On Tight Quantum Security of HMAC and NMAC in the Quantum Random Oracle Model

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

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

Abstract. HMAC and NMAC are the most basic and important constructions to convert Merkle-Damgård hash functions into message authentication codes (MACs) or pseudorandom functions (PRFs). In the quantum setting, at CRYPTO 2017, Song and Yun showed that HMAC and NMAC are quantum pseudorandom functions (qPRFs) under the standard assumption that the underlying compression function is a qPRF. Their proof guarantees security up to $O(2^{n/5})$ or $O(2^{n/8})$ quantum queries when the output length of HMAC and NMAC is n bits. However, there is a gap between the provable security bound and a simple distinguishing attack that uses $O(2^{n/3})$ quantum queries. This paper settles the problem of closing the gap. We show that the tight bound of the number of quantum queries to distinguish HMAC or NMAC from a random function is $\Theta(2^{n/3})$ in the quantum random oracle model, where compression functions are modeled as quantum random oracles. To give the tight quantum bound, based on an alternative formalization of Zhandry's compressed oracle technique, we introduce a new proof technique focusing on the symmetry of quantum query records.

Keywords: symmetric-key cryptography · post-quantum cryptography · provable security · quantum security · compressed oracle technique · HMAC · NMAC.

1 Introduction

In recent years, post-quantum cryptography is one of the most active research areas in cryptography. NIST is holding the standardization process for post-quantum *public-key* schemes such as public-key encryption, key-establishment algorithms, and signatures [28], and it is anticipated that currently used public-key schemes (such as RSA-based schemes) will be replaced with post-quantum ones in a near future. In the post-quantum era, it is desirable that we have some mathematical evidence that *symmetric-key* schemes also have post-quantum security. Studying post-quantum security of typical symmetric-key schemes is also an interesting problem from the view point of cryptographic theories, and there have been a significant number of recent papers that focus on this topic [32,21,18,14].

There exist two post-quantum security notions for cryptographic schemes: *standard security* and *quantum security* [33]. If a scheme $\mathcal{S}$ is proven to be secure in the setting where adversaries have quantum computers but they make only *classical* queries to keyed oracles, $\mathcal{S}$ is said to have *standard security*. If $\mathcal{S}$ is proven to be secure even if adversaries are allowed to make quantum superposed queries to keyed oracles, $\mathcal{S}$ is said to have *quantum security*. Quantum security is the ultimate security since, if $\mathcal{S}$ has quantum security, $\mathcal{S}$ satisfies arbitrary intermediate security notions between standard security and quantum security[3].

Message authentication codes (MACs) are the most important symmetric-key schemes to achieve data integrity. Some of them including block cipher based MACs such as CBC-MAC [5,7,22] and PMAC [8] do not have quantum security, since there exist polynomial time attacks on them [23]. However, they have standard security since their classical security proofs remain valid if adversaries are allowed to make only classical queries to keyed oracles and the underlying block ciphers are post-quantum secure.

On the other hand, classical security proofs are not necessarily applicable to the (post-quantum) standard security for hash based MACs where the proofs use idealized models such as the random oracle model (when underlying hash functions are built on the Merkle-Damgård construction, e.g., SHA-2 [26]) or the ideal permutation model (when underlying hash functions are built on the sponge construction, e.g., SHA-3 [27]). Since adversaries can implement compression functions and permutations used in the hash functions on their own quantum computers to make quantum queries, the security of hash based MACs should be proven in the corresponding idealized quantum models such as the quantum random oracle model (QROM) [9] or quantum ideal permutation model [2,21].

The main focus of this paper is to study the tight quantum pseudorandom function security (qPRF security) of HMAC and its variant NMAC [4], which are the most basic and important constructions to convert Merkle-Damgård hash functions into pseudorandom functions (PRFs) or MACs, in the QROM where compression functions are modeled as quantum random oracles (QROs).

HMAC and NMAC. For a compression function $h : \{0,1\}^{m+n} \to \{0,1\}^n$, the Merkle-Damgård construction MD^h is defined as follows[4]: Let $IV \in \{0,1\}^n$ be a fixed public initialization vector. For each input message $M \in \{0,1\}^*$, the construction pads M (with a fixed padding function) and splits it into m-bit message blocks $M[1], \ldots, M[\ell]$. The state is first set as $S_0 := IV$, and iteratively updated as $S_{i+1} := h(M[i+1]||S_i)$, and S_ℓ becomes the final output. We assume $m \geq n$, which is the case for usual concrete hash functions such as SHA-2.

For a key length $k \leq m$, HMAC is defined to be the keyed function $\mathsf{HMAC}^h : \{0,1\}^k \times \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ such that $\mathsf{HMAC}^h(K, IV, M) := \mathsf{MD}^h(IV,$

[3] Please do not confuse the notions of standard/quantum security with the standard model or the quantum random oracle model. The two notions are independent of the models, and it is possible that a scheme has quantum security in the standard model or standard security in the quantum random oracle model.

[4] n is the length of chaining values, and m is the length of message blocks.

$K_{out}||\mathsf{MD}^h(IV, K_{in}||M))$. Here, $K_{in} := (K||0^{m-k}) \oplus \mathsf{ipad}$, $K_{out} := (K||0^{m-k}) \oplus \mathsf{opad}$, and $\mathsf{ipad}, \mathsf{opad} \in \{0,1\}^m$ are fixed public constants such that $\mathsf{ipad} \neq \mathsf{opad}$. We sometimes write $\mathsf{HMAC}^h_K(IV, M)$ to denote $\mathsf{HMAC}^h(K, IV, M)$ for simplicity. See also Figure 1.

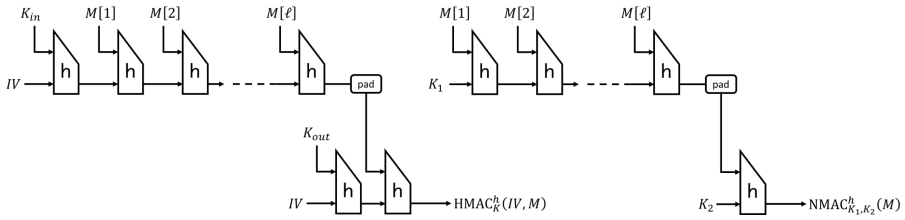


Fig. 1: HMAC and NMAC. Note that $\mathsf{pad}(M) = M[1]|| \cdots ||M[\ell]$.

NMAC is a two-key variant of HMAC. Mathematically, it is a keyed function $\mathsf{NMAC}^h : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n$ defined by $\mathsf{NMAC}^h(K_1, K_2, M) := \mathsf{MD}^h(K_2, \mathsf{MD}^h(K_1, M))$. Here, $K_1, K_2 \in \{0,1\}^n$ are chosen independently and uniformly at random.[5] We sometimes write $\mathsf{NMAC}^h_{K_1,K_2}(M)$ instead of $\mathsf{NMAC}^h($ $K_1, K_2, M)$ for simplicity. See also Figure 1.

Quantum security of HMAC and NMAC.

Simple Quantum distinguishing attacks on HMAC and NMAC. There are two simple quantum attacks to distinguish HMAC from a random function. Suppose that we are given an oracle $\mathcal{O}$ that is either of HMAC or a random function, in addition to the quantum random oracle h.

The first attack is the one that tries to recover the secret key K. Once we succeed in recovering the correct key K (when $\mathcal{O}$ is HMAC) or realizing that there is no plausible candidate for K (when $\mathcal{O}$ a random function), we can distinguish HMAC from a random function. Since the exhaustive key search of k-bit keys can be done with $O(2^{k/2})$ queries by using Grover's algorithm [17], we can distinguish HMAC from a random function with $O(2^{k/2})$ quantum queries.

The second attack uses a collision for $\mathcal{O}$. Suppose that the padding function pad in the Merkle-Damgård construction satisfies the condition that there exists a function $\mathsf{p} : \mathbb{Z}_{\geq 0} \to \{0,1\}^*$ such that $\mathsf{pad}(M) = M||\mathsf{p}(|M|)$, which is the case for usual hash functions such as SHA-2. First, we try to find $M, M' \in \{0,1\}^m$ such that $\mathcal{O}(M) = \mathcal{O}(M')$, which can be done with $O(2^{n/3})$ quantum queries by using the BHT algorithm [11]. When we find such messages, we check whether $\mathcal{O}(M||0^m) = \mathcal{O}(M'||0^m)$ holds. This equality holds with a high probability if $\mathcal{O}$ is HMAC, but it holds with a negligible probability if $\mathcal{O}$ is a random function.

[5] Note that there is no IV involved in NMAC and the key-length is always $n+n = 2n$.

Thus, we can distinguish HMAC from a random function with $O(2^{n/3})$ quantum queries.

In summary, HMAC can be distinguished with $O(\min\{2^{n/3}, 2^{k/2}\})$ quantum queries. The attacks are also applicable for NMAC, and $O(\min\{2^{n/3}, 2^{2n/2}\}) = O(2^{n/3})$ is an upper bound of the query complexity to distinguish NMAC.

Previous results on quantum security of HMAC and NMAC. Song and Yun proved that HMAC and NMAC become secure quantum pseudorandom functions (qPRFs) against polynomial-time quantum adversaries in the *standard model* under the assumption that $h(\cdot||K) : \{0,1\}^m \to \{0,1\}^n$ is a qPRF when $K \in \{0,1\}^n$ is randomly chosen [32]. They for the first time showed that HMAC and NMAC are secure even in the quantum setting, which has great importance in theory because it enables domain extension for qPRFs.

Roughly speaking, their proof guarantees security up to $O(2^{n/5})$ or $O(2^{n/8})$ quantum queries when the underlying function h_K is ideally random for each key K.[6] In other words, $\Omega(2^{n/5})$ or $\Omega(2^{n/8})$ is currently the best proven lower bound of quantum query complexity to distinguish HMAC or NMAC from a random function.

Results in standard models and those in (quantum) random oracles are not directly comparable, but there exists a large gap between the current best lower bound and the upper bound $O(2^{n/3})$ (when k is large enough) given in the above distinguishing attacks.

The gap between $\Omega(2^{n/5})$ (or $\Omega(2^{n/8})$) and $O(2^{n/3})$ may not be significant in an ideal world where adversaries are modeled as polynomial-time machines, but it is indeed significant in the real world applications, which we explain below.

Closing the gap. In the real world, closing the gap between $\Omega(2^{n/5})$ (or $\Omega(2^{n/8})$) and $O(2^{n/3})$ is relevant for the following reasons.

Recall that there exist two security notions in the quantum setting: quantum security and standard security. The standard security of HMAC will have practical importance in a very near future because it is quite reasonable to assume that an adversary has a quantum computer on which h is implemented, but the attack target (HMAC) is implemented on a classical device.

Now, the problem is that exiting results guarantee the security of HMAC and NMAC only up to $O(2^{n/5})$ or $O(2^{n/8})$ queries, not only for the quantum security but also for the standard security (in the QROM). This is problematic since when HMAC is instantiated with SHA-256, where $n = 256$, the security is not guaranteed after about $2^{n/5} \approx 2^{52}$ (or $2^{n/8} \approx 2^{32}$) *classical* queries. It is completely unacceptable in practice, as the number is modest even with the current standard, and is too small to guarantee a longer term security.

In theory, the security up to $O(2^{n/3})$ queries can be guaranteed with the previous result if the security parameter is changed from n to $5n/3$ (or $8n/3$),

[6] Actually, the previous work [32] did not give concrete security bound, but we can reasonably deduce that the security is guaranteed up to $O(2^{n/8})$ quantum queries. We have the bound $O(2^{n/5})$ instead of $O(2^{n/8})$ if we assume a conjecture. See Section A of this paper's full version [20] for details.

by replacing the underlying hash function with the one with a longer output length. However, in the real world, it requires many years to change parameters or primitives of widely used symmetric-key cryptosystems such as HMAC, or sometimes it is simply infeasible, as we illusrtare below:

- Some small IoT devices (e.g., RFID tags) need MACs but do not have enough area for hardware implementation of primitives with large parameters.
- Some banking systems are still using Triple-DES although 20 years have already passed after the standardization of AES [3]. This is because even a small change (changing the block cipher) in financial systems is too costly.
- Artificial satellites require MACs to prevent accepting commands from malicious attackers. Changing primitives embedded as hardware is infeasible after satellites are launched into the outer space [31].

Hence, giving a precise security bound is relevant from a practical view point, and is one of the most important topics to study in symmetric-key cryptography, even if the improvement will be from $O(2^{n/5})$ (or $O(2^{n/8})$) to $O(2^{n/3})$.

We also note that there has been a long line of research to close the gap for HMAC and NMAC in the classical setting, and it was eventually addressed by Gazi et al. at CRYPTO 2014 [16] showing the upper bound and the matching lower bound. However, the analysis in the quantum setting does not reach this point, and closing the gap is important also from a theoretical view point.

1.1 Our Contributions

The main result of this paper is the following theorem, which shows that the tight bound of the number of quantum queries to distinguish HMAC or NMAC from a random function is in $\Theta(2^{n/3})$ (when k is large enough).

