4-Round Luby-Rackoff Construction is a qPRP

Akinori Hosoyamada[1,2] and Tetsu Iwata[2]

[1] NTT Secure Platform Laboratories, Tokyo, Japan,
hosoyamada.akinori@lab.ntt.co.jp
[2] Nagoya University, Nagoya, Japan,
{hosoyamada.akinori,tetsu.iwata}@nagoya-u.jp

Abstract. The Luby-Rackoff construction, or the Feistel construction, is one of the most important approaches to construct secure block ciphers from secure pseudorandom functions. The 3- and 4-round Luby-Rackoff constructions are proven to be secure against chosen-plaintext attacks (CPAs) and chosen-ciphertext attacks (CCAs), respectively, in the classical setting. However, Kuwakado and Morii showed that a quantum superposed chosen-plaintext attack (qCPA) can distinguish the 3-round Luby-Rackoff construction from a random permutation in polynomial time. In addition, Ito et al. recently showed a quantum superposed chosen-ciphertext attack (qCCA) that distinguishes the 4-round Luby-Rackoff construction. Since Kuwakado and Morii showed the result, a problem of much interest has been how many rounds are sufficient to achieve provable security against quantum query attacks. This paper answers to this fundamental question by showing that 4-rounds suffice against qCPAs. Concretely, we prove that the 4-round Luby-Rackoff construction is secure up to $O(2^{n/12})$ quantum queries. We also give a query upper bound for the problem of distinguishing the 4-round Luby-Rackoff construction from a random permutation by showing a distinguishing qCPA with $O(2^{n/6})$ quantum queries. Our result is the first to demonstrate the security of a typical block-cipher construction against quantum query attacks, without any algebraic assumptions. To give security proofs, we use an alternative formalization of Zhandry's compressed oracle technique.

Keywords: symmetric-key cryptography · post-quantum cryptography · provable security · quantum security · the compressed oracle technique · quantum chosen plaintext attacks · Luby-Rackoff constructions.

1 Introduction

Post-quantum public-key cryptography has been one of the most actively researched areas in cryptography since Shor developed the polynomial-time integer factoring quantum algorithm [30]. NIST is working on a standardization process for post-quantum public-key schemes such as public-key encryption, key-establishment, and digital signature schemes [27].

On the other hand, for symmetric key cryptography, it was said that the security of symmetric-key schemes would not be much affected by quantum

computers. However, a series of recent results has shown that some symmetric key schemes are also broken in polynomial time by using Simon's algorithm [31] if quantum adversaries have access to quantum circuits that implement keyed primitives [18,20,8,6,21,29,13,12,11,17], though they are proven or assumed to be secure in the classical setting. Thus, the post-quantum security of symmetric-key schemes also needs to be studied.

Although many quantum query attacks on symmetric-key schemes have been proposed, post-quantum provable security of symmetric-key schemes has attracted little attention. There are two possible post-quantum security notions for symmetric-key schemes: *standard security* and *quantum security* [33]. The standard security assumes adversaries have quantum computers, but have access to keyed oracles in a classical manner. On the other hand, the quantum security assumes adversaries can make queries to keyed primitives in quantum superpositions. If a scheme is proven to have quantum security, then it will remain secure even in a far future where all computations and communications are done in quantum superpositions. Therefore, it is a problem of much interest whether a classically secure symmetric-key scheme also has quantum security.

The Luby-Rackoff construction. The Luby-Rackoff construction, or the Feistel construction, is one of the most important approaches to construct efficient and secure block ciphers, which are pseudorandom permutations (PRPs), from efficient and secure pseudorandom functions (PRFs). A significant number of block ciphers including commonly used ones such as DES [25] and Camellia [3] has been designed on the basis of this construction.

For families of functions $f_i := \{f_{i,k} : \{0,1\}^{n/2} \to \{0,1\}^{n/2}\}_{k\in\mathcal{K}}$ that are parameterized by k in a key space $\mathcal{K}$ $(1 \le i \le r)$, the r-round Luby-Rackoff construction $\mathsf{LR}_r(f_1,\ldots,f_r)$ is defined as follows: First, keys $k_1,\ldots,k_r$ are chosen independently and uniformly at random from $\mathcal{K}$. For each input $x_0 = x_{0L}\|x_{0R}$, where $x_{0L}, x_{0R} \in \{0,1\}^{n/2}$, the state is updated as

$$x_{(i-1)L}\|x_{(i-1)R} \mapsto x_{iL}\|x_{iR} := x_{(i-1)R} \oplus f_{i,k_i}(x_{(i-1)L})\|x_{(i-1)L} \tag{1}$$

for $i = 1,\ldots,r$ in a sequential order (see Fig. 1). The output is the final state $x_r = x_{rL}\|x_{rR}$. Then the resulting function becomes a keyed permutation over $\{0,1\}^n$ with keys in $(\mathcal{K})^r$.

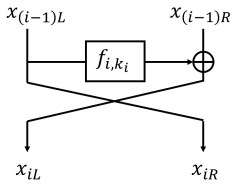

Fig. 1. The i-th round state update.

In the classical setting, if each f_i is a secure PRF, LR_r becomes a secure PRP against chosen-plaintext attacks (CPAs) for $r \geq 3$ and a secure PRP against chosen-ciphertext attacks (CCAs) for $r \geq 4$ [23], i.e., LR_r becomes a strong PRP. However, in the quantum setting, Kuwakado and Morii showed that LR_3 can be distinguished in polynomial time from a truly random permutation by a quantum superposed chosen-plaintext attack [20] (qCPA). [3] Moreover, Ito et al. recently showed that LR_4 can be distinguished in polynomial time by a quantum superposed chosen-ciphertext attack (qCCA) [17]. On the other hand, for any r, no post-quantum security proof of LR_r is known. A very natural question is then whether such a proof is feasible for some r, and if so, the minimum number of r such that we can prove the post-quantum security of LR_r needs to be determined.

1.1 Our Contributions

As the first step to giving post-quantum security proofs for the Luby-Rackoff constructions, this paper shows that the 4-round Luby-Rackoff construction LR_4 is secure against qCPAs. In particular, we give a security bound of LR_4 against qCPAs when all round functions are truly random functions. We also give a query upper bound for the problem of distinguishing LR_4 from a random permutation by showing a distinguishing attack. Concretely, we show the following theorems (see Table 1 for comparing security proofs and attacks for LR_4).

Theorem 1 (Lower bound and upper bound, informal). *If all round functions are truly random functions, then the following claims hold.*

1. LR_4 *cannot be distinguished from a truly random permutation by qCPAs up to* $O(2^{n/12})$ *quantum queries.*
2. *A quantum algorithm exists that distinguishes* LR_4 *from a truly random permutation with a constant probability by making* $O(2^{n/6})$ *quantum chosen-plaintext queries.*

Theorem 2 (Construction of PRP from PRF, informal). *Suppose that each* f_i *is a secure PRF against efficient quantum query attacks, for* $1 \leq i \leq 4$. *Then* $\mathsf{LR}_4(f_1, f_2, f_3, f_4)$ *is a secure PRP against efficient qCPAs.*

Technical details. To give a quantum security proof for LR_4 in the case that all round functions are truly random, we use the *compressed oracle technique* developed by Zhandry [37]. To be precise, we give an alternative formalization of the technique and use it.

One challenging obstacle to giving security proofs against quantum superposed query adversaries is that we cannot record *transcripts* of quantum queries

[3] Strictly speaking, the attack by Kuwakado and Morii works only when all round functions are keyed permutations. Kaplan et al. [18] showed that the attack works for more general cases.

Attack setting	Classical CPA	Classical CCA	Quantum CPA	Quantum CCA
Security proof	Secure up to $O(2^{n/4})$ queries [23]	Secure up to $O(2^{n/4})$ queries [23]	Secure up to $O(2^{n/12})$ queries **[Ours]** (Section 4)	No proofs (Insecure)
Distinguishing attack	$O(2^{n/4})$ queries [28]	$O(2^{n/4})$ queries [28]	$O(2^{n/6})$ queries **[Ours]** (Section 5)	$O(n)$ queries [17]

Table 1. Comparison of security proofs and attacks for the 4-round Luby-Rackoff construction LR_4 when all round functions are truly random. In the quantum CPA/CCA settings, adversaries can make quantum superposed queries.

and answers. Although it is trivial to store query-answer records in the classical setting, it is highly non-trivial to store them in the quantum setting, since measuring or copying (parts of) quantum states will lead to perturbing them, which may be detected by adversaries.

Zhandry's compressed oracle technique enables us to overcome the obstacle when oracles are truly random functions. The technique is so powerful that it can be used to show quantum indifferentiability of the Merkle-Damgård domain extender and quantum security for the Fujisaki-Okamoto transformation [37], in addition to the (tight) lower bounds for the multicollision-finding problems [22]. His crucial observation is that we can record queries and answers without affecting quantum states by appropriately forgetting previous records. In addition, he observed that transcripts of queries can be recorded in an compressed manner, which enables us to simulate random functions (random oracles) extremely efficiently.

The compressed oracle technique is a powerful tool, although the formalization of the technique is (necessarily) somewhat complex. A simpler alternative formalization would be better to have when we apply the technique to complex schemes that use multiple random functions, such as the Luby-Rackoff construction.

Zhandry's formalization enables us to both record transcripts and compress recorded data. We need the compression to efficiently simulate random functions but not when we focus on information theoretic security of cryptographic schemes.

With this in mind, we modify the construction of Zhandry's *compressed standard oracle* and give an alternative formalization of Zhandry's technique without compression of the database. Moreover, we scrutinize the properties of our modified oracle and observe that its behaviors can be described in an intuitively clear manner by introducing some *errors*. We also explicitly describe error terms, which enables us to give mathematically rigorous proofs. We name our alternative oracle the *recording standard oracle with errors*, because it records transcripts of queries and its behavior is described with errors. We believe that our alternative formalization and analyses for our oracle's behavior help us under-

stand Zhandry's technique better, which will lead to the technique being applied even more widely. See Section 3 for details on our alternative formalization.

By heavily using our recording standard oracle with errors, we complete the security proof of LR_4 against quantum superposed query attacks, taking advantage of classical proof intuitions to some extent. First, we consider LR_3, the 3-round Luby-Rackoff construction, which is easy to distinguish from a truly random permutation, and a slightly modified version of it, where the last-round state update of LR_3 is modified. Our observation is that even quantum (chosen-plaintext) query adversaries seem to have difficulty noticing the modification, and we are actually able to show that this is indeed the case. Intuitively, the proof is possible since even quantum query adversaries cannot feasibly produce collisions on the input of the third round. Second, we prove that a family of random permutations (i.e., a function $P : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ such that $P(x,\cdot)$ is a truly random permutation over $\{0,1\}^{n/2}$ for each x) is hard to distinguish from a truly random function. To show the first hardness result, we use our recording standard oracle with errors. On the other hand, for the second hardness result, we can show it by just combining some previous results. Once we prove these two hardness results, the rest of the proof can be done easily without any argument specific to the quantum setting. Our proof is much more complex than the classical one, though, we give rigorous and careful analyses. See Section 4 for details on the security proof of LR_4.

In contrast to the high complexity of the provable security result, our quantum distinguishing attack is a simple quantum polynomial speed-up of existing classical attacks. See Section 5 for details on the quantum distinguishing attack.

