Tighter Post-quantum Proof for Plain FDH, PFDH and GPV-IBE*

Yu Liu[1], Haodong Jiang[2,3], and Yunlei Zhao[1]

[1] School of Computer Science, Fudan University, Shanghai, China
yu_liu21@m.fudan.edu.cn, ylzhao@fudan.edu.cn
[2] State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, Henan, China
haodong2020@iscas.ac.cn
[3] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, Henan, China

Abstract. In CRYPTO 2012, Zhandry developed generic semi-constant oracle technique and proved security of an identity-based encryption scheme, GPV-IBE, and full domain hash (FDH) signature scheme in the quantum random oracle model (QROM). However, the reduction provided by Zhandry incurred a quadratic reduction loss. In this work, we provide a much tighter proof, with linear reduntion loss, for the FDH, probabilistc FDH (PFDH), and GPV-IBE in the QROM. Our proof is based on the measure-and-reprogram technique developed by Don, Fehr, Majenz and Schaffner.

Keywords: Quantum random oracle · Full domain hash · Identity-based encryption.

1 Introduction

1.1 Background

The (Quantum) Random Oracle Model. As is often the case, security proofs of practical cryptographic schemes are given in the random oracle model (ROM) [3], where a hash function is idealized as a publicly accessible oracle that evaluates a random function. However in 2011, Boneh et al. [5] pointed out that the ROM is not sufficient when considering security against quantum adversaries, who may be able to evaluate the oracle in superposition. Considering this fact, they proposed a new model named the quantum(-accessible) random oracle model (QROM) and called for new techniques to obtain the QROM counterparts of the existing security results in the ROM.

* This work differs from the previous version in that (1) the MaR predicate additionally includes verification that the final output m/id^* is never queried to the signing/extraction oracle before; (2) the term $q_H + q_S$ in the security bound is replaced with q_H.

Identity-Based Encryption in QROM. The identity-based encryption (IBE) was first envisioned by Shamir [19] and realized under various assumptions [6,7], among which the most efficient post-quantum one is GPV-IBE proposed by Gentry, Peikert and Vaikuntanathan[13]. Zhandry [20] first gave a security proof for generic PSF-based IBE in the QROM with quadratic loss. Katsumata et al. [15] provided a much tighter reduction from the security of GPV-IBE to the LWE assumption while only applying to certain lattice-based PSFs.

(Probabilistic) Full Domain Hash in QROM. In 1993, Bellare and Rogaway [3] formalized the well-known "hash-and-sign" paradigm for digital signature schemes, using the random oracle. Specifically, given a trapdoor permutation f and a random hash function H with the same range as f, the signature of a message m is defined as $f^{-1}(H(m))$. This signature scheme was subsequently called "Full Domain Hash" or FDH. To obtain a better security bound, Bellare and Rogaway[4] designed a new scheme, the probabilistic scheme (PSS), and then in 2002, Coron[10] described a variant of PSS, named as probabilistic full domain hash (PFDH), for the sake of simplicity. Zhandry[20] gave a reduction from the security of FDH to the onewayness of the underlying trapdoor permutation with $\epsilon' \approx \epsilon^2/(q_H + q_S)^4$ and $T' \approx T + (q_H + q_S)^2 \cdot \mathsf{poly}(\lambda)$, where q_H denotes the number of hash queries, q_S denotes the number of signing queries, λ denotes the security parameter, and poly denotes some fixed polynomial. If we consider the tightness of the reduction, the proof provided by Zhandry is not satisfactory. Indeed, Zhandry left it as an open problem to give a tighter reduction for the FDH, as well as the IBE. Moreover, NIST announced a new call for additional digital signature schemes for the PQC Standardization Process, especially schemes that are not based on structured lattices [18]. That means FDH and its variants can be promising candidates and thus their post-quantum security is worth reconsidered.

The Measure-and-Reprogram Technique. Don et al. [12] first introduced the measure-and-reprogram technique to reprogram the QROM adaptively at one input. More precisely, for any oracle quantum algorithm $\mathcal{A}^H$ making q quantum calls to a random oracle H and finally outputting a pair (x, z) so that some predicate $V(x, H(x), z)$ is satisfied, they showed the existence of a simulator $\mathcal{S}$ that mimics the random oracle, and then reprograms $H(x)$ to a given Θ so that z output by $\mathcal{A}^H$ now satisfies $V(x, \Theta, z)$, except with a multiplicative $O(q^2)$ loss in probability (plus a negligible additive loss). Then the result is further improved in [11] by Don et al, with a cleaner bound, i.e. a multiplicative $(2q+1)^2$ loss.

1.2 Our Contribution

We resolve the issues left by Zhandry [20] of improving the reduction to first-order in the adversary's advantage for the IBE scheme and Full Domain Hash in the QROM.