Theorem 1 (Lower bound, informal). *Assume $m \geq n$. Suppose that the maximum length of messages that we can query to HMAC, NMAC, or a random function* RF *(which is independent of h) is at most $m \cdot \ell$. Then, the following claims hold in the model where h is a quantum random oracle.*

1. *To distinguish HMAC from* RF *with a constant probability by making at most Q queries to HMAC or* RF *and at most q_h queries to h, $q_h \cdot \ell^{5/3} + Q \cdot \ell^{5/3} \geq \Omega(2^{n/3})$, or $q_h + Q \cdot \ell \geq \Omega(2^{k/2})$ have to be satisfied.*
2. *To distinguish NMAC from* RF *with a constant probability by making at most Q queries to NMAC or* RF *and at most q_h queries to h, $q_h \cdot \ell^{5/3} + Q \cdot \ell^{5/3} \geq \Omega(2^{n/3})$ has to be satisfied.*

Remark 1. Our tightness claim focuses on the number of quantum queries, neglecting the effect of the lengths of the queries (see also Figure 2). Nevertheless, our result still has practical importance. For instance, when HMAC-SHA-256 is used to authenticate TCP/IP packets on Ethernet, $\ell < 32$ always holds since Maximum Segment Size (MSS) is about 1500-byte. In such a use-case our result guarantees about 85-bit security ($2^{n/3} \approx 2^{85}$ for $n = 256$), while previous works do only about 52-bit security or 32-bit security (in the QROM).

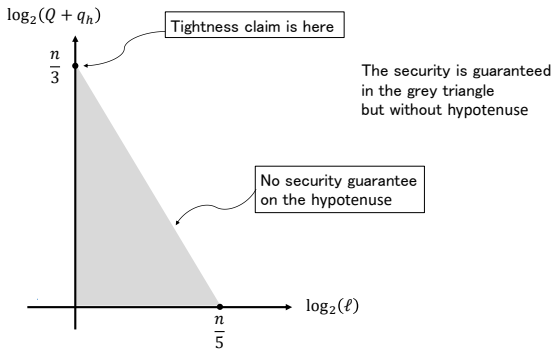


Fig. 2: The area that our result guarantees security (the grey triangle). We claim tightness of the bound for $\ell = O(1)$.

Remark 2. Some readers may think that results in the standard model are always superior to those in the (Q)ROM, but we emphasize that the standard model and (Q)ROM are theoretically incomparable.

To show the theorem, we use an alternative formalization [18,19] of Zhandry's *compressed oracle technique* [34]. One of the most difficult issues in proving security of cryptographic schemes against quantum adversaries is to record quantum queries to oracles. Zhandry [34] solved the issue by developing the compressed oracle technique, which can be used to record queries to QROs and efficiently simulate QROs. Intuitively, by using the technique, we can use the classical *lazy sampling* for quantum random oracles to some extent. The technique is so powerful that it is used to prove security of many cryptographic schemes [18,34,24,12,25,6]. However, efficient simulations of QROs are not necessary when we focus on the number of quantum queries made by adversaries and when their running time is irrelevant. Based on this observation, Hosoyamada and Iwata developed an alternative formalization of the compressed oracle technique that achieved a simpler formalization by ignoring efficient simulations of QROs and introducing notions of error terms, which is named *recording standard oracle with errors (*RstOE*)* [18,19]. Since our main focus is information theoretic adversaries of which computational resources are unlimited except for the number of quantum queries, we use RstOE instead of the original technique.

The technically hardest part to prove Theorem 1 is to show the indistinguishability of the function $F_1^h(u,v) := h(v, f(u))$ from a random function, where $u \in \{0,1\}^n, v \in \{0,1\}^m$, and $f : \{0,1\}^n \to \{0,1\}^n$ is a random function that is independent of h. (Adversaries have a direct oracle access to the quantum random oracle h, but only indirect access to f. That is, adversaries can query to f only through queries to F_1^h, and cannot observe the output values of f. See also Figure 3.) Once we show the indistinguishability of F_1^h, the remaining proofs can be done with simpler proof techniques.

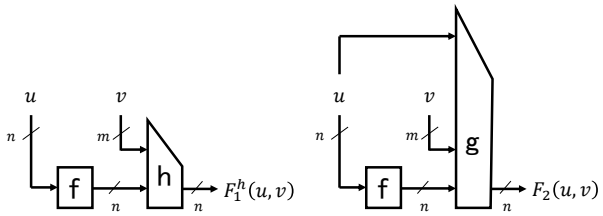


Fig. 3: F_1^h and F_2. h is a quantum random oracle that adversaries can directly access. f and g are random functions that are independent from h.

It turns out that previous techniques cannot be directly used to prove the indistinguishability of F_1^h. Thus we introduce a technique which we call *equivalent databases*. We explain the details in the next subsection.

1.2 Technical Overview

Let us denote the distinguishing advantage of an adversary $\mathcal{A}$ between (pair of) oracles $(\mathcal{O}_1^h, h)$ and $(\mathcal{O}_2, h)$ by $\mathbf{Adv}^{\mathrm{dist}}_{(\mathcal{O}_1^h,h),(\mathcal{O}_2,h)}(\mathcal{A})$, where h is a quantum random oracle and $\mathcal{O}_1^h$ depends on h. Let RF be a random function that is independent of h. As mentioned above, the technically hardest part to show the tight security bound of HMAC and NMAC is to show the following proposition[7].

Proposition 1 (Technically hardest proposition to show, informal). *If $\mathcal{A}$ makes at most q queries to each oracle, $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(\mathsf{RF},h)} \leq O(\sqrt{q^3/2^n})$ holds.*

Let F_2 be the function defined by $F_2(u,v) := g(u,v,f(u))$, where $g : \{0,1\}^n \times \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ is another random function (see also Figure 3). Then, since g is a random function, $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(\mathsf{RF},h)}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A})$ holds. In what follows, we present an overview of how we show

$$\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq O(\sqrt{q^3/2^n}), \tag{1}$$

instead of directly showing Proposition 1. For bit strings x and y, we identify the concatenation $x||y$ and the pair (x,y). [8]

Following usual terminology on provable security in symmetric-key cryptology, we call (direct) queries to h *offline queries* because h is an ideal model of a public function that adversaries can compute offline. In addition, we call queries to F_1^h and F_2 *online queries* because the oracles of F_1^h and F_2 model the keyed functions that adversaries can compute only by making online queries.

[7] In [34] Zhandry showed that F_1^h is indifferentiable from a QRO when h and g are QROs. His result implies qPRF security of F_1^h up to $O(2^{n/4})$ quantum queries, while Proposition 1 guarantees security up to $O(2^{n/3})$ queries.

[8] We consider F_2 instead of RF so that there exists a useful correspondence between "good" databases for F_1^h and those for F_2, which we will elaborate later.

Classical proof intuitions. If our goal were to show the indistinguishability of F_1^h and F_2 in the *classical* setting, we could show it based on the following idea by using the *lazy sampling* technique to f, g, and h:

> *If $\mathcal{A}$ cannot guess outputs of f, and outputs of f do not collide, then the outputs of F_1^h and F_2 seem completely random and indistinguishable.*

More precisely, a (classical) adversary $\mathcal{A}$ cannot distinguish F_1^h and F_2 as long as the following two bad events hit and coll do not happen.[9]

hit: $\mathcal{A}$ succeeds in guessing a previous output of f and queries it to h. That is, $\mathcal{A}$ has queried $u||v'$ to the online keyed oracle (F_1^h or F_2) before, and now $\mathcal{A}$ queries $v||f(u)$ to h (for some $v \in \{0,1\}^m$).

coll: A new output of f (which is sampled during an online query) happens to collide with either of (a) a previous output of f, or (b) the least significant n-bit ζ of a previous offline query $v||\zeta$ to h.

Our proof for the *classical* indistinguishability would be as follows: First, we show that F_1^h and F_2 are completely indistinguishable as long as hit and coll do not happen. Second, we show that $\Pr[\mathsf{hit}]$ and $\Pr[\mathsf{coll}]$ are small. Let coll_i denote the event that coll happens at the i-th query. Then, by using the randomness of outputs of f, we can show $\Pr[\mathsf{coll}_i] \leq O(i/2^n)$ for each i, which implies that $\Pr[\mathsf{coll}] \leq \sum_{1\leq i\leq q} \Pr[\mathsf{coll}_i] \leq \sum_{1\leq i\leq q} O(i/2^n) = O(q^2/2^n)$. Similarly, $\Pr[\mathsf{hit}] \leq O(q^2/2^n)$ can be shown. (Actually there exists a qualitative difference between the proof for $\Pr[\mathsf{coll}] \leq O(q^2/2^n)$ and that for $\Pr[\mathsf{hit}] \leq O(q^2/2^n)$, which will be explained later). Hence we can show $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq \Pr[\mathsf{coll}] + \Pr[\mathsf{hit}] \leq O(q^2/2^n)$ in the classical setting.

How to show quantum indistinguishability? When we show the *quantum* indistinguishability of F_1^h and F_2, it is natural to combine the above *classical* idea with some quantum proof techniques developed in previous works. Indeed, our first idea toward a quantum proof is to combine the above classical idea with a quantum technique introduced in [18,19]. [10] However, actually it turns out that they cannot be simply combined. The issue is attributed to our situation where we have to deal with the bad event hit that "$\mathcal{A}$'s *offline* query to h collides with a previous output of f in the *online* oracle".

Below, we explain (1) an overview of the previous quantum proof technique, (2) what kind of issue arises if we combine the above classical idea with the previous quantum technique, and that (3) we can solve the issue by introducing a new proof technique which we name *equivalent databases*.

[9] We use the symbols u and ζ to denote n-bit strings and v to denote an m-bit string.

[10] In Zhandry's paper that introduced the compressed oracle technique, quantum indifferentiability of the *fixed-input-length* Merkle-Damgård construction is proved [34]. Note that the *variable-input-length* Merkle-Damgård construction that is used in HMAC and NMAC is not indifferentiable in the random oracle model even in the classical setting [13]. In addition, the security bound of the indifferentiability is proved up to $O(2^{n/4})$ (but not $O(2^{n/3})$) quantum queries in [34]. Thus, we start from the proof technique used in [18,19] instead of [34].

Proof technique in [18,19]. The previous work [18,19] showed quantum indistinguishability (Proposition 4 in [19]) of certain two oracles as follows: [11]

1. Suppose that random functions from which the oracles are built (in our case, f, g, and h) are implemented by using RstOE so that we can use intuitions of classical lazy sampling in quantum proofs to some extent (let D_f, D_g, and D_h denote *databases* associated with RstOE for f, g, and h, respectively, which correspond to transcripts of queries in the classical setting).
2. Based on classical proof ideas of using good and bad events, define the notion of *good* and *bad* for tuples of databases (in our case, (D_f, D_h) for F_1^h and (D_f, D_g, D_h) for F_2) in such a way that
   (a) There exists a one-to-one correspondence between good databases for one oracle (in our case, good databases (D_f, D_h) for F_1^h) and good databases for the other oracle (in our case, good databases (D_f, D_g, D_h) for F_2).
   (b) The behavior of one oracle (in our case, F_1^h) on a good database is the same as that of the other oracle (in our case, F_2) on the corresponding good database.
3. By using (a) and (b), show that the oracles (in our case, the pairs of the oracles (F_1^h, h) and (F_2, h)) are completely indistinguishable as long as databases are good.
4. Show that the probability (in some sense) that good databases change to bad databases is very small at each query.

Note that, unlike the setting, even if the record "x has been queried to f and responded with y" is stored in a database D_f for f, there is a possibility that the record will be overwritten as "x has not been queried to f before", or "x has been queried to f and responded with y'" for some y' such that $y \neq y'$ [12]. Hence it is not necessarily trivial how to define good and bad databases in such a way that we can formally prove both of (a) and (b) hold.

Next, we explain what kind of issue happens when we apply the above idea to our situation. In short, the issue lies in the last one of the above four steps.

An issue with our situation. In the previous work [18,19], each adversary can access to only a single *keyed* oracle. Roughly speaking, a good database changes to bad only when a fresh value x is (indirectly) queried to a random function RF, and the newly sampled value $y := \mathsf{RF}(x)$ happens to collide with an existing record in a database (i.e., a bad event that correspond to coll in our situation).

[11] Some technical errors are contained in the Asiacrypt version of the previous work [18], which are corrected in the revised version [19]. Our technical overview in this section and formal proofs in later sections are based on the revised version. For completeness, we do not rely on any propositions in [18,19] that is related to the technical errors in [18]. The propositions from [18,19] that we use in this paper are the ones of which correctness can be confirmed just by straightforward algebraic calculation (Proposition 2 and Proposition 3).

[12] This may seem somewhat strange, but some differences between quantum oracles and classical oracles are explained by using this strange property.

On the other hand, in our situation, a good database also changes to bad when *an adversary succeeds to query $v||\zeta$ to h such that ζ collides with a previous output of f* (i.e., hit occurs).

This difference causes an issue to prove that the "bad" probability is small. Unlike the lazy sampling that always chooses values uniformly at random, (quantum) adversaries can choose offline (quantum) queries to h arbitrarily and *adaptively*. Thus, an adversary may have strong ability to succeed to cause hit, even if the probability of coll is small.

Note that how to deal with adaptive queries to offline queries is not an easy issue even in the classical setting. To reduce the arguments on adaptive queries into those on non-adaptive arguments, sophisticated proof techniques such as the coefficients H technique [29] are usually used.