1.2 Related Works

Other than the ones introduced above, security proofs against quantum query adversaries for symmetric key schemes include a proof for standard modes of operations by Targhi et al. [2], one for the Carter-Wegman message authentication codes (MACs) by Boneh and Zhandry [5], one for NMAC by Song and Yun [32], and one for Davies-Meyer and Merkle-Damgård constructions by Hosoyamada and Yasuda [16]. Zhandry showed the PRP-PRF switching lemma in the quantum setting [35] and demonstrated that quantum-secure PRPs can be constructed from quantum-secure PRFs by using a technique of format preserving encryption [36]. Czajkowski et al. showed that the sponge construction is *collapsing* (collapsing is a quantum extension of the classical notion of collision-resistance) when round functions are one-way random permutations or functions [9].[4] Alagic and Russell proved that polynomial-time attacks against symmetric-key schemes that use Simon's algorithm can be prevented by replacing XOR operations with modular additions on the basis of an algebraic hardness assumption [1]. However, Bonnetain and Naya-Plasecia showed that the countermeasure is not practical [7]. For standard security proofs (against quantum

[4] Note that the condition in which the round function of the sponge construction is one-way is unusual in the context of classical symmetric-key provable security.

adversaries that make only classical queries) for symmetric-schemes, Mennink and Szepieniec proved security for XOR of PRPs [24]. Czajkowski et al. [10] recently showed that the compressing technique can be extended to quantum oracles with non-uniform distributions such as a random permutation, and showed quantum indifferentiability of the sponge construction.

2 Preliminaries

This section describes notations and definitions. In this paper, all algorithms (or adversaries) are assumed to be quantum algorithms, and make quantum superposed queries to oracles. For any finite sets X and Y, let $\mathsf{Func}(X,Y)$ denote the set of all functions from X to Y. For any n-bit string x, we denote the left-half $n/2$-bits of x by x_L and the right-half $n/2$-bits by x_R, respectively. We identify the set $\{0,1\}^m$ with the set of the integers $\{0,1,\ldots,2^m-1\}$.

2.1 Quantum Computation

Throughout this paper, we assume that readers have basic knowledge about quantum computation and finite dimensional linear algebra (see textbooks such as [26,19] for an introduction). We use the computational model of quantum circuits. We measure complexity of quantum algorithms by the number of queries, and the number of basic gates in addition to oracle gates. In this paper, *basic gates* denote the gates in the standard basis of quantum circuits $\mathcal{Q}$ [19]. Let $\|\cdot\|$ and $\|\cdot\|_{\mathsf{tr}}$ denote the norm of vectors and the trace norm of operators, respectively. In addition, let $\mathsf{td}(\cdot,\cdot)$ denote the trace distance. For Hermitian operators ρ,σ on a Hilbert space $\mathcal{H}$, $\mathsf{td}(\rho,\sigma)=\frac{1}{2}\|\rho-\sigma\|_{\mathsf{tr}}$ holds. For a mixed state ρ of a joint quantum system $\mathcal{H}_A\otimes\mathcal{H}_B$, let $\mathsf{tr}_B(\rho)$ (resp., $\mathsf{tr}_A(\rho)$) denote the partial trace of ρ over $\mathcal{H}_B$ (resp., $\mathcal{H}_A$). Moreover, for a pure state $|\psi\rangle$ of the joint quantum system $\mathcal{H}_A\otimes\mathcal{H}_B$, we write $\mathsf{tr}_B(|\psi\rangle)$ (resp., $\mathsf{tr}_A(|\psi\rangle)$) instead of $\mathsf{tr}_B(|\psi\rangle\langle\psi|)$ (resp., $\mathsf{tr}_A(|\psi\rangle\langle\psi|)$), for simplicity. Similarly, for a pure state $|\psi\rangle$ and a mixed state ρ of a quantum system $\mathcal{H}$, we write $\mathsf{td}(|\psi\rangle,\rho)$ and $\mathsf{td}(\rho,|\psi\rangle)$ instead of $\mathsf{td}(|\psi\rangle\langle\psi|,\rho)$ and $\mathsf{td}(\rho,|\psi\rangle\langle\psi|)$, respectively. For an integer $n\geq 1$, I_n and $H^{\otimes n}$ denote the identity operator on n-qubit systems and the n-qubit Hadamard operator, respectively. If n is clear from the context, we just write I instead of I_n, for concision. By abuse of notation, for an operator V, we sometimes use the same notation V to denote $V\otimes I$ or $I\otimes V$ for simplicity, when it will cause no confusion. In addition, for a vector $|\phi\rangle$ and a positive integer m, we sometimes use the same notation $|\phi\rangle$ to denote $|\phi\rangle\otimes|0^m\rangle$ or $|0^m\rangle\otimes|\phi\rangle$ for simplicity, when it will cause no confusion.

Quantum oracle query algorithms. Following previous works (see [4], for example), any quantum oracle query algorithm $\mathcal{A}$ that makes at most q queries to oracles is modeled as a sequence of unitary operators $(U_0,\ldots,U_q)$, where each U_i is a unitary operator on an ℓ-qubit quantum system, for some integer ℓ. Here, U_0 can be regarded as the initialization process, and for $1\leq i\leq q-1$, U_i is

the process after the i-th query. U_q can be regarded as the finalization process. We only consider quantum algorithms that take no inputs and assume that the initial state of $\mathcal{A}$ is $|0^\ell\rangle$.

Stateless oracles. For a function $f : \{0,1\}^m \to \{0,1\}^n$, the quantum oracle of f is defined as the unitary operator $O_f : |x,y\rangle \mapsto |x, y \oplus f(x)\rangle$. When we run $\mathcal{A}$ relative to the oracle O_f, the unitary operators $U_0, O_f, \ldots, U_{q-1}, O_f, U_q$ act sequentially on the initial state $|0^\ell\rangle$. (We consider that O_f acts on the first $(m+n)$-qubits of $\mathcal{A}$'s quantum register.) Finally, $\mathcal{A}$ measures the resulting quantum state $U_q O_f U_{q-1} \cdots O_f U_0 |0^\ell\rangle$, and returns the measurement result as the output. f may be chosen in accordance with a distribution at the beginning of each game. Let us denote the event that $\mathcal{A}$ runs relative to the oracle O_f and returns an output α by $\alpha \leftarrow \mathcal{A}^{O_f}()$ or by $\mathcal{A}^{O_f}() \to \alpha$.

Stateful oracles. In this paper, we also consider more general cases in which quantum oracles are stateful, i.e., oracles have ℓ'-qubit quantum states for an integer $\ell' \geq 0$.[5] In these cases, an oracle $\mathcal{O}$ is modeled as a sequence of unitary operators $(\mathcal{O}_1, \ldots, \mathcal{O}_q)$ that acts on the first $(m+n)$-qubits of $\mathcal{A}$'s quantum register in addition to $\mathcal{O}$'s quantum register. When we run $\mathcal{A}$ relative to the oracle $\mathcal{O}$, the unitary operators $U_0 \otimes I_{\ell'}, \mathcal{O}_1, \ldots, (U_{q-1} \otimes I_{\ell'}), \mathcal{O}_q, (U_q \otimes I_{\ell'})$ act in a sequential order on the initial state $|0^\ell\rangle \otimes |\mathsf{init}_{\mathcal{O}}\rangle$, where $|\mathsf{init}_{\mathcal{O}}\rangle$ is the initial state of $\mathcal{O}$. Finally, $\mathcal{A}$ measures the resulting quantum state $(U_q \otimes I_{\ell'})\mathcal{O}_q(U_{q-1} \otimes I_{\ell'}) \cdots \mathcal{O}_1(U_0 \otimes I_{\ell'}) |0^\ell\rangle \otimes |\mathsf{init}_{\mathcal{O}}\rangle$, and returns the measurement result as the output. If $\mathcal{O}$ has no state and $\mathcal{O}_i = O_f$ holds for each i, the behavior of $\mathcal{A}$ relative to $\mathcal{O}$ precisely matches that of $\mathcal{A}$ relative to the stateless oracle O_f. Thus, our model of stateful oracles is an extension of the typical model of stateless oracles described above. $\mathcal{O}$ may be chosen in accordance with a distribution at the beginning of each game. We denote the event that $\mathcal{A}$ runs relative to the oracle $\mathcal{O}$ and returns an output α by $\alpha \leftarrow \mathcal{A}^{\mathcal{O}}()$ or by $\mathcal{A}^{\mathcal{O}}() \to \alpha$.

Quantum distinguishing advantages. Let $\mathcal{A}$ be a quantum algorithm that makes at most q queries and outputs 0 or 1 as the final output, and let $\mathcal{O}_1$ and $\mathcal{O}_2$ be some oracles. We consider the situation in which $\mathcal{O}_1$ and $\mathcal{O}_2$ are chosen randomly in accordance with some distributions. We define the *quantum distinguishing advantage* of $\mathcal{A}$ by

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathcal{O}_1,\mathcal{O}_2}(\mathcal{A}) := \left| \Pr_{\mathcal{O}_1}\left[\mathcal{A}^{\mathcal{O}_1}() \to 1\right] - \Pr_{\mathcal{O}_2}\left[\mathcal{A}^{\mathcal{O}_2}() \to 1\right] \right|. \tag{2}$$

When we are interested only in the number of queries and do not consider other complexities such as the number of gates (i.e., we focus on information

[5] Here we do not mean that our model captures all reasonable stateful quantum oracles. We use our model of stateful quantum oracles just for intermediate arguments to prove our main results, and the claims of the main results are described in the typical model of stateless oracles.

theoretic adversaries), we use the notation

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathcal{O}_1,\mathcal{O}_2}(q) := \max_{\mathcal{A}} \left\{ \mathbf{Adv}^{\mathrm{dist}}_{\mathcal{O}_1,\mathcal{O}_2}(\mathcal{A}) \right\}, \tag{3}$$

where the maximum is taken over all quantum algorithms that make at most q quantum queries.

Quantum PRF advantages. RF denotes the quantum oracle of random functions, i.e., the oracle such that a function $f \in \mathsf{Func}(\{0,1\}^m, \{0,1\}^n)$ is chosen uniformly at random, and an oracle access to O_f is given to adversaries.

Let $\mathcal{F} = \{F_k : \{0,1\}^m \to \{0,1\}^n\}_{k\in\mathcal{K}}$ be a family of functions. Let us use the same symbol $\mathcal{F}$ to denote the oracle such that k is chosen uniformly at random, and an oracle access to O_{F_k} is given to adversaries. In addition, let $\mathcal{A}$ be an oracle query algorithm that outputs 0 or 1. Then we define the quantum pseudorandom-function (qPRF) advantage by $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathcal{F}}(\mathcal{A}) := \mathbf{Adv}^{\mathrm{dist}}_{\mathcal{F},\mathsf{RF}}(\mathcal{A})$. Similarly, we define $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathcal{F}}(q)$ by $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathcal{F}}(q) := \max_{\mathcal{A}} \left\{ \mathbf{Adv}^{\mathrm{qPRF}}_{\mathcal{F}}(\mathcal{A}) \right\}$, where the maximum is taken over all quantum algorithms $\mathcal{A}$ that make at most q quantum queries.

Quantum PRP advantages. By RP we denote the quantum oracle of random permutations, i.e., the oracle such that a permutation $P \in \mathsf{Perm}(\{0,1\}^n)$ is chosen uniformly at random, and an oracle access to O_P is given to adversaries.

Let $\mathcal{P} = \{P_k : \{0,1\}^n \to \{0,1\}^n\}_{k\in\mathcal{K}}$ be a family of permutations. We use the same symbol $\mathcal{P}$ to denote the oracle such that k is chosen uniformly at random, and an oracle access to O_{P_k} is given to adversaries. Let $\mathcal{A}$ be an oracle query algorithm that outputs 0 or 1, and we define the quantum pseudorandom-permutation (qPRP) advantage by $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathcal{P}}(\mathcal{A}) := \mathbf{Adv}^{\mathrm{dist}}_{\mathcal{P},\mathsf{RP}}(\mathcal{A})$. Similarly, we define $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathcal{P}}(q)$ by $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathcal{P}}(q) := \max_{\mathcal{A}} \left\{ \mathbf{Adv}^{\mathrm{qPRP}}_{\mathcal{P}}(\mathcal{A}) \right\}$, where the maximum is taken over all quantum algorithms $\mathcal{A}$ that make at most q quantum queries.

