}qOWP}$*-breaks* $\mathcal{I}$*) if and only if*

$$\Pr\left[x \stackrel{\$}{\leftarrow} \{0,1\}^n; y \leftarrow f_n(x); x' \leftarrow \mathcal{A}(y) : x' = x\right] \tag{11}$$

is non-negligible.

Remark 7. Since there is no function generation algorithm Gen in the above definition, this captures "public-coin" one-way permutations. This makes the definition of one-way permutations stronger, and thus makes our negative result stronger.

Definition 10 (Trapdoor permutation). *Quantum-computable (resp., classical-computable) quantum-secure trapdoor permutation* $\mathsf{QC\text{-}qTDP}$ *(resp.,* $\mathsf{CC\text{-}qTDP}$*) is a quantum primitive defined as follows: Implementation of* $\mathsf{QC\text{-}qTDP}$ *(resp.,* $\mathsf{CC\text{-}qTDP}$*) is a triplet of efficient quantum (resp., classical) algorithms* $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$*. In addition, we require* $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ *to satisfy the following:*

1. *For any* $(\mathsf{pk}, \mathsf{td})$ *generated by* $\mathsf{Gen}(1^n)$*,* $\mathsf{Eval}(\mathsf{pk}, \cdot)$ *computes a permutation* $f_{\mathsf{pk},n}\{0,1\}^n \to \{0,1\}^n$*.*
2. *For any* $(\mathsf{pk}, \mathsf{td})$ *generated by* $\mathsf{Gen}(1^n)$ *and any* $x \in \{0,1\}^n$*, we have that the inequality* $\Pr[\mathsf{Inv}(\mathsf{td}, f_{\mathsf{pk},n}(x)) = x] > 2/3$ *holds (i.e.,* $\mathsf{Inv}(\mathsf{td}, \cdot)$ *computes* $f^{-1}_{\mathsf{pk},n}(\cdot)$*).*

For an implementation $\mathcal{I} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ *of* $\mathsf{QC\text{-}qTDP}$ *(resp.,* $\mathsf{CC\text{-}qTDP}$*) and a quantum algorithm* $\mathcal{A}$*, we say that* $\mathcal{A}$ $\mathsf{QC\text{-}qTDP}$*-breaks* $\mathcal{I}$ *(resp.,* $\mathsf{CC\text{-}qTDP}$*-breaks* $\mathcal{I}$*) if and only if*

$$\Pr\left[(\mathsf{pk}, \mathsf{td}) \leftarrow \mathsf{Gen}(1^n); x \stackrel{\$}{\leftarrow} \{0,1\}^n; y \leftarrow f_{\mathsf{pk},n}(x); x' \leftarrow \mathcal{A}(\mathsf{pk}, y) : x' = x\right] \tag{12}$$

is non-negligible.

Definition 11 (Collision-resistant hash function). *Quantum-computable (resp., classical-computable) quantum-collision-resistant hash function* $\mathsf{QC\text{-}qCRH}$ *(resp.,* $\mathsf{CC\text{-}qCRH}$*) is a quantum primitive defined as follows: Implementation of* $\mathsf{QC\text{-}qCRH}$ *(resp.,* $\mathsf{CC\text{-}qCRH}$*) is a pair of efficient quantum (resp., classical) algorithms* $(\mathsf{Gen}, \mathsf{Eval})$*.*

$\mathsf{Gen}(1^n)$**:** *This algorithm is given 1^n as input, and outputs a function index σ.*
$\mathsf{Eval}(\sigma, x)$**:** *This algorithm is given a function index σ and $x \in \{0,1\}^{m(n)}$ as input, and outputs $y \in \{0,1\}^{\ell(n)}$.*

In addition, we require $(\mathsf{Gen}, \mathsf{Eval})$ to satisfy the following:

1. *We have $m(n) > \ell(n)$ for all sufficiently large $n \in \mathbb{N}$.*
2. *$\mathsf{Eval}(\sigma, \cdot)$ computes a function $H_\sigma : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ for any σ generated by $\mathsf{Gen}(1^n)$.*

For an implementation $\mathcal{I} = (\mathsf{Gen}, \mathsf{Eval})$ of QC-qCRH *(resp.,* CC-qCRH*) and a quantum algorithm $\mathcal{A}$, we say that $\mathcal{A}$* QC-qCRH*-breaks $\mathcal{I}$ (resp.,* CC-qCRH*-breaks $\mathcal{I}$) if and only if*

$$\Pr\left[\sigma \leftarrow \mathsf{Gen}(1^n); (x, x') \leftarrow \mathcal{A}(\sigma) : H_\sigma(x) = H_\sigma(x')\right] \tag{13}$$

is non-negligible.

Remark 8. Though trapdoor permutations and collision-resistant hash functions are defined to be a tuple of algorithms, we can capture them as quantum primitives as defined in Definition 6 by considering a unified quantum algorithm that runs either of these algorithms depending on prefix of its input. We also remark that any classical algorithm can be seen as a special case of quantum computation, and thus classical-computable variants are also captured as quantum primitives.

4 Impossibility of Reduction from QC-qCRH to CC-qOWP

The goal of this section is to show the following theorem.

Theorem 3. *There exists no quantum fully-black-box reduction from* QC-qCRH *to* CC-qOWP.

To show this theorem, we define two (families of) oracles that separate QC-qCRH from CC-qOWP. That is, we define an oracle that implements CC-qOWP, in addition to an oracle that finds collisions of functions, and then apply the two oracle technique (Lemma 4). Our oracles are quantum analogues of those in previous works on impossibility results [Sim98,HHRS07,AS15] in the classical setting. Roughly speaking, we simply use random permutations f to implement one-way permutations. As for an oracle that finds collisions of functions, we use a randomized oracle ColFinder.

Remark 9. The statement of Theorem 3 is the strongest result among possible quantum (fully-black-box) separations of CRH from OWP, since it also excludes reductions from CC-qCRH to CC-qOWP, reductions from QC-qCRH to QC-qOWP, and reductions from CC-qCRH to QC-qOWP. [12]

[12] Note that it also excludes possible quantum (fully-black-box) reductions from collapsing hash functions to one-way permutations, since the notion of collapsing is stronger than collision-resistance.

Oracle $\mathsf{ColFinder}$**.**

Intuitive Idea. Intuitively, our oracle $\mathsf{ColFinder}^f$ works as follows for each fixed permutation f. As an input, $\mathsf{ColFinder}^f$ takes an an oracle-aided quantum circuit C. Note that, for each permutation f, a partially or totally defined function $F_C^f : \{0,1\}^m \to \{0,1\}^\ell \cup \{\perp\}$ is uniquely determined from C: Here, F_C^f is the function such that $F_C^f(x) = u \in \{0,1\}^\ell$ if and only if $\Pr\left[C^f(x) = u\right] > 2/3$ and $F_C^f(x) = \perp$ if and only if $\Pr\left[C^f(x) = u\right] \leq 2/3$ holds for all $u \in \{0,1\}^\ell$. First, $\mathsf{ColFinder}^f$ chooses $w_{C^f}^{(1)} \in \{0,1\}^m$ uniformly at random, and computes $u = F_C^f(w_{C^f}^{(1)})$ by running the circuit C on input $w_{C^f}^{(1)}$ relative to f. If $F_C^f(w_{C^f}^{(1)}) = \perp$, $\mathsf{ColFinder}^f$ sets $w_{C^f}^{(2)} := \perp$, and returns $(w_{C^f}^{(1)}, w_{C^f}^{(2)}, \perp)$. Second, if $F_C^f(w_{C^f}^{(1)}) \neq \perp$, $\mathsf{ColFinder}^f$ chooses $w_{C^f}^{(2)}$ from $(F_C^f)^{-1}(u)$ uniformly at random. Finally $\mathsf{ColFinder}^f$ returns $(w_{C^f}^{(1)}, w_{C^f}^{(2)}, u)$. If F_C^f is a totally defined function and has many collisions (for example, if $m > \ell$), $\mathsf{ColFinder}^f$ returns a collision of F_C^f with a high probability. The idea of the above oracle $\mathsf{ColFinder}$ originally comes from the seminal work by Simon [Sim98]. Below we give a formal description of $\mathsf{ColFinder}$, following the formalization of Asharov and Segev [AS15].

Formal Description. Here we give a formal description of $\mathsf{ColFinder}$. Let $\lambda : \mathbb{N} \to \mathbb{R}_{\geq 0}$ be a function, and $\mathsf{Circ}(\lambda(n))$ denote the set of oracle-aided quantum circuits C of which size is less than or equal to $\lambda(n)$. Note that $\mathsf{Circ}(\lambda(n))$ is a finite set for each n. Let $\mathit{\Pi}_n = \{\pi_C^{(1)}, \pi_C^{(2)}\}_{C \in \mathsf{Circ}(\lambda(n))}$ be a set of permutations. Here, for each permutation f, C computes a partially or totally defined function $F_C^f : \{0,1\}^m \to \{0,1\}^\ell \cup \{\perp\}$, and $\pi_C^{(1)}, \pi_C^{(2)}$ are permutations over $\{0,1\}^m$ (note that m is independent of f). It can be regarded that $\mathit{\Pi}_n$ assigns two permutations for each circuit in $\mathsf{Circ}(\lambda(n))$. Let $R_{\lambda,n}$ be the set of all possible such assignments $\mathit{\Pi}_n$, and R_λ be the product set $\prod_{n=1}^{\infty} R_{\lambda,n}$.

For each fixed permutation f and a function λ, we define a randomized quantum oracle $\mathsf{ColFinder}_\lambda^f = \{\mathsf{ColFinder}_{\lambda,\mathit{\Pi}}^f\}_{\mathit{\Pi} \leftarrow R_\lambda}$, where $\mathsf{ColFinder}_{\lambda,\mathit{\Pi}}^f = \{\mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^f\}_{n \in \mathbb{N}}$ is a fixed quantum oracle for each $\mathit{\Pi}$ (here by $\mathit{\Pi} \leftarrow R_\lambda$ we ambiguously denote the procedure that $\mathit{\Pi}$ is chosen uniformly at random before adversaries make queries to $\mathsf{ColFinder}_\lambda^f$). When we feed an algorithm $\mathcal{A}$ with an input $x \in \{0,1\}^n$ relative to $\mathsf{ColFinder}_\lambda^f$, first $\mathit{\Pi}_n \in R_{\lambda,n}$ is chosen uniformly at random (i.e., two permutations $\pi_C^{(1)}, \pi_C^{(2)}$ are chosen uniformly at random for each oracle-aided quantum circuit $C \in \mathsf{Circ}(\lambda(n))$), and then $\mathcal{A}$ runs the circuit $\mathcal{A}_n^{\mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^f}$ on the initial state $|x\rangle |0\rangle |0\rangle$. For each fixed n and $\mathit{\Pi}_n$, the deterministic function $\mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^f$ is defined by the following procedures:

1. Take an input C, where C is an oracle-aided quantum circuit.
2. Compute $w_{C^f}^{(1)} := \pi_C^{(1)}(0^m)$.
3. Compute $F_C^f(w_C^{(1)})$. That is, compute the output distribution of C^f on input $w_{C^f}^{(1)}$, find the element y such that $\Pr[C^f(w_{C^f}^{(1)}) = y] > 2/3$, and set $u \leftarrow y$. If there is no such y, set $u \leftarrow \perp$.

4. If $u = \perp$, set $w^{(2)}_{C^f} = \perp$. If $u \neq \perp$, search for the minimum $t \in \{0,1\}^m$ such that $F^f_C(\pi^{(2)}_C(t)) = u$ by checking whether
$$\Pr\left[C^f\left(\pi^{(2)}_C(i)\right) = u\right] > 2/3$$
holds for $i = 0, 1, 2, \ldots$ in a sequential order, and set $w^2_{C^f} := \pi^{(2)}_C(t)$ (note that such t always exists if $u \neq \perp$ since $F^f_C(w^{(1)}_{C^f}) = u$).
5. Return $(w^{(1)}_{C^f}, w^{(2)}_{C^f}, u)$.

Later we will apply Lemma 4 (the two oracle technique) with $\Gamma_1 := \mathsf{Perm}(\{0,1\}^*)$ and $\Gamma_2 := \{\mathsf{ColFinder}^f_\lambda\}_{f\in\Gamma_1,\lambda\in\Lambda}$, where Λ is the set of polynomials in n.

Proof of Theorem 3. It can be proven that Theorem 3 follows from the following proposition. Note that the oracle gate $\mathsf{ColFinder}^f_{\lambda,\Pi,n}$ is (and thus the circuit $\mathcal{A}_n^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}$ is) fixed once f_n and Π_n are fixed, since the output values of $\mathsf{ColFinder}^f_{\lambda,\Pi,n}$ are independent of f_m and Π_m for $m \neq n$.

Proposition 1. *Let λ, q, ϵ be functions such that $0 \leq \lambda(n), q(n)$ and $0 < \epsilon(n) \leq 1$. Let $\mathcal{A}$ be a q-query oracle-aided quantum algorithm. Suppose that there is a function $\eta(n) \leq \lambda(n)$ such that, for each circuit C that $\mathcal{A}_n$ queries to* $\mathsf{ColFinder}$*, C makes at most $\eta(n)$ queries. If*
$$\Pr_{\substack{f_n,\Pi_n\\ y\leftarrow\{0,1\}^n}}\left[x \leftarrow \mathcal{A}_n^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq \epsilon(n) \tag{14}$$
holds for infinitely many n, then there exists a constant const *such that*
$$\max\{q(n), \eta(n)\} \geq \mathsf{const} \cdot \epsilon(n) \cdot 2^{n/7} \tag{15}$$
holds for infinitely many n.

Now we show that Theorem 3 follows from Proposition 1.

Proof (of Theorem 3). Let $\Gamma_1 := \mathsf{Perm}(\{0,1\}^*)$ and $\Gamma_2 := \{\mathsf{ColFinder}^f_\lambda\}_{f\in\Gamma_1,\lambda\in\Lambda}$, where Λ is the set of all polynomials in n. (If $\lambda(n) \leq 0$ for some n, we assume that $\mathsf{ColFinder}^f_{\lambda,n}$ does not take any inputs.) Below we show that the two conditions of Lemma 4 are satisfied.

For the first condition of Lemma 4, we define an oracle-aided quantum algorithm $\mathcal{J}_0$ as follows: When we feed $\mathcal{J}_0$ with an input x relative to a permutation f, $\mathcal{J}_0$ queries x to f and obtains the output $f(x)$. Then $\mathcal{J}_0$ returns $f(x)$ as its output. We show that this algorithm $\mathcal{J}_0$ satisfies the first condition of Lemma 4 (existence of CC-qOWP). It is obvious that $\mathcal{J}_0^f \in F_{\mathsf{CC\text{-}qOWP}}$ for any permutation f, by definition of $\mathcal{J}_0$. Let $\mathcal{B}$ be an efficient oracle-aided quantum algorithm, and λ be a polynomial in n. Now we show the following claim.

Claim. For any efficient oracle-aided quantum algorithm $\mathcal{B}$ and for any polynomial λ, there exists a permutation $f : \{0,1\}^* \to \{0,1\}^*$ such that

$$\Pr_{y \leftarrow \{0,1\}^n}\left[x \leftarrow \mathcal{B}^{f, \mathsf{ColFinder}^f_\lambda}(y) : f(x) = y\right] < 2^{-n/8} \tag{16}$$

holds for all sufficiently large n.

Proof (of Claim). Without loss of generality we assume that there is a polynomial $\eta'(n)$ such that $\eta'(n) = |\mathcal{B}_n|$ holds, since $\mathcal{B}_n$ is an efficient algorithm. Then, for each circuit C that $\mathcal{B}_n$ queries to $\mathsf{ColFinder}$, C makes at most $\eta'(n)$ queries since $|C| \leq |\mathcal{B}_n|$ holds. It suffices to show the claim in the case that $\lambda(n) = |\mathcal{B}_n|$ holds since, in general, the ability of adversaries to invert permutations does not decrease as $\lambda(n)$ becomes large, and the size of quantum circuits that $\mathcal{B}_n$ can query to $\mathsf{ColFinder}$ does not exceed $|\mathcal{B}_n|$. Hence, below we consider the case that $\lambda(n) = \eta'(n) = |\mathcal{B}_n|$ holds. Note that $\mathcal{B}$ can be regarded as a λ-query algorithm in this case, since $\mathcal{B}_n$ cannot make more than $\lambda(n)$ queries.

Since $\mathcal{B}$ is an efficient algorithm and $\lambda(n)$ is a polynomial in n, it follows that

$$\Pr_{\substack{f_n, \Pi_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{B}_n^{f_n, \mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] < 2^{-n/8} \tag{17}$$

for all sufficiently large n, from Proposition 1. Thus, for all sufficiently large n, there exists a permutation f'_n on $\{0,1\}^n$ such that

$$\Pr_{\substack{\Pi_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{B}_n^{f'_n, \mathsf{ColFinder}^{f'}_{\lambda,\Pi,n}}(y) : f'_n(x) = y\right] < 2^{-n/8} \tag{18}$$

holds. Now, let $f' : \{0,1\}^* \to \{0,1\}^*$ be a permutation such that $f'|_{\{0,1\}^n} = f'_n$ for all sufficiently large n. Then

$$\Pr_{y \leftarrow \{0,1\}^n}\left[x \leftarrow \mathcal{B}^{f', \mathsf{ColFinder}^{f'}_\lambda}(y) : f(x) = y\right] < 2^{-n/8} \tag{19}$$

holds for all sufficiently large n. $\square$

From the above claim, it follows that, for any efficient oracle-aided quantum algorithm $\mathcal{B}$ and any $\lambda \in \Lambda$, there exists a permutation f such that

$$\Pr_{y \leftarrow \{0,1\}^n}\left[x \leftarrow \mathcal{B}^{f, \mathsf{ColFinder}^f_\lambda}(y) : f(x) = y\right] < \mathsf{negl}(n) \tag{20}$$

holds, which implies that $\mathcal{B}^{f, \mathsf{ColFinder}^f_\lambda}$ does not $\mathsf{CC\text{-}qOWP}$-break $\mathcal{J}^f_0$ relative to $(f, \mathsf{ColFinder}^f_\lambda)$. Hence the first condition (existence of $\mathsf{CC\text{-}qOWP}$) of Lemma 4 is satisfied.