- We give a reduction from the IND-ID-CPA-security of generic PSF-based IBE to the IND-CPA-security of the encryption scheme from which it is constructed in the QROM, with a $(2q+1)^2$ loss in advantage. We note that this technique is general and can apply to the random oracle hierarchical IBE schemes of Cash et al. [8] and Agrawal et al. [1].
- We also give a reduction from the UF-CMA-security of the FDH and PFDH signature schemes to the one-way security of the trapdoor permutation in the QROM, with a $(2q+1)^2$ loss in advantage. We also note that if the trapdoor permutation has some sort of homomorphic property, the security bound can be further tightened with $O(q_H^2)$ being replaced by $O(q_S)$, which is a significantly better result in practice since q_S is usually much smaller than q_H.

1.3 Technical Overview

Security Proofs in Classical ROM. We briefly recall the original security proof of FDH in the classical ROM given by Bellare and Rogaway [3] and give an insight into the role that a random oracle plays in the reduction algorithm. In the security proof, the reduction algorithm guesses $i \in [q_H]$ such that the adversary's i-th hash query is the m^* of its final forgery (m^*, σ^*), where q_H denotes the number of hash queries made by the adversary. Then for all but the i-th hash query, the reduction algorithm programs $H(m)$ by picking a random $x \leftarrow \mathsf{Dom}_f$ and returning $f_{pk}(x)$ and for the i-th query, it programs $H(m^*)$ to be the challenge $y := f_{pk}(x)$ to be inverted. Then, if the guess is correct and the forgery is valid, from $f_{pk}(\sigma^*) = H(m^*) = y$, the reduction algorithm can simply outputs σ^* and hopefully inverts $f_{sk}^{-1}(y)$. The reduction loses a factor of $1/q_H$ and the security proof for PFDH and PSF-based IBE in the ROM can be done similarly.

Security Proofs in QROM in [20]. Since a quantum adversary may evaluate a hash function on a superposition of inputs in a single query, the above reduction in the ROM cannot simply carry over to the QROM. To overcome the obstacle, Zhandry [20] developed generic semi-constant oracle technique. The semi-constant distribution with a parameter $0 < \lambda < 1$ is a distribution over functions from $\mathcal{X}$ to $\mathcal{Y}$ such that a function chosen from this distribution gives some fixed value y for uniformly random λ-fraction of all inputs, and behaves as a truly random function for the rest. Zhandry argued that an oracle drawn from the semi-constant distribution with parameter λ cannot be distinguished from a truly random one by an adversary that makes q_H queries with an advantage greater than $\frac{8}{3}q_H^4\lambda^2$. In the security proof, the reduction partitions the set of identities/messages $\mathcal{M}$ into two sets: $\mathcal{X}$ and $\mathcal{M}/\mathcal{X}$, where $\mathcal{X}$ is a uniformly random λ-fraction of $\mathcal{M}$. The basic idea is to plug the challenge c into this small fraction of inputs to the oracle. Then the adversary behaves as though the oracle is random. By appropriately setting λ, the reduction algorithm succeeds with probability $\epsilon' \approx \epsilon^2/(q_H + q_S)^4$, which is a quadratic loss.

Our Security Proofs in QROM. Our reduction is based on the measure-and-reprogram technique by Don, Fehr, Majenz and Schaffner [11,12]. For any oracle quantum algorithm $\mathcal{A}^H$ making q quantum calls to a random oracle H and finally outputting a pair (x, z) so that some predicate $V(x, H(x), z)$ is satisfied, the theorem states that there exists a simulator $\mathcal{S}$ that mimics the random oracle, and then reprograms $H(x)$ to a given Θ so that z output by $\mathcal{A}^H$ now satisfies $V(x, \Theta, z)$, except with a multiplicative $(2q^2 + 1)$ loss in probability. From any FDH forger $\mathcal{A}$ that tries to produce a forgery (m^*, σ^*), we obtain a reduction algorithm $\mathcal{S}$ that extracts m^* from $\mathcal{A}$ and uses a challenge $y = f_{pk}(x)$ to reprogram the RO, so that σ^* output by $\mathcal{A}$ will be a correct reply with respect to y with a probability not much smaller than the probability that $\mathcal{A}$ succeeds in forging. Concretely, the reduction loss is exactly a multiplicative $(2q^2 + 1)$. We achieve the same result with respect to PFDH and generic PSF-based IBE following similar discussion.

1.4 Related Work

Boneh et al. [5] introduced QROM and showed certain circumstances in which security in the classical RO implies security in the QROM . Zhandry [20] developed generic semi-constant technique and proved the security of GPV-IBE and FDH in the QROM. Katsumata et al. [15] provided much tighter security proofs for the GPV-IBE in the QROM in the single-challenge setting and also a multi-challenge tight variant of GPV-IBE that is secure both in the ROM and QROM. However, their reduction relies on certain properties of lattice-based PSFs and thus does not apply to generic PSF-based schemes. The measure-and-reprogram technique was developed and improved by Don et al. [11,12] originally to prove security of the Fiat Shamir transform in the QROM.