How to solve the issue. Our key intuition to solve the issue is, for arbitrary good database (D_f, D_h) for F_1^h that an adversary $\mathcal{A}$ is trying to change to be bad, there would be sufficiently many good databases (D'_f, D'_h) that $\mathcal{A}$ cannot distinguish from (D_f, D_h).

Suppose that (I) $\mathcal{A}$ is running relative to F_1^h and h, and has made $(i-1)$ queries in total, (II) both of the bad events coll and hit have not happened, and (III) now $\mathcal{A}$ chooses a bit string $\tilde{v}||\tilde{\zeta}$ to query to h, trying to cause hit at the i-th query.

Let D_f and D_h be the current databases for f and h (before the i-th query). Then there exist $u_1, \ldots, u_s, \alpha_1, \ldots, \alpha_s \in \{0,1\}^n$ $(s \leq i-1)$ such that $D_f = ((u_1, \alpha_1), \ldots, (u_s, \alpha_s))$. Intuitively, α_j is equal to $f(u_j)$. Since bad events have not happened yet, D_f does not contain any collision (i.e., $\alpha_i \neq \alpha_j$ for $i \neq j$).

Let hit_i denote the event that hit occurs at the i-th query (to h). Then, hit_i occurs when $\mathcal{A}$ successfully chooses a value $\tilde{v}||\tilde{\zeta}$ such that $\tilde{\zeta} = \alpha_j$ holds for some j. Our current goal is to prove that $\Pr[\mathsf{hit}_i]$ is very small.

To achieve this goal, we show that $\Pr\left[\mathsf{hit}_i \middle| \mathcal{A} \text{ chooses } \tilde{v}||\tilde{\zeta}\right]$ is very small for *arbitrary* $\tilde{v}||\tilde{\zeta}$, by focusing on the freedom of the choices of the values $f(u_1) = \alpha_1, \ldots, f(u_s) = \alpha_s$. Intuitively, even if the value $\alpha_j (= f(u_j))$ in the element $(u_j, \alpha_j) \in D_f$ is replaced with another value α'_j, $\mathcal{A}$ does not notice since $\mathcal{A}$ does not observe output values of f. This means that the choices of the values $f(u_1) = \alpha_1, \ldots, f(u) = \alpha_s$ have some degree of freedom, even after $\mathcal{A}$ has chosen which value $\tilde{v}||\tilde{\zeta}$ to query to h. We use this degree of freedom to bound the probability $\Pr\left[\mathsf{hit}_i \middle| \mathcal{A} \text{ chooses } \tilde{v}||\tilde{\zeta}\right]$ (actually we will show a stronger result).

To provide a proof based on the above intuition, we introduce the notion of *equivalent databases* as follows.

Definition 1 (Equivalent database, informal). *A (good) database (D'_f, D'_h) is said to be equivalent to (D_f, D_h) if $|D'_f| = |D_f|$, $|D'_h| = |D_h|$, and (D'_f, D'_h) is equal to (D_f, D_h) except for the choices of the output values of f.*

We present an example to illustrate the intuition on equivalent databases. Let $D_f := ((u_1, \alpha_1), (u_2, \alpha_2))$ and $D_h := ((v_1||\alpha_1, w_1), (v_2^{(1)}||\alpha_2, w_2^{(1)}), (v_2^{(2)}||\alpha_2, w_2^{(2)}),$

$(v_3||\zeta_3, w_3)$). This corresponds to the situation where $u_1||v_1$, $u_2||v_2^{(1)}$, $u_2||v_2^{(2)}$ have been queried to F_1^h, and $v_3||\zeta_3$ has been queried to h. See also Figure 4. The adversary observes that $F_1^h(u_1||v_1) = w_1$, $F_1^h(u_2||v_2^{(1)}) = w_2^{(1)}$, $F_1^h(u_2||v_2^{(2)}) = w_2^{(2)}$, and $h(v_3||\zeta_3) = w_3$, but does not know the values $\alpha_1 = f(u_1)$ and $\alpha_2 = f(u_2)$. Suppose $\alpha_1, \alpha_2, \zeta_3$ are distinct, which implies that (D_f, D_h) is a good database. Then, another good database (D'_f, D'_h) is equivalent to (D_f, D_h) if and only if there exist α'_1 and α'_2 such that $\alpha'_1, \alpha'_2, \zeta_3$ are distinct, $D'_f = ((u_1, \alpha'_1), (u_2, \alpha'_2))$, and $D'_h = ((v_1||\alpha'_1, w_1), (v_2^{(1)}||\alpha'_2, w_2^{(1)}), (v_2^{(2)}||\alpha'_2, w_2^{(2)}), (v_3||\zeta_3, w_3))$.

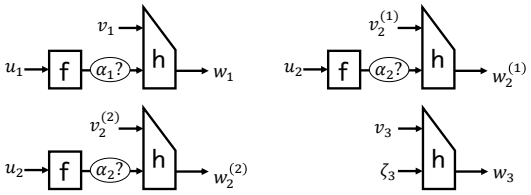


Fig. 4: The situation that corresponds to the good database (D_f, D_h). $\mathcal{A}$ has no information on α_1 and α_2 expect that $\alpha_1, \alpha_2, \zeta_3$ are distinct. We say that another good database (D'_f, D'_h) is equivalent to (D'_f, D'_h) if and only if (D_f, D_h) is equal to (D_f, D_h) except for the choice of the values for α_1 and α_2.

Let $\mathsf{Equiv}(D_f, D_h)$ be the set of good databases that are equivalent to (D_f, D_h). Then, intuitively, the following properties hold:

1. The probability that a database happens to become (D_f, D_h) (after $\mathcal{A}$ made $(i-1)$ queries) is equal to the probability that the database happens to become (D'_f, D'_h), for any $(D'_f, D'_h) \in \mathsf{Equiv}(D_f, D_h)$.
2. The ratio between (I) the number of $(D'_f, D'_h) \in \mathsf{Equiv}(D_f, D_h)$ that leads to the bad event hit_i (i.e., $\alpha_j = \tilde{\zeta}$ for some j) and (II) the size of the entire set $\mathsf{Equiv}(D_f, D_h)$ is at most about $\approx |D_f|/2^n \leq O(i/2^n)$. [13]

From the above two properties it follows that, for arbitrary $\tilde{v}||\tilde{\zeta}$ *and arbitrary good* (D_f, D_h), $\Pr\left[\mathsf{hit}_i \middle| \mathcal{A} \text{ chooses } \tilde{v}||\tilde{\zeta} \wedge \text{ database is equivalent to } (D_f, D_h)\right] \leq O\left(i/2^n\right)$ holds. This implies that $\Pr\left[\mathsf{hit}_i\right] \leq O(i/2^n)$.

The above explanations are in fact based on classical intuitions. To show they also work in the quantum setting, we carefully analyze quantum amplitude (complex coefficients) of state vectors.

[13] This holds due to the following reasoning. For simplicity, assume that nothing has been *directly* queried to h before, and D_f has $(i-1)$ entries $(u_1, \alpha_1), \ldots, (u_{i-1}, \alpha_{i-1})$ (other cases can be shown similarly). Then $|\mathsf{Equiv}(D_f, D_h)|$ is equal to the number of choices of the tuple $(\alpha_1, \ldots, \alpha_{i-1})$ such that $\alpha_j \neq \alpha_k$ for $j \neq k$. Hence $|\mathsf{Equiv}(D_f, D_h)| = \binom{2^n}{i-1}$. In addition, the number of $(D'_f, D'_h) \in \mathsf{Equiv}(D_f, D_h)$ such that $\alpha_j = \tilde{\zeta}$ for some j is $(i-1) \cdot \binom{2^n}{i-2}$. Thus the ratio is $(i-1) \cdot \binom{2^n}{i-2} / \binom{2^n}{i-1} = \frac{(i-1)}{(2^n - i + 2)} \leq O(i/2^n)$.

Finishing the proof. Now we have $\Pr[\mathsf{hit}_i] \leq O(\frac{i}{2^n})$ in the quantum setting. We can also show $\Pr[\mathsf{coll}_i] \leq O(\frac{i}{2^n})$ with the technique in the previous work [18].

In the classical setting, the distinguishing advantage is upper bounded by $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq \Pr[\mathsf{hit}] + \Pr[\mathsf{coll}] \leq \sum_{1\leq i\leq q}\Pr[\mathsf{hit}_i] + \sum_{1\leq i\leq q}\Pr[\mathsf{coll}_i]$. On the other hand, roughly speaking, the quantum distinguishing advantage is upper bounded by $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq \sum_{1\leq i\leq q}\sqrt{\Pr[\mathsf{hit}_i]} + \sum_{1\leq i\leq q}\sqrt{\Pr[\mathsf{coll}_i]}$. Therefore, we obtain the bound as $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq \sum_{1\leq i\leq q} O\left(\sqrt{i/2^n}\right) + \sum_{1\leq i\leq q} O\left(\sqrt{i/2^n}\right) \leq O\left(\sqrt{q^3/2^n}\right)$ in the quantum setting, instead of the classical bound $O(q^2/2^n)$.

The intuition behind the notion of equivalent databases might seem simple or even trivial, though, the important point is that we can provide a rigorous proof that the intuition actually works in the quantum setting through RstOE. (Recall that it was unclear how to record quantum queries before the development of the compressed oracle technique.)

As we mentioned before, it is quite important to show the tight security bound in symmetric cryptology because even the improvement from $O(2^{n/5})$ (or $O(2^{n/8})$) to $O(2^{n/3})$ has significant importance in the real world. Bad events like hit that an adversary succeeds to guess an output of a random function often appear in classical provable security for symmetric-key cryptosystems. To deal with such bad events when showing quantum tight security bounds, proof techniques like our equivalent databases seem indispensable. We believe that our technique broadens the applicability of quantum provable security in symmetric-key cryptology.

1.3 Limitations and Future Directions

Our security bound is tight and any further improvement is impossible in terms of the number of queries. However, there is a room for improvement in terms of the length of messages. When an adversary makes a single classical query of very long length (e.g., a message of $m \cdot 2^{n/5}$ bits, or equivalently $\ell = 2^{n/5}$) to the keyed oracle of HMAC or NMAC, our result no longer guarantees any security. (Note that this does not invalidate the practical importance of our result. See Remark 1 for details.) However, we do not find any quantum attack that actually breaks the security of HMAC or NMAC by making only a few queries of which length is $O(m \cdot 2^{n/5})$, and we expect that there does not exist such an attack. Improving the security bound in terms of message lengths is an interesting future work.

1.4 Related Works

There are various notions on quantum MAC security such as EUF-qCMA security [10] and blind unforgeability [1]. There also exists another security notion for one-time MAC security [15]. MACs built from qPRFs satisfy all these

security notions. Boneh and Zhandry showed that qPRFs become quantum secure MACs (in the sense of EUF-qCMA) and showed quantum security of the Carter-Wegman MACs [10]. Czajkowski et al. showed quantum security of random sponge, which can be seen as a variant of CBC-MAC [14].

1.5 Paper Organization

Section 2 describes notation, definitions, and some basic lemmas used in later sections. Section 3 gives an overview on the alternative formalization (RstOE) of Zhandry's compressed oracle technique. Section 4 gives the formal proof of the technically most hardest proposition (Proposition 1) and introduces the new proof technique. Section 5 shows quantum security bound of HMAC and NMAC.

2 Preliminaries

In this paper, all adversaries are quantum algorithms. I_n denotes the identity operator on n-qubit quantum states. We often write just I instead of I_n when it will cause no confusion. For a unitary operator U, we denote the operators $U \otimes I$ and $I \otimes U$ by the same symbol U, when it will cause no confusion. We identify the set of bit strings $\{0,1\}^n$ with the set of integers $\{0,1,\ldots,2^n-1\}$ for any positive integer n. In addition, we identify the pair $(x,y) \in \{0,1\}^m \times \{0,1\}^n$ with the concatenation $x||y \in \{0,1\}^{m+n}$. $\{0,1\}^*$ denotes the set $\coprod_{n=0}^{\infty}\{0,1\}^n$, where $\{0,1\}^0$ denotes the set that includes only the empty string. For a positive integer m, $(\{0,1\}^m)^+$ denotes the set $\coprod_{i=1}^{\infty}\{0,1\}^{im}$. $\mathsf{td}(\cdot,\cdot)$ denotes the trace distance function. For a vector $|\phi\rangle$ and a positive integer n, we also denote $|\phi\rangle \otimes |0^n\rangle$ and $|0^n\rangle \otimes |\phi\rangle$ by $|\phi\rangle$, when it will cause no confusion.

2.1 Quantum Algorithms and Quantum Oracles

When we consider the computational resources of adversaries, we focus on the number of queries made by adversaries, and we do not care about their running time and memory usage (i.e., we consider quantum information theoretic adversaries). Here we describe how we model (oracle-aided) quantum algorithms and quantum oracles in the case that each adversary is given an oracle access to a single quantum oracle.

Following previous works (e.g., [9]), we model an (oracle-aided) quantum algorithm $\mathcal{A}$ that makes at most q quantum queries to a single oracle as a sequence of unitary operators $(U_0, U_1, \ldots, U_q)$, where U_i corresponds to $\mathcal{A}$'s offline computation after the i-th oracle query for $i \geq 1$, and U_0 corresponds to $\mathcal{A}$'s initial computation. In addition, the quantum state space of $\mathcal{A}$ is a tensor product $\mathcal{H}_{\mathrm{query}} \otimes \mathcal{H}_{\mathrm{answer}} \otimes \mathcal{H}_{\mathrm{work}}$, where $\mathcal{H}_{\mathrm{query}}$, $\mathcal{H}_{\mathrm{answer}}$, and $\mathcal{H}_{\mathrm{work}}$ correspond to the register to make queries to the oracle, the register to receive answers from the oracle, and the register for $\mathcal{A}$'s offline computations, respectively. After the application of the final unitary operator U_q, $\mathcal{A}$'s entire state is measured, and (a part of) the measurement result (classical bit string) is returned as the output.