Next, we show that the second condition (non-existence of $\mathsf{QC\text{-}qCRH}$) of Lemma 4 is satisfied. For any efficient oracle-aided quantum algorithm $\mathcal{I} = (\mathsf{Gen}, \mathsf{Eval})$ such that $\mathcal{I}^f \in F_{\mathsf{CC\text{-}qCRH}}$ holds for any permutation f, let λ be a polynomial such that $\lambda(n) > |\mathcal{I}_n|$ for all n. We define a family of oracle-aided quantum algorithms $\mathcal{A}_{\mathcal{I}}$ as: Given an input σ which is generated by $\mathsf{Gen}(1^n)$, $\mathcal{A}_{\mathcal{I}}$ queries the oracle-aided quantum circuit $\mathsf{Eval}_n(\sigma, \cdot)$ to $\mathsf{ColFinder}^f_\lambda$, obtains an answer $(w^{(1)}, w^{(2)}, H_\sigma(w^{(1)}))$, and finally outputs $(w^{(1)}, w^{(2)})$. When $\mathcal{A}_{\mathcal{I}}^{\mathsf{ColFinder}^f_\lambda}$ is given an input σ, the output will be $(w^{(1)}, w^{(2)})$, where $w^{(1)}$ is uniformly distributed over the domain of $H_\sigma : \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}$ and $w^{(2)}$ is uniformly distributed over the set $H_\sigma^{-1}(H_\sigma(w^{(1)}))$. Since $m(n) > \ell(n)$ holds by definition of implementations of QC-qCRH, the probability that $w^{(1)} \neq w^{(2)}$, which implies that $(w^{(1)}, w^{(2)})$ is a collision of H_σ, is at least $1/4$. Thus it follows that there exists $\mathcal{A}_{\mathcal{I}}$ and $\lambda \in \Lambda$ such that $\mathcal{A}_{\mathcal{I}}^{\mathsf{ColFinder}^f_\lambda}$ CC-qCRH-breaks $\mathcal{I}^f$ for any permutation f. Hence the second condition of Lemma 4 is satisfied. $\square$

Remark 10. In this paper we formally treat only *efficient* reductions such that the circuit sizes of reduction algorithms are polynomial in n. However, the statement of Proposition 1 also excludes *sub-exponential* reductions from CRH to OWP in the quantum setting.

4.1 Proof of Proposition 1

This subsection proves Proposition 1. See Section 1.3 for an intuitive overview of our proof idea. We begin with describing some technical preparations.

Preparations. Without loss of generality we can assume that $q(n), \eta(n), \lambda(n) \geq 1$ holds, since increasing these numbers does not decrease the ability of $\mathcal{A}$ to invert f. We construct another algorithm $\hat{\mathcal{A}}$ that iteratively runs $\mathcal{A}$ to increase the success probability, and then apply the encoding technique to $\hat{\mathcal{A}}$.

Let c be a positive integer. Let $\mathcal{B}_c$ be an oracle-aided quantum algorithm that runs as follows, relative to the oracles f and $\mathsf{ColFinder}^f_\lambda$.

1. Take an input y. Set $\mathsf{guess} \leftarrow \perp$.
2. For $i = 1, \ldots, c\lceil 1/\epsilon(n) \rceil$ do:
3. Run $\mathcal{A}^{f, \mathsf{ColFinder}^f_\lambda}$ on the input y. Let x denote the output.
4. Query x to f. If $f(x) = y$, then set $\mathsf{guess} \leftarrow x$.
5. End For
6. Return guess.

Let $Q(n) := c\lceil 1/\epsilon(n) \rceil (\max\{q(n), \eta(n)\} + 1)$. Then $\mathcal{B}_c$ can be regarded as a Q-query algorithm, and for each quantum circuit C that $\mathcal{B}_c$ queries to $\mathsf{ColFinder}^f_{\lambda,n}$, C makes at most $Q(n)$ queries.

Remark 11. The randomness Π_n of $\mathsf{ColFinder}^f_\lambda$ is chosen before $\mathcal{B}_c$ starts, and unchanged while $\mathcal{B}_c$ is running (see Remark 4).

Lemma 5. *Let p_1, p_2 be any positive constant values such that $0 < p_1, p_2 < 1$. For a sufficiently large integer c, the following condition is satisfied for infinitely many n:*

Condition. *There exist* $X \subset \mathsf{Perm}(\{0,1\}^n)$ *and* Π_n *such that* $|X| \geq p_1 \cdot |\mathsf{Perm}(\{0,1\}^n)|$ *and*

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\Pr\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq 2/3\right] \geq p_2 \tag{21}$$

for all $f_n \in X$.

Proof. Let $p_0 := p_1 + (\frac{2}{3} + \frac{1}{3}p_2)(1 - p_1)$, and c be an integer that satisfies $e^{-c} \leq 1 - p_0$. In what follows, we show that this c satisfies the condition.

First, for each n such that

$$\Pr_{\substack{f_n,\Pi_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{A}_n^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq \epsilon(n) \tag{22}$$

holds, there exists Π_n such that

$$\Pr_{\substack{f_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{A}_n^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq \epsilon(n) \tag{23}$$

holds. Below we fix Π_n that satisfies inequality (23) for each n such that inequality (22) holds.

Now we have that

$$\begin{aligned}\Pr_{\substack{f_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] &\geq 1 - (1 - \epsilon(n))^{\frac{c}{\epsilon(n)}} \\ &= 1 - ((1 - \epsilon(n))^{-\frac{1}{\epsilon(n)}})^{-c} \end{aligned}\tag{24}$$

holds. If $\epsilon(n) = 1$, the right hand side of inequality (24) becomes 1, which is larger than p_0. If $\epsilon(n) < 1$, the right hand side of inequality (24) is lower bounded by $1 - e^{-c} \geq p_0$, here we used the fact that $(1-x)^{-\frac{1}{x}} \geq e$ holds for $0 < x < 1$. Therefore we have that

$$\Pr_{\substack{f_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq p_0 \tag{25}$$

holds.

Here it follows that

$$\Pr_{f_n}\left[\Pr_{y \leftarrow \{0,1\}^n}\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq \frac{2}{3} + \frac{1}{3}p_2\right] \geq p_1 \tag{26}$$

from inequality (25). In other words, there exists $X \subset \mathsf{Perm}(\{0,1\}^n)$ such that

$$|X| \geq p_1 |\mathsf{Perm}(\{0,1\}^n)|$$

and

$$\Pr_{y \leftarrow \{0,1\}^n}\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq \frac{2}{3} + \frac{1}{3}p_2 \tag{27}$$

holds for all $f_n \in X$. Now, from inequality (27), it follows that

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\Pr\left[x \leftarrow \mathcal{B}_{c,n}^{f_n,\mathsf{ColFinder}^f_{\lambda,\Pi,n}}(y) : f_n(x) = y\right] \geq 2/3\right] \geq p_2 \tag{28}$$

for all $f_n \in X$. □

In what follows, we fix constants p_1, p_2 such that $0 < p_1, p_2 < 1$ arbitrarily. Then, from the above lemma, it follows that there exists a constant c that satisfies the condition in Lemma 5 for infinitely many n. Let us denote $\mathcal{B}_c$ by $\hat{\mathcal{A}}$. We use the encoding technique to this Q-query algorithm $\hat{\mathcal{A}}$, here $Q(n) = c\lceil 1/\epsilon(n)\rceil(\max\{q(n), \eta(n)\} + 1)$. Below we fix a sufficiently large n in addition to Π_n and X such that the condition in Lemma 5 is satisfied. For simplicity, we write Q, q, ϵ, η, f, and $\mathsf{ColFinder}^f$ instead of $Q(n)$, $q(n)$, $\epsilon(n)$, $\eta(n)$, f_n, and $\mathsf{ColFinder}^f_{\lambda,\Pi,n}$ respectively, for simplicity.

Information Theoretic Property of Randomized Compression Scheme. Here we introduce an information theoretic property of a randomized compression scheme $(E_r : X \to Y \cup \{\bot\}, D_r : Y \to X \cup \{\bot\})$, where r is chosen according to a distribution $\mathcal{R}$. Generally, if the encoding and decoding success with a constant probability p, then $|Y|$ cannot be much smaller than $|X|$:

Lemma 6 ([DTT10], Fact 10.1). *If there exists a constant $0 \leq p \leq 1$ such that $\Pr_{r\sim\mathcal{R}}[D_r(E_r(x)) = x] \geq p$ holds for all $x \in X$, then $|Y| \geq p \cdot |X|$ holds.*

Below we formally define an encoder E and a decoder D that compress elements (truth tables of permutations) in X. In the encoder E, random coin r is chosen according to a distribution $\mathcal{R}$. On the other hand, we consider that D is deterministic rather than randomized, and regard r as a part of inputs to D. Note that we do not care whether encoding and decoding can be efficiently done, since Lemma 6 describes a purely information theoretic property.

Encoder *E*. Let δ be a sufficiently small constant ($\delta = (1/8)^4$ suffices). When we feed E with $f \in X$ as an input, E first chooses subsets $R, R' \subset \{0,1\}^n$ by the following sampling: For each $x \in \{0,1\}^n$, x is added to R with probability $\delta^{3/2}/Q^2$, and independently added to R' with probability $\delta^{5/2}/Q^4$. (The pair (R, R') is the random coin of E.)

According to the choice of R', "bad" inputs (oracle-aided quantum circuits) to $\mathsf{ColFinder}^f$ are defined for each $x \in \{0,1\}^n$ as follows. Note that now $\pi_C^{(1)}$ and $\pi_C^{(2)}$ have been fixed for each oracle-aided quantum circuit C, and thus the output $\mathsf{ColFinder}^f(C) = (w_{C^f}^{(1)}, w_{C^f}^{(2)}, F_C^f(w_{C^f}^{(1)}))$ is uniquely determined. For each oracle-aided quantum circuit C such that $F_C^f(w_{C^f}^{(1)}) \neq \bot$, we can define query magnitude of C to f on input $w_{C^f}^{(1)}$ and $w_{C^f}^{(2)}$ at $z \in \{0,1\}^n$ (see Definition 5). We say that a quantum circuit C such that $F_C^f(w_{C^f}^{(1)}) \neq \bot$ is *bad* relative to x if

$$\sum_{z \in R' \setminus \{x\}} \mu_z^{C,f}(w_{C^f}^{(1)}) > \frac{\delta}{Q} \tag{29}$$

or

$$\sum_{z\in R'\setminus\{x\}} \mu_z^{C,f}(w_{C^f}^{(2)}) > \frac{\delta}{Q} \tag{30}$$

hold, and otherwise we say that C is *good* relative to x. For quantum circuits C such that $F_C^f(w_{C^f}^{(1)}) = \perp$, we always say that C is *good*. Let $\mathsf{badC}(R', x)$ denote the set of bad circuits relative to x, for each $R' \subset \{0,1\}^n$.

Next, E constructs a set $G \subset \{0,1\}^n$ depending on the input f. Let $I \subset \{0,1\}^n$ be the set of elements x such that $\hat{\mathcal{A}}$ successfully inverts $f(x)$, i.e., $I := \{x \mid \Pr[x' \leftarrow \hat{\mathcal{A}}^{f,\mathsf{ColFinder}^f}(f(x)) : x' = x] \geq 2/3$. Then $|I| \geq p_2 \cdot 2^n$ holds by definition of X (Remember that X is chosen in such a way as to satisfy the condition in Lemma 5). Now, a set G is defined to be the set of elements $x \in I$ that satisfies the following conditions:

Conditions for G.

(Cond. 1) $x \in R \cap R'$.
(Cond. 2) $\sum_{z\in R\setminus\{x\}} \mu_z^{\hat{\mathcal{A}},f}(f(x)) \leq \delta/Q$.
(Cond. 3) $\sum_{C\in\mathsf{badC}(R',x)} \mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x)) \leq \delta/Q$.

Finally, E encodes f into $(f|_{\{0,1\}^n\setminus G}, f(G))$ if $|G| \geq \theta$, where $\theta = (1 - 60\sqrt{\delta})\delta^4 p_2 2^n/2Q^6$. Otherwise E encodes f into $\perp$.

In addition, here we formally define the set Y (the range of E) as

$$Y := \left\{(f|_{\{0,1\}^n\setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| \geq \theta\right\}. \tag{31}$$

In fact $E((R,R'), f) \in Y \cup \{\perp\}$ holds for any choice of (R,R') and any permutation $f \in X$.

Decoder *D*. D takes $(\tilde{f}, \tilde{G})$ as an input in addition to (R,R'), where $\tilde{G} \subset \{0,1\}^n$ and $\tilde{f}$ is a bijection from a subset of $\{0,1\}^n$ onto $\{0,1\}^n \setminus \tilde{G}$, and R, R' are subsets of $\{0,1\}^n$. If $\{0,1\}^n \setminus$ (the domain of $\tilde{f}$) $\not\subset R \cap R'$ holds, then D outputs $\perp$. Otherwise, D decodes $(\tilde{f}, \tilde{G})$ and reconstructs the truth table of a permutation $f \in \mathsf{Perm}(\{0,1\}^n)$ as follows.

For each x in the domain of $\tilde{f}$, D infers the value $f(x)$ as $f(x) := \tilde{f}(x)$. For other elements $x \in \{0,1\}^n$ which is not contained in the domain of $\tilde{f}$, what D now knows is only that $f(x)$ is contained in $\tilde{G}$. To determine the remaining part of the truth table of f, D tries to recover the value $f^{-1}(y)$ for each $y \in \tilde{G}$ by using $\hat{\mathcal{A}}$.

For each fixed $y \in \tilde{G}$, D could succeed to recover the value $f^{-1}(y)$ if D were able to determine the output distribution of $\hat{\mathcal{A}}$ on input y relative to oracles f and $\mathsf{ColFinder}^f$. However, D cannot determine the distribution even though D has no limitation on its running time, since f itself is the permutation of which D wants to reconstruct the truth table, and the behavior of $\mathsf{ColFinder}^f$ depends on f. Thus D instead prepares oracles h_y and SimCF^{h_y} which approximates f and $\mathsf{ColFinder}^f$, respectively, and computes the output distribution of $\hat{\mathcal{A}}^{h_y,\mathsf{SimCF}^{h_y}}$ on input y. SimCF^{h_y} uses a subroutine CalC_y that takes (C, w) as an input (C is an oracle-aided circuit that may make queries to f and computes a function F_C^f, and w is an element of the domain of F_C^f) and simulates the evaluation of $F_C^f(w)$. D finally infers that $f^{-1}(y)$ is the element which $\hat{\mathcal{A}}^{h_y, \mathsf{SimCF}^{h_y}}$ outputs with probability greater than $1/2$. (If there does not exist such an element, then D outputs $\perp$.) Below we describe h_y, CalC_y, and SimCF^{h_y}.

Oracle h_y. The oracle (function) $h_y : \{0,1\}^n \to \{0,1\}^n$ is defined by

$$h_y(z) = \begin{cases} \tilde{f}(z) \text{ if } z \notin R \cap R', \\ y \text{ otherwise.} \end{cases} \tag{32}$$

Subroutine CalC_y. Let $P_{\text{candidate}} := \{h' \in \mathsf{Perm}(\{0,1\}^n)) \mid \Delta(h', h_y) \subset R \cap R'\}$. CalC_y is defined as the following procedures.

1. Take an input (C, w), where C is an oracle-aided circuit and w is an element of the domain of the function F_C.
2. Compute the output distribution of the quantum circuit $C^{h'}$ on input w for each $h' \in P_{\text{candidate}}$, and find $u(C, w, h') \in \{0,1\}^\ell$ such that $\Pr[C^{h'}(w) = u(C, w, h')] > 1/2$. If there is no such value $u(C, w, h')$ for a fixed h', set $u(C, w, h') := \perp$.
3. If $u(C, w, h') = u(C, w, h'') \neq \perp$ for all $h', h'' \in P_{\text{candidate}}$, return the value $u(C, w, h')$. Otherwise return $\perp$.

Oracle SimCF^{h_y}. SimCF^{h_y} is defined as the following procedures:

1. Take an input C, where C is an oracle-aided quantum circuit.
2. Compute $\tilde{w}_{Cf}^{(1)} := \pi_C^{(1)}(0^m)$.
3. If $\mathsf{CalC}_y(C, \tilde{w}_{Cf}^{(1)}) = \perp$, set $\tilde{w}_{Cf}^{(2)} := \perp$.
4. Otherwise, search the minimum $t \in \{0,1\}^m$ such that $\mathsf{CalC}_y(C, \tilde{w}_{Cf}^{(1)}) = \mathsf{CalC}_y(C, \pi_C^{(2)}(t))$ by checking whether $\mathsf{CalC}_y(C, \tilde{w}_{Cf}^{(1)}) = \mathsf{CalC}_y(C, \pi_C^{(2)}(i))$ holds for $i = 0, 1, 2, \ldots$ in a sequential order, and set $\tilde{w}_{Cf}^{(2)} := \pi_C^{(2)}(t)$.
5. Return $(\tilde{w}_{Cf}^{(1)}, \tilde{w}_{Cf}^{(2)}, \mathsf{CalC}_y(C, \tilde{w}_{Cf}^{(1)}))$.

Note that D is an information theoretic decoder, and we do not care whether CalC_y and SimCF^{h_y} run efficiently.

Analyses. Here we give formal analyses. See Section 1.3 for an intuitive overview. The following lemma shows that h_y, CalC_y, and SimCF^{h_y} satisfy some suitable properties. Here we consider the situation that D takes an input $(\tilde{f}, \tilde{G})$ such that $(\tilde{f}, \tilde{G}) = E((R, R'), f)$ for some subsets $R, R' \subset \{0,1\}^n$ and a permutation $f \in \{0,1\}^n$, and tries to recover the value $f^{-1}(y)$ for some $y \in \tilde{G}$.

Lemma 7. *h_y, CalC_y, and SimCF_{h_y} satisfy the following properties.*

1. $\Delta(h_y, f) = R \cap R' \setminus \{f^{-1}(y)\}$ *holds.*
2. $\mathsf{CalC}_y(C, w) = F_C^f(w)$ *or* $\perp$ *holds for any* C *and* w.
3. For each circuit C *which is good relative to* $f^{-1}(y)$ *and satisfies* $F_C^f(w_{C^f}^{(1)}) \neq \perp$, $\mathsf{CalC}_y(C, w_{C^f}^{(1)}) = F_C^f(w_{C^f}^{(1)})$ *and* $\mathsf{CalC}_y(C, w_{C^f}^{(2)}) = F_C^f(w_{C^f}^{(2)})$ *hold. In addition, for each circuit* C *such that* $F_C^f(w_{C^f}^{(1)}) = \perp$, $\mathsf{CalC}_y(C, w_{C^f}^{(1)}) = \perp$ *holds.*
4. $\mathsf{SimCF}^{h_y}(C) = \mathsf{ColFinder}^f(C)$ *holds for each circuit* C *which is good relative to* $f^{-1}(y)$. *In particular,* $\Delta(\mathsf{ColFinder}^f, \mathsf{SimCF}^{h_y}) \subset \mathsf{badC}(R', f^{-1}(y))$ *holds.*

Proof. The first property is obviously satisfied by definition of h_y.