1.5 Comparison with Concurrent Results.

In concurrent and independent work [16], Kosuge and Xagawa showed a similar result based on measure-and-reprogram technique. However, our work differs from [16] in the following aspects. In [16], what they consider is the probabilistic hash-an-sign with retry based on non-PSF TDFs, while we focus on the plain FDH and PFDH as in [20]. We also show that if the trapdoor permutation has some sort of homomorphic property, the security bound can be further tightened with $O(q_H^2)$ being replaced by $O(q_S)$, which is a significantly better result in practice since q_S is usually much smaller than q_H. Besides, we also give QROM proofs for IBE and HIBE.

2 Preliminaries

For strings a and b, we denote the concatenation of these strings by $a||b$. For a positive integer n, we denote the set of integers ranging from 1 to n by $[n] := \{1, \cdots, n\}$. For a function f, we use the notation Dom_f and Ran_f to denote its domain and range. $\Pr[P : G]$ is the probability that the predicate holds true when free variables in P are assigned according to the program in G. If S is a finite set, we denote by $x \xleftarrow{\$} S$ the operation of sampling a value uniformly at random from the set S and assigning it to the variable x. For a quantum or randomized classical algorith $\mathcal{A}$, we denote $y \xleftarrow{\$} \mathcal{A}(x)$ to mean that $\mathcal{A}$ outputs y on input x and denote $y \in \mathcal{A}(x)$ to mean that y is in the support of $\mathcal{A}(x)$.

2.1 Cryptographic Primitives

Definition 1. *A preimage sampleable function (PSF) consists of four algorithms* $\mathsf{F} = (\mathsf{F.Gen}, \mathsf{F.Sample}, f, f^{-1})$ *where* $\mathsf{F.Gen}$ *generates secret/public keys* (sk, pk)*,* f_{pk} *is a function,* $\mathsf{F.Sample}$ *samples* x *from a distribution* D *such that* $f_{pk}(x)$ *is uniform, and* $f_{sk}^{-1}(y)$ *samples from* D *conditioned on* $f_{pk}(x) = y$*.*

Definition 2. *A trapdoor permutation (TDP) is a triple of algorithms* $\mathsf{F} = (\mathsf{Gen}, f, f^{-1})$ *where* Gen *generates secret/public keys* (sk, pk)*,* f_{pk} *is a permutation, and* f_{sk}^{-1} *is its inverse.*

We use the following security notion for trapdoor permutations. We say that a trapdoor permutation $\mathsf{F} = (\mathsf{Gen}, f, f^{-1})$ is hard to invert (one-way) if given pk and $y := f_{pk}(x)$ for a uniform x, it is hard to compute x. More formally, it is (t, ϵ)-hard to invert if for any adversary $\mathcal{A}$ running in time t, $\Pr[\mathcal{A}(pk, f_{pk}(x)) = x] \leq \epsilon$, where the probability is taken over $(sk, pk) \leftarrow \mathsf{Gen}, x \leftarrow \mathsf{Dom}_{f_{pk}}$, and the random coin tosses of $\mathcal{A}$.

Definition 3. *An identity-based encryption (IBE) scheme is a 4-tuple of PPT algorithms* $(\mathsf{IBESetup}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ *where*

- $\mathsf{IBESetup}(1^n) \to (msk, mpk)$*, outputs a master public key* mpk *and a master secret key* msk*.*
- $\mathsf{IBEExtract}_{msk}(id) \to sk_{id}$*, generates a secret key* sk_{id} *for given* msk *and identity* id*.*
- $\mathsf{IBEEnc}_{mpk}(id, m) \to c$*, given the master public key* mpk*, an identity* id*, and a message* m*, outputs a ciphertext* c*.*
- $\mathsf{IBEDec}_{sk}(c) \to m$*, given a secret key* sk*, and a ciphertext* c*, outputs a message* m*.*

We require the correctness of decryption that for all security parameters 1^n, all identities id, and all m in the specified message space,

$$\Pr[\mathsf{IBEDec}_{sk_{id}}(\mathsf{IBEEnc}_{mpk}(id, m)) \neq m] = negl(n),$$

where the probability is taken over the randomness used in $(mpk, msk) \leftarrow \mathsf{IBESetup}(1^n)$, $sk_{id} \leftarrow \mathsf{IBEExtract}_{msk}(id)$, and $\mathsf{IBEEnc}_{mpk}(id, m)$.

We use the indistinguishability under chosen plaintext attack (IND-ID-CPA) [6] notion of security.