Security against efficient adversaries. An algorithm $\mathcal{A}$ is called *efficient* if it can be realized as a quantum circuit that has a polynomial number of basic gates and oracle gates in n. A set of functions $\mathcal{F}$ (resp., a set of permutations $\mathcal{P}$) is a *quantumly secure PRF* (resp., a *quantumly secure PRP*) if the following properties are satisfied:

1. Uniform sampling $f \xleftarrow{\$} \mathcal{F}$ (resp., $P \xleftarrow{\$} \mathcal{P}$) and evaluation of each f (resp., each P) can be implemented on quantum circuits that have a polynomial number of basic gates in n.
2. $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathcal{F}}(\mathcal{A})$ (resp., $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathcal{P}}(\mathcal{A})$) is *negligible* (i.e., for any positive integer c, it is upper bounded by n^{-c} for all sufficiently large n) for any efficient algorithm $\mathcal{A}$.

2.2 The Luby-Rackoff Constructions

The Luby-Rackoff construction [23] is a construction of n-bit permutations from $n/2$-bit functions by using the Feistel network.

Fix $r \geq 1$, and for $1 \leq i \leq r$, let $f_i := \{f_{i,k} : \{0,1\}^{n/2} \to \{0,1\}^{n/2}\}_{k \in \mathcal{K}}$ be a family of functions parameterized by key k in a key space $\mathcal{K}$. Then, the Luby-Rackoff construction for $f_1, \ldots, f_r$ is defined as a family of n-bit permutations $\mathsf{LR}_r(f_1, \ldots, f_r) := \{\mathsf{LR}_r(f_{1,k_1}, \ldots, f_{r,k_r})\}_{k_1,\ldots,k_r \in \mathcal{K}}$ with the key space $(\mathcal{K})^r$. For each fixed key $(k_1, \ldots, k_r)$, $\mathsf{LR}_r(f_{1,k_1}, \ldots, f_{r,k_r})$ is defined by the following procedure: First, given an input $x_0 \in \{0,1\}^n$, divide it into $n/2$-bit strings x_{0L} and x_{0R}. Second, iteratively update n-bit states as

$$(x_{(i-1)L}, x_{(i-1)R}) \mapsto (x_{iL}, x_{iR}) := (x_{(i-1)R} \oplus f_{i,k_i}(x_{(i-1)L}), x_{(i-1)L}) \quad (4)$$

for $1 \leq i \leq r$. Finally, return the final state $x_r := x_{rL} \| x_{rR}$ as the output (see Fig. 2).

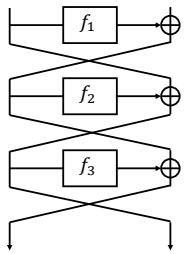

Fig. 2. The 3-round Luby-Rackoff construction.

The resulting function $\mathsf{LR}_r(f_{1,k_1}, \ldots, f_{r,k_r}) : x_0 \mapsto x_r$ becomes an n-bit permutation owing to the property of the Feistel network. Each f_{i,k_i} is called the i-th round function. When we say that an adversary is given an oracle access to $\mathsf{LR}_r(f_1, \ldots, f_r)$, we consider the situation in which keys $k_1, \ldots, k_r$ are first chosen independently and uniformly at random, and then the adversary runs relative to the stateless oracle $O_{\mathsf{LR}_r(f_{1,k_1},\ldots,f_{r,k_r})} : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus \mathsf{LR}_r(f_{1,k_1}, \ldots, f_{r,k_r})(x)\rangle$. When each round function is chosen from $\mathsf{Func}(\{0,1\}^{n/2}, \{0,1\}^{n/2})$ uniformly at random (i.e., each f_i is the set of all functions $\mathsf{Func}(\{0,1\}^{n/2}, \{0,1\}^{n/2})$ for all i), we use the notation LR_r for short.

3 An Alternative Formalization for the Compressed Oracle Technique

Many security proofs in the *classical* random oracle model (ROM), implicitly rely on the fact that transcripts of queries and answers can be recorded.

However, such proofs do not necessarily work in the *quantum random oracle model* (QROM) [4], since recording transcripts may significantly perturb quantum states, which might be detected by adversaries. To solve this issue, Zhandry introduced the "compressed oracle technique" [37] to enable us to record transcripts of queries and answers even in QROM. In addition to recording transcripts, Zhandry's technique enables us to simulate the random oracle extremely efficiently by compressing databases of transcripts.

Zhandry's technique was originally developed for QROM, in which adversaries can make direct queries to random functions, but it can also be applied when adversaries can make queries to random functions only indirectly. In particular, one may think that the technique is applicable to giving a security proof for the Luby-Rackoff constructions when all round functions are truly random.

The compressed oracle technique is very insightful and promising, but its formal description is somewhat (necessarily) complex. A simpler formalization would be better to have when we want to apply the technique to complex schemes that use multiple random functions, such as the Luby-Rackoff construction.

In provable security, especially for symmetric-key mode of operations, we often focus on security against information theoretic adversaries. When we are interested in such security, we do not care about efficient simulation of a random oracle, and thus do not have to compress databases. With this in mind, we modify the construction of Zhandry's *compressed standard oracle* and give an alternative formalization of his technique without compressing databases that can be used when we focus on (quantum) information theoretic security.

We also study the behavior of our oracle in detail and show that its properties can be described intuitively by introducing the notion of errors. Since our oracle records transcripts of queries and its behavior is described with errors, we call our oracle *recording standard oracle with errors* and denote it by RstOE.

We believe that our alternative formalization and analyses for its behavior help us understand Zhandry's technique better, which will lead to the technique begin applied even more widely.

In Section 3.1 we give an overview of the original technique by Zhandry, and describe which part of it can be improved. Then, in Section 3.2 we describe our alternative formalization for the technique.

3.1 An Overview of the Original Technique

First, Zhandry observed that the oracle O_f can be implemented with an encoding of f and an operator stO that is independent of f. In this subsection, we consider that each function $f : \{0,1\}^m \to \{0,1\}^n$ is encoded into the $(n2^m)$-qubit state $|f\rangle = |f(0)\|f(1)\|\cdots\|f(2^m-1)\rangle$. The operator stO is the unitary operator that acts on $(n+m+n2^m)$-qubit states defined as

$$\mathsf{stO} : |x\rangle|y\rangle \otimes |\alpha_0\rangle \cdots |\alpha_{2^m-1}\rangle \mapsto |x\rangle|y \oplus \alpha_x\rangle \otimes |\alpha_0\rangle \cdots |\alpha_{2^m-1}\rangle, \tag{5}$$

where $\alpha_x \in \{0,1\}^n$ for each $0 \le x \le 2^m - 1$. We can easily confirm that $\mathsf{stO}|x\rangle|y\rangle|f\rangle = |x\rangle|y \oplus f(x)\rangle|f\rangle$ holds. Here, we consider that $|x\rangle|y\rangle$ corresponds to the first $(m+n)$-qubits of adversaries' registers.

When f is chosen uniformly at random and $\mathcal{A}$ runs relative to stO and $|f\rangle$ (i.e., $\mathcal{A}$ runs relative to the quantum oracle of a random function), the whole quantum state before $\mathcal{A}$ makes the $(i+1)$-st quantum query becomes

$$|\phi_{f,i+1}\rangle = (U_i \otimes I)\mathsf{stO}(U_{i-1} \otimes I)\mathsf{stO} \cdots \mathsf{stO}(U_0 \otimes I)\,|0^\ell\rangle\,|f\rangle \tag{6}$$

with probability $1/2^{n2^m}$. Here, we assume that $\mathcal{A}$ has ℓ-qubit quantum states.

Random choice of f can be implemented by first making the uniform superposition of functions $\sum_f \frac{1}{\sqrt{2^{n2^m}}}|f\rangle = H^{\otimes n2^m}|0^{n2^m}\rangle$ and then measuring the state with the computational basis. So far we have considered that a random function f is chosen at the beginning of games, but the output distribution of $\mathcal{A}$ will not be changed even if we measure the $|f\rangle$ register at the same time as $\mathcal{A}$'s register. Thus, below we consider that all quantum registers including those of functions are measured only once at the end of each game.

Then the whole quantum state before $\mathcal{A}$ makes the $(i+1)$-st quantum query becomes

$$|\phi_{i+1}\rangle = \sum_f |\phi_{f,i+1}\rangle = (U_i \otimes I)\mathsf{stO} \cdots \mathsf{stO}(U_0 \otimes I)\left(|0^\ell\rangle \otimes \sum_f \frac{1}{\sqrt{2^{n2^m}}}|f\rangle\right). \tag{7}$$

Next, we change the basis of the y register and α_i registers in (5) from the standard computational basis $\{|u\rangle\}_{u\in\{0,1\}^n}$ to one called the *Fourier basis* $\{H^{\otimes n}|u\rangle\}_{u\in\{0,1\}^n}$[6] by Zhandry [37]. In what follows, we use the symbol " $\widehat{\ }$ " to denote the encoding of classical bit strings into quantum states by using the Fourier basis instead of the computational basis, and we ambiguously denote $H^{\otimes n}|u\rangle$ by $|\widehat{u}\rangle$ for each $u \in \{0,1\}^n$. Then, it can be easily confirmed that

$$\mathsf{stO}\,|x\rangle\,|\widehat{y}\rangle \otimes |\widehat{\alpha_0}\rangle \cdots |\widehat{\alpha_{2^m-1}}\rangle = |x\rangle\,|\widehat{y}\rangle \otimes |\widehat{\alpha_0}\rangle \cdots |\widehat{\alpha_x \oplus y}\rangle \cdots |\widehat{\alpha_{2^m-1}}\rangle \tag{8}$$

holds. Intuitively, the direction of data writing changes when we change the basis: When we use the standard computational basis, data is written from the function registers to adversaries' registers as in (5). On the other hand, when we use the Fourier basis, data is written in the opposite direction as in (8). With the Fourier basis, $|\phi_{i+1}\rangle$ can be written as

$$|\phi_{i+1}\rangle = (U_i \otimes I)\mathsf{stO}(U_{i-1} \otimes I)\mathsf{stO} \cdots \mathsf{stO}(U_0 \otimes I)\left(|0^\ell\rangle \otimes |\widehat{0^{n2^m}}\rangle\right). \tag{9}$$

Here, note that $\sum_f |f\rangle = H^{\otimes n2^m}|0^{n2^m}\rangle = |\widehat{0^{n2^m}}\rangle$ holds. In particular, the register of the functions are initially set as $|\widehat{0^{n2^m}}\rangle$, and at most one data is written (in superpositions) when an adversary makes a query. Thus

$$|\phi_{i+1}\rangle = \sum_{xyz\widehat{D}} a'_{xyz\widehat{D}}\,|xyz\rangle \otimes |\widehat{D}\rangle \tag{10}$$

[6] Note that the Hadamard operator $H^{\otimes n}$ corresponds to the Fourier transformation over the group $(\mathbb{Z}/2\mathbb{Z})^{\oplus n}$.

holds for some complex numbers $a'_{xyz\widehat{D}}$ such that $\sum_{xyz\widehat{D}} |a'_{xyz\widehat{D}}|^2 = 1$, where each x is an m-bit string that corresponds to $\mathcal{A}$'s query register, y is an n-bit string that corresponds to $\mathcal{A}$'s answer register, z corresponds to $\mathcal{A}$'s remaining register, and $\widehat{D} = \widehat{\alpha_0}\| \cdots \|\widehat{\alpha_{2^m-1}}$ is a concatenation of 2^m many n-bit strings.

Zhandry's key observation is that, since stO adds at most one data to the $\widehat{D}$-register in each query, $\widehat{\alpha}_x \neq 0^n$ holds for at most i many x, and thus $\widehat{D}$ can be regarded as a database with at most i many non-zero entries. (Note that $\widehat{D}$ may contain fewer than i non-zero entries. For example, if a state $|x\rangle |\widehat{y}\rangle$ is successively queried to stO twice, then the database will remain unchanged since $\mathsf{stO} \cdot \mathsf{stO} = I$.) We use the same notation $\widehat{D}$ to denote the database and call it the *Fourier database* since now we are using the Fourier basis for $\widehat{D}$. Each entry of the database $\widehat{D}$ has the form $(x, \widehat{\alpha}_x)$, where $x \in \{0,1\}^m$, $\widehat{\alpha}_x \in \{0,1\}^n$, and $\widehat{\alpha}_x \neq 0^n$.