When $\mathcal{A}$ does not take any initial input, we assume that $\mathcal{A}$'s initial state is set to be $|0^s\rangle$ for some positive integer s. When $\mathcal{A}$ takes a classical input $x \in \{0,1\}^m$, we assume that $\mathcal{A}$'s initial state is set to be $|x\rangle$ by convention. (This paper does not treat the situation that $\mathcal{A}$ takes quantum states as inputs.)

A quantum oracle $\mathcal{O}$ is modeled as a tuple of unitary operator O, quantum state space $\mathcal{H}_{\rm query} \otimes \mathcal{H}_{\rm answer} \otimes \mathcal{H}_{\rm state}$, and a vector (initial state) $|\mathsf{init}\rangle \in \mathcal{H}_{\rm state}$. Here, the state space $\mathcal{H}_{\rm query} \otimes \mathcal{H}_{\rm answer}$ (i.e., the registers to send queries and receive answers) is shared with adversaries, and $\mathcal{H}_{\rm state}$ is the oracle's private space that adversaries cannot access directly. O may be chosen randomly according to a distribution at the beginning of each game.

When the adversary $\mathcal{A}$ runs relative to the quantum oracle $\mathcal{O}$ on input x, the initial whole quantum state is $|x\rangle \otimes |\mathsf{init}\rangle$. The whole quantum state just before the i-th query is $U_{i-1}OU_{i-2}O\cdots OU_0 |x\rangle \otimes |\mathsf{init}\rangle$, and the whole quantum state just before the final measurement is $U_qOU_{q-1}O\cdots OU_0 |x\rangle \otimes |\mathsf{init}\rangle$. Let $z \leftarrow \mathcal{A}^{\mathcal{O}}(x)$ denote the event that the quantum algorithm $\mathcal{A}$ returns z as the final output when $\mathcal{A}$ takes x as an input and runs relative to $\mathcal{O}$.

Example: Quantum oracle of a fixed function and a quantum random oracle. According to the above model, the quantum oracle $\mathcal{O}_f$ of a fixed function $f : \{0,1\}^m \to \{0,1\}^n$ is modeled as follows: the state space of $\mathcal{O}_f$ is empty. The unitary operator O_f that processes queries made to $\mathcal{O}_f$ is defined by $O_f : |x\rangle |y\rangle \mapsto |x\rangle |y \oplus f(x)\rangle$ for all $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$.

In addition, a quantum random oracle (QRO) is defined to be the quantum oracle such that, $f : \{0,1\}^m \to \{0,1\}^n$ is chosen uniformly at random at the beginning of each game (for some m and n), and quantum oracle access to $\mathcal{O}_f$ is given to adversaries.

Even if a function f admits input messages M and M' of which lengths differ, we assume that the quantum oracle of O_f admits queries of superpositions of M and M'. In such a case, we assume that length $|M|$ of each message M is encoded with M. However, for ease of notation, we just write $|M\rangle$ instead of $|(|M|, M)\rangle$ for each message M.

2.2 How to Model Accesses to Multiple Quantum Oracles

Suppose that an adversary $\mathcal{A}$ is given oracle accesses to multiple quantum oracles $\mathcal{O}_1, \ldots, \mathcal{O}_s$, and $\mathcal{A}$ makes q queries to each oracle $\mathcal{O}_1, \ldots, \mathcal{O}_s$ in a sequential order. That is, for each $1 \leq j < s$, after $\mathcal{A}$ makes the i-th query to $\mathcal{O}_j$, $\mathcal{A}$ performs some offline computations, and then makes the i-th query to $\mathcal{O}_{j+1}$. Similarly, after $\mathcal{A}$ makes the i-th query to $\mathcal{O}_s$, $\mathcal{A}$ performs some offline computations, and then makes the $(i+1)$-th query to $\mathcal{O}_1$. Here we explain how to model the behavior of $\mathcal{A}$ and multiple quantum oracles $\mathcal{O}_1, \ldots, \mathcal{O}_s$ as sequential applications of unitary operators, in the case that $\mathcal{A}$ makes queries in a sequential order as above.

We assume that the oracles share a state space that is described as the tensor product $\mathcal{H}_{\rm query} \otimes \mathcal{H}_{\rm answer} \otimes \mathcal{H}_{\rm state}$. Here, $\mathcal{H}_{\rm query} \otimes \mathcal{H}_{\rm answer}$ is the partial state space of $\mathcal{A}$ (thus the adversary and the oracles share the registers to send queries and

receive answers). $\mathcal{H}_{\mathrm{state}}$ is oracles' private space that adversaries cannot access directly.

For each quantum oracle $\mathcal{O}_i$, let O_i denote the unitary operator to process queries. We assume that the initial state of $\mathcal{A}$ is set to be $|x\rangle$ when $\mathcal{A}$ takes x as an input (when $\mathcal{A}$ does not take any initial input, by convention we assume that the initial state of $\mathcal{A}$ is $|0^\alpha\rangle$ for some α). Let $|\mathsf{init}\rangle$ be the initial state of the oracles' private space $\mathcal{H}_{\mathrm{state}}$. Then we model that the quantum state of $\mathcal{A}$ and the oracles before the final measurement becomes $\left(\prod_{j=1}^{q} U_{s,j} O_s \cdots U_{1,j} O_1\right) U_0 |x\rangle \otimes |\mathsf{init}\rangle$, where the adversary $\mathcal{A}$ is modeled as the sequence of unitary operators $(U_0, U_{1,1}, \ldots, U_{s,1}, U_{1,2}, \ldots, U_{s,q})$, and $U_{i,j}$ corresponds to the offline computation by $\mathcal{A}$ after the j-th query to $\mathcal{O}_i$. By $z \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_s}(x)$, we denote the event that $\mathcal{A}$ finally outputs the classical string z when $\mathcal{A}$ takes x as an input and runs relative to the oracles $\mathcal{O}_1, \ldots, \mathcal{O}_s$.

The model of adversaries of which queries are not in a sequential order. In the above model we considered the special case that the adversary queries to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_s$ in a sequential order. However, even if an adversary $\mathcal{B}$ (given oracle accesses to $\mathcal{O}_1, \ldots, \mathcal{O}_s$) does not make queries in such a sequential order, the behavior of $\mathcal{B}$ can be captured with the above model: Suppose that $\mathcal{B}$ makes at most q_i quantum queries to $\mathcal{O}_i$ for each i, and s is a constant. Then, we can make another adversary $\mathcal{A}$ such that $\mathcal{A}$'s output distributions are the same as that of $\mathcal{B}$, and $\mathcal{A}$ makes $O(\max\{q_1, \ldots, q_s\})$ queries to each oracle in a sequential order as in the above model, by appropriately increasing the number of queries. Thus all reasonable adversaries are captured by the above model.

2.3 Security Advantages

Quantum distinguishing advantage. For quantum oracles $\mathcal{O}_1, \ldots, \mathcal{O}_s$ and $\mathcal{O}'_1, \ldots, \mathcal{O}'_s$, we define the quantum distinguishing advantage of an adversary $\mathcal{A}$ by $\mathbf{Adv}^{\mathrm{dist}}_{(\mathcal{O}_1,\ldots,\mathcal{O}_s),(\mathcal{O}'_1,\ldots,\mathcal{O}'_s)}(\mathcal{A}) := \left|\Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{O}_1,\ldots,\mathcal{O}_s}()\right] - \Pr\left[1 \leftarrow \mathcal{A}^{\mathcal{O}'_1,\ldots,\mathcal{O}'_s}()\right]\right|$.

qPRF advantage in QROM. Let h be a QRO and F^h_K be a keyed function that may depend on h. By the same symbol F^h_K we denote the quantum oracle such that the key K is chosen at random, and the quantum oracle access to F^h_K is given to adversaries. In addition, let RF be the quantum oracle of a random function that is independent of h. Then, we define the quantum pseudorandom function advantage (qPRF advantage) of $\mathcal{A}$ on F^h_K by $\mathbf{Adv}^{\mathrm{qPRF}}_{F^h_K}(\mathcal{A}) := \mathbf{Adv}^{\mathrm{dist}}_{(F^h_K,h),(\mathsf{RF},h)}(\mathcal{A})$.

Here we introduce a basic proposition from a previous work [30] for later use.

Lemma 1 (Lemma 2.2 of [30]). *Let $h : \{0,1\}^{m+n} \to \{0,1\}^n$ be a quantum random oracle. For a random key $K \in \{0,1\}^k$ ($k < m+n$), define $F^h_K : \{0,1\}^{m+n-k} \to \{0,1\}^n$ by $F^h_K(x) = h(x||K)$. Then, for each adversary $\mathcal{A}$ that makes at most q_h quantum queries to h, $\mathbf{Adv}^{\mathrm{qPRF}}_{F^h_K}(\mathcal{A}) \leq O\left(q_h/2^{k/2}\right)$ holds.*

qPRG advantage. Let h be a quantum random oracle and $\rho^h : \{0,1\}^{k_1} \to \{0,1\}^{k_2}$ be a function that may depend on h. Then, we define the quantum PRG advantage $\mathbf{Adv}^{\mathrm{qPRG}}_{\rho^h}(\mathcal{A})$ of $\mathcal{A}$ on ρ^h by $\mathbf{Adv}^{\mathrm{qPRG}}_{\rho^h}(\mathcal{A}) := \left|\Pr\left[K_1 \xleftarrow{\$} \{0,1\}^{k_1} : 1 \leftarrow \mathcal{A}^h(\rho^h(K_1))\right] - \Pr\left[K_2 \xleftarrow{\$} \{0,1\}^{k_2} : 1 \leftarrow \mathcal{A}^h(K_2)\right]\right|$. In addition, we introduce the following lemma for later use.

Lemma 2. *Let $h : \{0,1\}^{m+n} \to \{0,1\}^n$ be a quantum random oracle, and $k \leq m$. Let $\Delta \in \{0,1\}^m$ and $IV \in \{0,1\}^n$ be public constants such that $\Delta \neq 0^m$. Define $\rho^h : \{0,1\}^k \to \{0,1\}^{2n}$ by $\rho^h(K) = h(K||0^{m-k}||IV)||h((K||0^{m-k} \oplus \Delta)||IV)$. Then, for any quantum adversary $\mathcal{A}$ that makes at most q_h quantum queries to h, $\mathbf{Adv}^{\mathrm{qPRG}}_{\rho^h}(\mathcal{A}) \leq O\left(q_h/2^{k/2}\right)$ holds.*

Lemma 2 can easily be shown by slightly modifying the proof of Lemma 1 (Lemma 2.2 in [30]). See Section B of this paper's full version [20] for details.

3 An Overview on How to Record Quantum Queries

Here, we give an overview of the recording standard oracle with errors [18,19], which is an alternative formalization of Zhandry's compressed oracle technique [34].

The primal definition of QRO. Let us begin with recalling the primal definition of QRO (see Section 2 for details). A QRO is the quantum oracle such that

1. a function f is chosen from $\mathsf{Func}(\{0,1\}^m, \{0,1\}^n)$, the set of all functions from $\{0,1\}^m$ to $\{0,1\}^n$, uniformly at random, and
2. a quantum oracle access to f is given to adversaries.

Here, m and n are positive integers. Note that the quantum oracle of f is described as the unitary operator O_f that is defined by $O_f : |x\rangle\,|y\rangle \mapsto |x\rangle\,|y \oplus f(x)\rangle$ for all $x \in \{0,1\}^m$ and $y \in \{0,1\}^n$. In the QROM, an adversary $\mathcal{A}$ makes quantum queries to a QRO (and quantum queries to additional oracles that may depend on the QRO) and finally returns some outputs.

An alternative view of QRO: the standard oracle. Here, let us define a quantum oracle named the *standard oracle*, which is an alternative view of QRO. First, suppose that each function $f : \{0,1\}^m \to \{0,1\}^n$ is encoded into the $2^m \cdot (n+1)$-bit string $(0||f(0))|| \cdots ||(0||f(2^m-1))$, and identify f with this bit string[14]. Second, let stO be the unitary operator defined by

$$\mathsf{stO} : |x\rangle\,|y\rangle \otimes |S\rangle \mapsto |x\rangle\,|y \oplus S_x\rangle \otimes |S\rangle\,, \tag{2}$$

[14] Here, the bit "0" concatenated with each $f(i)$ is redundant, but it is necessary so that the notation for stO is compatible with that for the recording standard oracle with errors introduced later.

where $x \in \{0,1\}^m$, $y \in \{0,1\}^n$, and $S = (b_0||S_0)||\cdots||(b_{2^m-1}||S_{2^m-1})$ ($b_i \in \{0,1\}$ and $S_i \in \{0,1\}^n$ for each i. Essentially, the operator stO does not act on the register for b_i for each i). Then we have $\mathsf{stO}\,|x\rangle\,|y\rangle \otimes |f\rangle = |x\rangle\,|y \oplus f(x)\rangle \otimes |f\rangle$ for each function f.