Definition 4 (IND-ID-CPA). *An adversary $\mathcal{A}$ is said to (t, q_H, q_E, ϵ)-break the identity-based encryption scheme* (IBESetup, IBEExtract, IBEEnc, IBEDec) *if $\mathcal{A}$ runs in time at most t, makes at most q_H hash queries and at most q_E extracting queries, and furthermore*

$$\Pr[b' = b \wedge id^* \notin Q : b' \leftarrow \mathcal{A}^{\mathsf{IBEExtract}^H_{msk}(\cdot), H(\cdot), \mathsf{Chall}(id^*, m_0, m_1)}(mpk)] \geq \epsilon,$$

where Q is the set of extracting queries made by $\mathcal{A}$ and the challenge query $\mathsf{Chall}(id^*, m_0, m_1)$ *answers as follows: pick a random bit $b \xleftarrow{\$} \{0,1\}$ and return* $\mathsf{IBEEnc}^H_{mpk}(id^*, m_b)$*. The probability is taken over the random choice of the oracle H and all the randomness used in the probabilistic algorithms involved. An identity-based encryption scheme is (t, q_H, q_E, ϵ)-secure if no adversary can (t, q_H, q_E, ϵ)-break it.*

Definition 5. *A signature scheme consists of three probabilistic polynomial-time algorithms* (Gen, Sign, Vrfy) *such that:*

- Gen *takes as input a security parameter 1^n, and outputs a public key pk and a private key sk.*
- Sign *takes as input a private key sk and a message m, and outputs a signature σ. We write this as $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.*
- Vrfy *takes as input a public key pk, a message m, and a signature σ, and outputs a bit b, with $b = 1$ meaning* accept *and $b = 0$ meaning* reject*. We write this as $b := \mathsf{Vrfy}_{pk}(m, \sigma)$.*

We make the standard correctness require: for all (sk, pk) generated by Gen and all messages $m \in \mathcal{M}$ we have $\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1$. We use the existential unforgeablility under chosen message attack (UF-CMA) notion of security [14].

Definition 6 (UF-CMA[14]). *A forger $\mathcal{F}$ is said to (t, q_H, q_S, ϵ)-break the signature scheme* (Gen, Sign, Vrfy) *if $\mathcal{F}$ runs in time at most t, makes at most q_H hash queries and at most q_S signing queries, and furthermore*

$$\Pr[\mathsf{Vrfy}_{pk}(m, \sigma) = 1 \wedge m \notin Q : (pk, sk) \leftarrow \mathsf{Gen}, H \leftarrow \Omega, \sigma \leftarrow \mathcal{F}^{\mathsf{Sign}^H_{sk}(\cdot), H(\cdot)}(pk)] \geq \epsilon,$$

where Ω is the space from which the random oracle H is selected, and Q is the set of signing queries made by $\mathcal{F}$. A signature scheme is (t, q_H, q_S, ϵ)-secure if no forger can (t, q_H, q_S, ϵ)-break it.

2.2 Quantum Computation

We give a brief introduction to quantum computation and refer to [17] for more detailed information. A quantum system A is associated to finite-dimentional complex Hilbert space $\mathcal{H}_A$ with an inner product $\langle \cdot | \cdot \rangle$. A state of the system is described by a vector $|\phi\rangle \in \mathcal{H}_A$ such that the Euclidean norm of $|\phi\rangle$ is 1. Any classical bit string x can be encoded into a quantum state as $|x\rangle$. An arbitatry pure n-qubit state can be expressed in the computational basis as $|\phi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$, where α_x are complex amplitudes satisfying $\sum_{x\in\{0,1\}^n}|\alpha_x|^2 = 1$. An evolution of quantum state can be described by a unitary matrix $U: |x\rangle \to U|x\rangle$. Information can be extracted from a quantum state by performing a measurement. Take the measurement in the computational basis as an example. This measuring of a qubit $|\phi\rangle = \sum_{x\in\{0,1\}^n}\alpha_x|x\rangle$ results in x with probability α_x. A quantum algorithm is composed of quantum evolutions described by unitary matrices and measurements. Following [2,20], we view a quantum oracle O as a mapping $|x\rangle|y\rangle \to |x\rangle|y \oplus O(x)\rangle$, and model adversary $\mathcal{A}$ with quantum access to O by a sequence of unitaries $U_1, O, U_2, \cdots, O, U_q$. We recall the following results that we will be using. As shown by Zhandry[20], a quantum random oracle can be simulated by a family of $2q$-wise independent hash functions indistinguishably with respect to any adversary that makes at most q quantum query to that oracle. Specifically, he obtained the following result.

Lemma 1 (Theorem 6.1 in [20]). *Any quantum algorithm $\mathcal{A}$ making quantum queries to random oracles can be efficiently simulated by a quantum algorithm $\mathcal{B}$, which has the same output distribution, but makes no queries. In detail, if $\mathcal{A}$ makes at most q queries to a random oracle $H : \{0,1\}^a \to \{0,1\}^b$, then $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A}) + q \cdot T_{a,b}^{2q\text{-}wise}$, where $T_{a,b}^{2q\text{-}wise}$ denotes the time to evaluate a $2q$-wise independent hash function from $\{0,1\}^a$ to $\{0,1\}^b$.*

Definition 7 (Reprogrammed Functions). *For a given function $H : \mathcal{X} \to \mathcal{Y}$ and for fixed $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, the reprogrammed function $H_{x\to\Theta} : \mathcal{X} \to \mathcal{Y}$ coincides with H on $\mathcal{X}/\{x\}$ but maps x to Θ.*