For the second property, since $f \in P_{\text{candidate}}$, if $\mathsf{CalC}_y(C, w) \neq \perp$ then we have $\mathsf{CalC}_y(C, w) = u(C, w, f) \neq \perp$ by definition of CalC_y, and $u(C, w, f) = F_C^f(w)$ always holds. Hence the second property holds.

For the third property, let C be a quantum circuit is good relative to $f^{-1}(y)$ and satisfies $F_C^f(w_{C^f}^{(1)}) \neq \perp$. For each $h' \in P_{\text{candidate}}$, from Lemma 2 we have

$$\Pr\left[C^{h'}(w_{C^f}^{(1)}) = F_C^f(w_{C^f}^{(1)})\right] \geq \Pr\left[C^f(w_{C^f}^{(1)}) = F_C^f(w_{C^f}^{(1)})\right] - \left\| C^f \left| w_{C^f}^{(1)}, 0, 0 \right\rangle - C^{h'} \left| w_{C^f}^{(1)}, 0, 0 \right\rangle \right\|. \tag{33}$$

From the swapping lemma (Lemma 3) it follows that

$$\left\| C^f \left| w_{C^f}^{(1)}, 0, 0 \right\rangle - C^{h'} \left| w_{C^f}^{(1)}, 0, 0 \right\rangle \right\| \leq 2\sqrt{Q \sum_{z \in \Delta(f, h')} \mu_z^{C,f}(w_{C^f}^{(1)})}. \tag{34}$$

Since $\Delta(f, h') \subset R \cap R' \setminus \{f^{-1}(y)\} \subset R' \setminus \{f^{-1}(y)\}$ holds for all $h' \in P_{\text{candidate}}$, and C is a good circuit relative to $f^{-1}(y)$, the right hand side of the above inequality is upper bounded by $2\sqrt{\delta}$ Thus, for a sufficiently small δ we have

$$\Pr\left[C^{h'}(w_{C^f}^{(1)}) = F_C^f(w_{C^f}^{(1)})\right] \geq \frac{2}{3} - 2\sqrt{\delta} > \frac{1}{2}, \tag{35}$$

which implies that $u(C, w_{C^f}^{(1)}, h') = F_C^f(w_{C^f}^{(1)})$ holds for every $h' \in P_{\text{candidate}}$. Thus $\mathsf{CalC}_y(C, w_{C^f}^{(1)}) = F_C^f(w_{C^f}^{(1)})$ holds if C is good relative to $f^{-1}(y)$. The equality $\mathsf{CalC}_y(C, w_{C^f}^{(2)}) = F_C^f(w_{C^f}^{(2)})$ can be shown in the same way. In addition, for a circuit C such that $F_C^f(w_{C^f}^{(1)}) = \perp$, $\mathsf{CalC}_y(C, w_{C^f}^{(1)}) = \perp$ holds since $u(C, w_{C^f}^{(1)}, f) = F_C^f(w_{C^f}^{(1)}) = \perp$ holds. Therefore the third property follows.

The fourth property follows from the definition of SimCF^{h_y}, the second property, and the third property. □

The following lemma shows that the decoding always succeeds if the encoding succeeds.

Lemma 8. *If* $E((R, R'), f) \neq \perp$, *then* $D((R, R'), E((R, R'), f)) = f$ *holds.*

Proof (of Lemma 8). Let $\tilde{f} := f|_{\{0,1\}^n \setminus G}$ and $\tilde{G} := f(G)$. We show that D can correctly recover $x = f^{-1}(y)$ for each $y \in \tilde{G}$.

We apply the swapping lemma (Lemma 3) to the oracle pairs $(f, \mathsf{ColFinder}^f)$ and $(h_y, \mathsf{SimCF}^{h_y})$. Then we have

$$\begin{aligned}&\left\|\hat{\mathcal{A}}_n^{f,\mathsf{ColFinder}^f}|f(x),0,0\rangle - \hat{\mathcal{A}}_n^{h_y,\mathsf{SimCF}^{h_y}}|f(x),0,0\rangle\right\| \\ &\quad\le 2\sqrt{Q\sum_{z\in\Delta(f,h_y)}\mu_z^{\hat{\mathcal{A}},f}(f(x))} + 2\sqrt{Q\sum_{C\in\Delta(\mathsf{ColFinder}^f,\mathsf{SimCF}^{h_y})}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))}.\end{aligned} \tag{36}$$

Since $\Delta(f,h_y) = R\cap R'\setminus\{f^{-1}(y)\} \subset R\setminus\{f^{-1}(y)\} = R\setminus\{x\}$ and $\Delta(\mathsf{ColFinder}^f, \mathsf{SimCF}^{h_y}) \subset \mathsf{badC}(R', f^{-1}(y)) = \mathsf{badC}(R',x)$ from Lemma 7, the right hand side of inequality (36) is upper bounded by

$$2\sqrt{Q\sum_{z\in R\setminus\{x\}}\mu_z^{\hat{\mathcal{A}},f}(f(x))} + 2\sqrt{Q\sum_{C\in\mathsf{badC}(R',x)}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))}. \tag{37}$$

Due to the conditions (Cond. 2) and (Cond. 3) (see p. 25), each term of the above expression is upper bounded by $2\sqrt{\delta}$. Thus, eventually we have

$$\left\|\hat{\mathcal{A}}_n^{f,\mathsf{ColFinder}^f}|f(x),0,0\rangle - \hat{\mathcal{A}}_n^{h_y,\mathsf{SimCF}^{h_y}}|f(x),0,0\rangle\right\| \le 4\sqrt{\delta} \tag{38}$$

Finally, from Lemma 2, for sufficiently small δ it follows that

$$\begin{aligned}&\Pr\left[\hat{\mathcal{A}}^{h_y,\mathsf{SimCF}^{h_y}}(f(x)) = x\right] \\ &\quad\ge \Pr\left[\hat{\mathcal{A}}^{f,\mathsf{ColFinder}^f}(f(x)) = x\right] \\ &\qquad - \left\|\hat{\mathcal{A}}_n^{f,\mathsf{ColFinder}^f}|f(x),0,0\rangle - \mathcal{A}_n^{h_y,\mathsf{ColFinder}^h}|f(x),0,0\rangle\right\| \\ &\quad\ge 2/3 - 4\sqrt{\delta} > 1/2,\end{aligned} \tag{39}$$

which implies that D correctly recovers $x = f^{-1}(y)$. □

The following lemma is a generalization of a claim showed by Nayebi et al [NABT15, Claim 8], which shows that our E and D work well with a constant probability.

Lemma 9. *If $Q^6 \le \delta^4 p_2 2^n/32$,*

$$\Pr_{(R,R')}\left[D((R,R'),E((R,R'),f) = f\right] \ge 0.7 \tag{40}$$

holds for each $f \in X$.

Proof (of Lemma 9). If $|G| \ge \theta$ holds, then it follows that $E((R,R'),f) \ne \perp$ by definition of E, which leads to $D((R,R'),E((R,R'),f) = f$ by Lemma 8.

Therefore, in what follows, we show that $|G| \geq \theta$ holds with a high probability. Let H be the set defined as $H := \{x \in I \mid x \text{ satisfies } (\textsf{Cond. 1}) \}$, J_1 be the set defined as $J_1 := \{x \in I \mid x \text{ satisfies } (\textsf{Cond. 1}) \text{ but does notsatisfy } (\textsf{Cond. 2}) \}$, and J_2 be the set defined as $J_2 := \{x \in I \mid x \text{ satisfies } (\textsf{Cond. 1}) \text{ but does not satisfy } (\textsf{Cond. 3})\}$. Then $|G| \geq |H| - |J_1| - |J_2|$ holds.

First, we show that $|H|$ becomes large with a high probability: Since we have $\mathbf{E}_{R,R'}[|H|] = \delta^4|I|/Q^6$,

$$\Pr_{R,R'}\left[|H| \geq \frac{1}{2} \cdot \frac{\delta^4|I|}{Q^6}\right] \geq 1 - \exp\left[-\frac{1}{8} \cdot \frac{\delta^4|I|}{Q^6}\right] \tag{41}$$

follows from the multiplicative Chernoff bound. Since $|I| \geq p_2 2^n$ holds by definition of I, and $Q^6 \leq \delta^4 p_2 2^n/32$ is assumed, we have

$$\exp\left[-\frac{1}{8} \cdot \frac{\delta^4|I|}{Q^6}\right] \leq \exp[-4] \leq 0.1. \tag{42}$$

Therefore

$$\Pr_{R,R'}\left[|H| \geq \frac{1}{2} \cdot \frac{\delta^4|I|}{Q^6}\right] \geq 0.9 \tag{43}$$

holds.

Second, we show that $|J_1|$ becomes large only with a small probability: For each $x \in I$, we have that

$$\mathbf{E}_R\left[\sum_{z \in R\setminus\{x\}} \mu_z^{\hat{\mathcal{A}},f}(f(x))\right] = \sum_{z \in \{0,1\}^n\setminus\{x\}} \frac{\delta^{3/2}}{Q^2}\mu_z^{\hat{\mathcal{A}},f}(f(x)) \leq \frac{\delta^{3/2}}{Q} \tag{44}$$

holds, where we used the property that $\sum_z \mu_z^{\hat{\mathcal{A}},f}(f(x)) \leq Q$ holds since $\hat{\mathcal{A}}$ is a Q-query algorithm. Hence

$$\Pr_R\left[\sum_{z \in R\setminus\{x\}} \mu_z^{\hat{\mathcal{A}},f}(f(x)) \geq \frac{\delta}{Q}\right] \leq \sqrt{\delta} \tag{45}$$

follows from Markov's inequality. Since the conditions ($\textsf{Cond. 1}$) and ($\textsf{Cond. 2}$) are independent (note that the condition ($\textsf{Cond. 2}$) does not depend on whether $x \in R \cap R'$),

$$\begin{aligned}\Pr_{R,R'}[x \in J_1] &= \Pr_{R,R'}[x \text{ satisfies } (\textsf{Cond. 1})] \cdot \Pr_{R,R'}[x \text{ does not satisfy } (\textsf{Cond. 2})] \\ &\leq (\delta^4/Q^6) \cdot \sqrt{\delta} = \frac{\delta^{9/2}}{Q^6} \end{aligned} \tag{46}$$

holds for each $x \in I$. Now we can show the following claim.

Claim. It holds that

$$\mathbf{E}_{R,R'}[|J_1|] \leq \delta^{9/2}|I|/Q^6. \tag{47}$$

Proof (of Claim). Note that the set J_1 is determined once R and R' are fixed. Let $J_1^{(R,R')}$ denote the set J_1 that corresponds to (R,R'). Let 2^I be the set of subsets of I. For each $x \in I$, define a function $\xi_x : 2^I \to \{0,1\}$ by $\xi_x(J) = 1$ if and only if $x \in J$. Then we have

$$\begin{aligned}
\mathbf{E}_{R,R'}\left[|J_1|\right] &= \mathbf{E}_{R,R'}\left[\sum_{x\in I}\xi_x\left(J_1^{(R,R')}\right)\right] \\
&= \sum_{R_0,R_0'}\left(\sum_{x\in I}\xi_x\left(J_1^{(R_0,R_0')}\right)\right)\cdot \Pr_{R_0,R_0'}\left[(R,R')=(R_0,R_0')\right] \\
&= \sum_{x\in I}\left(\sum_{R_0,R_0'}\xi_x\left(J_1^{(R_0,R_0')}\right)\cdot \Pr_{R_0,R_0'}\left[(R,R')=(R_0,R_0')\right]\right) \\
&= \sum_{x\in I}\Pr_{R,R'}\left[x\in J_1\right] \le |I|\cdot\frac{\delta^{9/2}}{Q^6}, \qquad (48)
\end{aligned}$$

where the last inequality follows from inequality (46). □

From the above claim and Markov's inequality, it follows that

$$\Pr_{R,R'}\left[|J_1| \ge \frac{10\delta^{9/2}|I|}{Q^6}\right] \le 0.1 \qquad (49)$$

holds.

Third, we show that $|J_2|$ becomes large only with a small probability: Remember that, for each $x \in I$, a quantum circuit C becomes bad relative to x if and only if $F_C^f(w_{C^f}^{(1)}) \neq \perp$, and inequalities (29) or (30) hold. Here, for any fixed C and w we have

$$\mathbf{E}_{R'}\left[\sum_{z\in R'\setminus\{x\}}\mu_z^{C,f}(w)\right] = \sum_{z\in\{0,1\}^n\setminus\{x\}}\frac{\delta^{5/2}}{Q^4}\mu_z^{C,f}(w) \le \frac{\delta^{5/2}}{Q^3}, \qquad (50)$$

where we used the property that $\sum_z \mu_z^{C,f}(w) \le Q$ holds since C makes at most Q queries. Thus, the probability that a fixed C such that $F_C^f(w_{C^f}^{(1)}) \neq \perp$ becomes bad relative to x is upper bounded as

$$\Pr_{R'}\left[C \in \mathsf{badC}(R',x)\right] \le \sum_{i=1,2}\Pr_{R'}\left[\sum_{z\in R'\setminus\{x\}}\mu_z^{C,f}(w_{C^f}^i) > \delta/Q\right] \le \frac{2\delta^{3/2}}{Q^2} \qquad (51)$$

by Markov's inequality. Moreover, if C satisfies $F_C^f(w_{C^f}^{(1)}) = \perp$ holds, it follows that $\Pr_{R'}\left[C \in \mathsf{badC}(R',x)\right] = 0$ holds. Therefore

$$\Pr_{R'}\left[C \in \mathsf{badC}(R',x)\right] \le \frac{2\delta^{3/2}}{Q^2} \qquad (52)$$

holds for any quantum circuit C.

Since R' is chosen independently of $\hat{\mathcal{A}}$, we have

$$\begin{aligned}
&\mathbf{E}_{R'}\left[\sum_{C\in\mathsf{badC}(R',x)}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\right]\\
&=\sum_{R_0}\sum_{C\in\mathsf{badC}(R_0,x)}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\cdot\Pr_{R'}[R'=R_0]\\
&=\sum_{R_0}\sum_{C}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\cdot\mathcal{X}_{\mathsf{badC}(R_0,x)}(C)\cdot\Pr_{R'}[R'=R_0]\\
&=\sum_{C}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\cdot\left(\sum_{R_0}\mathcal{X}_{\mathsf{badC}(R_0,x)}(C)\cdot\Pr_{R'}[R'=R_0]\right)\\
&=\sum_{C}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\cdot\Pr_{R'}[C\in\mathsf{badC}(R',x)]\\
&\leq\sum_{C}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\cdot(2\delta^{3/2}/Q^2)\leq 2\delta^{3/2}/Q,
\end{aligned}\tag{53}$$

where $\mathcal{X}_{\mathsf{badC}(R_0,x)}$ is the boolean function such that $\mathcal{X}_{\mathsf{badC}(R_0,x)}(C)=1$ if and only if $C\in\mathsf{badC}(R_0,x)$, and we used the property that $\sum_C\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))\leq Q$ holds since $\hat{\mathcal{A}}$ is a Q-query algorithm. Therefore

$$\Pr_{R'}\left[\sum_{C\in\mathsf{badC}(R',x)}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^f}(f(x))>\delta/Q\right]\leq 2\sqrt{\delta}\tag{54}$$

follows from Markov's inequality. Since the conditions (Cond. 1) and (Cond. 3) are independent (note that the condition (Cond. 3) does not depend on whether $x\in R\cap R'$),

$$\begin{aligned}
\Pr_{R,R'}[x\in J_2]&=\Pr_{R,R'}[x\text{ satisfies }(\mathsf{Cond.\ 1})]\cdot\Pr_{R,R'}[x\text{ does not satisfy }(\mathsf{Cond.\ 3})]\\
&\leq(\delta^4/Q^6)\cdot 2\sqrt{\delta}=\frac{2\delta^{9/2}}{Q^6}
\end{aligned}\tag{55}$$

holds for each $x\in I$. Now we can show the following claim in the same way as we showed that Claim 4.1 holds.

Claim. It holds that

$$\mathbf{E}_{R,R'}[|J_2|]\leq 2\delta^{9/2}|I|/Q^6\tag{56}$$

From the above claim and Markov's inequality, it follows that

$$\Pr_{R,R'}\left[|J_2|\geq\frac{20\delta^{9/2}|I|}{Q^6}\right]\leq 0.1\tag{57}$$

holds.

Finally, we show that $|G|$ becomes large with a high probability: From inequalities (43), (49), and (57) it follows that

$$\Pr_{R,R'}\left[|H| < \frac{1}{2}\cdot\frac{\delta^4|I|}{Q^6} \vee |J_1| \geq \frac{10\delta^{9/2}|I|}{Q^6} \vee |J_2| \geq \frac{20\delta^{9/2}|I|}{Q^6}\right] \leq 0.3. \tag{58}$$

holds. Therefore, with a probability at least $1-0.3=0.7$ it holds that

$$\begin{aligned} |G| \geq |H| - |J_1| - |J_2| &\geq \frac{\delta^4|I|}{2Q^6} - \frac{10\delta^{9/2}|I|}{Q^6} - \frac{20\delta^{9/2}|I|}{Q^6} \\ &= \frac{\delta^4|I|}{2Q^6}\left(1-60\sqrt{\delta}\right) \geq \delta^4(1-60\sqrt{\delta})\frac{p_2 2^n}{2Q^6} = \theta. \end{aligned} \tag{59}$$

Thus we have that

$$\Pr_{R,R'}\left[|G| \geq \theta\right] \geq 0.7, \tag{60}$$

which completes the proof. □

Finally, we show that Proposition 1 follows from the above lemmas.

Proof (of Proposition 1). First, remember that the set Y is defined as

$$Y := \left\{(f|_{\{0,1\}^n\setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| \geq \theta\right\}. \tag{61}$$

For each fixed positive integer $\theta \leq M \leq 2^n$, the cardinality of the set

$$Y_M := \{(f|_{\{0,1\}^n\setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| = M\} \tag{62}$$

is equal to $(2^n-M)!\cdot\binom{2^n}{M} = (2^n)!/M!$. Thus $|Y|$ is upper bounded as

$$|Y| = \sum_{M=\lceil\theta\rceil}^{2^n} \frac{(2^n)!}{M!} \leq 2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!} \tag{63}$$

for sufficiently large n. Here we show the following claim.

Claim. If $Q^6 \leq \delta^4 p_2 2^n/32$, there exists a constant const_1 such that $Q^6 \geq \mathsf{const}_1 \cdot 2^n/n$ holds. We can choose const_1 independently of n.