Definition 2 (Standard oracle). *The standard oracle is the quantum oracle such that the initial state of the oracle is* $\sum_f \sqrt{1/2^{n2^m}}\,|f\rangle$ *and each quantum query is processed with the unitary operator* stO.

By the same symbol stO we denote not only the unitary operator (2) but also the standard oracle if it will cause no confusion. The following lemma clearly holds.

Lemma 3. *For any quantum algorithm* $\mathcal{A}$ *and any possible output* z *(classical bit string),* $\Pr\left[z \leftarrow \mathcal{A}^{\mathrm{QRO}}\right] = \Pr\left[z \leftarrow \mathcal{A}^{\mathsf{stO}}\right]$ *holds.*

The recording standard oracle with errors. Let IH, U_{toggle}, and CH be the unitary operators that act on $2^m \cdot (n+1)$-qubit states defined by $\mathsf{IH} := (I \otimes H^{\otimes n})^{2^m}$, $U_{\mathrm{toggle}} := (I_1 \otimes |0^n\rangle\langle 0^n| + X \otimes (I_n - |0^n\rangle\langle 0^n|))^{2^m}$, and $\mathsf{CH} := (CH)^{2^m}$. Here, X is the 1-qubit bit-flip operation such that $X\,|b\rangle = |b \oplus 1\rangle$ and $CH := |0\rangle\langle 0| \otimes I_n + |1\rangle\langle 1| \otimes H^{\otimes n}$. Let $U_{\mathrm{enc}} := \mathsf{CH} \cdot U_{\mathrm{toggle}} \cdot \mathsf{IH}$ and $U_{\mathrm{dec}} := U_{\mathrm{enc}}^*$, and define the unitary operator RstOE that acts on $(m+n+(n+1)\cdot 2^m)$-qubit quantum states by

$$\mathsf{RstOE} := (I_{m+n} \otimes U_{\mathrm{enc}}) \cdot \mathsf{stO} \cdot (I_{m+n} \otimes U_{\mathrm{dec}}). \tag{3}$$

Then the recording standard oracle with errors RstOE is defined as follows.

Definition 3 (Recording standard oracle with errors). *The recording standard oracle with errors is the quantum oracle such that its initial state is* $|0^{2^m(n+1)}\rangle$ *and each quantum query is processed with the unitary operator* RstOE.

By the same symbol RstOE we denote not only the unitary operator (3) but also the recording standard oracle with errors if it will cause no confusion.

Intuition behind the definition of RstOE. RstOE is the composition of U_{dec}, stO, and U_{enc}. The first operator U_{dec} decodes superpositions of databases into the uniform superposition of all functions $\sum_f \sqrt{1/2^{n2^m}}\,|f\rangle$. The second stO responds to queries in the same way as the original standard oracle. Finally, U_{enc} encodes the uniform superposition of functions into a superposition of databases. Recall that $U_{\mathrm{enc}} = \mathsf{CH} \cdot U_{\mathrm{toggle}} \cdot \mathsf{IH}$. Intuitively, after the action of the first unitary operator IH, the register of the function f that corresponds to the value $f(x)$ changes to $|0^n\rangle$ if adversary has no information on $f(x)$, and changes to some non-zero value if adversary has some information on $f(x)$. If the value of the register is non-zero, database should record the value of $f(x)$. The second operator U_{toggle} checks if the register is non-zero, and set $b_x := 1$ to indicate that "the value of $f(x)$ should be recorded". Finally, the third operator CH constructs a (superposition of) database D in such a way that the value $f(x)$ is recorded in D if and only if $b_x = 1$.

Next, we give some notation used to describe the property of RstOE. Let $D := (b_0||y_0)||\cdots||(b_{2^m-1}||y_{2^m-1})$ be a $2^m\cdot(n+1)$-bit bit string, where $b_i \in \{0,1\}$ and $y_i \in \{0,1\}^n$ for $0 \le i \le 2^m-1$. We call D a *valid database* if $\neg(b_i = 0 \wedge y_i \neq 0^n)$ holds for all i. If $b_i = 0 \wedge y_i \neq 0^n$ holds for some i, we call D an *invalid database*. Intuitively, a valid database D will be a quantum version of "transcript" for a random oracle: $b_x = 1 \wedge y_x = y$ implies that "the adversary queried x to the random oracle before, and the query was responded with y".

Let $D = (b_0||y_0)||\cdots||(b_{2^m-1}||y_{2^m-1})$ be a valid database, and $I_D \subset \{0,1\}^m$ be the set of indices such that $i \in I_D$ if and only if $b_i = 1$. Then, we can define a set $S_D \subset \{0,1\}^m \times \{0,1\}^n$ from D by $S_D := \{(i, y_i)\}_{i\in I_D}$. Similarly, if a subset $S \subset \{0,1\}^m \times \{0,1\}^n$ satisfies the condition

$$x \neq x' \text{ for distinct elements } (x, s_x), (x', s_{x'}) \in S, \tag{4}$$

we can define a valid database D_S from S by $D_S := (b_0||y_0)||\cdots||(b_{2^m-1}||y_{2^m-1})$, where $b_x = 1$ and $y_x = s_x$ if $(x, s_x) \in S$ and $b_x = 0$ and $y_x = 0^n$ otherwise. Each of the maps $D \mapsto S_D$ and $S \mapsto D_S$ is the inverse of the other, and we identify valid databases and the subsets that satisfy (4). Furthermore, we identify a set $S \subset \{0,1\}^m \times \{0,1\}^n$ that satisfies (4) with the partially defined function f_S such that $f_S(x) = y$ if and only if $(x, y) \in S$, and $f_S(x) = \perp$ if $(x, y) \notin S$ for any y. Particularly, we use the same symbol D to denote S_D and f_{S_D}.

Remark 3. Pay attention not to confuse the (valid) databases with the encoding of functions $f : \{0,1\}^m \to \{0,1\}^n$ that is used when we defined the standard oracle stO. The encoding of functions are used only in the definition of stO, but the notion of databases are used throughout the rest of the paper.

By definition of RstOE, the proposition below immediately follows (see arguments in Section 3 of [18,19] for details).

Proposition 2. *The recording standard oracle with errors* RstOE *is completely indistinguishable from the quantum random oracle. That is, for any quantum algorithm* $\mathcal{A}$ *and any possible output* z, $\Pr\left[z \leftarrow \mathcal{A}^{\mathrm{QRO}}\right] = \Pr\left[z \leftarrow \mathcal{A}^{\mathsf{RstOE}}\right]$ *holds. In addition, if we measure the database register of* RstOE *just before* $\mathcal{A}$ *makes the* i*-th query, the database after the measurement contains at most* $(i-1)$ *entries.*

The following proposition shows the main properties of RstOE that are shown in the previous work [18,19].

Proposition 3 (Proposition 1 in [18,19]). *Let* $x \in \{0,1\}^m$ *and* $D = (b_0||y_0)||\cdots||(b_{2^m-1}||y_{2^m-1})$ *be a valid database such that* $D(x) = \perp$ *(in particular,* $b_x = 0$ *and* $y_x = 0^n$ *hold). In addition, for* $z \neq 0^n$ *let* $D \cup (x,z)^{\mathsf{invalid}}$ *be the* invalid *database* $D \cup (x,z)^{\mathsf{invalid}} := (b'_0||y'_0)||\cdots(b'_{2^m-1}||y'_{2^m-1})$ *such that* $b'_t = b_t \wedge y_t = y'_t$ *if* $t \neq x$*, and* $b_x = 0 \wedge y_x = z$.

1. *For any* $y, \alpha \in \{0,1\}^n$*, there exists a vector* $|\epsilon_1\rangle$ *such that*

$$\mathsf{RstOE}\,|x,y\rangle \otimes |D \cup (x,\alpha)\rangle = |x, y\oplus\alpha\rangle \otimes |D \cup (x,\alpha)\rangle + |\epsilon_1\rangle \tag{5}$$

and $\||\epsilon_1\rangle\| \leq O(\sqrt{1/2^n})$ *hold. More precisely,*

$$|\epsilon_1\rangle = \frac{1}{\sqrt{2^n}}|x, y\oplus\alpha\rangle\left(|D\rangle - \left(\sum_{\beta\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|D\cup(x,\beta)\rangle\right)\right) \tag{6}$$

$$-\frac{1}{\sqrt{2^n}}\sum_{\beta\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|x, y\oplus\beta\rangle\left(|D\cup(x,\beta)\rangle - |D^{\mathsf{invalid}}_\beta\rangle\right) \tag{7}$$

$$+\frac{1}{2^n}|x\rangle|\widehat{0^n}\rangle\left(2\sum_{\beta\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|D\cup(x,\beta)\rangle - |D\rangle\right) \tag{8}$$

holds, where $|D^{\mathsf{invalid}}_\beta\rangle$ *is a superposition of invalid databases for each* β *defined by* $|D^{\mathsf{invalid}}_\beta\rangle = \sum_{\gamma\neq 0^n}\frac{(-1)^{\beta\cdot\gamma}}{\sqrt{2^n}}|D\cup(x,\gamma)^{\mathsf{invalid}}\rangle$ *and* $|\widehat{0^n}\rangle := H^{\otimes n}|0^n\rangle$.

2. *For any* y*, there exists a vector* $|\epsilon_2\rangle$ *such that*

$$\mathsf{RstOE}|x, y\rangle\otimes|D\rangle = \sum_{\alpha\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|x, y\oplus\alpha\rangle\otimes|D\cup(x,\alpha)\rangle + |\epsilon_2\rangle \tag{9}$$

and $\||\epsilon_2\rangle\| \leq O(\sqrt{1/2^n})$ *hold. More precisely,*

$$|\epsilon_2\rangle = \frac{1}{\sqrt{2^n}}|x\rangle|\widehat{0^n}\rangle\left(|D\rangle - \sum_{\beta\in\{0,1\}^n}\frac{1}{\sqrt{2^n}}|D\cup(x,\beta)\rangle\right) \tag{10}$$

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

The first and second properties (especially, (5) and (9)) in this proposition correspond to the classical intuition for lazy sampling such that, when x is queried to a random function, (i) if x has been queried before and responded with α, respond with α again, and (ii) if x has not been queried before, sample α uniformly at random and respond with α, respectively. This intuition works well when the initial state $|x, y\rangle\otimes|D\cup(x,\alpha)\rangle$ or $|x, y\rangle\otimes|D\rangle$ are not superposed. When the initial states are superposed, the effect of the error terms $|\epsilon_1\rangle$ and $|\epsilon_2\rangle$ become significant, and quantum-specific property such that "an entry $(x,\alpha)\in D$ is deleted from D at a query" or "an entry $(x,\alpha)\in D$ is overwritten with another data (x,α') at a query" emerge.

4 Technical Proposition

The goal of this section is to show the following proposition, which is the technically hardest part to show quantum security of HMAC and NMAC.[15] Once we prove it, the remaining proofs for HMAC and NMAC can be shown by using simpler techniques. See also Section 1.2 for proof intuition.

[15] The proposition is a formal restatement of Proposition 1 in Section 1.2 for the case $u\in\{0,1\}^{n+m'}$.

Proposition 4. *Let $h : \{0,1\}^{m+n} \to \{0,1\}^n$ be a quantum random oracle. Let $f : \{0,1\}^{n+m'} \to \{0,1\}^n$ be a random function, and $F_1^h : \{0,1\}^{n+m'} \times \{0,1\}^m \to \{0,1\}^n$ be the function defined by $F_1^h(u,v) := h(v, f(u))$. Let $\mathcal{A}$ be an algorithm that runs relative to the quantum oracle of F_1^h and the quantum random oracle h, or the quantum oracle of a random function RF and the quantum random oracle h. Suppose that $\mathcal{A}$ makes at most q_h quantum queries to h and Q quantum queries to F_1^h or RF. Let $q := \max\{Q, q_h\}$, and suppose that q is in $o(2^{n/3})$. Then*

$$\mathbf{Adv}^{\mathrm{qPRF}}_{F_1^h}(\mathcal{A}) \leq O\left(\sqrt{q^3/2^n}\right) \tag{11}$$

holds.

Let F_2 be the function defined by $F_2(u,v) := g(u,v,f(u))$, where $g : \{0,1\}^{n+m'} \times \{0,1\}^m \times \{0,1\}^n \to \{0,1\}^n$ is another random function. Then, since g is a random function, $\mathbf{Adv}^{\mathrm{qPRF}}_{F_1^h}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A})$ holds. To simplify proofs, instead of directly showing (11), we show that $\mathbf{Adv}^{\mathrm{dist}}_{(F_1^h,h),(F_2,h)}(\mathcal{A}) \leq O\left(\sqrt{q^3/2^n}\right)$ holds.

4.1 Proof of Proposition 4

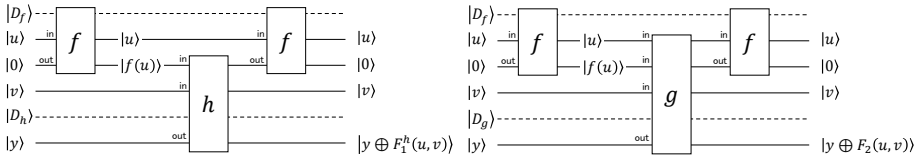
Here we give a proof for the case $m' = 0$. The claims for $m' > 0$ can be shown in the same way. We assume that $\mathcal{A}$ makes queries to F_1^h and h (or, F_2 and h) in a sequential order and model the adversary and oracles as in Section 2.2. In particular, by convention we assume that $\mathcal{A}$'s $(2i-1)$-th query is made to F_1^h (or F_2) and $2i$-th query is made to h for $1 \leq i \leq q$. (For instance, $\mathcal{A}$ first queries to F_1^h (or F_2) and second queries to h.) We call queries to F_1^h and F_2 *online queries* and queries to h *offline queries* since, in practical settings, computations of h are done offline on adversaries' (quantum) computers.