Intuitively, if the Fourier database $\widehat{D}$ contains an entry $(x, \widehat{\alpha}_x)$, it means that $\mathcal{A}$ has queried x to a random function f and holds some information about the value $f(x)$. Hence $\widehat{D}$ can be seen as a record of transcripts for queries and answers. However, it is still not clear what kind of information $\mathcal{A}$ has about the value $f(x)$, since we are now using the Fourier basis. To clarify this information, let the Hadamard operator $H^{\otimes n}$ act on each $\widehat{\alpha}_x$ in $\widehat{D}$ and obtain another (superposition of) database D. Then, intuitively, D satisfies the condition in which "$(x, \alpha_x) \in D$ corresponds to the condition that $\mathcal{A}$ has queried x to the oracle and received the value α_x in response." We call D a *standard database*.

In summary, Zhandry observed that the quantum random oracle can be described as a stateful quantum oracle CstO. The whole quantum state of an adversary $\mathcal{A}$ and the oracle just before the $(i+1)$-st query is

$$|\phi_{i+1}\rangle = \sum_{xyzD} a_{xyzD} |xyz\rangle \otimes |D\rangle , \tag{11}$$

where each D is a standard database that contains at most i entries. Initially, the database D is empty. Intuitively, when $\mathcal{A}$ makes a query $|x, y\rangle$ to the oracle, CstO does the following three-step procedure. [7]

The three-step procedure of CstO.

1. Look for a tuple $(x, \alpha_x) \in D$. If one is found, respond with $|x, y \oplus \alpha_x\rangle$.
2. If no tuple is found, create new registers initialized to the state $\frac{1}{\sqrt{2^n}} \sum_{\alpha_x} |\alpha_x\rangle$. Add the registers (x, α_x) to D. Then respond with $|x, y \oplus \alpha_x\rangle$.
3. Finally, regardless of whether the tuple was found or added, there is now a tuple (x, α_x) in D, which may have to be removed. To do so, test whether the registers containing α_x contain 0^n in the Fourier basis. If so, remove the tuple from D. Otherwise, leave the tuple in D.

[7] Note that this three-step procedure is a quoted verbatim from the original paper [37] of version 20180814:183812, except that the symbol y' and 0 are used instead of α_x and 0^n, respectively, in the original procedure.

Intuitively, the first and second steps correspond to the classical *lazy sampling*, which do the following procedure: When an adversary makes a query x to the oracle, look for a tuple (x, α_x) in the database. If one is found, respond with α_x (this part corresponds to the first procedure of CstO). If no tuple is found, *choose* α_x *uniformly at random from* $\{0,1\}^n$ (this part corresponds to creating the superposition $\frac{1}{\sqrt{2^n}}\sum_{\alpha_x}|\alpha_x\rangle$ in the second step of CstO), respond with α_x, and add (x, α_x) to the database.

The third "test and forget" step is crucial and specific to the quantum setting. Intuitively, the third step forgets data that is no longer used by the adversary from the database. By appropriately forgetting information, we can record transcripts of queries and answers without perturbing quantum states.

Formalization with compression. On the basis of above clever intuitions, Zhandry gave a formalized description of the compressed standard oracle CstO (although we do not give the explicit description here). Note that, since each database D has at most i entries before the $(i+1)$-st query, D can be encoded in a compressed manner by using only $O(i(m+n))$ qubits. With this observation, CstO is formalized in such a way that it has $O(i(m+n))$-qubit states before the $(i+1)$-st query for each i, which enables us to simulate a random oracle very efficiently on the fly, without an a priori bound on the number of queries (which required computational assumption before Zhandry's work).

3.2 Our Alternative Formalization

Next we give our alternative formalization. The original oracle CstO maintains only a $O(i(m+n))$-qubit state by compressing databases. On the other hand, in our alternative formalization, we do not consider any compression to focus on recording transcripts of queries, and our oralce always has $(n+1)2^m$-qubit states.

From now on, we represent each function $f : \{0,1\}^m \to \{0,1\}^n$ as $(n+1)2^m$-bit strings $(0\|f(0))\|(0\|f(1))\|\cdots\|(0\|f(2^m-1))$. Remember that the whole quantum state before $\mathcal{A}$ makes the $(i+1)$-st query is described as

$$|\phi_{i+1}\rangle = (U_i\otimes I)\mathsf{stO}(U_{i-1}\otimes I)\mathsf{stO}\cdots\mathsf{stO}(U_0\otimes I)\left(|0^\ell\rangle\otimes\sum_f \frac{1}{\sqrt{2^{n2^m}}}|f\rangle\right). \quad (12)$$

At each query, unlike the original technique that adds/deletes at most one entry to/from each database, we first "decode" superpositions of databases to superpositions of functions when an adversary makes a query, then respond to the adversary, and finally "encode" again superpositions of functions to superpositions of databases. Below we describe our encoding.

Encoding functions to databases: Intuitive descriptions. Modifying the idea of Zhandry, we apply the following operations to the $|f\rangle$-register of $|\phi_{i+1}\rangle$.

1. Let the Hadamard operator $H^{\otimes n}$ act on the $f(x)$ register for all x. Now the state becomes
$$\sum_{xyz\widetilde{D}} a'_{xyz\widetilde{D}} |xyz\rangle \otimes |\widetilde{D}\rangle \tag{13}$$
for some complex numbers $a'_{xyz\widetilde{D}}$, where each $\widetilde{D} = (0\|\widehat{\alpha}_0)\| \cdots \|(0\|\widehat{\alpha}_{2^m-1})$ is a concatenation of 2^m many $(n+1)$-bit strings, and $\widehat{\alpha}_x \neq 0^n$ at most i-many x.
2. For each x, if $\widehat{\alpha}_x \neq 0^n$, flip the bit just before $\widehat{\alpha}_x$. Now each $\widetilde{D}$ changes to the bit strings $(b_0\|\widehat{\alpha}_0)\| \cdots \|(b_{2^m-1}\|\widehat{\alpha}_{2^m-1})$, where $b_x \in \{0,1\}$, and $b_x = 1$ if and only if $\widehat{\alpha}_x \neq 0^n$.
3. For each $x \in \{0,1\}^n$, let the n-bit Hadamard transformation $H^{\otimes n}$ act on $|\widehat{\alpha}_x\rangle$ if and only if $b_x = 1$. Then the quantum state becomes
$$|\psi_{i+1}\rangle := \sum_{xyzD} a_{xyzD} |xyz\rangle \otimes |D\rangle \tag{14}$$
for some complex numbers a_{xyzD}, where each D is a concatenation of 2^m many $(n+1)$-bit strings $(b_0\|\alpha_0)\| \cdots \|(b_{2^m-1}\|\alpha_{2^m-1})$ such that $b_x \neq 0$ holds for at most i many x, and intuitively $b_x \neq 0$ means that $\mathcal{A}$ has queried x to a random function f and has information that $f(x) = \alpha_x$.

Encoding functions to databases: Formal descriptions. The above three operations can be formally realized as actions of unitary operators on $|f\rangle$-registers. The first one is realized as $\mathsf{IH} := (I_1 \otimes H^{\otimes n})^{\otimes 2^m}$. The second one is realized as $U_{\mathrm{toggle}} := (I_1 \otimes |0^n\rangle\langle 0^n| + X \otimes (I_n - |0^n\rangle\langle 0^n|))^{\otimes 2^m}$, where X is the 1-qubit operator such that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$. The third one is realized by the operator $\mathsf{CH} := (CH^{\otimes n})^{\otimes 2^m}$, where $CH := |0\rangle\langle 0| \otimes I_n + |1\rangle\langle 1| \otimes H^{\otimes n}$.

We call the action of unitary operator $U_{\mathrm{enc}} := \mathsf{CH} \cdot U_{\mathrm{toggle}} \cdot \mathsf{IH}$ and its conjugate U^*_{enc} *encoding* and *decoding*, respectively. By using our encoding and decoding, the recording standard oracle with errors is defined as follows.

Definition 1 (Recording standard oracle with errors). *The* recording standard oracle with errors *is the stateful quantum oracle such that queries are processed with the unitary operator* RstOE *defined by* $\mathsf{RstOE} := (I \otimes U_{\mathrm{enc}}) \cdot \mathsf{stO} \cdot (I \otimes U^*_{\mathrm{enc}})$.

Note that $|\psi_{i+1}\rangle = (U_i \otimes I)\mathsf{RstOE}(U_{i-1} \otimes I)\mathsf{RstOE} \cdots \mathsf{RstOE}(U_0 \otimes I)(|0^\ell\rangle \otimes |0^{(n+1)2^m}\rangle)$ and $|\phi_{i+1}\rangle = (I \otimes U^*_{\mathrm{enc}})|\psi_{i+1}\rangle$ hold for each i.

Next, we introduce notations related to our recording standard oracle with errors that are required to describe properties of RstOE.

Notations related to RstOE. We call a bit string $D = (b_0\|\alpha_0)\| \cdots \|(b_{2^m-1}\|\alpha_{2^m-1})$, where $b_x \in \{0,1\}$ and $\alpha_x \in \{0,1\}^n$ for each $x \in \{0,1\}^m$, is a *valid database* if $\alpha_x \neq 0^n$ holds only if $b_x \neq 0$. We call D an *invalid database* if it is not a valid database. Note that, in a valid database, b_x can be 0 or 1

if $\alpha_x = 0^n$. We identify a valid database D with the partially defined function from $\{0,1\}^m$ to $\{0,1\}^n$ of which the value on $x \in \{0,1\}^m$ is defined to be y if and only if $b_x \neq 0$ and $\alpha_x = y$. We use the same notation D for this function. Moreover, we identify D with the set $\{(x, D(x))\}_{x \in \mathrm{dom}(D)} \subset \{0,1\}^m \times \{0,1\}^n$. We say that *an entry of x is in D* if $(x, y) \in D$ for some y. Unless otherwise noted, we always assume that D is valid.

We say that a valid database D is compatible with a function $f : \{0,1\}^m \to \{0,1\}^n$ if $D(x) = f(x)$ holds for each x in the domain of D. For each valid database D, let $\mathsf{comp}(D)$ denote the set of functions that are compatible with D.

If $\| \left|\psi\right\rangle - \left|\psi'\right\rangle \|$ is in $O(\epsilon)$ for two vectors $\left|\psi\right\rangle, \left|\psi'\right\rangle$, and some parameter ϵ (which will be a function of n in later applications), then we say that $\left|\psi\right\rangle$ *is equal to* $\left|\psi'\right\rangle$ *with an error in* $O(\epsilon)$, or just write $\left|\psi\right\rangle = \left|\psi'\right\rangle$ *with an error in* $O(\epsilon)$.

The following proposition describes the core properties of RstOE.