Proof (of Claim). By definition of X, $|X| \geq p_1(2^n)!$ holds. In addition, from inequality (63), we have $|Y| \leq 2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!}$. Moreover, since now we are assuming that $Q^6 \leq \delta^4 p_2 2^n/32$ holds, it follows that $|Y| \geq 0.7|X|$ from Lemma 6 and Lemma 9. Hence we have $2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!} \geq 0.7\cdot p_1(2^n)!$, which is equivalent to

$$\frac{2^n}{0.7p_1} \geq \lceil\theta\rceil!. \tag{64}$$

Since p_1 is a constant and $n! \geq 2^n$ holds for $n \geq 4$, there exists a constant const_2, which can be taken independently of n, such that $\lceil\mathsf{const}_2\cdot n\rceil! \geq 2^n/(0.7p_1)$ holds. Now we have $\lceil\mathsf{const}_2\cdot n\rceil \geq \lceil\theta\rceil$, which implies that

$$\mathsf{const}_2\cdot n + 1 \geq \theta = \delta^4\left(1-60\sqrt{\delta}\right)\frac{p_2 2^n}{2Q^6} \tag{65}$$

holds. Moreover, since δ and p_2 are also constants, there exists a constant const_1 that is independent of n and

$$Q^6 \geq \mathsf{const}_1 \cdot 2^n/n \tag{66}$$

holds, which completes the proof of the claim. □

Let $\mathsf{const}_3 := \min\{\delta^4 p_2/32, \mathsf{const}_1\}$. Then, from the Claim 4.1, it follows that

$$Q^6 \geq \mathsf{const}_3 \cdot 2^n/n \tag{67}$$

holds. Since $Q = c\left\lceil\frac{1}{\epsilon}\right\rceil(\max\{q,\eta\}+1)$ by definition of Q, we have

$$c^6\left\lceil\frac{1}{\epsilon}\right\rceil^6(\max\{q,\eta\}+1)^6 \geq \mathsf{const}_3 \cdot 2^n/n. \tag{68}$$

Hence there exists a constant const such that

$$\max\{q,\eta\} \geq \mathsf{const} \cdot \epsilon \cdot 2^{n/6}/n^{1/6} \geq \mathsf{const} \cdot \epsilon \cdot 2^{n/7} \tag{69}$$

holds for sufficiently large n, which completes the proof. □

5 Impossibility of Reduction from QC-qCRH to CC-qTDP

The goal of this section is to show the following theorem.

Theorem 4. *There exists no quantum fully-black-box reduction from* QC-qCRH *to* CC-qTDP*.*

To show this theorem, we define two (families of) oracles that separate QC-qCRH from CC-qTDP. That is, we define an oracle that implements trapdoor permutations, in addition to an oracle that finds collisions of functions, and then apply the two oracle technique (Lemma 4).

Remark 12. The statement of Theorem 4 is the strongest result among possible quantum (fully-black-box) separations of CRH from TDP, since it also excludes reductions from CC-qCRH to CC-qTDP, reductions from QC-qCRH to QC-qTDP, and reductions from CC-qCRH to QC-qTDP. [13]

Oracles that separates QC-qCRH from CC-qTDP. Suppose, for each n, we have a permutation $g_n : \{0,1\}^n \to \{0,1\}^n$ and a function $f_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ for each n, where $f_n(z,\cdot) : \{0,1\}^n \to \{0,1\}^n$ is a permutation for each $z \in \{0,1\}^n$. Define $f_n^{\mathrm{inv}} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ by $f_n^{\mathrm{inv}}(z,\cdot) := (f_n(g_n(z),\cdot))^{-1}$ for each z. Let $g := \{g_n\}_{n\in\mathbb{N}}$, $f := \{f_n\}_{n\in\mathbb{N}}$, and $f^{\mathrm{inv}} := \{f_n^{\mathrm{inv}}\}_{n\in\mathbb{N}}$. Define efficient oracle-aided quantum algorithms $(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ relative to (g, f, f^{inv}) as follows.

[13] Note that it also excludes possible quantum (fully-black-box) reductions from collapsing hash functions to one-way permutations, since the notion of collapsing is stronger than collision-resistance.

1. When we feed Gen^g with 1^n as an input, first $\mathsf{td} \in \{0,1\}^n$ is chosen uniformly at random, and then pk is set as $\mathsf{pk} := g_n(\mathsf{td})$. Finally Gen^g outputs $(\mathsf{pk}, \mathsf{td})$.
2. Given an input $(\mathsf{pk}, x) \in \{0,1\}^n \times \{0,1\}^n$, Eval^f queries (pk, x) to f_n, and output $f_n(\mathsf{pk}, x)$.
3. Given an input $(\mathsf{td}, x) \in \{0,1\}^n \times \{0,1\}^n$, $\mathsf{Inv}^{f^{\mathrm{inv}}}$ queries (td, x) to f_n^{inv}, and output $f_n^{\mathrm{inv}}(\mathsf{td}, x)$.

$(\mathsf{Gen}, \mathsf{Eval}, \mathsf{Inv})$ implements $\mathsf{CC\text{-}qTDP}$ relative to (g, f, f^{inv}).

For each fixed g, f and a function λ, define the randomized oracle $\mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}$ in the same way as we defined $\mathsf{ColFinder}$ in Section 4. Note that now each input to $\mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}$ is an oracle-aided quantum circuit C of which circuit size is at most $\lambda(n)$, and that may make queries to g, f, and f^{inv}. Note that, for each permutations f and g, a partially or totally defined function $F_C^{g,f,f^{\mathrm{inv}}} : \{0,1\}^m \to \{0,1\}^\ell \cup \{\perp\}$ is uniquely determined from C: Here, $F_C^{g,f,f^{\mathrm{inv}}}$ is the function such that $F_C^{g,f,f^{\mathrm{inv}}}(x) = y \in \{0,1\}^\ell$ if and only if $\Pr\left[C^{g,f,f^{\mathrm{inv}}}(x) = y\right] > 2/3$ holds, and $F_C^{g,f,f^{\mathrm{inv}}}(x) =\perp$ if and only if $\Pr\left[C^{g,f,f^{\mathrm{inv}}} f(x) = y\right] \leq 2/3$ holds for any $y \in \{0,1\}^\ell$.

We can show that Theorem 4 follows from Proposition 2 below by applying the two oracle technique (Lemma 4) with $\Gamma_1 := \{(g, f, f^{\mathrm{inv}})\}$ and $\Gamma_2 := \{\mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}\}_{(g,f,f^{\mathrm{inv}}) \in \Gamma_1, \lambda \in \Lambda}$, where Λ is the set of polynomials in n, in the same way as Theorem 3 follows from Proposition 1.

Proposition 2. *Let λ, q, ϵ be functions such that $0 \leq \lambda(n), q(n)$ and $0 < \epsilon(n) \leq 1$. Let $\mathcal{A}$ be a q-query oracle-aided quantum algorithm. Suppose that there is a function $\eta(n) \leq \lambda(n)$ such that, for each circuit C that $\mathcal{A}_n$ queries to $\mathsf{ColFinder}$, C makes at most $\eta(n)$ queries. If*

$$\Pr_{\substack{g_n, f_n, \Pi_n \\ y, \mathsf{td} \leftarrow \{0,1\}^n}}\left[\mathsf{pk} \leftarrow g_n(\mathsf{td}), x \leftarrow \mathcal{A}_n^{g_n, f_n, f_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\Pi,n}^{g,f,f^{\mathrm{inv}}}}(\mathsf{pk}, y) : f_n(\mathsf{pk}, x) = y\right] \geq \epsilon(n) \qquad (70)$$

holds for infinitely many n, then there exists a constant const *such that*

$$\max\{q(n), \eta(n)\} \geq \mathsf{const} \cdot \epsilon(n)^3 \cdot 2^{n/42} \qquad (71)$$

holds for infinitely many n. [14]

Remark 13. In this paper we formally treat only *efficient* reductions such that the circuit sizes of reduction algorithms are polynomial in n. However, the statement of Proposition 2 also excludes *sub-exponential* reductions from CRH to TDP in the quantum setting.

[14] Strictly speaking, when we feed an input $(\mathsf{pk}, y) \in \{0,1\}^n \times \{0,1\}^n$, $\mathcal{A}$ should run a quantum circuit denoted by $\mathcal{A}_{2n}$ in our definition of quantum circuits (see Definition 1 and Definition 2). However, in this section we abuse the notation $\mathcal{A}_n$ to denote $\mathcal{A}_{2n}$, for simplicity.

Intuitive Overview of Proof Idea. Here we explain an intuition of our proof idea. We consider three separate cases. In the first and second cases, we can show that the claim of Proposition 2 is reduced to Proposition 1. In the third case, we again use the arguments about randomized compressing schemes to show permutations are hard to invert.

The first case is the one that $\mathcal{A}$ queries td to f^{inv} with a high probability (we denote this event by TDHIT_1). In this case, we can make an oracle-aided quantum query algorithm $\mathcal{B}_1$ that inverts the permutation g, given oracle access to $(g, \mathsf{ColFinder}^g)$. Given $\mathsf{pk} = g(\mathsf{td})$ as an input and oracle access to $(g, \mathsf{ColFinder}^g)$, $\mathcal{B}_1$ runs $\mathcal{A}$ simulating oracles f and f^{inv} itself, and simulating $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}$ by making queries to $\mathsf{ColFinder}^g$. Then $\mathcal{B}_1$ measures a query of $\mathcal{A}$ to f^{inv}. Since $\mathcal{A}$ queries td to f^{inv} with a high probability, $\mathcal{B}_1$ can obtain td with a high probability, which implies that $\mathcal{B}_1$ can invert pk in g. Thus the claim can be reduced to Proposition 1 in this case. From Proposition 1, it follows that $\mathcal{B}_1$ has to make many queries if $\epsilon(n)$ is non-negligible, which implies that $\mathcal{A}$ also has to make many queries.

The second case is the one that $\mathcal{A}$ queries a *trapdoor-hitting* circuit C to $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}$ with a high probability (we denote this event by TDHIT_2). Intuitively, a circuit C is called *trapdoor-hitting* if it queries td to f^{inv} with a high probability on input $w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}$ or $w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}$ (here, $w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}$ and $w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}$ are defined in the same way as $w^{(1)}_{C^f}$ and $w^{(2)}_{C^f}$ in Section 4). In this case, again we can make an oracle-aided quantum query algorithm $\mathcal{B}_2$ that inverts the permutation g, given oracle access to $(g, \mathsf{ColFinder}^g)$. Given $\mathsf{pk} = g(\mathsf{td})$ as an input and oracle access to $(g, \mathsf{ColFinder}^g)$, $\mathcal{B}_2$ runs $\mathcal{A}$ simulating oracles f and f^{inv}, and simulating $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}$ by making queries to $\mathsf{ColFinder}^g$. Then $\mathcal{B}_2$ measures a query of $\mathcal{A}$ to $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}$. Since $\mathcal{A}$ queries a trapdoor-hitting circuit C to $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}$ with a high probability, $\mathcal{B}_2$ can obtain a trapdoor-hitting circuit C with a high probability. Once $\mathcal{B}_2$ obtains a trapdoor-hitting circuit C, $\mathcal{B}_2$ computes the value $\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}(C) = (w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}, w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}, u)$ by simulating f, f^{inv} itself and making queries to its own oracle $\mathsf{ColFinder}^g$. Then $\mathcal{B}_2$ runs C relative to the oracles g, f, and f^{inv} on inputs $w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}$ and $w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}$, and measures some queries of C to f^{inv}. Since the trapdoor-hitting circuit C queries td to f^{inv} with a high probability, $\mathcal{B}_2$ can obtain td with a high probability, which implies that $\mathcal{B}_2$ can invert pk in g. Thus the claim can be reduced to Proposition 1 in this case as well.

The third case is the one that either of TDHIT_1 and TDHIT_2 does not occur (that is, the case that $\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ occurs). In this case, intuitively, we can construct a randomized compressing scheme that compresses the truth table of $f(\mathsf{pk}, \cdot)$ without the oracle $f^{\mathrm{inv}}(\mathsf{td}, \cdot)$ since the query magnitude to $f^{\mathrm{inv}}(\mathsf{td}, \cdot)$ is almost always small if $\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ occurs. In this section, we only describe the difference between the proof for the third case and the proof in Section 4. The complete proof of the third case can be found in Section A.

Formal Proof. Below we give a formal proof. We begin with formally defining trapdoor-hitting circuits, and the events TDHIT_1 and TDHIT_2. Let δ be a sufficiently small constant ($\delta = (1/8)^4$ suffices), and c be a sufficiently large positive constant integer (actually $c = 2$ suffices.) Let $Q(n) := c\left\lceil \frac{12}{\epsilon(n)} \right\rceil (\max\{q(n), \eta(n)\}+1)$, and $\tilde{Q}(n) := c\left\lceil \frac{12}{\epsilon(n)} \right\rceil \cdot Q(n)$. (We will use δ, c, and $Q(n)$ for the compressing technique in the third case, in the almost same way as we did in Section 4. $\eta(n)$ is the upper bound of the number of queries made by the circuits that $\mathcal{A}$ queries to $\mathsf{ColFinder}$.)

Definition of trapdoor-hitting Circuits. For each fixed n, Π, (g, f, f^{inv}), and td, we say that an oracle-aided quantum circuit C is *trapdoor-hitting* if

$$\begin{aligned} &F_C^{g,f,f^{\mathrm{inv}}}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) \neq \perp \\ &\wedge \left(\sum_{z\in\{0,1\}^n} \mu^{C,f^{\mathrm{inv}}}_{(\mathsf{td},z)}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{Q(n)} \vee \sum_{z\in\{0,1\}^n} \mu^{C,f^{\mathrm{inv}}}_{(\mathsf{td},z)}(w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{Q(n)} \right) \end{aligned} \tag{72}$$

holds, or

$$F_C^{g,f,f^{\mathrm{inv}}}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) = \perp \wedge \sum_{z\in\{0,1\}^n} \mu^{C,f^{\mathrm{inv}}}_{(\mathsf{td},z)}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{Q(n)} \tag{73}$$

holds. If C is not trapdoor-hitting, we say that it is a *non-trapdoor-hitting* circuit.

Definition of the events TDHIT_1 *and* TDHIT_2*.* For each n, we define TDHIT_1 as the event that

$$\sum_z \mu^{\mathcal{A},f^{\mathrm{inv}}}_{(\mathsf{td},z)}(\mathsf{pk}, y) > \frac{\delta}{\tilde{Q}(n)} \tag{74}$$

occurs. In addition, for each n, we define TDHIT_2 as the event that

$$\sum_{C:\text{trapdoor-hitting}} \mu_C^{\mathcal{A},\mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}}(\mathsf{pk}, y) > \frac{\delta}{\tilde{Q}(n)} \tag{75}$$

occurs. Below we give a proof of Proposition 2.

Remark 14. Once g, f, td, y, and Π_n are fixed, whether or not the events TDHIT_1 and TDHIT_2 occur is determined, since the left hand side of inequalities (74) and (75) are completely determined.

Proof (Proof of Proposition 2). Let E denote the event that $\mathcal{A}$ inverts the trapdoor permutation, i.e., $f_n(\mathsf{pk}, x) = y$ holds. If $\Pr[E] \geq \epsilon(n)$ holds, then one of the three conditions holds: (1) TDHIT_1 occurs with a high probability, i.e., $\Pr[E \wedge \mathsf{TDHIT}_1] \geq \epsilon(n)/3$ holds, (2)TDHIT_2 occurs with a high probability, i.e., $\Pr[E \wedge \mathsf{TDHIT}_2] \geq \epsilon(n)/3$ holds, or (3)$\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ occurs with a high probability, i.e., $\Pr[E \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)] \geq \epsilon(n)/3$ holds. Below we show that the claim of the proposition holds in each case.

Case 1: The Event TDHIT_1 Occurs. Here we consider the case that TDHIT_1 occurs. That is, we consider the case that

$$\Pr_{\substack{g_n,f_n,\Pi_n \\ y,\mathsf{td}\leftarrow\{0,1\}^n}}\Bigg[\mathsf{pk}\leftarrow g_n(\mathsf{td}), x \leftarrow \mathcal{A}_n^{g_n,f_n,f_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{g,f,f^{\mathrm{inv}}}}(\mathsf{pk},y) : f_n(\mathsf{pk},x) = y \wedge \mathsf{TDHIT}_1\Bigg] \geq \frac{\epsilon(n)}{3} \tag{76}$$

holds for infinitely many n. In this case, for each n such that (76) holds, there exist $y_0 \in \{0,1\}^n$ and $\hat{f}_n$ such that

$$\Pr_{\substack{g_n,\Pi_n \\ \mathsf{td}\leftarrow\{0,1\}^n}}\Bigg[\mathsf{pk}\leftarrow g_n(\mathsf{td}), x \leftarrow \mathcal{A}_n^{g_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}}(\mathsf{pk},y_0) : \hat{f}_n(\mathsf{pk},x) = y_0 \wedge \mathsf{TDHIT}_1\Bigg] \geq \frac{\epsilon(n)}{3}. \tag{77}$$

Under the condition that TDHIT_1 occurs, we have that

$$\sum_z \mu^{\mathcal{A},\hat{f}^{\mathrm{inv}}}_{(\mathsf{td},z),i_0}(\mathsf{pk},y_0) > \frac{\delta}{q(n)\cdot\tilde{Q}(n)} \geq \frac{\delta}{\tilde{Q}(n)^2} \tag{78}$$

holds for some $1 \leq i_0 \leq q(n)$. Below we construct an oracle-aided quantum algorithm $\mathcal{B}_1$ relative to oracles $g \in \mathsf{Perm}(\{0,1\}^n)$ and $\mathsf{ColFinder}^g_{\lambda'}$ (defined in Section 4), where λ' is a function that $\lambda'(n)$ is sufficiently large for each n.

Before describing the algorithm $\mathcal{B}_1$, here we explain that we can simulate the oracles $\hat{f}^{\mathrm{inv}}$ and $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$, given the truth table of $\hat{f}$ and oracle access to g and $\mathsf{ColFinder}^g_{\lambda'}$, with knowing pk but without knowing td.

We begin with explaining how to simulate the oracle $\hat{f}^{\mathrm{inv}}$. Remember that $\hat{f}^{\mathrm{inv}}(z,x) = (\hat{f}(g(z),\cdot))^{-1}(x)$ holds. Thus we can evaluate $\hat{f}^{\mathrm{inv}}$ once by using the truth table of $\hat{f}$ and making two queries to g.