We assume that the unitary operators to process queries to F_1^h and F_2 are implemented as follows:

Quantum oracle of F_1^h.

1. Take $|u,v\rangle\,|y\rangle$ as an input, where $u, y \in \{0,1\}^n$ and $v \in \{0,1\}^m$.
2. Query u to f and obtain
$$|u,v\rangle\,|y\rangle \otimes |f(u)\rangle\,. \tag{12}$$
3. Query $(v, f(u))$ to h and add the answer into the y register to obtain
$$|u,v\rangle\,|y \oplus F_1^h(u,v)\rangle \otimes |f(u)\rangle\,. \tag{13}$$
4. Uncompute Step 2 to obtain $|u,v\rangle\,|y \oplus F_1^h(u,v)\rangle\,.$

We assume that the quantum oracle of F_2 is implemented in the same way as F_1^h, except that the query $(v, f(u))$ to h in Step 3 is replaced with the query $(u, v, f(u))$ to g. See also Figure 5.



(a) Implementation of F_1^h. (b) Implementation of F_2.

Fig. 5: Implementations of F_1^h and F_2. "in" and "out" denote the registers to send queries and receive answers, respectively. The dotted lines (and $|D_f\rangle, |D_h\rangle, |D_g\rangle$) appear only when f, h, g are implemented with RstOE, which correspond to the database registers.

We show the hardness of distinguishing F_1^h and F_2 by using the recording standard oracle with errors (RstOE): We assume that the quantum oracles of f, g, and h are implemented by using RstOE (quantum queries are processed with RstOE). Let RstOE_f, RstOE_g, and RstOE_h be the recording standard oracle with errors for f, g, and h, respectively. We use the symbols D_f, D_g, and D_h to denote databases for f, g, and h, respectively. Then the unitary operator $O_{F_1^h}$ (resp., O_{F_2}) to process queries to F_1^h (resp., F_2) can be decomposed as $O_{F_1^h} = \mathsf{RstOE}_f^* \cdot \mathsf{RstOE}_h \cdot \mathsf{RstOE}_f$ (resp., $O_{F_2} = \mathsf{RstOE}_f^* \cdot \mathsf{RstOE}_g \cdot \mathsf{RstOE}_f$). See also Figure 5 for the intuition about which registers the different RstOEs act.

Good and bad databases. Here we introduce the notion of good and bad databases for F_1^h and F_2. When we use the symbols u, ζ, v, w, we assume that $u, \zeta, w \in \{0,1\}^n$ and $v \in \{0,1\}^m$. We say that a pair of valid database (D_f, D_h) for F_1^h is *good* if and only if the following properties are satisfied.

1. For each $(u, \zeta) \in D_f$, there exist $v \in \{0,1\}^m$ and $w \in \{0,1\}^n$ such that $((v, \zeta), w) \in D_h$.
2. For (u, ζ) and (u', ζ') in D_f such that $u \neq u'$, $\zeta \neq \zeta'$ holds (there is no collision for f).

We say that (D_f, D_h) is *bad* if it is not good.

Similarly, we say that a tuple of valid databases (D_f, D_g, D_h) for F_2 is *good* if and only if the following properties are satisfied.

1. For each $(u, \zeta) \in D_f$, there exist $v \in \{0,1\}^m$ and $w \in \{0,1\}^n$ such that $((u, v, \zeta), w) \in D_g$.
2. For each $((u, v, \zeta), w) \in D_g$, $(u, \zeta) \in D_f$.
3. For (u, ζ) and (u', ζ') in D_f such that $u \neq u'$, $\zeta \neq \zeta'$ holds (i.e., there is no collision for f).
4. For each $((v, \zeta), w) \in D_h$ and $(u', \zeta') \in D_f$, $\zeta \neq \zeta'$ holds (i.e., the most significant n bits of inputs to h and the outputs of f do not collide).

We say that (D_f, D_g, D_h) is *bad* if it is not good.

Intuition behind good databases. Intuitively, a database (D_f, D_h) for F_1^h is defined to be good if and only if D_f does not contain collisions (the second condition on F_1^h). The first condition on F_1^h is included so that a weird situation such as "u has been queried to f, but $(v, f(u))$ has not been queried to h for any v" will not happen for good databases. Similarly, a database (D_f, D_g, D_h) for F_2 is defined to be good if and only if D_f does not contain collisions (the third condition condition on F_2) and the least significant n bits of inputs to h do not collide with outputs of f (the fourth condition on F_2). The first and second conditions on F_2 is included so that weird situations such as "u has been queried to f, but $(u, v, f(u))$ has not been queried to g for any v" or "(u, v, ζ) has been queried to g, but u has not been queried to f" will not happen for good databases.

One-to-one correspondence for good databases. For a good database (D_f, D_g, D_h) for F_2, let $D_g \star D_h$ be the valid database for h such that $((v, \zeta), w) \in D_g \star D_h$ if and only if $((v, \zeta), w) \in D_h$ or $((u, v, \zeta), w) \in D_g$ for some u. Then $(D_f, D_g \star D_h)$ becomes a good database for F_1^h. Let us denote $(D_f, D_g \star D_h)$ by $[(D_f, D_g, D_h)]_1$. Then, it can easily be shown that the map $[\cdot]_1 : (D_f, D_g, D_h) \mapsto [(D_f, D_g, D_h)]_1 = (D_f, D_g \star D_h)$ is a bijection between the set of good databases for F_2 and that for F_1^h. Let $[\cdot]_2$ denote the inverse map of $[\cdot]_1$.

The bijections extend to (partially defined) isometries between the state spaces. Let $\mathcal{H}_{\mathcal{A}}$ be the state space of the adversary, and $\mathcal{H}_{D_f D_h}$ (resp., $\mathcal{H}_{D_f D_g D_h}$) be the state space of the databases for F_1^h (resp., F_2^h). In addition, let $V_{\mathsf{good}}^{(1)} \subset \mathcal{H}_{D_f D_h}$ (resp., $V_{\mathsf{good}}^{(2)} \subset \mathcal{H}_{D_f D_g D_h}$) be the subspace spanned by good databases. Then, the linear map from $\mathcal{H}_{\mathcal{A}} \otimes V_{\mathsf{good}}^{(1)}$ to $\mathcal{H}_{\mathcal{A}} \otimes V_{\mathsf{good}}^{(2)}$ that maps $|\eta\rangle \otimes |D_f, D_h\rangle$ to $|\eta\rangle \otimes |[D_f, D_h]_2\rangle$ for $|\eta\rangle \in \mathcal{H}_{\mathcal{A}}$ and a good database (D_f, D_h) becomes an isometry. We denote this isometry and its inverse also by $[\cdot]_2$ and $[\cdot]_1$, respectively.

Equivalent good databases. Next, we define the notion of *equivalent databases*. First, we define the notion for equivalent good databases for F_1^h. Let (D_f, D_h) be a good database for F_1^h, and let

$$S := \{\zeta \in \{0,1\}^n | \exists v, w \text{ s.t. } ((v, \zeta), w) \in D_h \text{ and } (u, \zeta) \notin D_f \text{ for all } u\}.$$

We say that another good database (D'_f, D'_h) is equivalent to (D_f, D_h) if and only if they are the same except for the output values of f, i.e., there exists a permutation π on $\{0,1\}^n$ such that

1. $\pi(\zeta) = \zeta$ for all $\zeta \in S$,
2. $(u, \zeta) \in D_f$ if and only if $(u, \pi(\zeta)) \in D'_f$, and
3. $((v, \zeta), w) \in D_h$ if and only if $((v, \pi(\zeta)), w) \in D'_h$ holds.

We define that a good database (D'_f, D'_g, D'_h) for F_2 is equivalent to another good database (D_f, D_g, D_h) in the same way, except that S is defined as $S := \{\zeta \in \{0,1\}^n | \exists v, w \text{ s.t. } ((v, \zeta), w) \in D_h\}$ and the following condition is additionally imposed.

3^+. $((u, v, \zeta), w) \in D_g$ if and only if $((u, v, \pi(\zeta)), w) \in D'_g$ hold.

As explained in Section 1.2, intuitively, two good databases are defined to be equivalent if and only if any adversary cannot distinguish them. By definition of equivalent databases, if a good database (D_f, D_g, D_h) for F_2 is equivalent to another good database (D'_f, D'_g, D'_h), then $D'_h = D_h$ holds.

Notations for state vectors. Let $|\phi_{2i-1}\rangle$ be the whole quantum state just before $\mathcal{A}$'s i-th query to F_1^h when $\mathcal{A}$ runs relative to F_1^h and h. In addition, let $|\phi_{2i}\rangle$ be the whole quantum state just before $\mathcal{A}$'s i-th query to h when $\mathcal{A}$ runs relative to F_1^h and h. Define $|\psi_{2i-1}\rangle$ and $|\psi_{2i}\rangle$ similarly when $\mathcal{A}$ runs relative to F_2 and h. For ease of notation, let $|\phi_{2q+1}\rangle$ and $|\psi_{2q+1}\rangle$ be the quantum states just before the final measurement when $\mathcal{A}$ runs relative to (F_1^h, h) and (F_2, h), respectively.

We will show that Proposition 4 follows from the proposition below.

Proposition 5. *For each $j = 1, \ldots, 2q+1$, there exist $|\phi_j^{\mathsf{good}}\rangle$, $|\phi_j^{\mathsf{bad}}\rangle$, $|\psi_j^{\mathsf{good}}\rangle$, and $|\psi_j^{\mathsf{bad}}\rangle$ that satisfy the following properties:*

1. *$|\phi_j\rangle = |\phi_j^{\mathsf{good}}\rangle + |\phi_j^{\mathsf{bad}}\rangle$ and $|\psi_j\rangle = |\psi_j^{\mathsf{good}}\rangle + |\psi_j^{\mathsf{bad}}\rangle$.*
2. *$|\phi_j^{\mathsf{good}}\rangle \in \mathcal{H}_{\mathcal{A}} \otimes V_{\mathsf{good}}^{(1)}$ and $|\psi_j^{\mathsf{good}}\rangle \in \mathcal{H}_{\mathcal{A}} \otimes V_{\mathsf{good}}^{(2)}$.*
3. *$|\phi_j^{\mathsf{good}}\rangle = \left[|\psi_j^{\mathsf{good}}\rangle\right]_1$.*
4. *There exists a complex number $a^{(j)}_{uvyzD_fD_gD_h}$ such that*

$$|\psi_j^{\mathsf{good}}\rangle = \sum_{\substack{u,v,y,z,D_f,D_g,D_h;\\ (D_f,D_g,D_h):good}} a^{(j)}_{uvyzD_fD_gD_h} |u,v\rangle |y\rangle |z\rangle \otimes |D_f, D_g, D_h\rangle \quad (14)$$

   and $a^{(j)}_{uvyzD_fD_gD_h} = a^{(j)}_{uvyzD'_fD'_gD'_h}$ if (D_f, D_g, D_h) and (D'_f, D'_g, D'_h) are equivalent, where (u, v), y, and z correspond to $\mathcal{A}$'s register to send queries, to receive answers from oracles, and for offline computations, respectively. [16]
5. *For a good database (D_f, D_g, D_h) with non-zero coefficient in $|\psi_{2i-1}^{\mathsf{good}}\rangle$ (resp., in $|\psi_{2i}^{\mathsf{good}}\rangle$), $|D_g| \leq i-1$, $|D_f| \leq 2(i-1)$, and $|D_h| \leq i-1$ hold (resp., $|D_g| \leq i$, $|D_f| \leq 2i$, and $|D_h| \leq i-1$ hold).*
6. *$\| |\phi_j^{\mathsf{bad}}\rangle \| \leq \| |\phi_{j-1}^{\mathsf{bad}}\rangle \| + O\left(\sqrt{j/2^n}\right)$ and $\| |\psi_j^{\mathsf{bad}}\rangle \| \leq \| |\psi_{j-1}^{\mathsf{bad}}\rangle \| + O\left(\sqrt{j/2^n}\right)$ hold (we regard that $\| |\phi_0^{\mathsf{bad}}\rangle \| = \| |\psi_0^{\mathsf{bad}}\rangle \| = 0$).*

[16] To be precise, we have to use the symbol (v, ζ) instead of (u, v) when $j = 2i$ since we always use the symbol $v||\zeta$ to denote an input to h. However, here we use (u, v) to simplify notations. In the proof we use the symbol $a^{(2i)}_{v\zeta yzD_fD_gD_h}$ instead of $a^{(2i)}_{uvyzD_fD_gD_h}$.