Proposition 1 (Core Properties). *Let D be a valid database. Then, the following properties hold.*

1. *Suppose that $|D| \leq i$ holds. Then*

$$U^*_{\mathrm{enc}} \left|D\right\rangle = \sum_{f \in \mathsf{comp}(D)} \sqrt{\frac{1}{|\mathsf{comp}(D)|}} \left|f\right\rangle \tag{15}$$

   holds with an error in $O(\sqrt{i^2/2^n})$.
2. *Suppose that there is no entry of x in D. Then, for any y and α,*

$$\mathsf{RstOE} \left|x\right\rangle \left|y\right\rangle \otimes \left|D \cup (x, \alpha)\right\rangle = \left|x\right\rangle \left|y \oplus \alpha\right\rangle \otimes \left|D \cup (x, \alpha)\right\rangle$$

   with an error in $O(1/\sqrt{2^n})$. More precisely,

$$\begin{aligned}
&\mathsf{RstOE} \left|x, y\right\rangle \otimes \left|D \cup (x, \alpha)\right\rangle \\
&\quad = \left|x, y \oplus \alpha\right\rangle \otimes \left|D \cup (x, \alpha)\right\rangle \\
&\qquad + \frac{1}{\sqrt{2^n}} \left|x, y \oplus \alpha\right\rangle \left(\left|D\right\rangle - \left(\sum_{\gamma \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} \left|D \cup (x, \gamma)\right\rangle \right) \right) \\
&\qquad - \frac{1}{\sqrt{2^n}} \sum_{\gamma} \frac{1}{\sqrt{2^n}} \left|x, y \oplus \gamma\right\rangle \otimes \left(\left|D \cup (x, \gamma)\right\rangle - \left|D^{\mathsf{invalid}}_{\gamma}\right\rangle \right) \\
&\qquad + \frac{1}{2^n} \left|x\right\rangle \left|\widehat{0^n}\right\rangle \otimes \left(2 \sum_{\delta \in \{0,1\}^n} \frac{1}{\sqrt{2^n}} \left|D \cup (x, \delta)\right\rangle - \left|D\right\rangle \right)
\end{aligned} \tag{16}$$

   holds, where $\left|D^{\mathsf{invalid}}_{\gamma}\right\rangle$ is a superposition of invalid databases for each γ, and $\left|\widehat{0^n}\right\rangle = H^{\otimes n} \left|0^n\right\rangle$.

3. *Suppose that there is no entry of x in D. Then, for any y,*

$$\mathsf{RstOE}\,|x\rangle\,|y\rangle \otimes |D\rangle = \sum_{\alpha\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}\,|x\rangle\,|y\oplus\alpha\rangle \otimes |D\cup(x,\alpha)\rangle$$

with an error in $O(1/\sqrt{2^n})$. To be more precise,

$$\begin{aligned}\mathsf{RstOE}\,|x\rangle\,|y\rangle \otimes |D\rangle = &\sum_{\alpha\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}\,|x, y\oplus\alpha\rangle \otimes |D\cup(x,\alpha)\rangle \\ &+ \frac{1}{\sqrt{2^n}}\,|x\rangle\,|\widehat{0^n}\rangle \otimes \left(|D\rangle - \sum_{\gamma\in\{0,1\}^n} \frac{1}{\sqrt{2^n}}\,|D\cup(x,\gamma)\rangle\right)\end{aligned} \tag{17}$$

holds, where $|\widehat{0^n}\rangle = H^{\otimes n}\,|0^n\rangle$.

Proposition 1 can be shown by straightforward calculations. For completeness, a proof of Proposition 1 is given in Section A in this paper's full version [14].

An intuitive interpretation of Proposition 1. The first property is a subsidiary one, which will be useful in later applications. When we ignore error terms, the second and third properties correspond to the first and second procedures of CstO, respectively: When an adversary makes a query x to the oracle, RstOE looks for a tuple (x,α) in the database. If one is found, respond with α (the second property in the above proposition). If no tuple is found, create the superposition $\frac{1}{\sqrt{2^n}}\sum_{\alpha_x}|\alpha_x\rangle$, respond with α_x, and add (x,α_x) to the database (the third property in the above proposition).

Note that we do not need any "test and forget" step to describe the second and third properties in Proposition 1. Thus we can intuitively capture time evolutions of databases with only the (classical) lazy-sampling-like arguments.

To get rid of the "test and forget" step, we have to introduce some errors. The error increases as the number of adversaries' queries q increases, but it remains negligible as long as $q \ll 2^{n/2}$. Thus the error will not be problematic when we focus on the situation $q \ll 2^{n/2}$, which is the case for showing the security bound of the 4-round Luby-Rackoff construction.

In later applications, similarly to classical proofs, we introduce *good* and *bad* transcripts. The explicit formulas of the second and third properties will be used to show that, intuitively, adversaries cannot distinguish two oracles if transcripts are "good". Moreover, the first property and the descriptions with errors of the second and third properties will be used to show that the probability that transcripts become "bad" is negligible.

4 Security Proofs

The goal of this section is to show the following theorem, which gives the quantum query lower bound for the problem of distinguishing the 4-round Luby-

Rackoff construction LR_4 from random permutations RP, when all round functions are truly random functions.

Theorem 3. *Let q be a positive integer. Let $\mathcal{A}$ be an adversary that makes at most q quantum queries. Then, $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathsf{LR}_4}(\mathcal{A})$ is in $O\left(\sqrt{q^6/2^{n/2}}\right)$.*

Since we can efficiently simulate truly random functions against efficient quantum algorithms [34], the following corollary follows from Theorem 3.

Corollary 1. *Let f_i be a quantumly secure PRF for each $1 \leq i \leq 4$. Then, the 4-round Luby-Rackoff construction $\mathsf{LR}_4(f_1, f_2, f_3, f_4)$ is a quantumly secure PRP.*

In the rest of this section, we assume that all round functions in the Luby-Rackoff constructions are truly random functions, and we focus on the number of queries when we consider computational resources of adversaries. To have a good intuition on our proof in the quantum setting, it would be better to intuitively capture how LR_3 is proven to be secure against classical CPAs, how the quantum attack on LR_3 works, and what problem will be hard even for quantum adversaries. Thus, before giving a formal proof for the above theorem, in what follows we give some observations about these questions, and then explain where to start.

An overview of a classical security proof for LR_3. Here we give an overview of a *classical* proof for the security of LR_3 against chosen plaintext attacks in the classical setting. For simplicity, we consider a proof for PRF security of LR_3.

Let bad_2 be the event that an adversary makes two distinct plaintext queries $(x_{0L}, x_{0R}) \neq (x'_{0L}, x'_{0R})$ to the real oracle LR_3 such that the corresponding inputs x_{1L} and x'_{1L} to the second round function f_2 are equal, i.e., inputs to f_2 collide. In addition, let bad_3 be the event that inputs to f_3 collide, and define $\mathsf{bad} := \mathsf{bad}_2 \vee \mathsf{bad}_3$.

If bad_2 (resp., bad_3) does not occur, then the right-half (resp., left-half) $n/2$ bits of LR_3's outputs cannot be distinguished from truly random $n/2$-bit strings. Thus, unless the event bad occurs, adversaries cannot distinguish LR_3 from random functions.

If the number of queries of an adversary $\mathcal{A}$ is at most q, we can show that the probability that the event bad occurs when $\mathcal{A}$ runs relative to the oracle LR_3 is in $O(q^2/2^{n/2})$. Thus we can deduce that LR_3 is indistinguishable from a random function up to $O(2^{n/4})$ queries.

Quantum chosen plaintext attack on LR_3. Next, we give an overview of the quantum chosen plaintext attack on LR_3 by Kuwakado and Morii [20]. Note that we consider the setting in which adversaries can make quantum superposition queries. The attack distinguishes LR_3 from a random permutation with only $O(n)$ queries.

Fix $\alpha_0 \neq \alpha_1 \in \{0,1\}^{n/2}$ and for $i = 0, 1$, define $g_i : \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ by $g_i(x) = (\mathsf{LR}_3(\alpha_i, x))_R \oplus \alpha_i$, where $(\mathsf{LR}_3(\alpha_i, x))_R$ denote the right half $n/2$-bits of $\mathsf{LR}_3(\alpha_i, x)$. In addition, define $G : \{0,1\} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ by $G(b, x) = g_b(x)$. Then, $g_0(x) = g_1(x \oplus s)$ can be easily confirmed to hold for any $x \in \{0,1\}^{n/2}$, where $s = f_1(\alpha_0) \oplus f_1(\alpha_1)$. Thus $G(b, x) = G((b, x) \oplus (1, s))$ holds for any b and x, i.e., the function G has the period $(1, s)$.

If we can make quantum superposed queries to G, then we can find the period $(1, s)$ by using Simon's period finding algorithm [31], making $O(n)$ queries to G. In fact G can be implemented on an oracle-querying quantum circuit $\mathcal{C}^{\mathsf{LR}_3}$ by making $O(1)$ queries to LR_3. [8]

Roughly speaking, Simon's algorithm outputs the periods with a high probability by making $O(n)$ queries if applied to periodic functions, and outputs the result that "this function is not periodic" if applied to functions without periods.

If we are given the oracle of a random permutation RP, the circuit $\mathcal{C}^{\mathsf{RP}}$ will implement an almost random function, which does not have any period with a high probability. Thus, if we run Simon's algorithm on $\mathcal{C}^{\mathsf{RP}}$, with a high probability, it does not output any period. Therefore, we can distinguish LR_3 from RP by checking if Simon's period finding algorithm outputs a period.

Observation: Why the classical proof does not work? Here we give an observation about why quantum adversaries can distinguish LR_3 from random permutations even though LR_3 is proven to be indistinguishable from a random permutation in the classical setting.

We observe that quantum adversaries can make the event bad_2 occur: Once we find the period $1 \| s = 1 \| f_1(\alpha_0) \oplus f_2(\alpha_1)$ given the real oracle LR_3, we can force collisions on the input of f_2. Concretely, take $x \in \{0,1\}^{n/2}$ arbitrarily and set $(x_{0L}, x_{0R}) := (\alpha_0, x)$, $(x'_{0L}, x'_{0R}) := (\alpha_1, x \oplus s)$. Then the corresponding inputs to f_2 become $f_1(\alpha_0) \oplus x$ for both plaintexts. Thus the classical proof idea does not work in the quantum setting.

Quantum security proof for LR_4: The idea. As we explained above, the essence of the quantum attack on LR_3 is finding collisions for inputs to the second round function f_2. On the other hand, finding collisions for inputs to the third round function f_3 seems difficult even for quantum (chosen-plaintext) query adversaries.

Having these observations, our idea is that even quantum adversaries would have difficulty in noticing that the third state update $(x_{2L}, x_{2R}) \mapsto (x_{2R} \oplus f_3(x_{2L}), x_{2L})$ of LR_3 is modified as $(x_{2L}, x_{2R}) \mapsto (F(x_{2L}, x_{2R}), x_{2L})$, where $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ is a random function. We denote this modified function by LR'_3 (see Fig. 3) and begin by showing that it is hard to distinguish LR'_3 from LR_3.

[8] Here we have to truncate outputs of $\mathcal{O}$ without destroying quantum states, which is pointed out to be non-trivial in the quantum setting [18]. However, this "truncation" issue can be overcome by using a technique described in [15].

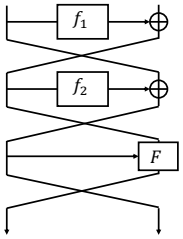

Fig. 3. LR'_3

We will show this by combining the classical proof idea and our recording standard oracle with errors. Roughly speaking, we define "bad" databases as the ones that contain "collisions at left-half inputs to the third round function". Then we show that the probability that we measure bad databases is very small, and that adversaries cannot distinguish LR'_3 from LR_3 when databases are not bad.

Next, let $\mathsf{FamP}(\{0,1\}^{n/2})$ be the set of functions $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ such that $F(x,\cdot)$ is a permutation for each x. If P is chosen uniformly at random from $\mathsf{FamP}(\{0,1\}^{n/2})$, we say that P is a *family of random permutations* (FRP). Then, we intuitively see that FRP is hard to distinguish from a random function RF from $\{0,1\}^n$ to $\{0,1\}^{n/2}$.

Once we show the above two properties, i.e.,

1. LR'_3 is hard to distinguish from LR_3, and
2. FRP is hard to distinguish from RF,

we can prove Theorem 3 with simple and easy arguments. In other words, those two properties are technically the most difficult parts to show in our proof for Theorem 3. To show the first property, we use our recording standard oracle with errors. On the other hand, to show the second property, we can just combine some previous results.

Organization of the rest of Section 4. Section 4.1 shows that LR'_3 is hard to distinguish from LR_3. Section 4.2 shows that FRP is hard to distinguish from RF. Section 4.3 proves Theorem 3 by combining the results in Sections 4.1 and 4.2.