Next we explain how to simulate the oracle $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$. Given an oracle-aided circuit C which may make queries to $g, \hat{f}$, and $\hat{f}^{\mathrm{inv}}$, first we replace each $\hat{f}$ oracle gate in C with the concrete quantum circuit that computes $\hat{f}$, by using the truth table of $\hat{f}$. (Note that here we do not care whether calculations can be done efficiently, and we focus only on the number of queries to g.) Second, we replace each $\hat{f}^{\mathrm{inv}}$ oracle gate in C with an oracle-aided quantum circuit that computes $\hat{f}^{\mathrm{inv}}$ by using the truth table of $\hat{f}$ and making two queries to g, in the same way as we simulate the $\hat{f}^{\mathrm{inv}}$ oracle.

Let C_{fill} denote the resulting circuit. If C is an η-query circuit, then C_{fill} makes at most 3η-queries. By definition of C_{fill}, obviously $\mathsf{ColFinder}^g_{\lambda'}(C_{\mathrm{fill}}) = \mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}(C)$ holds. Thus we can simulate the oracles of $\hat{f}^{\mathrm{inv}}$ and $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$.

Next we give the description of $\mathcal{B}_1$.

Algorithm $\mathcal{B}_1$.

1. $\mathcal{B}_1$ takes $\mathsf{pk} \in \{0,1\}^n$ as an input and is given oracle access to a permutation $g \in \mathsf{Perm}(\{0,1\}^n)$. The truth table of $\hat{f}$ is hardcoded in the description of $\mathcal{B}_1$. Set $\mathsf{guess} \leftarrow \perp$.
2. Repeat the following procedures $\tilde{Q}(n)^2$ times.
   (a) Run the algorithm $\mathcal{A}$ on input y_0 relative to the oracles $g, \hat{f}, \hat{f}^{\mathrm{inv}}$, and $\mathsf{ColFinder}_{\lambda}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}$ before the i_0-th query to $\hat{f}^{\mathrm{inv}}$, and measure the i_0-th query. $\mathcal{B}_1$ simulates the oracles $g, \hat{f}, \hat{f}^{\mathrm{inv}}$, and $\mathsf{ColFinder}_{\lambda}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}$ as we described above. Let $(\tilde{\mathsf{td}}, \tilde{z}) \in \{0,1\}^n \times \{0,1\}^n$ be the measurement result.
   (b) Query $\tilde{\mathsf{td}}$ to g. If $\mathsf{pk} = g(\tilde{\mathsf{td}})$ holds, set $\mathsf{guess} \leftarrow \tilde{\mathsf{td}}$.
3. Return guess.

Analysis of $\mathcal{B}_1$. The number of queries to each of g and $\mathsf{ColFinder}^g$ made by $\mathcal{B}_1$ is at most $\tilde{Q}(n)^2(3q(n)+1) \leq 4\tilde{Q}(n)^3$. In addition, for each oracle aided circuit C that $\mathcal{A}$ queries to $\mathsf{ColFinder}_{\lambda}^{g,f,\hat{f}^{\mathrm{inv}}}$, the number of queries to each oracle made by C is at most $\eta(n)$, by assumption. Hence, for each oracle aided circuit C_{fill} that $\mathcal{B}_1$ queries to $\mathsf{ColFinder}_{\lambda'}^{g}$, the number of queries to g made by C_{fill} is at most $3\eta(n)$.

From inequality (78), under the condition that TDHIT_1 occurs, it follows that the probability that $\mathcal{B}_1$ finds $\tilde{\mathsf{td}}$ such that $\mathsf{pk} = g(\tilde{\mathsf{td}})$ is at least $1-(1-\delta/\tilde{Q}(n)^2)^{\tilde{Q}(n)^2} \geq 1-e^{-\delta}$. (Here we used the fact that $(1-x)^{-\frac{1}{x}} \geq e$ for $0<x<1$.) That is, we have that

$$\Pr_{\substack{g_n,\Pi_n\\ \mathsf{td}\leftarrow\{0,1\}^n}}\left[\mathsf{pk}\leftarrow g_n(\mathsf{td}), \tilde{\mathsf{td}} \leftarrow \mathcal{B}_{1,n}^{g_n,\mathsf{ColFinder}^g_{\lambda',\Pi,n}}(\mathsf{pk}) : \tilde{\mathsf{td}} = \mathsf{td} \,\middle|\, \mathsf{TDHIT}_1\right] \geq 1-e^{-\delta} \tag{79}$$

holds for the $4\tilde{Q}(n)^3$-query algorithm $\mathcal{B}_1$. From inequality (77), it follows that

$$\Pr_{\substack{g_n,\Pi_n,\\ \mathsf{td}\leftarrow\{0,1\}^n}}[\mathsf{TDHIT}_1] \geq \frac{\epsilon(n)}{3} \tag{80}$$

holds for infinitely many n. Therefore we have

$$\Pr_{\substack{g_n,\Pi_n\\ \mathsf{td}\leftarrow\{0,1\}^n}}\left[\mathsf{pk}\leftarrow g_n(\mathsf{td}), \tilde{\mathsf{td}} \leftarrow \mathcal{B}_{1,n}^{g_n,\mathsf{ColFinder}^g_{\lambda',\Pi,n}}(\mathsf{pk}) : \tilde{\mathsf{td}} = \mathsf{td}\right] \geq (1-e^{-\delta})\cdot\frac{\epsilon(n)}{3} \tag{81}$$

for infinitely many n.

Now we can show that there exists a constant const_1 such that

$$\max\left\{4\tilde{Q}(n)^3, 3\eta(n)\right\} \geq \mathsf{const}_1 \cdot \epsilon(n) \cdot 2^{n/7} \tag{82}$$

holds for infinitely many n in the same way as we showed Proposition 1.

Moreover, since $\tilde{Q}(n) = c^2 \left\lceil \frac{12}{\epsilon(n)} \right\rceil^2 (\max\{q(n), \eta(n)\} + 1)$, we have that

$$4c^6 \left\lceil \frac{12}{\epsilon(n)} \right\rceil^6 (\max\{q(n), \eta(n)\} + 1)^3 \geq \mathsf{const}_1 \cdot \epsilon(n) \cdot 2^{n/7}, \tag{83}$$

which implies that there exists a constant const_2 such that

$$\max\{q(n), \eta(n)\} \geq \mathsf{const}_2 \cdot \epsilon(n)^3 \cdot 2^{n/21} \tag{84}$$

for infinitely many n. Therefore the claim holds in this case.

Case 2: The Event TDHIT_2 Occurs. Here we consider the case that TDHIT_2 occurs. That is, we consider the case that

$$\Pr_{\substack{g_n, f_n, \Pi_n \\ y, \mathsf{td} \leftarrow \{0,1\}^n}} \left[\mathsf{pk} \leftarrow g_n(\mathsf{td}), x \leftarrow \mathcal{A}_n^{g_n, f_n, f_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\Pi,n}^{g,f,f^{\mathrm{inv}}}}(\mathsf{pk}, y) : \right.$$
$$\left. f_n(\mathsf{pk}, x) = y \wedge \mathsf{TDHIT}_2 \right] \geq \frac{\epsilon(n)}{3} \tag{85}$$

holds for infinitely many n. In this case, for each n such that inequality (85) holds, again there exist $y_0 \in \{0,1\}^n$ and $\hat{f}_n$ such that

$$\Pr_{\substack{g_n, \Pi_n \\ \mathsf{td} \leftarrow \{0,1\}^n}} \left[\mathsf{pk} \leftarrow g_n(\mathsf{td}), x \leftarrow \mathcal{A}_n^{g_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\Pi,n}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}}(\mathsf{pk}, y_0) : \right.$$
$$\left. \hat{f}_n(\mathsf{pk}, x) = y_0 \wedge \mathsf{TDHIT}_2 \right] \geq \frac{\epsilon(n)}{3}, \tag{86}$$

and we can construct an adversary $\mathcal{B}_2$ that inverts random permutation g_n. Under the condition that TDHIT_2 occurs, we have that

$$\sum_{C:\text{trapdoor-hitting}} \mu_{C,i_0}^{\mathcal{A},\mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}}(\mathsf{td}, y) > \frac{\delta}{\tilde{Q}(n) \cdot q(n)} \geq \frac{\delta}{\tilde{Q}(n)^2} \tag{87}$$

holds for some $1 \leq i_0 \leq q(n)$. In addition, for each trapdoor-hitting circuit C such that $F_C^{g,f,f^{\mathrm{inv}}}(\ w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) \neq \perp$, we have that

$$\sum_{z \in \{0,1\}^n} \mu_{(\mathsf{td},z),j_0}^{C,f^{\mathrm{inv}}}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{\tilde{Q}(n)^2} \text{ or } \sum_{z \in \{0,1\}^n} \mu_{(\mathsf{td},z),j_0}^{C,f^{\mathrm{inv}}}(w^{(2)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{\tilde{Q}(n)^2} \tag{88}$$

for some $1 \leq j_0 \leq \eta(n)$, by definition of trapdoor-hitting circuits and since $\eta(n) \leq Q(n)$. Similarly, for each trapdoor-hitting circuit C such that $F_C^{g,f,f^{\mathrm{inv}}}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) = \perp$, we have that

$$\sum_{z \in \{0,1\}^n} \mu_{(\mathsf{td},z),j_0}^{C,f^{\mathrm{inv}}}(w^{(1)}_{C^{g,f,f^{\mathrm{inv}}}}) > \frac{\delta}{\tilde{Q}(n)^2} \tag{89}$$

for some $1 \leq j_0 \leq \eta(n)$.

Below we construct an oracle-aided quantum algorithm $\mathcal{B}_2$ relative to oracles $g \in \mathsf{Perm}(\{0,1\}^n)$ and $\mathsf{ColFinder}^g_{\lambda'}$ (defined in Section 4), where λ' is a function that $\lambda'(n)$ is sufficiently large for each n. In what follows, without loss of generality we assume that each circuit C that $\mathcal{A}$ queries to $\mathsf{ColFinder}$ makes $\eta(n)$ queries.

Algorithm $\mathcal{B}_2$.

1. $\mathcal{B}_2$ takes $\mathsf{pk} \in \{0,1\}^n$ as an input and is given oracle access to a permutation $g \in \mathsf{Perm}(\{0,1\}^n)$ and $\mathsf{ColFinder}^g_{\lambda'}$. The truth table of $\hat{f}$ is hardcoded in the description of $\mathcal{B}_2$. Set $\mathsf{guess} \leftarrow \perp$.
2. Repeat the following procedures $\tilde{Q}(n)^2$ times.
   (a) Run the algorithm $\mathcal{A}$ on input y_0 relative to the oracles $g, \hat{f}, \hat{f}^{\mathrm{inv}}$, and $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$ before the i_0-th query to $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$, and measure the i_0-th query. $\mathcal{B}_2$ simulates the oracles $g, \hat{f}, \hat{f}^{\mathrm{inv}}$, and $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}$ as we described in the proof of Case 1. Let C be the measurement result.
   (b) Query C_{fill} to $\mathsf{ColFinder}^g_{\lambda'}$ to compute $\mathsf{ColFinder}^{g,\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda}(C) = (w^{(1)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}}, w^{(2)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}}, u)$ (see p. 37 for the definition of C_{fill}).
   (c) For $1 \leq i \leq \eta(n)$, do:
      i. Repeat the following procedures $\tilde{Q}(n)^2$ times.
         A. Run the circuit C on the input $w^{(1)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}}$ relative to $g, \hat{f}, \hat{f}^{\mathrm{inv}}$ before the i-th query to $\hat{f}^{\mathrm{inv}}$, and measure the i-th query. $\mathcal{B}_2$ simulates the oracles $(g, \hat{f}, \hat{f}^{\mathrm{inv}})$ as we described in the proof of Case 1. Let $(\tilde{\mathsf{td}}, z) \in \{0,1\}^n \times \{0,1\}^n$ be the measurement result.
         B. Query $\tilde{\mathsf{td}}$ to g. If $\mathsf{pk} = g(\tilde{\mathsf{td}})$ holds, set $\mathsf{guess} \leftarrow \tilde{\mathsf{td}}$.
         C. If $w^{(2)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}} \neq \perp$, do Steps A and B by using $w^{(2)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}}$ instead of $w^{(1)}_{C^{(g,\hat{f},\hat{f}^{\mathrm{inv}})}}$.
3. Return guess.

Analysis of $\mathcal{B}_2$. First we analyze the number of queries made by $\mathcal{B}_2$. Steps (a) and (b) require at most $3i_0 \leq 3q(n)$ and 1 queries to each oracle, respectively, and the maximum number of queries made by each circuit C_{fill} that $\mathcal{B}_2$ queries to $\mathsf{ColFinder}^g_{\lambda'}$ is at most $3\eta(n)$.

In Step A, C makes at most $\eta(n)$ queries to each oracle. Since $\mathcal{B}_2$ makes at most two queries to g in order to simulate one evaluation of $\hat{f}^{\mathrm{inv}}$, $\mathcal{B}_2$ makes at most $3\eta(n)$ queries in Step A. In Step B, $\mathcal{B}_2$ makes 1 query. Thus, in Step (c), $\mathcal{B}_2$ makes at most $\eta(n) \cdot (\tilde{Q}(n))^2 \cdot 2 \cdot (3\eta(n)+1) \leq 8\tilde{Q}(n)^4$ queries.

Therefore $\mathcal{B}_2$ makes at most $\tilde{Q}(n)^2 \cdot (8\tilde{Q}(n)^4 + (3q(n)+1)) \leq 12\tilde{Q}(n)^6$ queries, and the maximum number of queries made by each circuit C_{fill} that $\mathcal{B}_2$ queries to $\mathsf{ColFinder}^g_{\lambda'}$ is at most $3\eta(n)$.

Second we analyze success probability of $\mathcal{B}_2$. Since inequality (87) holds, under the condition that TDHIT_2 occurs, the probability that $\mathcal{B}_2$ obtains a trapdoor-hitting circuit C in Step 2-(a) at least once while $\mathcal{B}_2$ is running (below we call this event succ_1) is lower bounded by $1-(1-\delta/\tilde{Q}(n)^2)^{\tilde{Q}(n)^2} \geq 1-e^{-\delta}$. Since inequalities (88) or (89) hold for each trapdoor-hitting circuit, under the condition that succ_1 occurs, the probability that $\mathcal{B}_2$ obtains $\tilde{\mathsf{td}}$ such that $\mathsf{pk} = g(\tilde{\mathsf{td}})$ in Step 2-(c)-i at least once while $\mathcal{B}_2$ is running under the condition that succ_1 occurs is lower bounded by $1-(1-\delta/(\tilde{Q}(n))^2)^{(\tilde{Q}(n))^2} \geq 1-e^{-\delta}$. Hence it follows that $\mathcal{B}_2$ finds $\tilde{\mathsf{td}}$ such that $\mathsf{pk} = g(\tilde{\mathsf{td}})$ with a probability at least $(1-e^{-\delta})^2$, under the condition that TDHIT_2 occurs.

Now we have that

$$\Pr_{\substack{g_n,\Pi_n\\ \mathsf{td}\leftarrow\{0,1\}^n}}\left[\mathsf{pk}\leftarrow g_n(\mathsf{td}), \tilde{\mathsf{td}} \leftarrow \mathcal{B}_{2,n}^{g_n,\mathsf{ColFinder}^g_{\lambda',\Pi,n}}(\mathsf{pk}) : \tilde{\mathsf{td}} = \mathsf{td} \,\middle|\, \mathsf{TDHIT}_2\right] \geq (1-e^{-\delta})^2 \tag{90}$$

holds for a $12\tilde{Q}^6$-query quantum algorithm $\mathcal{B}_2$. Moreover, from inequality (86), it follows that

$$\Pr_{\substack{g_n,\Pi_n,\\ \mathsf{td}\leftarrow\{0,1\}^n}}[\mathsf{TDHIT}_2] > \frac{\epsilon(n)}{3} \tag{91}$$

holds for infinitely many n. Therefore we have that

$$\Pr_{\substack{g_n,\Pi_n\\ \mathsf{td}\leftarrow\{0,1\}^n}}\left[\mathsf{pk}\leftarrow g_n(\mathsf{td}), \tilde{\mathsf{td}} \leftarrow \mathcal{B}_{2,n}^{g_n,\mathsf{ColFinder}^g_{\lambda',\Pi,n}}(\mathsf{pk}) : \tilde{\mathsf{td}} = \mathsf{td}\right] \geq (1-e^{-\delta})^2 \cdot \frac{\epsilon(n)}{3} \tag{92}$$

holds for infinitely many n. Thus we can show that there exists a constant const_1 such that

$$\max\left\{12\tilde{Q}(n)^6, 3\eta(n)\right\} \geq \mathsf{const}_1 \cdot \epsilon(n) \cdot 2^{n/7} \tag{93}$$

holds for infinitely many n, in the almost same way as we showed Proposition 1.

Moreover, since $\tilde{Q}(n) = c^2\left\lceil\frac{12}{\epsilon(n)}\right\rceil^2(\max\{q(n),\eta(n)\}+1)$, we have that

$$12c^{12}\left\lceil\frac{12}{\epsilon(n)}\right\rceil^{12}(\max\{q(n),\eta(n)\}+1)^6 \geq \mathsf{const}_1 \cdot \epsilon(n) \cdot 2^{n/7}, \tag{94}$$

which implies that there exists a constant const_2 such that

$$\max\{q(n),\eta(n)\} \geq \mathsf{const}_2 \cdot \epsilon(n)^3 \cdot 2^{n/42} \tag{95}$$

for infinitely many n. Therefore the claim also holds in this case.

Case 3: The Event $\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ Occurs. Here we consider the case that $\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ occurs. That is, we consider the case that

$$\Pr_{\substack{g_n,f_n,\Pi_n\\ y,\mathsf{td}\leftarrow\{0,1\}^n}}\Big[\mathsf{pk}\leftarrow g_n(\mathsf{td}), x\leftarrow \mathcal{A}_n^{g_n,f_n,f_n^{\mathrm{inv}},\mathsf{ColFinder}^{g,f,f^{\mathrm{inv}}}_{\lambda,\Pi,n}}(\mathsf{pk},y) : f_n(\mathsf{pk},x) = y \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\Big] \geq \frac{\epsilon(n)}{3} \tag{96}$$

holds for infinitely many n. In this case, for each n such that inequality (96) holds, there exist an n-bit string $\mathsf{td}_0 \in \{0,1\}^n$, a permutation $\hat{g}_n \in \mathsf{Perm}(\{0,1\}^n)$, and a family of permutations $\{\hat{f}(\mathsf{pk}, \cdot)\}_{\mathsf{pk} \neq \mathsf{pk}_0}$ such that

$$\Pr_{\substack{\hat{f}_n(\mathsf{pk}_0,\cdot),\Pi_n \\ y \leftarrow \{0,1\}^n}} \Big[\mathsf{pk}_0 \leftarrow \hat{g}_n(\mathsf{td}_0), x \leftarrow \mathcal{A}_n^{\hat{g}_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(\mathsf{pk}_0, y) : \\ \hat{f}_n(\mathsf{pk}_0, x) = y \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2) \Big] \geq \frac{\epsilon(n)}{3}, \quad (97)$$

Here we can construct a randomized compressing scheme (E, D) that compresses the truth table of $\hat{f}(\mathsf{pk}_0, \cdot)$, and can show that

$$\max\{q(n), \eta(n)\} \geq \mathsf{const} \cdot \epsilon(n)^3 \cdot 2^{n/7} \quad (98)$$

for infinitely many n, which implies that the claim also holds in this case.