Intuitive interpretation of Proposition 5. The first and second properties show that $|\phi_j\rangle$ and $|\psi_j\rangle$ are divided into good and bad components. The third property shows that the good component of $|\phi_j\rangle$ matches to that of $|\psi_j\rangle$ through the isometry $[\cdot]_1$, which intuitively means that $\mathcal{A}$ cannot distinguish the two oracles as long as databases are good. The fourth property shows that the coefficients of equivalent databases are perfectly equal, which intuitively means that $\mathcal{A}$ cannot distinguish equivalent good databases. The fifth property shows the upper bound of the size of databases. The sixth property shows that the chance for good databases change to bad is very small at each query.

Overview of the proof of Proposition 5. The proposition is shown by induction on j. The claim for $j = 1$ obviously holds by setting $|\phi_1^{\mathsf{bad}}\rangle = |\psi_1^{\mathsf{bad}}\rangle = 0$. Inductive steps are separated into two cases.

> (Online queries): If the claim for $j = 2i - 1$ (i.e., before the i-th query to F_1^h or F_2) holds, then the claim for $j = 2i$ (i.e., after the query) holds.
> (Offline queries): If the claim for $j = 2i$ (i.e., before the i-th query to h) holds, then the claim for $j = 2i + 1$ (i.e., after the query) holds.

Proof for online queries. Recall that $O_{F_1^h}$ (resp., O_{F_2}) are decomposed as $O_{F_1^h} = \mathsf{RstOE}_f^* \cdot \mathsf{RstOE}_h \cdot \mathsf{RstOE}_f$ (resp., $O_{F_2} = \mathsf{RstOE}_f^* \cdot \mathsf{RstOE}_g \cdot \mathsf{RstOE}_f$). We show that Properties 1–6 listed in Proposition 5 hold at each action of RstOE_f, RstOE_h (resp., RstOE_g), and RstOE_f^*. A state vector after an action of RstOE can be decomposed into three components. [17]

(i) The one that was (pre-)good before the action and still remains (pre-)good.
(ii) The one that was (pre-)good before the action but changed to bad.
(iii) The one that was already bad before the action.

Roughly speaking, we define (i) to be a new good vector, and the sum of (ii) and (iii) to be a new bad vector.[18] Then Properties 1 and 5 of Proposition 5 can easily be shown.

Intuitively, we defined good databases so that the behavior of the oracle of F_1^h on good databases will be the same for that of F_2 on the corresponding good databases. Thus we can show that Property 3 still holds for the new good vectors by keeping track of how the coefficients of basis vectors change, using Proposition 3.

The intuition for the proof of Property 4 is as follows. Let $\mathcal{DB}_0 := (D_f, D_g, D_h)$ and $\mathcal{DB}_1 := (D'_f, D'_f, D'_h)$ (resp., $\widetilde{\mathcal{DB}}_0 := (\tilde{D}_f, \tilde{D}_g, \tilde{D}_h)$ and $\widetilde{\mathcal{DB}}_1 := (\tilde{D}'_f, \tilde{D}'_f, \tilde{D}'_h)$) be equivalent good databases in $|\psi_{2i-1}^{\mathsf{good}}\rangle$ (resp., $|\psi_{2i}^{\mathsf{good}}\rangle$). In addition, by p_{ij} we ambiguously denote the "probability" that $\mathcal{DB}_i$ changes to $\widetilde{\mathcal{DB}}_j$ for $i, j \in \{0, 1\}$ (p_{ij} has the information on the ratio of the coefficient of the vector corresponding to $\mathcal{DB}_i$ and that of $\widetilde{\mathcal{DB}}_j$). Then we can show $p_{ij} = p_{i'j'}$ holds for all

[17] Pre-good databases are defined in the complete proof of Proposition 5 presented in Section C of this paper's full version [20].

[18] To be more precise, we sometimes include small "good" terms into the new bad vector so that the analysis will be easier.

$(i,j),(i',j') \in \{0,1\} \times \{0,1\}$ by using symmetry of equivalent databases and Proposition 3. Since the coefficients corresponding to $\mathcal{DB}_0$ and $\mathcal{DB}_1$ are equal due to Property 4 on $|\psi^{\mathsf{good}}_{2i-1}\rangle$, this implies that Property 4 also holds for $|\psi^{\mathsf{good}}_{2i}\rangle$.

Property 6 is proven by showing the norm of the component (iii) is in $O(\sqrt{i/2^n})$. Intuitively, this corresponds to showing the probability that the event coll in Section 1.2 happens at the query is $O(i/2^n)$. We carefully prove it by using Proposition 3, taking into account that records in databases may be deleted or overwritten.

Proof for offline queries. The proof for offline queries are similar[19], except that showing $\left\|\text{(iii)}\right\| \leq O(\sqrt{i/2^n})$ corresponds to showing $\Pr[\mathsf{hit}_i] \leq O(i/2^n)$ in Section 1.2. See the explanations around page 11 for the intuition on $\Pr[\mathsf{hit}_i] \leq O(i/2^n)$. To formally prove the bound, we use the inductive hypothesis that Property 4 holds for $j = 2i$.

See Section C of this paper's full version [20] for a complete proof.

Proof (of Proposition 4). Let tr_{D1} (resp., tr_{D2}) denote the partial trace operations over the quantum states of the databases for (F_1^h, h) (resp., (F_2, h)). Then

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{dist}}_{F_1^h,F_2}(\mathcal{A}) &\leq \mathsf{td}\left(\mathsf{tr}_{D1}(|\phi_{2q+1}\rangle\langle\phi_{2q+1}|), \mathsf{tr}_{D2}(|\psi_{2q+1}\rangle\langle\psi_{2q+1}|)\right) \\
&\leq \mathsf{td}\left(\mathsf{tr}_{D1}(|\phi^{\mathsf{good}}_{2q+1}\rangle\langle\phi^{\mathsf{good}}_{2q+1}|), \mathsf{tr}_{D2}(|\psi^{\mathsf{good}}_{2q+1}\rangle\langle\psi^{\mathsf{good}}_{2q+1}|)\right) \qquad (15) \\
&\quad + \left\||\phi^{\mathsf{bad}}_{2q+1}\rangle\right\| + \left\||\psi^{\mathsf{bad}}_{2q+1}\rangle\right\| \qquad (16)
\end{aligned}$$

holds. By Property 3 of Proposition 5, the term (15) is equal to zero. In addition, $(16) \leq \sum_{1\leq j\leq 2q+1} O\left(\sqrt{j/2^n}\right) + \sum_{1\leq j\leq 2q+1} O\left(\sqrt{j/2^n}\right) \leq O\left(\sqrt{q^3/2^n}\right)$ follows from Property 6 of Proposition 5. Hence Proposition 4 follows. □

5 Quantum Security Proofs for HMAC and NMAC

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

Proposition 6. *Let $h : \{0,1\}^{m+n} \to \{0,1\}^n$ be a quantum random oracle. Assume $m \geq n$. Suppose that the padding function pad for the Merkle-Damgård construction is injective and there exists a function $\mathsf{p} : \mathbb{Z}_{\geq 0} \to \{0,1\}^*$ such that $\mathsf{pad}(M) = M||\mathsf{p}(|M|)$[20]. Let $\mathcal{A}$ be a quantum adversary that runs relative to two quantum oracles $\mathcal{O}^h$ and h [21] such that (i) $|\mathsf{pad}(M)| \leq m \cdot \ell$ for arbitrary M that $\mathcal{A}$ queries to $\mathcal{O}^h$ when $\mathcal{O}^h$ is HMAC^h_K or $\mathsf{NMAC}^h_{K_1,K_2}$, and (ii) $\mathcal{A}$ makes at most Q queries to $\mathcal{O}^h$ and q_h queries to h. Then $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathsf{HMAC}^h_K}(\mathcal{A}) \leq O\left(\sqrt{\frac{(q_h+Q)^3\ell^5}{2^n}} + \frac{q_h+Q\ell}{2^{k/2}}\right)$ and $\mathbf{Adv}^{\mathrm{qPRF}}_{\mathsf{NMAC}^h_{K_1,K_2}}(\mathcal{A}) \leq O\left(\sqrt{\frac{(q_h+Q)^3\ell^5}{2^n}}\right)$ hold.*

[19] Actually the proof for offline queries are even simpler because the offline oracle is just a single random oracle h while the online oracles consist of two random functions.

[20] These conditions are satisfied for usual concrete hash functions such as SHA-2. Recall that $(\{0,1\}^m)^+$ is the set of bit strings of length positive multiple of m bits.

[21] $\mathcal{O}^h$ will be HMAC^h_K, $\mathsf{NMAC}^h_{K_1,K_2}$, or a random function.

Recall that HMAC^h_K (resp., $\mathsf{NMAC}^h_{K_1,K_2}$) is the composition of the functions $\mathsf{MD}^h(IV, K_{in}||\cdot)$ and $\mathsf{MD}^h(IV, K_{out}||\cdot)$ (resp., $\mathsf{MD}^h(K_1,\cdot)$ and $\mathsf{MD}^h(K_2,\cdot)$). Let us call the first (resp., second) function the *inner function* (resp., *outer function*). In addition, let $\mathsf{MD}'^h : \{0,1\}^n \times (\{0,1\}^m)^+ \to \{0,1\}^n$ be the function that is defined in the same way as MD^h but without padding. Then, to prove Proposition 6, it suffices to prove the claim in the case that the inner function of HMAC^h_K (resp., $\mathsf{NMAC}^h_{K_1,K_2}$) is replaced with $\mathsf{MD}'^h(IV, K_{in}||\cdot)$ (resp., $\mathsf{MD}'^h(K_1,\cdot)$) and the lengths of messages queried by $\mathcal{A}$ is always a multiple of m and at most $\ell \cdot m$, since this change does not decrease adversaries' ability to distinguish.

Thus, in what follows, we prove Proposition 6 in the case where HMAC^h_K and $\mathsf{NMAC}^h_{K_1,K_2}$ are modified as above. We show it by introducing $(2\ell+2)$ games $G_{0,H}, G_{0,N}, G_i$ $(1 \le i \le \ell)$, G'_i $(1 \le i \le \ell)$.

Game $G_{0,H}$. This is the game that the adversary is given oracle access to the quantum oracle of HMAC^h_K, in addition to h.

Game $G_{0,N}$. This is the game that the adversary is given oracle access to the quantum oracle of $\mathsf{NMAC}^h_{K_1,K_2}$, in addition to h.

Game G_i for $1 \le i \le \ell$. In the game G_i, the adversary is given quantum oracle access to the function H^h_i (in addition to h) that is defined as follows. Let $M := M[1]||\cdots||M[j]$ ($M[t] \in \{0,1\}^m$ for each t) be an input message for H^h_i.

1. If $j < i$, $H^h_i(M) := g_j(M)$ for a random function $g_j : \{0,1\}^{mj} \to \{0,1\}^n$.
2. If $j = i$, $H^h_i(M) := f_{out}(f_i(M))$ for a random function $f_i : \{0,1\}^{mi} \to \{0,1\}^n$ and $f_{out} : \{0,1\}^n \to \{0,1\}^n$.
3. If $j > i$, first $S_i := f_i(M[1]||\cdots||M[i])$ is computed, and then $S_t := h(M[t]||S_{t-1})$ is iteratively computed for $i < t \le j$, and finally $H^h_i(M)$ is set as $H^h_i(M) := f_{out}(S_j)$.

See also Figure 6.

Game G'_i for $1 \le i \le \ell$. In the game G'_i, the adversary is given quantum oracle access to the function H'^h_i (in addition to h) that is defined as follows. Let $M := M[1]||\cdots||M[j]$ ($M[t] \in \{0,1\}^m$ for each t) be an input for H'^h_i.

1. If $j \le i$, $H'^h_i(M) := g_j(M)$ for a random function $g_j : \{0,1\}^{mj} \to \{0,1\}^n$.
2. If $j > i$, first $S_i := f_i(M[1]||\cdots||M[i])$ is computed, and then $S_t := h(M[t]||S_{t-1})$ is iteratively computed for $i < t \le j$, and finally $H'^h_i(M)$ is set as $H'^h_i(M) := f_{out}(S_j)$. Here, $f_i : \{0,1\}^{mi} \to \{0,1\}^n$ and $f_{out} : \{0,1\}^n \to \{0,1\}^n$ are random functions.

See also Figure 7. Since the lengths of messages queried by $\mathcal{A}$ is at most $m \cdot \ell$, G'_ℓ becomes the ideal game that $\mathcal{A}$ runs relative to a random function and h.

For the distinguishing advantage between $G_{0,N}$ and G_1 and the distinguishing advantage between $G_{0,H}$ and G_1, the following two lemmas hold.

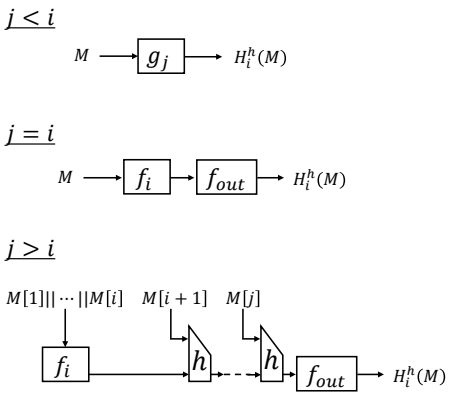


Fig. 6: $H_i^h(M)$ in game G_i.