4.1 Hardness of Distinguishing LR'_3 from LR_3

Here we show the following proposition.

Proposition 2. *Let q be a positive integer. Let $\mathcal{A}$ be an adversary that makes at most q quantum queries. Then, $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3,\mathsf{LR}'_3}(\mathcal{A})$ is in $O\left(\sqrt{q^3/2^{n/2}}\right)$.*

First, let us discuss the behavior of the quantum oracles of LR_3 and LR'_3.

Quantum oracle of LR_3. Let O_{f_i} denote the quantum oracle of each round function f_i. In addition, let us define the unitary operator $O_{\mathrm{UP}.i}$ that computes the state update of the i-th round by

$$\begin{aligned} O_{\mathrm{UP}.i} : |x_{(i-1)L}, x_{(i-1)R}\rangle\,|y_L, y_R\rangle \\ \mapsto |x_{(i-1)L}, x_{(i-1)R}\rangle\,|(y_L, y_R) \oplus (f_i(x_{(i-1)L}) \oplus x_{(i-1)R}, x_{(i-1)L})\rangle\,. \end{aligned}$$

$O_{\mathrm{UP}.i}$ can be implemented by making one query to f_i (see Fig. 4).

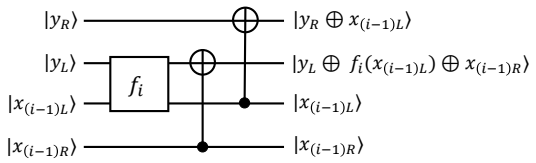

Fig. 4. Implementation of $O_{\mathrm{UP}.i}$. f_i will be implemented by using the recording standard oracle with errors.

Now O_{LR_3} can be implemented as follows by using $\{O_{\mathrm{UP}.i}\}_{1\le i\le 3}$:

1. Take $|x\rangle\,|y\rangle = |x_{0L}, x_{0R}\rangle\,|y_L, y_R\rangle$ as an input.
2. Compute the state (x_{1L}, x_{1R}) by querying $|x_{0L}, x_{0R}\rangle\,|0^n\rangle$ to $O_{\mathrm{Up}.1}$, and obtain
$$|x_{0L}, x_{0R}\rangle\,|y_L, y_R\rangle \otimes |x_{1L}, x_{1R}\rangle\,. \tag{18}$$
3. Compute the state (x_{2L}, x_{2R}) by querying $|x_{1L}, x_{1R}\rangle\,|0^n\rangle$ to $O_{\mathrm{Up}.2}$, and obtain
$$|x_{0L}, x_{0R}\rangle\,|y_L, y_R\rangle \otimes |x_{1L}, x_{1R}\rangle \otimes |x_{2L}, x_{2R}\rangle\,. \tag{19}$$
4. Query $|x_{2L}, x_{2R}\rangle\,|y_L, y_R\rangle$ to $O_{\mathrm{Up}.3}$, and obtain
$$|x\rangle\,|y \oplus \mathsf{LR}_3(x)\rangle \otimes |x_{1L}, x_{1R}\rangle \otimes |x_{2L}, x_{2R}\rangle\,. \tag{20}$$
5. Uncompute Steps 2 and 3 to obtain
$$|x\rangle\,|y \oplus \mathsf{LR}_3(x)\rangle\,. \tag{21}$$
6. Return $|x\rangle\,|y \oplus \mathsf{LR}_3(x)\rangle$.

The above implementation is illustrated in Fig. 5.

Quantum oracle of LR'_3. The quantum oracle of LR'_3 is implemented in the same way as LR_3, except that the third round state update oracle $O_{\mathrm{UP}.3}$ is replaced with another oracle $O'_{\mathrm{UP}.3}$ defined as

$$O'_{\mathrm{UP}.3} : |x_{2L}, x_{2R}\rangle\,|y_L, y_R\rangle \mapsto |x_{2L}, x_{2R}\rangle\,|(y_L, y_R) \oplus (F(x_{2L}, x_{2R}) \oplus x_{2R}, x_{2L})\rangle\,.$$

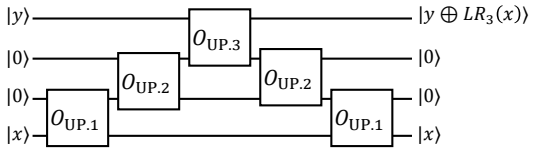

Fig. 5. Implementation of LR_3.

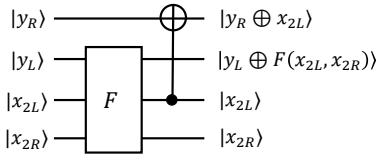

Fig. 6. Implementation of $O'_{\mathrm{UP.3}}$. F will be implemented by using the recording standard oracle with errors.

$O'_{\mathrm{UP.3}}$ is implemented by making one query to O_F, i.e., the quantum oracle of F (see Fig. 6).

Below, we show the claim of the proposition by using the recording standard oracle with errors for f_1, f_2, f_3, and F. We consider that the oracles of these functions are implemented as the recording standard oracle with errors, and we use D_1, D_2, D_3, and D_F to denote (valid) databases for f_1, f_2, f_3, and F, respectively. In particular, after the i-th query of an adversary to LR_3, the joint quantum states of the adversary and functions can be described as

$$\sum_{xyzD_1D_2D_3} a_{xyzD_1D_2D_3} |xyz\rangle \otimes |D_1\rangle |D_2\rangle |D_3\rangle \tag{22}$$

for some complex numbers $a_{xyzD_1D_2D_3}$ such that $\sum_{xyzD_1D_2D_3} |a_{xyzD_1D_2D_3}|^2 = 1$. Here, x, y, and z correspond to the adversary's query, answer, and output registers, respectively. (If the oracle is LR'_3, then the register $|D_3\rangle$, which corresponds to f_3, is replaced with $|D_F\rangle$, which corresponds to F.)

Next, we define good and bad databases for LR_3 and LR'_3. Intuitively, we say that a tuple (D_1, D_2, D_3) (resp., (D_1, D_2, D_F)) for LR_3 (resp., LR'_3) is bad if and only if it contains the information that some inputs to f_3 (resp., the left halves of some inputs to F) collide. Roughly speaking, we define good and bad databases in such a way that a one-to-one correspondence exists between good databases for LR_3 and those for LR'_3, so that adversaries will not be able to distinguish LR'_3 from LR_3 as long as databases are good.

Good and bad databases for LR_3. Here we introduce the notion of *good* and *bad* for each tuple (D_1, D_2, D_3) of valid database for LR_3. We say that

(D_1, D_2, D_3) is good if, for each entry $(x_{2L}, \gamma) \in D_3$, there exists exactly one pair $((x_{0L}, \alpha), (x_{1L}, \beta)) \in D_1 \times D_2$ such that $\beta \oplus x_{0L} = x_{2L}$. We say that (D_1, D_2, D_3) is bad if it is not good.

Good and bad databases for LR'_3. Next we introduce the notion of *good* and *bad* for each tuple (D_1, D_2, D_F) of valid database for LR'_3. We say that a valid database D_F is *without overlap* if each pair of distinct entries (x_{2L}, x_{2R}, γ) and $(x'_{2L}, x'_{2R}, \gamma')$ in D_F satisfies $x_{2L} \neq x'_{2L}$. We say that (D_1, D_2, D_F) is good if D_F is without overlap, and for each entry $(x_{2L}, x_{2R}, \gamma) \in D_F$, there exists exactly one pair $((x_{0L}, \alpha), (x_{1L}, \beta)) \in D_1 \times D_2$ such that $\beta \oplus x_{0L} = x_{2L}$ and $x_{2R} = x_{1L}$. We say that (D_1, D_2, D_F) is bad if it is not good.

Compatibility of D_F with D_3. Let D_F be a valid database for F without overlap and D_3 be a valid database for f_3. We say that D_F is compatible with D_3 if the following conditions are satisfied:

1. If $(x_{2L}, x_{2R}, \gamma) \in D_F$, then $(x_{2L}, x_{2R} \oplus \gamma) \in D_3$.
2. If $(x_{2L}, \gamma) \in D_3$, there is a unique x_{2R} and $(x_{2L}, x_{2R}, x_{2R} \oplus \gamma) \in D_F$.

For each valid D_F without overlap, the unique valid database exists for f_3, which we denote by $[D_F]_3$.

Remark 1. For each good database (D_1, D_2, D_3) for LR_3, a unique D_F without overlap exists such that $[D_F]_3 = D_3$ and (D_1, D_2, D_F) is a good database for LR'_3, by the definition of good databases. Similarly, for each good database (D_1, D_2, D_F) for LR'_3, $(D_1, D_2, [D_F]_3)$ becomes a good database for LR_3.

Next we define regular and irregular quantum states for the oracles O_{LR_3} and $O_{\mathsf{LR}'_3}$. Roughly speaking, we will treat irregular states as some small error terms, and focus on regular states.

Regular and irregular states of oracles. Recall that, in addition to database registers, the quantum oracle O_{LR_3} uses ancillary $2n$-qubit registers to compute the intermediate state after the first and second rounds (see (19) and (20)). We say that a state vector $|D_1\rangle |D_2\rangle |D_3\rangle \otimes |x_1\rangle \otimes |x_2\rangle$ for O_{LR_3}, where $|x_1\rangle \otimes |x_2\rangle$ is the ancillary $2n$ qubits, is *irregular* if $x_1 \neq 0^n \vee x_2 \neq 0^n$ holds, or at least one of the three databases (D_1, D_2, or D_3) is invalid. We say that the state vector is *regular* if it is not irregular. We define regular and irregular states for $O_{\mathsf{LR}'_3}$ similarly.

Next we define some modified versions of LR_3 and LR'_3, which we denote by LR_3-det and LR'_3-det, respectively ("det" is an abbreviation of "detection of bad database").

The oracles LR_3-det and LR'_3-det. The oracle LR_3-det is defined in the same way as LR_3, except that the oracle checks whether the database is bad (or the state of the oracle is irregular) after each query, and writes the result to an additional qubit. Note that we define regular and irregular states for LR_3-det in the same way as for LR_3. Additional qubits are prepared before an adversary $\mathcal{A}$ runs (q additional qubits are sufficient if $\mathcal{A}$ is a q query adversary). If $i \neq j$, the results of "detection of bad database" for the i-th and j-th queries are written in distinct qubits.

Intuitively, LR_3-det behaves as follows when $\mathcal{A}$ makes the i-th query:

1. Check if the j-th additional qubit is 1 for $1 \leq j \leq i-1$ (i.e., check if the database has been bad before the i-th query). If so, do nothing. If not, go to the next step.
2. Make a query to O_{LR_3}.
3. Check if the database is bad, or the quantum state of O_{LR_3} is irregular. If so, flip the i-th additional qubit.

Next, we formally explain how the above procedures can be realized as a unitary operator. Let Π_{bad} be the projection to the space spanned by the vectors of *bad* databases, and irregular state vectors. In addition, let $\Pi^{[i-1]}_{\mathsf{flipped}}$ be the projection onto the space spanned by the vectors such that the j-th additional qubit is 1 for some $1 \leq j \leq i-1$.

Formally, for the i-th query, the behavior of the quantum oracle of LR_3-det is described by the unitary operator

$$O_{\mathsf{LR}_3\text{-det}} := \Big((\Pi_{\mathsf{bad}} \otimes I_{i-1} \otimes X + (I - \Pi_{\mathsf{bad}}) \otimes I_{i-1} \otimes I_1) \cdot (O_{\mathsf{LR}_3} \otimes I_{i-1} \otimes I_1)\Big) \cdot ((I - \Pi^{[i-1]}_{\mathsf{flipped}}) \otimes I_1) + \Pi^{[i-1]}_{\mathsf{flipped}} \otimes I_1, \quad (23)$$

where I_{i-1} is the identity operator which acts on the first $(i-1)$ additional qubits. In addition, I_1 and X are the identity operator and the operator such that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$, respectively, which act on the i-th additional qubit.