The compressing scheme is an analogue of that in Section 4. Below we describe only the difference between the randomized compressing scheme here and that in Section 4. See Appendix A for a complete proof.

Difference from the Proof in Section 4. The constructions of E and D are almost the same as that of Section 4, except that in this section D uses the dummy oracle that always returns $\perp$ to simulate the oracle $\hat{f}^{\mathrm{inv}}(\mathsf{td}_0, \cdot)$.

The main difference from the proof in Section 4 is that, roughly speaking, we take X (the domain of encoder E) and G (subset of $\{0,1\}^n$ on which E "forgets" values of permutation $f \in X$) in such a way that, for any $f = \hat{f}(\mathsf{pk}_0, \cdot) \in X$ and $x \in G$, (i) $\hat{\mathcal{A}}$ inverts $f(x)$ in f with probability at least $2/3$ and *(ii) the event* $\neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)$ *always occurs with respect to* $\hat{\mathcal{A}}$*,* $y = f(x)$*, and* $f = \hat{f}(\mathsf{pk}_0, \cdot)$. We use $\epsilon(n)/6$ and $\epsilon(n)/12$, which may not be constants, instead of constants p_1 and p_2 so that the condition (ii) will hold. Hence we have to change Lemma 5.

Accordingly, the statement of Lemma 7 and Lemma 8 will be slightly changed: In Lemma 7, it is claimed that CalC_y satisfies some suitable properties for good circuits, but in this section CalC_y satisfies the corresponding properties for good *and non-trapdoor-hitting* circuits. For Lemma 8, the statement will not be changed in this section, but we will make full use of the condition (ii) above in the proof.

Moreover, since we use $\epsilon(n)/6$ and $\epsilon(n)/12$ instead of constants p_1 and p_2, the factor $\epsilon(n)^3$, instead of $\epsilon(n)$, appears in the final bound (98).

6 Concluding Remarks

In this paper we studied black-box impossibility in the quantum setting. We first formalized a quantum counterpart of the classical fully-black-box reduction [RTV04], and then proved that there is no quantum fully-black box reduction from collision-resistant hash functions to one-way permutations, or even trapdoor permutations. Our result is an extension to the quantum setting of the work of Simon [Sim98] who showed a similar result in the classical setting. We used compressing arguments to show the impossibility results, which is based on the work by Nayebi et al. [NABT15] and extends the work by Asharov and Segev [AS15].

Future direction. Here, we give two possible future directions. The first is to strengthen the black-box separation for CRH from other cryptographic primitives. In the classical setting, Asharov and Segev [AS15] proved that there does not exist a black-box reduction from CRH to OWP (or TDP) and indistinguishability obfuscations (IO) [GGH+13].[15] Since IO and OWP implies many strong cryptographic primitives including functional encryption [GGH+13], witness encryption [GGSW13], deniable encryption [SW14] etc., their result means that it is difficult to construct CRH from these primitives. Though it would be nice if we obtain a similar result in the quantum setting, it is not clear how we can define IO and "black-box access" to it in the quantum setting. Thus we considered simpler cases to separate CRH from OWP (or TDP) as a first step. We leave it as an interesting open problem to extend our result to separate CRH from OWP (or TDP) and IO.

The second is to give quantum analogues of black-box impossibility results shown in the classical setting. As seen in Section 1.4, there are many known black-box impossibility results shown in the classical setting. However, we observe that many of them crucially relies on the fact that all algorithms are classical, and it seems not easy to extend them to ones in the quantum setting. Especially, a theoretically important question is if we can rule out a quantum black-box reduction from classical-communication key-exchanges to OWP (or OWF) in the quantum setting. (If quantum communications are allowed, then the protocol in [BB84] is unconditionally secure. Therefore we only consider the case of classical-communication for making the question meaningful.) We note that this can be done if we prove that there does not exist a classical-communication key-exchange protocol (with super-polynomial security) in the quantum random oracle model (QROM). In the classical setting, a similar statement was proven by Impagliazzo and Rudich [IR89], followed by Barak and Mahmoody [BMG09] who gave the optimal security bound. On the other hand, in the quantum setting, we do not know any non-trivial security bound. We note that though Brassard et al. [BHK+11] gave a classical-communication key-exchange protocol in the QROM that is secure against adversary making $q^{5/3}$ queries to the random oracle where q is the number of queries by honest parties, they did not show their protocol is optimal in regard to security.

[15] Since a certain type of non-black-box construction is inherent in many IO-based constructions, they actually also ruled out reductions using "commonly used" non-black-box techniques.

References

Aar09. Scott Aaronson. Quantum copy-protection and quantum money. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 229–242, 2009.

ABF$^+$16. Gorjan Alagic, Anne Broadbent, Bill Fefferman, Tommaso Gagliardoni, Christian Schaffner, and Michael St. Jules. Computational security of quantum encryption. In Anderson C. A. Nascimento and Paulo Barreto, editors, *ICITS 16*, volume 10015 of *LNCS*, pages 47–71. Springer, Heidelberg, August 2016.

AC12. Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 41–60. ACM Press, May 2012.

AGM18. Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 489–519. Springer, Heidelberg, April / May 2018.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

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

AS15. Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In Venkatesan Guruswami, editor, *56th FOCS*, pages 191–209. IEEE Computer Society Press, October 2015.

BB84. Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, India, 1984.

BBBV97. Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

BBCS92. Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 351–366. Springer, Heidelberg, August 1992.

BBF13. Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 296–315. Springer, Heidelberg, December 2013.

BBHT98. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

BD19. Nir Bitansky and Akshay Degwekar. On the complexity of collision resistant hash functions: New and old black-box separations. *To appear at TCC 2019*, 2019.

BHK$^+$11. Gilles Brassard, Peter Høyer, Kassem Kalach, Marc Kaplan, Sophie Laplante, and Louis Salvail. Merkle puzzles in a quantum world. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 391–410. Springer, Heidelberg, August 2011.

BHT98. Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *Latin American Symposium on Theoretical Informatics*, pages 163–169. Springer, 1998.

BJ15. Anne Broadbent and Stacey Jeffery. Quantum homomorphic encryption for circuits of low T-gate complexity. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 609–629. Springer, Heidelberg, August 2015.

BL17. Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549:188–194, 2017.

BLP$^+$13. Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 575–584. ACM Press, June 2013.

BMG09. Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 374–390. Springer, Heidelberg, August 2009.

Bra18. Zvika Brakerski. Quantum FHE (almost) as secure as classical. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 67–95. Springer, Heidelberg, August 2018.

BS16. Anne Broadbent and Christian Schaffner. Quantum cryptography beyond quantum key distribution. *Des. Codes Cryptography*, 78(1):351–382, 2016.

BV98. Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 59–71. Springer, Heidelberg, May / June 1998.

CHS18. Nai-Hui Chia, Sean Hallgren, and Fang Song. On basing one-way permutations on NP-hard problems under quantum reductions. *CoRR*, abs/1804.10309, 2018.

CLMP13. Kai-Min Chung, Huijia Lin, Mohammad Mahmoody, and Rafael Pass. On the power of nonuniformity in proofs of security. In Robert D. Kleinberg, editor, *ITCS 2013*, pages 389–400. ACM, January 2013.

Cor02. Jean-Sébastien Coron. Security proof for partial-domain hash signature schemes. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 613–626. Springer, Heidelberg, August 2002.

DFG13. Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The Fiat-Shamir transformation in a quantum world. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part II*, volume 8270 of *LNCS*, pages 62–81. Springer, Heidelberg, December 2013.

DFLS16. Frédéric Dupuis, Serge Fehr, Philippe Lamontagne, and Louis Salvail. Adaptive versus non-adaptive strategies in the quantum setting with applications. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 33–59. Springer, Heidelberg, August 2016.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 356–383, 2019.

DOP05. Yevgeniy Dodis, Roberto Oliveira, and Krzysztof Pietrzak. On the generic insecurity of the full domain hash. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 449–466. Springer, Heidelberg, August 2005.

DTT10. Anindya De, Luca Trevisan, and Madhur Tulsiani. Time space tradeoffs for attacks against one-way functions and PRGs. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 649–665. Springer, Heidelberg, August 2010.

Fis12. Marc Fischlin. Black-box reductions and separations in cryptography. In *Progress in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings*, pages 413–422, 2012.

FKS$^+$13. Serge Fehr, Jonathan Katz, Fang Song, Hong-Sheng Zhou, and Vassilis Zikas. Feasibility and completeness of cryptographic tasks in the quantum world. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 281–296. Springer, Heidelberg, March 2013.

FLR$^+$10. Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 303–320. Springer, Heidelberg, December 2010.

FS12. Dario Fiore and Dominique Schröder. Uniqueness is a different story: Impossibility of verifiable random functions from trapdoor permutations. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 636–653. Springer, Heidelberg, March 2012.

GC01. Daniel Gottesman and Isaac Chuang. Quantum digital signatures. *CoRR*, abs/quant-ph/0105032, 2001.

GGH$^+$13. Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *54th FOCS*, pages 40–49. IEEE Computer Society Press, October 2013.

GGSW13. Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. Witness encryption and its applications. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *45th ACM STOC*, pages 467–476. ACM Press, June 2013.

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*, pages 212–219. ACM, 1996.

GT00. Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st FOCS*, pages 305–313. IEEE Computer Society Press, November 2000.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011.

HHRS07. Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - a tight lower bound on the round complexity of statistically-hiding commitments. In *48th FOCS*, pages 669–679. IEEE Computer Society Press, October 2007.

HL18. Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 850–858, 2018.

Hof11. Dennis Hofheinz. Possibility and impossibility results for selective decommitments. *Journal of Cryptology*, 24(3):470–516, July 2011.

HR04. Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 92–105. Springer, Heidelberg, August 2004.

HXY19. Minki Hhan, Keita Xagawa, and Takashi Yamakawa. Quantum random oracle model with auxiliary input. *To appear at Asiacrypt 2019*, 2019.

IR89. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

JF11. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, pages 19–34, 2011.

KSVV02. Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

LZ19. Qipeng Liu and Mark Zhandry. Revisiting post-quantum Fiat-Shamir. In *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, pages 326–355, 2019.

Mah18. Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 332–338, 2018.

McE78. Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 44:114–116, 1978.

NABT15. Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. Quantum lower bound for inverting a permutation with advice. *Quantum Information & Computation*, 15(11&12):901–913, 2015.

NC10. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

NIS16. NIST. Post-quantum cryptography standardization. 2016. See `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`.

Pas11. Rafael Pass. Limits of provable security from standard assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd ACM STOC*, pages 109–118. ACM Press, June 2011.

Pei09. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 333–342. ACM Press, May / June 2009.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

RS18. Lior Rotem and Gil Segev. Injective trapdoor functions via derandomization: How strong is rudich's black-box barrier? In Amos Beimel and Stefan Dziembowski, editors, *TCC 2018*, volume 11239 of *LNCS*, pages 421–447. Springer, Heidelberg, November 2018.

RTV04. Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 1–20. Springer, Heidelberg, February 2004.

Rud88. Steven Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, University of California, Berkeley, 1988.

Rud92. Steven Rudich. The use of interaction in public cryptosystems (extended abstract). In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 242–251. Springer, Heidelberg, August 1992.

Sho94. Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134. IEEE Computer Society Press, November 1994.

Sim98. Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 334–345. Springer, Heidelberg, May / June 1998.

Son14. Fang Song. A note on quantum security for post-quantum cryptography. In *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings*, pages 246–265, 2014.

SW14. Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In David B. Shmoys, editor, *46th ACM STOC*, pages 475–484. ACM Press, May / June 2014.

Unr16. Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 497–527. Springer, Heidelberg, May 2016.

Vaz98. Umesh Vazirani. On the power of quantum computation. *PHILOSOPHICAL TRANSACTIONS-ROYAL SOCIETY OF LONDON SERIES A MATHEMATICAL PHYSICAL AND ENGINEERING SCIENCES*, pages 1759–1767, 1998.

Wie83. Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

Yao93. Andrew Chi-Chih Yao. Quantum circuit complexity. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 352–361, 1993.

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

Zha19. Mark Zhandry. Quantum lightning never strikes the same state twice. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, pages 408–438, 2019.

A A Complete Proof for the Case 3 of Proposition 2.

The goal of this section is to show the following proposition.

Proposition 3. *Suppose that, for infinitely many n, there exist an n-bit string $\mathsf{td}_0 \in \{0,1\}^n$, a permutation $\hat{g}_n \in \mathsf{Perm}(\{0,1\}^n)$, and a family of permutations*

$\{\hat{f}_n(\mathsf{pk}, \cdot)\}_{\mathsf{pk} \neq \mathsf{pk}_0}$ *such that*

$$\Pr_{\substack{\hat{f}_n(\mathsf{pk}_0, \cdot), \Pi_n \\ y \leftarrow \{0,1\}^n}} \left[\mathsf{pk}_0 \leftarrow \hat{g}_n(\mathsf{td}_0), x \leftarrow \mathcal{A}_n^{\hat{g}_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda, \Pi, n}^{\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}}}(\mathsf{pk}_0, y) : \hat{f}_n(\mathsf{pk}_0, x) = y \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2) \right] \geq \frac{\epsilon(n)}{3}, \quad (99)$$

holds. Then there exists a constant const *such that*

$$\max\{q(n), \eta(n)\} \geq \mathsf{const} \cdot \epsilon(n)^3 \cdot 2^{n/7} \quad (100)$$

holds for infinitely many n.

Preparations. Here we describe some technical preparations before using the encoding technique. Without loss of generality we can assume $q(n), \eta(n), \lambda(n) \geq 1$ holds, since increasing these numbers does not decrease the ability of $\mathcal{A}$ to invert $\hat{f}$. In a similar way as we did in Section 4, we construct another algorithm $\hat{\mathcal{A}}$ that iteratively runs $\mathcal{A}$ to increase the success probability, and then apply the encoding technique to $\hat{\mathcal{A}}$.

Remember that c is a sufficiently large positive integer in Section 5. Let $\mathcal{B}_c$ be an oracle-aided quantum algorithm that runs as follows, relative to the oracles $\hat{g}$, $\hat{f}$, $\hat{f}^{\mathrm{inv}}$, $\mathsf{ColFinder}_{\lambda}^{\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}}$.

1. Take an input y. Set $\mathsf{guess} \leftarrow \perp$.
2. For $i = 1, \ldots, c\lceil 12/\epsilon(n) \rceil$ do:
3. Run $\mathcal{A}^{\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda}^{\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}}}$ on the input (pk_0, y). Let x be the output.
4. Query (pk_0, x) to $\hat{f}$. If $\hat{f}(\mathsf{pk}_0, x) = y$, then set $\mathsf{guess} \leftarrow x$.
5. End For
6. Return guess.

Remember that $Q(n)$ is defined as $c\lceil 12/\epsilon(n) \rceil(\max\{q(n), \eta(n)\} + 1)$ in Section 5. $\mathcal{B}_c$ can be regarded as a Q-query algorithm, and for each quantum circuit C that $\mathcal{B}_c$ queries to $\mathsf{ColFinder}_{\lambda, n}^{\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}}$, C makes at most $Q(n)$ queries.

Lemma 10 that will be shown below corresponds to Lemma 5 in Section 4. The main difference between Lemma 10 and Lemma 5 is that Lemma 10 uses $\epsilon(n)/6$ and $\epsilon(n)/12$, which may not be constants, instead of constants p_1 and p_2, respectively. We use $\epsilon(n)/6$ and $\epsilon(n)/12$ so that, for $x \in G$ (G is the set we will use in our encoder and decoder) and $f = \hat{f}(\mathsf{pk}_0, \cdot) \in X$, $\mathcal{B}_c$ will invert $y = f(x) = \hat{f}(\mathsf{pk}_0, x)$ in $f = \hat{f}(\mathsf{pk}_0, \cdot)$ *and* the event $\neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)$ occurs with respect to $\mathcal{B}_c$, y, and f. Here, TDHIT'_1 and TDHIT'_2 are the events defined as follows.

Definition of the events TDHIT'_1 *and* TDHIT'_2. For each n, we define TDHIT'_1 as the event that

$$\sum_z \mu_{(\mathsf{td}, z)}^{\mathcal{B}_c, f^{\mathrm{inv}}}(\mathsf{pk}, y) > \frac{\delta}{Q(n)} \quad (101)$$

occurs. In addition, for each n, we define TDHIT'_2 as the event that

$$\sum_{C\text{:trapdoor-hitting}} \mu_C^{\mathcal{B}_c, \mathsf{ColFinder}_\lambda^{g,f,f^{\mathrm{inv}}}}(\mathsf{pk}, y) > \frac{\delta}{Q(n)} \tag{102}$$

occurs. Note that, in the definitions of TDHIT_1 and TDHIT_2, we used $\tilde{Q}(n)$ instead of $Q(n)$. We need not only TDHIT_1 and TDHIT_2 but also TDHIT'_1 and TDHIT'_2 since $\mathcal{B}_c$ makes more queries than $\mathcal{A}$, and thus the query magnitudes of $\mathcal{B}_c$ is larger than those of $\mathcal{A}$. (See (74) and (75) for the definitions of TDHIT_1 and TDHIT_2.)

Lemma 10. *For a sufficiently large positive integer c, the following condition is satisfied for infinitely many n:*
Condition. *There exist $X \subset \mathsf{Perm}(\{0,1\}^n)$ and $\mathit{\Pi}_n$ such that $|X| \geq \frac{\epsilon(n)}{6} \cdot |\mathsf{Perm}(\{0,1\}^n)|$ and*

$$\Pr_{y \leftarrow \{0,1\}^n}\left[\Pr\left[x \leftarrow \mathcal{B}_{c,n}^{\hat{g}_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0, x) = y\right] \geq 2/3 \right.$$
$$\left. \wedge \neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)\right] \geq \frac{\epsilon(n)}{12} \tag{103}$$

for all $\hat{f}_n(\mathsf{pk}_0, \cdot) \in X$. (Note that whether or not the event $\neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)$ occurs is determined once y, $\hat{f}_n(\mathsf{pk}_0, \cdot)$ $\hat{g}$, $\{\hat{f}_n(z, \cdot)\}_{z \neq \mathsf{pk}_0}$, td_0, pk_0, and $\mathit{\Pi}_n$ are all fixed.)