As shown by J. Don et al. [11], queries made by an arbitrary quantum oracle algorithm $\mathcal{A}$ can be read out by defining a two-stage algorithm $\mathcal{S}$ with black-box access to $\mathcal{A}$, with the corresponding hash value being reprogrammed. In [11], $\mathcal{S}$ works by running $\mathcal{A}$ with the following modifications. First, one of the $q+1$ queries of $\mathcal{A}$ (also counting the final output) is selected uniformly at random and measured, with the measurement result x being output by the first stage of $\mathcal{S}$. Then, this very query of $\mathcal{A}$ is answered either using the original H or using the reprogrammed oracle $H_{x\to\Theta}$, with the choice being made at random, while all the remaining queries of $\mathcal{A}$ are answered using $H_{x\to\Theta}$. Finally, $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs. As a result, they obtain the following theorem.

Lemma 2 (Measure-and-reprogram, theorem 2 in [11]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box two-stage quantum algorithm $\mathcal{S}$ with the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs some $x \in \mathcal{X}$ and a (possibly quantum) output z. Then, the two-stage algorithm $\mathcal{S}^{\mathcal{A}}$ outputs some $x \in \mathcal{X}$ in the first stage and, upon a random $\Theta \in \mathcal{Y}$ as input to the second stage, a (possibly quantum) output z, so that for any $x_0 \in \mathcal{X}$ and any (possibly quantum) predicate V:*

$$\Pr_{\Theta}[x = x_0 \wedge V(x, \Theta, z) : \ (x, z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \Theta \rangle] \\ \geq \frac{1}{(2q+1)^2} \Pr_{H}[x = x_0 \wedge V(x, H(x), z) : \ (x, z) \leftarrow \mathcal{A}^H].$$

Furthermore, $\mathcal{S}$ runs in time polynomial in q, $\log|\mathcal{X}|$, and $\log|\mathcal{Y}|$.

3 Tighter Security Proof for GPV-IBE

Here we prove the security of IBE scheme from Gentry et al. [13]. Their scheme is constructed from a dual cryptosystem $(\mathsf{DualGen}, \mathsf{DualEnc}, \mathsf{DualDec})$ whose key generation algorithm $\mathsf{DualGen}$ is associated with a PSF $\mathsf{F} = (\mathsf{F.Gen}, \mathsf{F.Sample}, f, f^{-1})$ and works as follows: generate $(msk, mpk) \leftarrow \mathsf{F.Gen}(1^n)$, sample $sk \leftarrow \mathsf{F.Sample}(1^n)$, compute $pk = f_{mpk}(sk)$, and output $(sk, (pk, mpk))$. Then, using a random oracle $H : \mathcal{ID} \rightarrow \mathsf{Ran}_f$ that maps the identities to the range of f, the GPV-IBE scheme $\mathsf{IBE} = (\mathsf{IBESetup} = \mathsf{F.Gen}, \mathsf{IBEExtract}, \mathsf{IBEEnc}, \mathsf{IBEDec})$ is defined as follows.

- $\mathsf{IBEExtract}^H_{msk}(id) := f^{-1}_{msk}(H(id))$,
- $\mathsf{IBEEnc}^H_{mpk}(id, m) := \mathsf{DualEnc}_{H(id),mpk}(m)$,
- $\mathsf{IBEDec}_{sk}(c) := \mathsf{DualDec}_{sk}(c)$.

Theorem 1. *Suppose that the dual cryptosystem is quantum IND-CPA-secure. Then the GPV-IBE scheme defined as above is quantum IND-ID-secure when we model H as a random oracle. Detailedly, for any quantum PPT adversary $\mathcal{A}$ making at most q_H random oracle queries to H and q_E extraction queries that breaks IBE with advantage ϵ, there exists a quantum PPT algorithm $\mathcal{B}$ that breaks the dual cryptosystem with probability ϵ' such that*

$$\epsilon \leq (2q_H + 1)^2 \epsilon'.$$

Proof. Let $\mathcal{A}_0$ be a quantum adversary making q_H hash queries, q_E extracting queries, that breaks IBE with advantage ϵ.

Let $\mathtt{Game}_0$ be the standard attack game for IBE: the challenger generates (msk, mpk) from $\mathsf{IBESetup}$, and sends mpk to the adversary. The adversary can make (classical) extraction queries on identities id_i, and (quantum) hash queries to the random oracle H. $\mathcal{A}_0$ then produces an identity id^*, along with two messages m_0 and m_1. The challenger chooses a random bit b, and responds with $\mathsf{IBEEnc}^H_{mpk}(id^*, m_b)$. $\mathcal{A}_0$ is allowed to make further extracting and hash queries, except that we make sure $\mathcal{A}_0$ never queries $\mathsf{IBEExtract}^H_{msk}(id^*)$. Finally, $\mathcal{A}_0$ outputs a bit b' and we report $\mathcal{A}_0$ wins if $b' = b$. By definition, this happens with probability $\frac{1}{2} + \epsilon$.