LR'_3-det is constructed from LR'_3 in the same way as LR_3-det is constructed from LR_3-det as above. The behaviors of the oracles of LR'_3-det and LR_3-det depend on i, though for simplicity, we always use the notations $O_{\mathsf{LR}'_3\text{-det}}$ and $O_{\mathsf{LR}_3\text{-det}}$ without i.

Below we first show that LR_3-det is hard to distinguish from LR'_3-det, and second show that LR_3-det (resp., LR'_3-det) is hard to distinguish from LR_3 (resp., LR'_3).

Hardness of distinguishing LR_3-det from LR'_3-det. Let $|\psi_i\rangle$ and $|\psi'_i\rangle$ be the state just before the i-th query to LR_3-det and LR'_3-det, respectively. By abuse of notation, we let $|\psi_{(q+1)}\rangle, |\psi'_{(q+1)}\rangle$ denote the quantum states $(U_q \otimes I) O_{\mathsf{LR}_3\text{-det}} |\psi_q\rangle$ and $(U_q \otimes I) O_{\mathsf{LR}'_3\text{-det}} |\psi'_q\rangle$, respectively.

We need the following lemma. Intuitively, the lemma claims that no adversary can distinguish LR_3-det from LR'_3-det if databases are "good".

Lemma 1. *For each j, let $|\psi_j^{\mathsf{good}}\rangle$ and $|\psi_j^{'\mathsf{good}}\rangle$ denote $(I - \Pi_{\mathsf{flipped}}^{[i-1]})|\psi_j\rangle$ and $(I - \Pi_{\mathsf{flipped}}^{[i-1]})|\psi'_j\rangle$, respectively. Let $\mathsf{tr}_{\mathcal{D}_{123}}$ and $\mathsf{tr}_{\mathcal{D}_{12F}}$ denote the partial trace over databases and additional qubits for LR_3-det and LR'_3-det, respectively. Then, $\mathsf{tr}_{\mathcal{D}_{123}}\left(|\psi_i^{\mathsf{good}}\rangle\right) = \mathsf{tr}_{\mathcal{D}_{12F}}\left(|\psi_i^{'\mathsf{good}}\rangle\right)$ holds for $1 \leq i \leq q+1$.*

Proof intuition. Lemma 1 can be shown by straightforward algebraic calculations using the strict formulas of the second and third properties in Proposition 1. The equality holds owing to the one-to-one correspondences between good databases for LR_3 and those for LR'_3 (see Remark 1). More precisely, for every $x, y, x', y' \in \{0,1\}^n$ and for every good databases $(D_1, D_2, D_F), (D'_1, D'_2, D'_F)$ for LR'_3, the "probability" (in the quantum meaning) that

$$O_{\mathsf{LR}'_3} \text{ changes the vector } |x,y\rangle|D_1,D_2,D_F\rangle \text{ to } |x',y'\rangle|D'_1,D'_2,D'_F\rangle \tag{24}$$

is equal to the probability that

$$O_{\mathsf{LR}_3} \text{ changes the vector } |x,y\rangle|D_1,D_2,[D_F]_3\rangle \text{ to } |x',y'\rangle|D'_1,D'_2,[D'_F]_3\rangle, \tag{25}$$

where $(D_1, D_2, [D_F]_3)$ and $(D'_1, D'_2, [D'_F]_3)$ are the good databases for LR_3 that correspond to (D_1, D_2, D_F) and (D'_1, D'_2, D'_F), respectively. By linearity of unitary operations, this equality shows that if $\mathsf{tr}_{\mathcal{D}_{123}}\left(|\psi_j^{\mathsf{good}}\rangle\right) = \mathsf{tr}_{\mathcal{D}_{12F}}\left(|\psi_j^{'\mathsf{good}}\rangle\right)$ (i.e., the good probabilities before the j-th queries are equal) then $\mathsf{tr}_{\mathcal{D}_{123}}\left(|\psi_{j+1}^{\mathsf{good}}\rangle\right) = \mathsf{tr}_{\mathcal{D}_{12F}}\left(|\psi_{j+1}^{'\mathsf{good}}\rangle\right)$ (i.e., the good probabilities are still equal after the j-th queries) holds. A complete proof of Lemma 1 is given in Section B in this paper's full version [14].

We also need the following lemma, which intuitively claims that "good" states change to "bad" states only with a negligible probability.

Lemma 2. *For each j, $\left\|\Pi_{\mathsf{bad}} \cdot O_{\mathsf{LR}_3}|\psi_j^{\mathsf{good}}\rangle\right\|$ and $\left\|\Pi_{\mathsf{bad}} \cdot O_{\mathsf{LR}'_3}|\psi_j^{'\mathsf{good}}\rangle\right\|$ are in $O(\sqrt{j/2^{n/2}})$.*

Proof intuition. Here we give a proof intuition for LR_3. Owing to the second and third properties of Proposition 1 with errors, we can use classical lazy-sampling intuition (see explanations below Proposition 1). Roughly speaking, good databases change to bad if and only if a fresh query is made to f_1 or f_2, and the corresponding input to f_3 collides with some existing record in the database for f_3.

Since each database of $|\psi_j^{\mathsf{good}}\rangle$ has at most $(j-1)$ entries and outputs of f_1 and f_2 are $(n/2)$-bits, the input to f_3 that corresponds to a fresh input to f_1 or f_2 collides with one of the existing records in D_3 with a probability at most $O(j/2^{n/2})$. This corresponds to the claim that $\left\|\Pi_{\mathsf{bad}} \cdot O_{\mathsf{LR}_3}|\psi_j^{\mathsf{good}}\rangle\right\|^2 \leq O(j/2^{n/2})$ holds. This argument actually ignores some errors, but the errors will be in $O(\sqrt{1/2^{n/2}})$ due to Proposition 1. The claim for LR'_3 can be shown

similarly. A complete proof of Lemma 2 is given in Section C in this paper's full version [14].

The following proposition guarantees that LR_3-det is hard to distinguish from LR'_3-det.

Proposition 3. $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3\text{-det},\mathsf{LR}'_3\text{-det}}(\mathcal{A})$ *is in* $O\left(\sqrt{q^3/2^{n/2}}\right)$.

Proof intuition. Due to Lemma 1, $\mathcal{A}$ cannot distinguish LR_3-det from LR'_3-det as long as databases are good. Thus, intuitively, the distinguishing advantage is upper bounded by the square root of the probability that databases become bad while $\mathcal{A}$ makes q queries, which is further upper bounded by $\sum_{1\le j\le q}\|\Pi_{\mathsf{bad}}\cdot O_{\mathsf{LR}_3\text{-det}}|\psi_j^{\mathsf{good}}\rangle\|+\sum_{1\le j\le q}\|\Pi_{\mathsf{bad}}\cdot O_{\mathsf{LR}'_3\text{-det}}|\psi_j^{'\mathsf{good}}\rangle\|$. From Lemma 2, this can be upper bounded by $\sum_{1\le j\le q}O(\sqrt{j/2^{n/2}})+\sum_{1\le j\le q}O(\sqrt{j/2^{n/2}})=O(\sqrt{q^3/2^{n/2}})$. A complete proof of Proposition 3 is given in Section D in this paper's full version [14].

Hardness of distinguishing LR_3-det and LR'_3-det from LR_3 and LR'_3. The following proposition guarantees that LR_3-det and LR'_3-det are hard to distinguish from LR_3 and LR'_3, respectively.

Proposition 4. $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3,\mathsf{LR}_3\text{-det}}(\mathcal{A})$ *and* $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}'_3,\mathsf{LR}'_3\text{-det}}(\mathcal{A})$ *are in* $O\left(\sqrt{q^3/2^{n/2}}\right)$.

Proof intuition. We give a proof intuition for LR_3-det and LR_3. Since the databases of round functions for LR_3-det are the same as those for LR_3, $\mathcal{A}$ cannot distinguish LR_3-det from LR'_3-det as long as databases are good. Thus, roughly speaking, the distinguishing advantage is upper bounded by the square root of the probability that databases become bad while $\mathcal{A}$ makes q queries. Owing to Lemma 2, we can show the claim in the same way as the proof intuition for Proposition 3. The claim for LR'_3-det and LR'_3 can be shown in a similar way. A complete proof of Proposition 4 is given in Section E in this paper's full version [14].

Proof of Proposition 2. Finally, we show Proposition 2.

Proof (of Proposition 2). $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3,\mathsf{LR}'_3}(\mathcal{A})$ is upper bounded by $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3,\mathsf{LR}_3\text{-det}}(\mathcal{A})$ + $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_3\text{-det},\mathsf{LR}'_3\text{-det}}(\mathcal{A})$ + $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}'_3\text{-det},\mathsf{LR}'_3}(\mathcal{A})$. Thus, the claim of Proposition 2 follows from Proposition 3 and Proposition 4. □

4.2 Hardness of Distinguishing FRP from RF

Recall that $\mathsf{FamP}(\{0,1\}^{n/2})$ is the set of functions $F:\{0,1\}^{n/2}\times\{0,1\}^{n/2}\to\{0,1\}^{n/2}$ such that $F(x,\cdot)$ is a permutation for each x, and if P is chosen uniformly at random from $\mathsf{FamP}(\{0,1\}^{n/2})$, we say that P is a *family of random permutations* (FRP). The following proposition claims that FRP is hard to distinguish from RF.

Proposition 5. *For any quantum adversary $\mathcal{A}$ that makes at most q quantum queries,* $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{FRP},\mathsf{RF}}(\mathcal{A})\le O\left(\sqrt{q^6/2^{n/2}}\right)$ *holds.*

Proof intuition. This proposition can be proven by just combining the two previous results: The first one is the indistinguishability of a random function and a random permutation shown by Zhandry [35], and the second one is the equivalence of oracle-indistinguishability and indistinguishability, which was first shown by Zhandry [33] and later generalized by Song and Yun [32]. If a function $F : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ is a random function RF (resp., a family of random permutations FRP), $F(x,\cdot)$ is a random function (resp., a random permutation) for each $x \in \{0,1\}^{n/2}$. Roughly speaking, F can be regarded as an "oracle" that returns a random function (resp., random permutation) for each x. Then, from the equivalence of indistinguishability and oracle-indistinguishability, indistinguishability of RF and FRP (which is, intuitively, "oracle"-indistinguishability of a random function and a random permutation) follows from the indistinguishability of a random function and a random permutation from $\{0,1\}^{n/2}$ to $\{0,1\}^{n/2}$, which is already shown as the first result above. See Section F in this paper's full version [14] for a formal proof.

4.3 Proof of Theorem 3

This subsection finishes our proof of Theorem 3, by using the results given in Sections 4.1 and 4.2.

Proof (of Theorem 3). First, let us modify LR_4 in such a way that the state updates of the third and fourth rounds are replaced with $(x_{2L}, x_{2R}) \mapsto (x_{3L}, x_{3R}) := (F(x_{2L}, x_{2R}), x_{2L})$ and $(x_{3L}, x_{3R}) \mapsto (x_{4L}, x_{4R}) := (F'(x_{3L}, x_{3R}), x_{3L})$, respectively, where $F, F' : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ are random functions. Let us denote the modified function by LR''_4. In addition, by $\mathsf{LR}''_2(F, F')$ we denote the function defined by $(x_L, x_R) \mapsto (F'(F(x_L, x_R), x_L), F(x_L, x_R))$ (see Fig. 7).

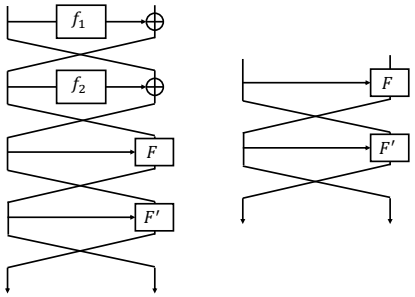

Fig. 7. LR''_4 and $\mathsf{LR}''_2(F, F')$.