Proof. Let c be an integer that satisfies $e^{-c} \leq 1/3$. In what follows, we show that this c satisfies the condition.

First, for each n such that

$$\Pr_{\substack{\hat{f}_n(\mathsf{pk}_0,\cdot), \mathit{\Pi}_n \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0, x) = y \right.$$
$$\left. \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\right] \geq \frac{\epsilon(n)}{3} \tag{104}$$

holds, there exists $\mathit{\Pi}_n$ such that

$$\Pr_{\substack{\hat{f}_n(\mathsf{pk}_0,\cdot), \\ y \leftarrow \{0,1\}^n}}\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n, \hat{f}_n, \hat{f}_n^{\mathrm{inv}}, \mathsf{ColFinder}_{\lambda,\mathit{\Pi},n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0, x) = y \right.$$
$$\left. \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\right] \geq \frac{\epsilon(n)}{3} \tag{105}$$

holds. Below we fix $\mathit{\Pi}_n$ that satisfies inequality (105) for each n such that inequality (104) holds.

Now we have that

$$\Pr_{\hat{f}_n(\mathsf{pk}_0,\cdot)}\left[\Pr_{y\leftarrow\{0,1\}^n}\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0,x)=y \right.\right.$$
$$\left.\left. \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\right] \geq \frac{\epsilon(n)}{6}\right] \geq \frac{\epsilon(n)}{6} \qquad (106)$$

from inequality (105). In other words, there exists $X \subset \mathsf{Perm}(\{0,1\}^n)$ such that $|X|$ is lower bounded by $\frac{\epsilon(n)}{6}|\mathsf{Perm}(\{0,1\}^n)|$ and

$$\Pr_{y\leftarrow\{0,1\}^n}\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0,x)=y \right.$$
$$\left. \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\right] \geq \frac{\epsilon(n)}{6} \qquad (107)$$

holds for all $\hat{f}_n(\mathsf{pk}_0,\cdot) \in X$. Hence, for each $\hat{f}_n(\mathsf{pk}_0,\cdot) \in X$, from inequality (107) it follows that

$$\Pr_{y\leftarrow\{0,1\}^n}\left[\Pr\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0,x)=y \right.\right.$$
$$\left.\left.\right] \geq \frac{\epsilon(n)}{12} \wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2)\right] \geq \frac{\epsilon(n)}{12} \qquad (108)$$

For each pair $(f(\mathsf{pk}_0,\cdot),y) \in X \times \{0,1\}^n$ such that

$$\Pr\left[x \leftarrow \mathcal{A}_n^{\hat{g}_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0,x)=y\right] \geq \frac{\epsilon(n)}{12}$$
$$\wedge \neg(\mathsf{TDHIT}_1 \vee \mathsf{TDHIT}_2), \qquad (109)$$

we have that

$$\Pr\left[x \leftarrow \mathcal{B}_{c,n}^{\hat{g}_n,\hat{f}_n,\hat{f}_n^{\mathrm{inv}},\mathsf{ColFinder}_{\lambda,\Pi,n}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(y) : \hat{f}_n(\mathsf{pk}_0,x)=y\right] \quad \geq 1-\left(1-\frac{\epsilon(n)}{12}\right)^{\frac{12c}{\epsilon(n)}}$$
$$= 1-\left(\left(1-\frac{\epsilon(n)}{12}\right)^{-\frac{1}{\frac{\epsilon(n)}{12}}}\right)^{-c}. \qquad (110)$$

The right hand side of inequality (110) is equal to 1 if $\epsilon(n)=1$, and lower bounded by $1-e^{-c} \geq \frac{2}{3}$ if $\epsilon(n)<1$ (here we used the fact that $(1-x)^{-\frac{1}{x}} \geq e$ holds for $0<x<1$). In addition, for each pair $(f(\mathsf{pk}_0,\cdot),y) \in X \times \{0,1\}^n$ such that (109) holds, the event $\neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)$ occurs with respect to $\mathcal{B}_c$ by definition of the events TDHIT_1, TDHIT_2, TDHIT'_1, and TDHIT'_2 since $\mathcal{B}_c$ iteratively runs $\mathcal{A}$ just $c\lceil 12/\epsilon(n)\rceil$ times, and $\tilde{Q}(n)=c\lceil 12/\epsilon(n)\rceil Q(n)$ holds. Therefore the claim holds. $\square$

Then, from the above lemma, it follows that there exists a constant c that satisfies the condition in Lemma 10 for infinitely many n. Let us denote $\mathcal{B}_c$ by $\hat{\mathcal{A}}$. We use the encoding technique to this Q-query algorithm $\hat{\mathcal{A}}$, here $Q(n) = c\lceil 12/\epsilon(n)\rceil(\max\{q(n), \eta(n)\} + 1)$. Below we fix a sufficiently large n in addition to Π_n and X such that the condition in Lemma 10 is satisfied. For simplicity, we write Q, ϵ, $\hat{g}$, $\hat{f}$, $\hat{f}^{\mathrm{inv}}$, and $\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}$ instead of $Q(n)$, $\epsilon(n)$, $\hat{g}_n$, $\hat{f}_n$, $\hat{f}^{\mathrm{inv}}_n$, and $\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}_{\lambda,\Pi,n}$, respectively, for simplicity. Moreover, sometimes we write f instead of $\hat{f}(\mathsf{pk}_0,\cdot)$.

Below we describe an encoder E and a decoder D that compress elements (truth tables of permutations) in X. The encoder in this section has to deal with more oracles than the encoder in Section 4 does, but there is no essential difference between them. The decoder in this section has to simulate the oracle $\hat{f}^{\mathrm{inv}}(\mathsf{td}_0,\cdot) = (\hat{f}(\mathsf{pk}_0,\cdot))^{-1}$ since $\hat{\mathcal{A}}$ may make queries to it. However, $\hat{f}(\mathsf{pk}_0,\cdot)$ itself is the permutation that our decoder want to invert. Thus we use the dummy oracle that returns $\perp$ for any input instead of $f^{\mathrm{inv}}(\mathsf{td}_0,\cdot)$. Since the sets X and G will be constructed in such a way that the event $\neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)$ occurs with respect to $\hat{\mathcal{A}}$, $f = \hat{f}(\mathsf{pk}_0,\cdot) \in X$, and $y \in G$, $\hat{\mathcal{A}}$ will not be able to distinguish the dummy oracle and $\hat{f}^{\mathrm{inv}}(\mathsf{td}_0,\cdot)$.

Encoder *E*. When we feed E with $f = \hat{f}(\mathsf{pk}_0,\cdot) \in X$ as an input, E first chooses subsets $R, R' \subset \{0,1\}^n$ by the following sampling: For each $x \in \{0,1\}^n$, x is added to R with probability $\delta^{3/2}/Q^2$, and independently added to R' with probability $\delta^{5/2}/Q^4$. (The pair (R, R') is the random coin of E.)

According to the choice of R', "bad" inputs (oracle-aided quantum circuits) to $\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}$ are defined for each $x \in \{0,1\}^n$ as follows. Note that now $\pi^{(1)}_C$ and $\pi^{(2)}_C$ have been fixed for each C, and the output $\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(C) = (w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, F^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}_C(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}))$ is uniquely determined. For each oracle-aided quantum circuit C such that $F^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}_C(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) \neq \perp$, we can define query magnitude of C to $f = \hat{f}(\mathsf{pk}_0,\cdot)$ on input $w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}$ and $w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}$ at $z \in \{0,1\}^n$ (see Definition 5). We say a quantum circuit C such that $F^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}_C(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) \neq \perp$ is *bad* relative to x if

$$\sum_{z \in R' \setminus \{x\}} \mu_z^{C,\hat{f}(\mathsf{pk}_0,\cdot)}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) > \frac{\delta}{Q} \tag{111}$$

or

$$\sum_{z \in R' \setminus \{x\}} \mu_z^{C,\hat{f}(\mathsf{pk}_0,\cdot)}(w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) > \frac{\delta}{Q} \tag{112}$$

hold, and otherwise we say C is *good* relative to x. For quantum circuits C such that $F^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}_C(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = \perp$, we always say that C is *good*. Let $\mathsf{badC}(R', x)$ denote the set of bad circuits relative to x for each $R' \subset \{0,1\}^n$.

Next, E construct a set $G \subset \{0,1\}^n$ depending on the input $f = \hat{f}(\mathsf{pk}_0, \cdot)$. Let $I \subset \{0,1\}^n$ be the set of elements x such that $\hat{\mathcal{A}}$ successfully inverts $f(x) = \hat{f}(\mathsf{pk}_0, x)$, i.e., $I := \{x \mid \Pr[x' \leftarrow \hat{\mathcal{A}}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}},\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(\hat{f}(\mathsf{pk}_0,x)) : x' = x] \geq 2/3$. Then $|I| \geq \frac{\epsilon}{12} \cdot 2^n$ holds by definition of X (Remember that X is chosen in such a way as to satisfy the condition in Lemma 10). Now, a set G is defined to be the set of elements $x \in I$ that satisfies the following conditions:

Conditions for G.

(Cond. 1) $x \in R \cap R'$.
(Cond. 2) $\sum_{z \in R \setminus \{x\}} \mu_z^{\hat{\mathcal{A}},\hat{f}(\mathsf{pk}_0,\cdot)}(\hat{f}(\mathsf{pk}_0, x)) \leq \delta/Q$.
(Cond. 3) $\sum_{C \in \mathsf{badC}(R',x)} \mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(\hat{f}(\mathsf{pk}_0, x)) \leq \delta/Q$.

Finally, E encodes $f = \hat{f}(\mathsf{pk}_0, \cdot)$ into $(f|_{\{0,1\}^n \setminus G}, f(G))$ if $|G| \geq \theta$, where $\theta = (1 - 60\sqrt{\delta})\delta^4 \cdot (\frac{\epsilon}{12}) \cdot 2^n / 2Q^6$. Otherwise E encodes $f = \hat{f}(\mathsf{pk}_0, \cdot)$ into $\perp$.

In addition, here we formally define the set Y (the range of E) as

$$Y := \left\{ (f|_{\{0,1\}^n \setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| \geq \theta \right\}. \quad (113)$$

In fact $E((R,R'), f) \in Y \cup \{\perp\}$ holds for any choice of (R, R') and any permutation $f \in X$.

Decoder D. D takes $(\tilde{f}, \tilde{G})$ as input in addition to (R, R'), where $\tilde{G} \subset \{0,1\}^n$ and $\tilde{f}$ is a bijection from a subset of $\{0,1\}^n$ onto $\{0,1\}^n \setminus \tilde{G}$, and R, R' are subsets of $\{0,1\}^n$. If $\{0,1\}^n \setminus$(the domain of $\tilde{f}$) $\not\subset R \cap R'$ holds, then D outputs $\perp$. Otherwise, D decodes $(\tilde{f}, \tilde{G})$ and reconstruct the truth table of a permutation $f = \hat{f}(\mathsf{pk}_0, \cdot) \in \mathsf{Perm}(\{0,1\}^n)$ as follows.

For each x in the domain of $\tilde{f}$, D infers the value $f(x) = \hat{f}(\mathsf{pk}_0, x)$ as $f(x) := \tilde{f}(x)$. For other elements $x \in \{0,1\}^n$ which is not contained in the domain of $\tilde{f}$, what D now knows is only that $f(x)$ is contained in $\tilde{G}$. To determine the remaining part of the truth table of $f = \hat{f}(\mathsf{pk}_0, \cdot)$, D tries to recover the value $f^{-1}(y)$, which is equal to $(\hat{f}(\mathsf{pk}_0, \cdot))^{-1}(y) = \hat{f}^{\mathrm{inv}}(\mathsf{td}_0, y)$, for each $y \in \tilde{G}$ by using $\hat{\mathcal{A}}$ and without the oracle $\hat{f}^{\mathrm{inv}}(\mathsf{td}_0, \cdot)$.

In a similar way as we did in Section 4, D prepares oracles h_y and SimCF^{h_y} which approximates $f(\mathsf{pk}_0, \cdot)$ and $\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}$, respectively, and computes the output distribution of $\hat{\mathcal{A}}^{\hat{g},(h_y,\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(\perp,\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0}),\mathsf{SimCF}^{h_y}}$ on input y. Here, the pair $(h_y, \hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0})$ is the oracle that returns $h_y(x)$ on input (pk_0, x), and returns $\hat{f}(z, x)$ on input (z, x) such that $z \neq \mathsf{pk}_0$. $(\perp, \hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})$ is the oracle that returns $\perp$ on input (td_0, x) and returns $\hat{f}^{\mathrm{inv}}(z, x)$ on input (z, x) such that $z \neq \mathsf{td}_0$.

SimCF^{h_y} uses a subroutine CalC_y that takes (C, w) as an input (C is an oracle-aided circuit that may make queries to $\hat{g}, \hat{f}, \hat{f}^{\mathrm{inv}}$ and computes a function $F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}$, and w is an element of the domain of $F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}$) and simulates the evaluation of $F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w)$. D finally infers that $f^{-1}(y)$, which is equal to

$(f(\mathsf{pk}_0,\cdot))^{-1}(y) = f^{\mathrm{inv}}(\mathsf{td}_0, y)$, is the element which $\hat{\mathcal{A}}^{\hat{g},(h_y,\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(\perp,\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0}),\mathsf{SimCF}^{h_y}}$ outputs with probability greater than 1/2. (If there does not exist such an element, then D outputs $\perp$.) Below we describe h_y, CalC_y, and SimCF^{h_y}.

Oracle h_y. The oracle (function) $h_y : \{0,1\}^n \to \{0,1\}^n$ is defined by

$$h_y(z) = \begin{cases} \tilde{f}(z) \text{ if } z \notin R \cap R', \\ y \text{ otherwise.} \end{cases} \tag{114}$$

Subroutine CalC_y. Let $P_{\mathrm{candidate}} := \{h' \in \mathsf{Perm}(\{0,1\}^n)) \mid \Delta(h', h_y) \subset R \cap R'\}$. CalC_y is defined as the following procedures. For $h' \in P_{\mathrm{candidate}}$, let $({h'}^{-1}, \hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})$ denote the oracle that returns ${h'}^{-1}(x)$ on input (td_0, x) and returns $\hat{f}^{\mathrm{inv}}(z, x)$ on input (z, x) such that $z \neq \mathsf{td}_0$.

1. Take an input (C, w), where C is an oracle-aided circuit and w is an element of the domain of the function F_C.
2. Compute the output distribution of the quantum circuit $C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),({h'}^{-1},\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})}$ on input w for each $h' \in P_{\mathrm{candidate}}$, and find the corresponding output $u(C, w, h')$ such that $\Pr\left[C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),({h'}^{-1},\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})}(w) = u(C, w, h')\right] > 1/2$. If there are no such $u(C, w, h')$ for a fixed h', set $u(C, w, h') :=\perp$.
3. If $u(C, w, h') = u(C, w, h'') \neq\perp$ for all $h', h'' \in P_{\mathrm{candidate}}$, return the value $u(C, w, h')$. Otherwise return $\perp$.

Oracle SimCF^{h_y}. SimCF^{h_y} is defined as the following procedures:

1. Take an input C, where C is an oracle-aided quantum circuit.
2. Compute $\tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} := \pi^{(1)}_C(0^m)$.
3. If $\mathsf{CalC}_y(C, \tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) =\perp$, set $\tilde{w}^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} :=\perp$.
4. Otherwise, search the minimum $t \in \{0,1\}^m$ such that $\mathsf{CalC}_y(C, \tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = \mathsf{CalC}_y(C, \pi^{(2)}_C(t))$ by checking whether $\mathsf{CalC}_y(C, \tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = \mathsf{CalC}_y(C, \pi^{(2)}_C(i))$ holds for $i = 0, 1, 2, \ldots$ in a sequential order. If the minimum number t is found, set $\tilde{w}^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} := \pi^{(2)}_C(t)$. Otherwise set $\tilde{w}^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} :=\perp$.
5. Return $(\tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, \tilde{w}^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, \mathsf{CalC}_y(C, \tilde{w}^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}))$.

Note that D is an information theoretic decoder, and we do not care whether CalC_y and SimCF^{h_y} run efficiently.

Analyses. The following lemma, which corresponds to Lemma 7 in Section 4, shows that h_y, CalC_y, and SimCF^{h_y} satisfy some suitable properties. Here we consider the situation that D takes an input $(\tilde{f}, \tilde{G})$ such that $(\tilde{f}, \tilde{G}) = E((R, R'), f)$ for some subsets $R, R' \subset \{0,1\}^n$ and a permutation $f = \hat{f}(\mathsf{pk}_0, \cdot) \in \{0,1\}^n$, and tries to recover the value $f^{-1}(y)$ for some $y \in \tilde{G}$.

In Lemma 7, some suitable properties are satisfied for good circuits. On the other hand, in Lemma 11, to satisfy the corresponding suitable properties, a circuit have to be good *and* non-trapdoor-hitting (see (72) for the definition of non-trapdoor-hitting circuits). This is the main difference between Lemma 7 and Lemma 11.

Lemma 11. *h_y, CalC_y, and SimCF^{h_y} satisfy the following properties.*

1. *$\Delta(h_y, f) = R \cap R' \setminus \{f^{-1}(y)\}$ holds.*
2. *$\mathsf{CalC}_y(C, w) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w)$ or $\perp$ holds for any C and w.*
3. *For each non-trapdoor-hitting circuit C which is good relative to $f^{-1}(y)$ and satisfies $F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(\, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) \neq \perp$, it holds that $\mathsf{CalC}_y(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})$ and $\mathsf{CalC}_y(C, w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})$. In addition, for each circuit C such that $F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = \perp$ holds, we have that $\mathsf{CalC}_y(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = \perp$ holds.*
4. *$\mathsf{SimCF}^{h_y}(C) = \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(C)$ holds for each circuit C which is good relative to $f^{-1}(y)$ and non-trapdoor-hitting. In particular,*

$$\Delta(\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}, \mathsf{SimCF}^{h_y}) \subset \mathsf{badC}(R', f^{-1}(y)) \cup \mathsf{hitC}$$

holds, where hitC is the set of trapdoor-hitting circuits.

Proof. The first property is obviously satisfied by definition of h_y.