Let $\mathcal{A}$ be the following algorithm that makes quantum queries to another oracle $H' : \mathcal{ID} \rightarrow \mathsf{Dom}_f$, and simulates the interaction between $\mathcal{A}_0$ and the challenger: generate (msk, mpk) from $\mathsf{IBESetup}$, send mpk to $\mathcal{A}_0$, and run $\mathcal{A}_0$.

When $\mathcal{A}_0$ makes an extraction query $\mathsf{IBEExtract}^H_{msk}(id)$, $\mathcal{A}$ returns $H'(id)$. In response to a random oracle query on id, $\mathcal{A}$ first forwards id to H', gets x, and then returns $f_{mpk}(x)$. Similarly, answer the challenge query (id^*, m_0, m_1) by choosing a random bit b and encrypting m_b to the identity id^*. The output of $\mathcal{A}$ is (id^*, c), where $c = b \oplus b'$ and b' is the guess produced by $\mathcal{A}_0$. We can now think of $\mathtt{Game}_1$ as follows: run $\mathcal{A}$ with a random oracle to obtain (id^*, c). Report that the game is won if and only if $c = 0$. The number of queries to H' made by $\mathcal{A}$ is q_H for random queries, q_E for queries through the extraction algorithm, and 1 for the encryption of m_b, for a total of $q_H + q_E + 1$ queries.

Thus, we can apply Lemma 2, with $id^*,\ c,\ \mathcal{ID},\ \mathsf{Dom}_f,\ H'$ playing the role of what is referred to as $x,\ z,\ \mathcal{X},\ \mathcal{Y},\ H$, respectively, in the theorem statement, to obtain the existence of an algorithm $\mathcal{S}^{\mathcal{A}}$ that produces id^* in the first stage, and upon receiving a random $sk \in \mathsf{Dom}_f$ produces c, such that for any $id \in \mathcal{ID}$

$$\Pr_{sk}[id^* = id \wedge V(id^*, sk, c) :\ (id^*, c) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, sk\rangle] \\ \geq \frac{1}{(2q+1)^2} \Pr_H[id^* = id \wedge V(id^*, H'(id^*), c) :\ (id^*, c) \leftarrow \mathcal{A}^{H'}],$$

where $V(id^*, sk, c)$ and $V(id^*, H'(id^*), c)$ both specify $c = 0$ and id^* is never queried to the extraction oracle before. Summed over all $(m_0, r_0) \in \mathcal{M} \times \{0,1\}^{k_0}$, this in particular implies that

$$\Pr_{sk}[c = 0 \wedge id^* \notin Q :\ (id^*, c) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, sk\rangle] \\ \geq \frac{1}{(2q+1)^2} \Pr_H[c = 0 \wedge id^* \notin Q :\ (id^*, c) \leftarrow \mathcal{A}^{H'}].$$

where Q is the list of extraction queries made by $\mathcal{A}_0$. Let $\mathtt{Game}_2$ be $\mathtt{Game}_1$ with the following modifications. During the process, one unique RO query from $\mathcal{A}_0$ is chosen uniformly at random, and measured to hopefully obtain the very id^* that $\mathcal{A}_0$ will produce in its final forgery. Subsequently, the RO is reprogrammed, so as to answer $H(id^*)$ with $pk = f_{mpk}(sk)$ for some $sk \in \mathsf{Dom}_f$, either from this point on or from the following query on, with the binary choice made at random. Since the messages yielded by measuring on these H'-queries cannot pass the MaR predicate V, the reprogram operation on H'-queries that are used for simulating the extraction oracle can be removed. Thus, the $\mathcal{A}$ for instantiation of MaR can be transformed into $\mathtt{Game}_2$ with $\mathcal{A}_0$, where the measure-and-reprogram is performed only on the H-queries. Then, the inequality becomes

$$\Pr_{sk}[b' = b \wedge id^* \notin Q :\ b' \leftarrow \langle \mathcal{A}_0, \mathtt{Game}_2\rangle] \geq \frac{\epsilon}{(2q_H+1)^2}.$$

Now we are ready to define an algorithm $\mathcal{B}$ that breaks the IND-CPA-security of the dual cryptosystem. Give $\mathcal{B}$ access to the random oracle $H' : \mathcal{ID} \to \mathsf{Dom}_f$. On input (pk, mpk), $\mathcal{B}$ works as follows.

- Send mpk to $\mathcal{A}_0$, simulate $\mathcal{A}_0$, and play the role of challenger to $\mathcal{A}_0$.
- Choose a uniformly random $i \xleftarrow{\$} \{1, \ldots, q_H + 1\}$ and $t \xleftarrow{\$} \{0, 1\}$.