Then, by applying Proposition 2 twice we can show that

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_4, \mathsf{LR}''_4}(q) \leq O\left(\sqrt{\frac{q^3}{2^{n/2}}}\right) \tag{26}$$

holds.

Let us modify $\mathsf{LR}_2''(F, F')$ in such a way that F is replaced with a family of random permutations P, and denote the resulting function by $\mathsf{LR}_2''(P, F')$. Then, from Proposition 5 it follows that $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_2''(F,F'),\mathsf{LR}_2''(P,F')}(q) \leq O(\sqrt{q^6/2^{n/2}})$ holds. Next, let us define a function G by $G(x_L, x_R) = F'(x_L, x_R) \| P(x_L, x_R)$, where F' is a random function and P is a family of random permutations (see Fig. 8). Then, the function distribution of $\mathsf{LR}_2''(P, F')$ is the same as that of G.

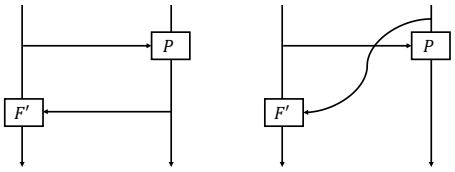

Fig. 8. $\mathsf{LR}_2''(P, F')$ and G.

(Note that $P(x_L, x_R) \neq P(x_L, x'_R)$ always holds if $x_R \neq x'_R$. Thus, if $(x_L, x_R) \neq (x'_L, x'_R)$, the corresponding inputs to F' will be distinct.) Therefore we have that $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_2''(P,F'),G}(q) = 0$ holds. Moreover, from Proposition 5 $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{RF},G}(q) \leq O\left(\sqrt{q^6/2^{n/2}}\right)$ holds. Therefore $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_2''(P,F'),\mathsf{RF}}(q) \leq O\left(\sqrt{q^6/2^{n/2}}\right)$ follows, which implies that

$$\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_4'',\mathsf{RF}}(q) \leq O\left(\sqrt{\frac{q^6}{2^{n/2}}}\right) \tag{27}$$

holds.

Hence, from (26) and (27), it follows that $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_4,\mathsf{RF}}(\mathcal{A}) \leq O\left(\sqrt{q^6/2^{n/2}}\right)$ holds for any quantum adversary $\mathcal{A}$ that makes at most q quantum queries. In addition, $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{RP},\mathsf{RF}}(\mathcal{A}) \leq O(q^6/2^n)$ follows from a quantum version of the PRP-PRF switch [35]. (See Proposition 7 in this paper's full version [14] for details.) Therefore $\mathbf{Adv}^{\mathrm{dist}}_{\mathsf{LR}_4,\mathsf{RP}}(\mathcal{A}) \leq O\left(q^6/2^{n/2}\right)$ follows for any quantum adversary $\mathcal{A}$ that makes at most q quantum queries, which completes the proof of the theorem. □

Remark 2. In the above proof, we went back and forth between random functions and (families of) random permutations, which may seem unnatural. The motivation for our proof strategy was to avoid complex arguments that are specific to the quantum setting as much as possible.

5 A Query Upper Bound

Here we give a query upper bound for the problem of distinguishing LR_4 from a random permutation by showing a distinguishing attack. Again, we consider

the case that all round functions of LR_4 are truly random functions, and show the following theorem.

Theorem 4. *A quantum algorithm $\mathcal{A}$ exists that makes $O(2^{n/6})$ quantum queries and satisfies $\mathbf{Adv}^{\mathrm{qPRP}}_{\mathsf{LR}_4}(\mathcal{A}) = \Omega(1)$.*

Proof intuition. Intuitively, our distinguishing attack is just a quantum version of a classical collision-finding-based distinguishing attack [28]. A classical attack distinguishes LR_4 from a random permutation by finding a collision of a function that takes values in $\{0,1\}^{n/2}$, which requires $O(\sqrt{2^{n/2}}) = O(2^{n/4})$ queries in the quantum setting. However, finding a collision of the function requires only $O(\sqrt[3]{2^{n/2}}) = O(2^{n/6})$ queries in the quantum setting, which enables us to make a $O(2^{n/6})$-query quantum distinguisher. (Note that, we can generally find a collision of random functions from $\{0,1\}^{n/2}$ to $\{0,1\}^{n/2}$ with $O(\sqrt[3]{2^{n/2}}) = O(2^{n/6})$ quantum queries [35].) See Section G in this paper's full version [14] for a complete proof.

6 Concluding Remarks

This paper showed that $\Omega(2^{n/12})$ quantum queries are required to distinguish the (n-bit block) 4-round Luby-Rackoff construction from a random permutation by qCPAs. In particular, the 4-round Luby-Rackoff construction becomes a quantumly secure PRP against qCPAs if round functions are quantumly secure PRFs. We also gave a qCPA that distinguishes the 4-round Luby-Rackoff construction from a random permutation with $O(2^{n/6})$ quantum queries. To give security proofs, we gave an alternative formalization of the compressed oracle technique by Zhandry and applied it.

An important future work is to give the tight bound for the problem of distinguishing the 4-round Luby-Rackoff construction from a random permutation. [9] It would be interesting to see if the provable security bound improves when we increase the number of rounds. Also, analyzing the security of the Luby-Rackoff constructions against *qCCAs* is left as an interesting open question. It would be a challenging problem since we have to treat inverse (decryption) queries to quantum oracles. Oracles that allow inverse quantum queries are usually much harder to deal with than the ones that allow only forward quantum queries, and some other new techniques would be required for the analysis.

Acknowledgments

The authors thank Qipeng Liu and anonymous reviewers for pointing out an issue of Proposition 5 in a previous version of this paper.

[9] See Section H in this paper's full version [14] for the reason that closing the gap is important.

References

1. Alagic, G., Russell, A.: Quantum-secure symmetric-key cryptography based on hidden shifts. In: EUROCRYPT 2017, Proceedings, Part III. LNCS, vol. 11693, pp. 65–93. Springer (2017)
2. Anand, M.V., Targhi, E.E., Tabia, G.N., Unruh, D.: Post-quantum security of the CBC, CFB, OFB, CTR, and XTS modes of operation. In: PQCrypto 2016, Proceedings. LNCS, vol. 11505, pp. 44–63. Springer (2016)
3. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In: SAC 2000, Proceedings. LNCS, vol. 2012, pp. 39–56. Springer (2000)
4. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT 2011, Proceedings. LNCS, vol. 7073, pp. 41–69. Springer (2011)
5. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: EUROCRYPT 2013, Proceedings. LNCS, vol. 7881, pp. 592–608. Springer (2013)
6. Bonnetain, X.: Quantum key-recovery on full AEZ. In: SAC 2017, Proceedings. LNCS, vol. 10719, pp. 394–406. Springer (2017)
7. Bonnetain, X., Naya-Plasencia, M.: Hidden shift quantum cryptanalysis and implications. In: ASIACRYPT 2018, Proceedings, Part I. LNCS, vol. 11272, pp. 560–592. Springer (2018)
8. Bonnetain, X., Naya-Plasencia, M., Schrottenloher, A.: On quantum slide attacks, Appeared at SAC 2019
9. Czajkowski, J., Bruinderink, L.G., Hülsing, A., Schaffner, C., Unruh, D.: Post-quantum security of the sponge construction. In: PQCrypto 2018, Proceedings. LNCS, vol. 11505, pp. 185–204. Springer (2018)
10. Czajkowski, J., Majenz, C., Schaffner, C., Zur, S.: Quantum lazy sampling and game-playing proofs for quantum indifferentiability. IACR Cryptology ePrint Archive **2019**, 428 (2019)
11. Dong, X., Dong, B., Wang, X.: Quantum attacks on some Feistel block ciphers. IACR Cryptology ePrint Archive **2018**, 504 (2018)
12. Dong, X., Li, Z., Wang, X.: Quantum cryptanalysis on some generalized feistel schemes. SCIENCE CHINA Information Sciences **62**(2), 22501:1–22501:12 (2019)
13. Dong, X., Wang, X.: Quantum key-recovery attack on Feistel structures. SCIENCE CHINA Information Sciences **61**(10), 102501:1–102501:7 (2018)
14. Hosoyamada, A., Iwata, T.: 4-Round Luby-Rackoff Construction is a qPRP. IACR Cryptology ePrint Archive **2019**, 243 (2019)
15. Hosoyamada, A., Sasaki, Y.: Quantum Demiric-Selçuk meet-in-the-middle attacks: Applications to 6-round generic Feistel constructions. In: SCN 2018, Proceedings. LNCS, vol. 11035, pp. 386–403. Springer (2018)
16. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: ASIACRYPT 2018, Proceedings, Part I. LNCS, vol. 11272, pp. 275–304. Springer (2018)
17. Ito, G., Hosoyamada, A., Matsumoto, R., Sasaki, Y., Iwata, T.: Quantum chosen-ciphertext attacks against feistel ciphers. In: CT-RSA 2019, Proceedings. LNCS, vol. 11405, pp. 391–411. Springer (2019)
18. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: CRYPTO 2016, Proceedings, Part II. LNCS, vol. 11693, pp. 207–237. Springer (2016)

19. Kitaev, A.Y., Shen, A.H., Vyalyi, M.N.: Classical and Quantum Computation. American Mathematical Society, Boston, MA, USA (2002)
20. Kuwakado, H., Morii, M.: Quantum distinguisher between the 3-round Feistel cipher and the random permutation. In: ISIT 2010, Proceedings. pp. 2682–2685. IEEE (2010)
21. Kuwakado, H., Morii, M.: Security on the quantum-type Even-Mansour cipher. In: ISITA 2012, Proceedings. pp. 312–316. IEEE (2012)
22. Liu, Q., Zhandry, M.: On finding quantum multi-collisions. In: EUROCRYPT 2019, Proceedings, Part III. LNCS, vol. 11478, pp. 189–218. Springer (2019)
23. Luby, M., Rackoff, C.: How to construct pseudo-random permutations from pseudo-random functions (abstract). In: CRYPTO '85, Proceedings. LNCS, vol. 218, p. 447. Springer (1985)
24. Mennink, B., Szepieniec, A.: XOR of PRPs in a quantum world. In: PQCrypto 2017, Proceedings. LNCS, vol. 10346, pp. 367–383. Springer (2017)
25. National Bureau of Standards: Data encryption standard. FIPS 46 (January 1977)
26. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition (2010)
27. NIST: Announcing request for nominations for public-key post-quantum cryptographic algorithms. National Institute of Standards and Technology (2016)
28. Patarin, J.: New results on pseudorandom permutation generators based on the DES scheme. In: CRYPTO '91, Proceedings. LNCS, vol. 576, pp. 301–312. Springer (1991)
29. Santoli, T., Schaffner, C.: Using Simon's algorithm to attack symmetric-key cryptographic primitives. Quantum Information & Computation **17**(1&2), 65–78 (2017)
30. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: FOCS 1994, Proceedings. pp. 124–134. IEEE (1994)
31. Simon, D.R.: On the power of quantum computation. SIAM J. Comput. **26**(5), 1474–1483 (1997)
32. Song, F., Yun, A.: Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In: CRYPTO 2017, Proceedings, Part II. LNCS, vol. 10402, pp. 283–309. Springer (2017)
33. Zhandry, M.: How to construct quantum random functions. In: FOCS 2012, Proceedings. pp. 679–687. IEEE (2012)
34. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: CRYPTO 2012, Proceedings. LNCS, vol. 7417, pp. 758–775. Springer (2012)
35. Zhandry, M.: A note on the quantum collision and set equality problems. Quantum Information & Computation **15**(7&8), 557–567 (2015)
36. Zhandry, M.: A note on quantum-secure PRPs. IACR Cryptology ePrint Archive **2016**, 1076 (2016)
37. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: CRYPTO 2019, Proceedings, Part II. LNCS, vol. 11693, pp. 239–268. Springer (2019)