For the second property, since $f = \hat{f}(\mathsf{pk}_0, \cdot) \in P_{\mathrm{candidate}}$, if $\mathsf{CalC}_y(C, w) \neq \perp$ then we have $\mathsf{CalC}_y(C, w) = u(C, w, f)$ by definition of CalC_y, and $u(C, w, f) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w)$ always holds. Hence the second property holds.

For the third property, for each $h' \in P_{\mathrm{candidate}}$, from Lemma 2 we have

$$\begin{aligned}
&\Pr\left[C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(h'^{-1},\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})\right] \\
&\quad \geq \Pr\left[C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})\right] \\
&\qquad - \left\| C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}} |w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, 0, 0\rangle - C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(h'^{-1},\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})} |w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, 0, 0\rangle \right\|.
\end{aligned} \tag{115}$$

From the swapping lemma (Lemma 3) it follows that

$$\begin{aligned}
&\left\| C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}} |w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, 0, 0\rangle - C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(h'^{-1},\hat{f}^{\mathrm{inv}}_{\mathsf{td}\neq\mathsf{td}_0})} |w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}, 0, 0\rangle \right\| \\
&\quad \leq 2\sqrt{Q \sum_{z \in \Delta(f(\mathsf{pk}_0,\cdot),h')} \mu_z^{C,\hat{f}(\mathsf{pk}_0,\cdot)}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})} \\
&\qquad + 2\sqrt{Q \sum_{z \in \{0,1\}^n} \mu_z^{C,\hat{f}^{\mathrm{inv}}(\mathsf{td}_0,\cdot)}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}})}.
\end{aligned} \tag{116}$$

Since $\varDelta(f(\mathsf{pk}_0,\cdot),h') = \varDelta(f,h') \subset R \cap R' \setminus \{f^{-1}(y)\} \subset R' \setminus \{f^{-1}(y)\}$ holds for all $h' \in P_{\text{candidate}}$, and C is good relative to $f^{-1}(y)$ and non-trapdoor-hitting, the right hand side of the above inequality is upper bounded by $2\sqrt{\delta} + 2\sqrt{\delta} = 4\sqrt{\delta}$. Thus, for a sufficiently small δ we have

$$\Pr\left[C^{\hat{g},(h',\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(h'^{-1},\hat{f}^{\text{inv}}_{\mathsf{td}\neq\mathsf{td}_0})}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}})\right] \geq \frac{2}{3} - 4\sqrt{\delta} > \frac{1}{2}, \tag{117}$$

which implies that $u(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}, h') = F_C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}})$ holds for every $h' \in P_{\text{candidate}}$. Thus $\mathsf{CalC}_y(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}) = F_C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}})$ holds if C is good relative to $f^{-1}(y)$ and non-trapdoor-hitting. It can be shown that the corresponding property also holds for $w^{(2)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}$ in the same way. In addition, for a circuit C such that $F_C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}) = \perp$, $\mathsf{CalC}_y(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}) = \perp$ holds since $u(C, w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}, f) = F_C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}(w^{(1)}_{C^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}) = \perp$ holds. Therefore the third property follows.

The fourth property follows from the definition of SimCF^{h_y}, the second property, and the third property. □

The following lemma shows that the decoding always succeeds if the encoding succeeds. In the proof below, we make full use of the condition that the sets X and G are constructed in such a way that the event $\neg(\mathsf{TDHIT}'_1 \vee \mathsf{TDHIT}'_2)$ occurs with respect to $\hat{\mathcal{A}}$, $f = \hat{f}(\mathsf{pk}_0,\cdot) \in X$, and $y \in G$.

Lemma 12. *If $E((R,R'),f) \neq \perp$, then $D((R,R'),E((R,R'),f)) = f$ holds for each $f = \hat{f}(\mathsf{pk}_0,\cdot) \in X$.*

Proof (of Lemma 8). Let $\tilde{f} := f|_{\{0,1\}^n \setminus G}$ and $\tilde{G} := f(G)$. We show that D can correctly recover $x = f^{-1}(y)$ for each $y \in \tilde{G}$.

By applying Lemma 3 (the swapping lemma) to $(\hat{g},\hat{f},\hat{f}^{\text{inv}},\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\text{inv}}})$ and $(\hat{g},(h_y,\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(\perp,\hat{f}^{\text{inv}}_{\mathsf{td}\neq\mathsf{td}_0}),\mathsf{SimCF}^{h_y})$, we obtain

$$\begin{aligned}
&\left\|\hat{\mathcal{A}}_n^{\hat{g},\hat{f},\hat{f}^{\text{inv}},\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}|f(x),0,0\rangle - \hat{\mathcal{A}}_n^{\hat{g},(h_y,\hat{f}_{\mathsf{pk}\neq\mathsf{pk}_0}),(\perp,\hat{f}^{\text{inv}}_{\mathsf{td}\neq\mathsf{td}_0}),\mathsf{SimCF}^{h_y}}|f(x),0,0\rangle\right\| \\
&\quad\leq 2\sqrt{Q\sum_{z\in\varDelta(\hat{f}(\mathsf{pk}_0,\cdot),h_y)}\mu_z^{\hat{\mathcal{A}},\hat{f}(\mathsf{pk}_0,\cdot)}(f(x))} + 2\sqrt{Q\sum_{z\in\{0,1\}^n}\mu_z^{\hat{\mathcal{A}},\hat{f}^{\text{inv}}(\mathsf{td}_0,\cdot)}(f(x))} \\
&\quad\quad + 2\sqrt{Q\sum_{C\in\varDelta(\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\text{inv}}},\mathsf{SimCF}^{h_y})}\mu_C^{\hat{\mathcal{A}},\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\text{inv}}}}(f(x))}.
\end{aligned} \tag{118}$$

Since $\varDelta(\hat{f}(\mathsf{pk}_0,\cdot),h_y) = \varDelta(f,h_y) = R \cap R' \setminus \{f^{-1}(y)\} \subset R \setminus \{f^{-1}(y)\} = R \setminus \{x\}$ hold, the first term of the right hand side of inequality (118) is upper bounded by

$$2\sqrt{Q \sum_{z \in R \setminus \{x\}} \mu_z^{\hat{\mathcal{A}}, \hat{f}(\mathsf{pk}_0, \cdot)}(f(x))}, \tag{119}$$

which is upper bounded by $2\sqrt{\delta}$ due to the condition (Cond. 2) (see p. 53).

In addition, since TDHIT'_1 does not occur for $f = \hat{f}(\mathsf{pk}_0, \cdot) \in X$ and $y \in \tilde{G}$ by definition of X and $\tilde{G}$, the second term of the right hand side of inequality (118) is also upper bounded by $2\sqrt{\delta}$.

Moreover, since $\Delta(\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}, \mathsf{SimCF}^{h_y}) \subset \mathsf{badC}(R', f^{-1}(y)) \cup \mathsf{hitC} = \mathsf{badC}(R', x) \cup \mathsf{hitC}$ holds from Lemma 11, it follows that

$$\begin{aligned}
&\sum_{C \in \Delta(\mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}, \mathsf{SimCF}^{h_y})} \mu_C^{\hat{\mathcal{A}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(f(x)) \\
&\leq \sum_{C \in \mathsf{badC}(R', x)} \mu_C^{\hat{\mathcal{A}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(f(x)) + \sum_{C \in \mathsf{hitC}} \mu_C^{\hat{\mathcal{A}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(f(x)) \\
&\leq \frac{\delta}{Q} + \frac{\delta}{Q},
\end{aligned} \tag{120}$$

here we used the condition (Cond. 3) (see p. 53) and that TDHIT'_2 does not occur for $f = \hat{f}(\mathsf{pk}_0, \cdot) \in X$ and $x \in G$ by definition of X and G for the last inequality. Hence the third term of the right hand side of eq. (118) is upper bounded by $8\sqrt{\delta}$.

Thus, eventually we have

$$\begin{aligned}
&\Big\| \hat{\mathcal{A}}_n^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} |f(x), 0, 0\rangle \\
&\quad - \hat{\mathcal{A}}_n^{\hat{g},(h_y, \hat{f}_{\mathsf{pk} \neq \mathsf{pk}_0}),(\perp, \hat{f}^{\mathrm{inv}}_{\mathsf{td} \neq \mathsf{td}_0}), \mathsf{SimCF}^{h_y}} |f(x), 0, 0\rangle \Big\| \leq 8\sqrt{\delta}.
\end{aligned} \tag{121}$$

Finally, from Lemma 2, for sufficiently small δ it follows that

$$\begin{aligned}
&\Pr\left[\hat{\mathcal{A}}^{\hat{g},(h_y, \hat{f}_{\mathsf{pk} \neq \mathsf{pk}_0}),(\perp, \hat{f}^{\mathrm{inv}}_{\mathsf{td} \neq \mathsf{td}_0}), \mathsf{SimCF}^{h_y}}(f(x)) = x\right] \\
&\geq \Pr\left[\hat{\mathcal{A}}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}}(f(x)) = x\right] \\
&\quad - \Big\| \hat{\mathcal{A}}_n^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}, \mathsf{ColFinder}^{\hat{g},\hat{f},\hat{f}^{\mathrm{inv}}}} |f(x), 0, 0\rangle \\
&\qquad - \hat{\mathcal{A}}_n^{\hat{g},(h_y, \hat{f}_{\mathsf{pk} \neq \mathsf{pk}_0}),(\perp, \hat{f}^{\mathrm{inv}}_{\mathsf{td} \neq \mathsf{td}_0}), \mathsf{SimCF}^{h_y}} |f(x), 0, 0\rangle \Big\| \\
&\geq 2/3 - 8\sqrt{\delta} > 1/2,
\end{aligned} \tag{122}$$

which implies that D correctly recovers $x = f^{-1}(y)$. □

The following lemma shows that our E and D works well with a constant probability.

Lemma 13. *If* $Q^6 \le \delta^4 \cdot \frac{\epsilon}{12} \cdot 2^n/32$,

$$\Pr_{(R,R')}\left[D((R,R'),E((R,R'),f) = f\right] \ge 0.7 \tag{123}$$

holds for each $f = \hat{f}(\mathsf{pk}_0, \cdot) \in X$.

Since it can be proven in the almost same way as Lemma 9 is proven (by replacing $\frac{\epsilon(n)}{6}$ and $\frac{\epsilon(n)}{12}$ with p_1 and p_2, respectively), here we omit the proof of Lemma 13.

Finally, we show that Proposition 3 follows from the above lemmas.

Proof (of Proposition 3). First, remember that the set Y is defined as

$$Y := \left\{(f|_{\{0,1\}^n \setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| \ge \theta\right\}. \tag{124}$$

For each fixed positive integer $\theta \le M \le 2^n$, the cardinality of the set

$$Y_M := \{(f|_{\{0,1\}^n \setminus G}, f(G)) \mid f \in \mathsf{Perm}(\{0,1\}^n), G \subset \{0,1\}^n, |G| = M\} \tag{125}$$

is equal to $(2^n - M)! \cdot \binom{2^n}{M} = (2^n)!/M!$. Thus $|Y|$ is upper bounded as

$$|Y| = \sum_{M=\lceil\theta\rceil}^{2^n} \frac{(2^n)!}{M!} \le 2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!} \tag{126}$$

for sufficiently large n. Here we show the following claim.

Claim. If $Q^6 \le \delta^4 \cdot \frac{\epsilon}{12} \cdot 2^n/32$, there exists a constant const_1 such that $Q^6 \ge \mathsf{const}_1 \cdot \epsilon^2 \cdot 2^n/n$ holds. We can choose const_1 independently of n.

Proof (of Claim). By definition of X, $|X| \ge \frac{\epsilon}{6} \cdot (2^n)!$ holds. In addition, from inequality (126), we have $|Y| \le 2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!}$. Moreover, since now we are assuming that $Q^6 \le \delta^4 \cdot \frac{\epsilon}{12} \cdot 2^n/32$ holds, it follows that $|Y| \ge 0.7|X|$ from Lemma 6 and Lemma 13. Hence we have $2^n \cdot \frac{(2^n)!}{(\lceil\theta\rceil)!} \ge 0.7 \cdot \frac{\epsilon}{6} \cdot (2^n)!$, which is equivalent to

$$\frac{6 \cdot 2^n}{0.7 \cdot \epsilon} \ge \lceil\theta\rceil!. \tag{127}$$

Since $n! \ge 2^n$ holds for $n \ge 4$, we have that

$$\left\lceil \frac{6 \cdot n}{0.7 \cdot \epsilon} \right\rceil! \ge \frac{6 \cdot 2^n}{0.7 \cdot \epsilon} \tag{128}$$

for sufficiently large n. Hence we have $\lceil \frac{6 \cdot n}{0.7 \cdot \epsilon} \rceil \ge \lceil\theta\rceil$, which implies that

$$\frac{6n}{0.7 \cdot \epsilon} + 1 \ge \theta = \delta^4 \left(1 - 60\sqrt{\delta}\right) \cdot \frac{\epsilon}{12} \cdot \frac{2^n}{2Q^6} \tag{129}$$

holds. Moreover, since δ is a constant, there exists a constant const_1 that is independent of n and

$$Q^6 \ge \mathsf{const}_1 \cdot \epsilon^2 \cdot 2^n/n \tag{130}$$

holds, which completes the proof of the claim. □

From the above claim, it follows that there exists a constant const_2 such that

$$Q^6 \geq \min\left\{\delta^4 \cdot \frac{\epsilon}{12} \cdot 2^n/32, \mathsf{const}_1 \cdot \epsilon^2 \cdot 2^n/n\right\} \geq \mathsf{const}_2 \cdot \epsilon^2 2^n/n \tag{131}$$

holds.

Since $Q = c\left\lceil\frac{12}{\epsilon}\right\rceil(\max\{q,\eta\}+1)$ by definition of Q and $\frac{1}{\epsilon} \geq 1$, we have

$$c^6\left\lceil\frac{12}{\epsilon}\right\rceil^6(\max\{q,\eta\}+1)^6 \geq \mathsf{const}_2 \cdot \epsilon^2 \cdot 2^n/n. \tag{132}$$

Hence there exists a constant const such that

$$\max\{q,\eta\} \geq \mathsf{const} \cdot \epsilon^3 \cdot 2^{n/7} \tag{133}$$

holds for sufficiently large n, which completes the proof. $\square$

B Technical Difference from the Previous Version

Here we describe the technical difference from this paper's previous version. The previous version contained a technical error and failed to show the main results. Below we explain only the difference in Section 4 (the separation result for CRH and OWP), since the difference in Section 5 (the separation result for CRH and TDP) is almost the same.

Roughly speaking, in the previous version, the oracle $\mathsf{ColFinder}^f$ was too weak, and one cannot break collision-resistance of all (compressing) hash functions with that oracle contrary to our claim.[16] In the current version, we have modified $\mathsf{ColFinder}^f$ so that it will certainly break collision-resistance of all (compressing) hash functions. The construction of $\mathsf{ColFinder}^f$ has been changed, but other parts of the proof remains almost the same. Below we explain details about what was wrong with $\mathsf{ColFinder}^f$ and how we have corrected it, with an example.

In the previous version, we defined that an input (quantum circuit) C to $\mathsf{ColFinder}^f$ is *valid* if C^f computes a *totally defined* function for *all* permutations $f \in \mathsf{Perm}(\{0,1\}^n)$ (in addition, we defined that C is *invalid* if it is not valid). We constructed $\mathsf{ColFinder}^f$ in such a way that, on each input C, it first checks whether it is a valid input, and reject (i.e., outputs $\perp$) if it is invalid. The previous $\mathsf{ColFinder}^f$ failed to find collisions of some hash functions because of this checking procedure.

For example, let $(\mathsf{Gen}^f, \mathsf{Eval}^f)$ be an oracle-aided implementation of hash function (a pair of oracle-aided quantum circuits) that makes queries to a permutation f. Fix a positive integer n. Assume that outputs of Gen^f on the input 1^n are always in $\{0,1\}^n$ and f is an n-bit permutation, for simplicity. In addition, suppose that $\mathsf{Eval}^f(\sigma,\cdot)$ computes a function $H^f(\sigma,\cdot) : \{0,1\}^{n+1} \to \{0,1\}^n$ for each σ returned by $\mathsf{Gen}^f(1^n)$. Now, consider to construct another implementation of hash function $(\mathsf{Gen}'^f, \mathsf{Eval}'^f)$ as follows.

[16] This was pointed out by a reviewer of STOC 2019.

Algorithm Gen'^{f}.

1. Take 1^n as an input.
2. Run Gen^f on the input 1^n and obtain an output $\sigma \in \{0,1\}^n$.
3. Choose r from $\{0,1\}^n$ uniformly at random and compute $f(r)$ by querying r to f.
4. Return $\sigma' := (\sigma, r, f(r)) \in \{0,1\}^{3n}$.

Algorithm Eval'^{f}.

1. Take (σ', x) as an input, where $\sigma' = (\sigma, r, v)$ and $\sigma, r, v \in \{0,1\}^n$.
2. Check if $f(r) = v$ holds by querying r to f. If it does not hold, return $\perp$.
3. If $f(r) = v$, compute $y = H^f(\sigma, x)$ by running Eval^f on the input (σ, x), and return y.

The pair $(\mathsf{Gen}'^{f}, \mathsf{Eval}'^{f})$ is in fact an (oracle-aided) implementation of hash function. Let $\sigma' = (\sigma, r, f(r))$ be an output of $\mathsf{Gen}'^{f}(1^n)$. The previous $\mathsf{ColFinder}^f$ should have been constructed in such a way that it would return a collision of $H^f(\sigma', \cdot)$ when the (oracle-aided) quantum circuit of $\mathsf{Eval}'^{(\cdot)}(\sigma', \cdot)$ is queried. However, since there exists a permutation g such that $g(r) \neq f(r)$ and $\mathsf{Eval}'^{g}(\sigma, \cdot)$ outputs $\perp$ for any input x, the previous $\mathsf{ColFinder}^f$ judges that the input $\mathsf{Eval}'^{(\cdot)}(\sigma', \cdot)$ is invalid. In particular, $\mathsf{ColFinder}^f(\mathsf{Eval}'^{(\cdot)}(\sigma', \cdot)) = \perp$ holds, and thus we failed to prove the main theorem in the previous version.

On the other hand, in the current version, we just removed the checking procedure from $\mathsf{ColFinder}^f$ so that it will correctly return collisions for all possible implementations of hash function. Though this modification may seem significant, the remaining parts of the proof remain almost the same. Namely, we did not use the fact that $\mathsf{ColFinder}^f$ immediately returns $\perp$ for invalid inputs in the proof at all, and the checking procedure was just unnecessary and extra one.