- Construct the (quantum) oracle H such that $H(id) = f_{mpk}(H'(id))$. Answer the first $i-1$ random oracle queries that $\mathcal{A}_0$ makes by H. Measure the i-th query, get id^*, and answer this query by H for $t = 1$ and by the reprogrammed function $H_{id^* \to pk}$ for $t = 0$. The remaining queries are answered using $H_{id^* \to pk}$.
- When $\mathcal{A}_0$ asks for the secret key for id, return $H'(id)$.
- When $\mathcal{A}_0$ produces the challenge query (id^*, m_0, m_1), forward (m_0, m_1) to $\mathcal{B}$'s challenger and send the response to $\mathcal{A}_0$.
- When $\mathcal{A}_0$ outputs its guess b', output b'.

Note that by reprogramming $H(id^*)$ to pk, the challenge $c = \mathsf{DualEnc}_{pk,mpk}(m_b)$ is exactly $\mathsf{IBEEnc}^{H}_{mpk}(id^*, m_b)$, so that the view of $\mathcal{A}_0$ when ran as a subroutine by $\mathcal{B}$ is identical to the view of $\mathcal{A}_0$ in $\texttt{Game}_2$ and $\mathcal{B}$ wins if and only if $\mathcal{A}_0$ wins $\texttt{Game}_2$. We get that the advantage of $\mathcal{B}$ is at least

$$\frac{\epsilon}{(2q_H + 1)^2}.$$

Note that by Lemma 1 the quantum random oracle H' can be efficiently simulated by a family of $2q$-wise independent hash functions.

This completes the proof.

Remark 1. In [20], Zhandry showed how to prove the security of the hierarchical IBE (HIBE) of Agrawal et al. [1] and Cash et al. [8] by repeatedly applying the arguments of the IBE result. We note that Theorem 1 can also be applied to the random oracle HIBE schemes.

In an HIBE scheme, identities are structured as a directed tree in which every node contains its parent as a prefix and can produce secret keys for its children. Specifically, instead of an extraction algorithm $sk_{id} \leftarrow \mathsf{Extract}_{msk}(id)$, in an HIBE scheme, identities are vectors and there is an algorithm named Derive, which takes an identity $\boldsymbol{id} = (id_1, \cdots, id_k)$ and a secret key $sk_{\boldsymbol{id}_{|l}}$ of a parent $\boldsymbol{id}_{|l} = (id_1, \cdots, id_l)$ for some $l < k$, and outputs a secret key $sk_{\boldsymbol{id}}$ for the identity $\boldsymbol{id}$. The adversary $\mathcal{A}$ is allowed to adaptively take control of an arbitrary number of nodes in the tree and obtain the associated secret keys. Suppose d and $\boldsymbol{id}^*$ denote the max hierarchy depth and the identity that $\mathcal{A}$ produces in the challenge query, respectively. In [20], Zhandry highly generalized the reduction of Agrawal et al. [1] as:

Setup. $\mathcal{B}$ prepares a simulated attack environment for $\mathcal{A}$.

- Select d uniformly random integers $q_1^*, \cdots, q_d^* \in [q_H]$, and hopefully the q_i^*-th query to H will contain the hash of the level-i parent if $\boldsymbol{id}^*$.
- Sample d random quantities $R_1^*, \cdots, R_d^*$.
- Choose a random $\omega \in [d]$, a guess at the level that contains the targeted identity $\boldsymbol{id}^*$.

Random oracle queries. $\mathcal{A}$ may query the random oracle H on any $\boldsymbol{id}$ adaptively at any time. Let $i = |\boldsymbol{id}|$ be the depth of $\boldsymbol{id}$. $\mathcal{B}$ answers the q-th query as follows.

- Simulate a separate random oracle for identities at each level.
- If $q = q_i^*$, return $H(\boldsymbol{id}) \leftarrow R_i^*$, and otherwise return a random value $H(\boldsymbol{id}) \leftarrow R$.

Secret key queries. Secret key queries are answered in a certain way to match with the RO queries. If $\mathcal{A}$ makes a query on $\boldsymbol{id} = (id_1, \cdots, id_k)$ such that $H(\boldsymbol{id}_{|i}) = R_i^*$ for all $i \leq k$, then the simulator aborts and fails.

Finally, $\mathcal{B}$ succeeds if $\mathcal{A}$ succeeds, $\boldsymbol{id}^*$ is at level ω, and no abortion is triggered. The reduction can be transplanted to the QROM version by repeatedly applying the arguments of Theorem 1. We iterate over level i, and use R_i^* to reprogram the separate random oracle for identities at that level. In iteration i, we say the adversary wins if it won in the previous iteration, the level-i prefix of the challenge identity $\boldsymbol{id}^*$ is reprogrammed (i.e. $H(\boldsymbol{id}^*_{|i}) = R_i^*$), and no signature query is. Let ϵ_i denote the iteration i advantage, then using the same techniques as in Theorem 1, we get

$$\epsilon_i \geq \frac{\epsilon_{i-1}}{(2q_H + 1)^2}.$$

In iteration 0, the adversary wins if it wins the standard game and we guess correctly which level $\boldsymbol{id}^*$ belongs to. Then, we have $\epsilon_0 = \epsilon/d$, where ϵ is the adversary's advantage in the standard game and the total advantage after iteration d is at least

$$\frac{\epsilon/d}{(2q_H + 1)^{2d}} = \frac{\epsilon}{d}\left(\frac{1}{2q_H + 1}\right)^{2d}.$$

Recall that the result $l(\epsilon/dl)^{2^d}$ in [20] is doubly-exponential in the depth d, whereas our result is singly exponential as in the classical proof. This is an even more significant improvement than the one in original IBE.