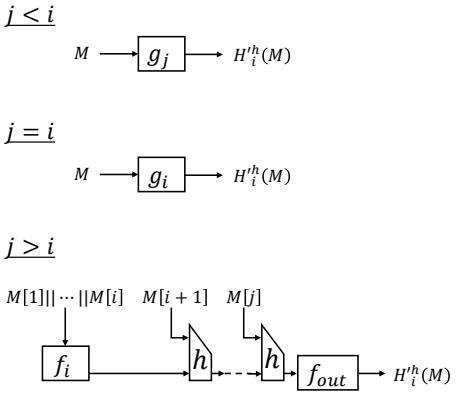


Fig. 7: $H_i^{'h}(M)$ in game G'_i.

Lemma 4 ($G_{0,N}$ **and** G_1)**.** *It holds that* $\mathbf{Adv}^{\mathrm{dist}}_{(\mathsf{NMAC}^h_{K_1,K_2},h),(H_1^h,h)}(\mathcal{A})$ *is in* $O\big(\sqrt{(q_h+Q\ell)^3/2^n}\big)$.

Lemma 5 ($G_{0,H}$ **and** G_1)**.** $\mathbf{Adv}^{\mathrm{dist}}_{(\mathsf{HMAC}^h_K,h),(H_1^h,h)}(\mathcal{A})$ *is in* $O(\sqrt{(q_h+Q\ell)^3/2^n} + (q_h+Q\ell)/2^{k/2})$.

It is straightforward to show that these lemmas follow from Lemma 1, Lemma 2, and Proposition 4. See Section D and Section E of this paper's full version [20] for complete proofs.

For the distinguishing advantage between G_i and G'_i for $1 \le i \le \ell$, the following lemma holds.

Lemma 6 (G_i **and** G'_i)**.** $\mathbf{Adv}^{\mathrm{dist}}_{(H_i^h,h),(H_i^{'h},h)}(\mathcal{A})$ *is in* $O(\sqrt{q^3\ell^3/2^n})$*, where* $q = \max\{Q, q_h\}$.

Here we provide a rough proof overview. See Section F of this paper's full version [20] for details.

Proof Overview. First, let us slightly modify the definition of $H_i^{'h}$. For a message $M = M[1]||\cdots||M[i]$ of length $m \cdot i$, the value $H_i^{'h}(M)$ was defined as $H^{'h}(M) := g_i(M)$ for a random function g_i, but here we re-define $H_i^{'h}(M) := f'_{out}(M, f_i(M))$, where $f'_{out} : \{0,1\}^{mi} \times \{0,1\}^n \to \{0,1\}^n$ is another random function. This modification does not change the distribution of $H_i^{'h}$ since f'_{out} is random.

Our proof strategy for Lemma 6 is similar to that for Proposition 4, and we use RstOE to show the indistinguishability. In fact proving Lemma 6 is easier than proving Proposition 4 because the following difference exists between Proposition 4 and Lemma 6.

1. In the proof of Proposition 4, a function to which adversaries can *directly* query in one construction (i.e., h in F_1^h) is replaced with another function to

which adversaries can query *only indirectly* in the other construction (i.e., g in F_2).
2. On the other hand, in Lemma 6, a function to which adversaries can query *only indirectly* in one construction (i.e., f_{out} in H_i^h of G_i) is replaced with another function to which adversaries can query *only indirectly* in the other construction (i.e., f'_{out} in $H_i^{'h}$ of G'_i).

In the proof of Proposition 4, we had to assure that the probability that an adversary directly queries to h a value that is recorded in a database is very small (i.e., the probability of the bad event hit in Section 1.2 is very small). This is the reason that we introduced the notion of equivalent databases. On the other hand, in Lemma 6, adversaries can query to both of f_{out} and f'_{out} only indirectly (adversaries do not have full control on inputs to f_{out} and f'_{out}). In particular, we can define bad events in Lemma 6 in such a way that whether they happen or not do not depend on the values of $\mathcal{A}$'s queries, and their probability can be bounded by using the randomness of outputs of random functions (like coll in Section 1.2). Therefore we do not have to introduce the notion of equivalent databases in Lemma 6. Hence it easier to prove Lemma 6 than to prove Proposition 4.

For the distinguishing advantage between G'_i and G_{i+1} for $1 \leq i < \ell$, the following lemma holds.

Lemma 7 (G'_i and G_{i+1}). $\mathbf{Adv}^{\mathrm{dist}}_{(H_i^{'h},h),(H_{i+1}^h,h)}(\mathcal{A})$ *is in* $O\left(\sqrt{(q_h + Q\ell)^3/2^n}\right)$.

Proof. Let $f_{i+1}^{'h} : \{0,1\}^{m(i+1)} \to \{0,1\}^n$ be the function defined by $f_{i+1}^{'h}(M[1]|| \cdots ||M[i+1]) := h(M[i+1]||f_i(M[1]|| \cdots ||M[i]))$.

For an adversary $\mathcal{A}$ to distinguish $(H_i^{'h}, h)$ from (H_{i+1}^h, h) that makes at most Q quantum queries to $H_i^{'h}$ or H_{i+1}^h and at most q_h quantum queries to h, we construct another adversary $\mathcal{B}$ to distinguish $(f_{i+1}^{'h}, h)$ and (f_{i+1}, h) by making $O(Q)$ queries to $f_{i+1}^{'h}$ or f_{i+1} and $O(q_h + Q\ell)$ queries to h, as follows.

$\mathcal{B}$ is given a quantum oracle access to $\mathcal{O}^h$, which is $f_{i+1}^{'h}$ or f_{i+1}, in addition to a quantum oracle access to h. First, $\mathcal{B}$ chooses functions $\tilde{g}_j : \{0,1\}^{jm} \to \{0,1\}^n$ for $j = 1, \ldots, i$ and $f_{out} : \{0,1\}^n \to \{0,1\}^n$ uniformly at random, and runs $\mathcal{A}$. When $\mathcal{A}$ makes a query to the second oracle (which is supposed to be h), $\mathcal{B}$ responds by querying to h. When $\mathcal{A}$ queries $M = M[1]|| \cdots ||M[j]$ to the first oracle (which is supposed to be $H_i^{'h}$ or H_{i+1}^h), $\mathcal{B}$ responds to $\mathcal{A}$ as follows:

1. If $j \leq i$, $\mathcal{B}$ computes $T = \tilde{g}_j(M)$ by itself, and responds to $\mathcal{A}$ with T.
2. If $j > i$, $\mathcal{B}$ computes $S_{i+1} := \mathcal{O}^h(M)$, $S_u := h(M[u]||S_{u-1})$ for $u = i+2, \ldots, j$, and $T := f_{out}(S_j)$, by making queries to $\mathcal{O}^h$ and h. Then $\mathcal{B}$ responds to $\mathcal{A}$ with T.

Finally, $\mathcal{B}$ returns $\mathcal{A}$'s output as its own output.

Then $\mathcal{B}$ perfectly simulates $H_i^{'h}$ or H_{i+1}^h depending on whether $\mathcal{O}^h = f_{i+1}^{'h}$ or $\mathcal{O}^h = f_{i+1}$, which implies that $\mathbf{Adv}^{\mathrm{dist}}_{(H_i^{'h},h),(H_{i+1}^h,h)}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{dist}}_{(f_{i+1}^{'h},h),(f_{i+1},h)}(\mathcal{B})$. In addition, $\mathcal{B}$ makes at most $O(Q)$ quantum queries to $f_{i+1}^{'h}$ or f_{i+1} and $O(q_h +$

$Q\ell$) quantum queries to h. Therefore

$$\mathbf{Adv}^{\mathrm{dist}}_{(H_i'^h,h),(H_{i+1}^h,h)}(\mathcal{A}) = \mathbf{Adv}^{\mathrm{dist}}_{(f_{i+1}'^h,h),(f_{i+1},h)}(\mathcal{B}) \leq O\left(\sqrt{\frac{(q_h + Q\ell)^3}{2^n}}\right) \quad (17)$$

follows from Proposition 4. □

Proof (of Proposition 6). The claim of the proposition immediately follows from Lemma 4, Lemma 5, Lemma 6, and Lemma 7. □

Acknowledgements. The second author was supported in part by JSPS KAKENHI Grant Number JP20K11675.

References

1. Alagic, G., Majenz, C., Russell, A., Song, F.: Quantum-access-secure message authentication via blind-unforgeability. In: EUROCRYPT 2020, Proceedings, Part III. pp. 788–817 (2020)
2. Alagic, G., Russell, A.: Quantum-secure symmetric-key cryptography based on hidden shifts. In: EUROCRYPT 2017, Proceedings, Part III. pp. 65–93 (2017)
3. ANSI: Retail Financial Services Symmetric Key Management Part 1: Using Symmetric Techniques. ANSI X9.24-1-2017 (2017)
4. Bellare, M., Canetti, R., Krawczyk, H.: Keying hash functions for message authentication. In: CRYPTO 1996, Proceedings. pp. 1–15 (1996)
5. Bellare, M., Kilian, J., Rogaway, P.: The security of the cipher block chaining message authentication code. In: CRYPTO 1994, Proceedings. pp. 341–358 (1994)
6. Bindel, N., Hamburg, M., Hövelmanns, K., Hülsing, A., Persichetti, E.: Tighter proofs of CCA security in the quantum random oracle model. In: TCC 2019, Proceedings, Part II. pp. 61–90 (2019)
7. Black, J., Rogaway, P.: CBC macs for arbitrary-length messages: The three-key constructions. In: CRYPTO 2000, Proceedings. pp. 197–215 (2000)
8. Black, J., Rogaway, P.: A block-cipher mode of operation for parallelizable message authentication. In: EUROCRYPT 2002, Proceedings. pp. 384–397 (2002)
9. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: ASIACRYPT 2011, Proceedings. pp. 41–69 (2011)
10. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. In: EUROCRYPT 2013, Proceedings. pp. 592–608 (2013)
11. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. In: LATIN 1998, Proceedings. pp. 163–169 (1998)
12. Chiesa, A., Manohar, P., Spooner, N.: Succinct arguments in the quantum random oracle model. In: TCC 2019, Proceedings, Part II. pp. 1–29 (2019)
13. Coron, J., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-Damgård revisited: How to construct a hash function. In: CRYPTO 2005, Proceedings. pp. 430–448 (2005)
14. Czajkowski, J., Hülsing, A., Schaffner, C.: Quantum indistinguishability of random sponges. In: CRYPTO 2019, Proceedings, Part II. pp. 296–325 (2019)
15. Garg, S., Yuen, H., Zhandry, M.: New security notions and feasibility results for authentication of quantum data. In: CRYPTO 2017, Proceedings, Part II. pp. 342–371 (2017)

16. Gazi, P., Pietrzak, K., Rybár, M.: The exact prf-security of NMAC and HMAC. In: CRYPTO 2014, Proceedings, Part I. pp. 113–130 (2014)
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: ACM STOC 1996, Proceedings. pp. 212–219 (1996)
18. Hosoyamada, A., Iwata, T.: 4-round luby-rackoff construction is a qprp. In: ASIACRYPT 2019, Proceedings, Part I. pp. 145–174 (2019)
19. Hosoyamada, A., Iwata, T.: 4-round Luby-Rackoff construction is a qPRP: Tight quantum security bound. IACR Cryptol. ePrint Arch. 2019/243, version 20200720:101411 (2020), (A revised version of [18].)
20. Hosoyamada, A., Iwata, T.: On tight quantum security of HMAC and NMAC in the quantum random oracle model (2021), to appear on IACR Cryptology ePrint Archive
21. Hosoyamada, A., Yasuda, K.: Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions. In: ASIACRYPT 2018, Proceedings, Part I. pp. 275–304 (2018)
22. Iwata, T., Kurosawa, K.: OMAC: one-key CBC MAC. In: FSE 2003, Proceedings. pp. 129–153 (2003)
23. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Breaking symmetric cryptosystems using quantum period finding. In: CRYPTO 2016, Proceedings, Part II. pp. 207–237 (2016)
24. Liu, Q., Zhandry, M.: On finding quantum multi-collisions. In: EUROCRYPT 2019, Proceedings, Part III. pp. 189–218 (2019)
25. Liu, Q., Zhandry, M.: Revisiting post-quantum fiat-shamir. In: CRYPTO 2019, Proceedings, Part II. pp. 326–355 (2019)
26. NIST: Secure Hash Standard (SHS). NIST FIPS PUB 180-4 (2015)
27. NIST: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. NIST FIPS PUB 202 (2015)
28. NIST: Announcing request for nominations for public-key post-quantum cryptographic algorithms. National Institute of Standards and Technology (2016)
29. Patarin, J.: The "coefficients H" technique. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers. Lecture Notes in Computer Science, vol. 5381, pp. 328–345. Springer (2008)
30. Saito, T., Xagawa, K., Yamakawa, T.: Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In: EUROCRYPT 2018, Proceedings, Part III. pp. 520–551 (2018)
31. Sanchez, I.A., Fischer, D.: Authenticated encryption in civilian space missions: context and requirements. DIAC - Directions in Authenticated Ciphers (2012)
32. Song, F., Yun, A.: Quantum security of NMAC and related constructions - PRF domain extension against quantum attacks. In: CRYPTO 2017, Proceedings, Part II. pp. 283–309 (2017)
33. Zhandry, M.: How to construct quantum random functions. In: FOCS 2012, Proceedings. pp. 679–687. IEEE (2012)
34. Zhandry, M.: How to record quantum queries, and applications to quantum indifferentiability. In: CRYPTO 2019, Proceedings, Part II. pp. 239–268 (2019)