4 Tighter Security Proof for (P)FDH

Definition 8 (FDH Signatures[3]). *Let* $\mathsf{F} = (\mathsf{F.Gen}, f, f^{-1})$ *be a trapdoor permutation with* $f: \mathcal{X} \to \mathcal{Y}$*, and* $H: \mathcal{M} \to \mathcal{Y}$ *be a hash function. The FDH signature scheme introduced by* F *and* H *is a triple* $(\mathsf{GenFDH} = \mathsf{F.Gen}, \mathsf{SignFDH}^H, \mathsf{VrfyFDH}^H)$*, defined as follows.*

- $\mathsf{SignFDH}^H_{sk}(m) := f^{-1}_{sk}(H(m))$.
- $\mathsf{VrfyFDH}^H_{pk}(m, \sigma) := \begin{cases} \mathsf{accept} & \textit{if } f_{pk}(\sigma) = H(m) \\ \mathsf{reject} & \textit{otherwise.} \end{cases}$

Theorem 2. *Suppose that the trapdoor permutation* F *is quantum one-way. Then the signature scheme FDH is UF-CMA-secure in the quantum random* *oracle model. Detailedly, for any quantum PPT adversary $\mathcal{A}$ making at most q_H random oracle queries to H and q_S signature queries that breaks FDH with advantage ϵ, there exists a quantum PPT algorithm $\mathcal{B}$ that inverts* F *with probability ϵ' such that*

$$\epsilon \leq (2q_H + 1)^2 \epsilon'.$$

Proof. Suppose towards contradiction that there is a quantum adversary $\mathcal{A}_0$ making q_H hash queries, q_S signature queries, that breaks FDH with probability ϵ.

Let $\mathtt{Game}_0$ be the standard attack game for FDH: the challenger generates (pk, sk) from GenFDH, and sends pk to the adversary. The adversary can make (quantum) hash queries to the random oracle H, and (classical) signature queries on messages m_i, to which the challenger responds with $\mathsf{SignFDH}^H_{sk}(m_i)$. $\mathcal{A}_0$ wins if it can produce a pair (m, σ) such that $m \neq m_i$ for any i, and $\mathsf{VrfyFDH}^H_{pk}(m, \sigma) = \mathsf{accept}$. The success probability in $\mathtt{Game}_0$ is ϵ.

Let $\mathcal{A}$ be the following algorithm that makes quantum queries to another random oracle $H' : \mathcal{M} \to \mathcal{X}$, and simulates the interaction between $\mathcal{A}_0$ and the challenger: generate (pk, sk) from GenFDH, send pk to $\mathcal{A}_0$, and run $\mathcal{A}_0$. Further, when $\mathcal{A}_0$ makes a signature query $\mathsf{SignFDH}^H_{sk}(m)$, $\mathcal{A}$ returns $H'(m)$. In response to a random oracle query on m, $\mathcal{A}$ first forwards m to H', gets x, and then returns $f_{pk}(x)$. Finally, $\mathcal{A}$ outputs the forgery (m, σ) that $\mathcal{A}_0$ outputs, and the total number of queries $\mathcal{A}$ makes to H' is $q = q_S + q_H$. We can now think of $\mathtt{Game}_1$ as follows: run $\mathcal{A}$ with a random oracle to obtain (m, σ). Report that the game is won if and only if $\mathsf{VrfyFDH}^H_{pk}(m, \sigma) = \mathsf{accept}$ and this happens with the probability ϵ.

Thus, we can apply Lemma 2, with $m,\ \sigma,\ \mathcal{M},\ \mathcal{X},\ H'$ playing the role of what is referred to as $x,\ z,\ \mathcal{X},\ \mathcal{Y},\ H$, respectively, in the theorem statement, to obtain the existence of an algorithm $\mathcal{S}^{\mathcal{A}}$ that produces m in the first stage, and upon receiving a random $x \in \mathcal{X}$ produces σ, such that for any $m_0 \in \mathcal{M}$

$$\begin{aligned}
&\Pr_x[m = m_0 \wedge V(m, x, \sigma) :\ (m, \sigma) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, x\rangle] \\
&\qquad \geq \frac{1}{(2q+1)^2} \Pr_H[m = m_0 \wedge V(m, H'(m), \sigma) :\ (m, \sigma) \leftarrow \mathcal{A}^{H'}],
\end{aligned}$$

where $V(m, x, \sigma)$ (or $V(m, H'(m), \sigma)$) specifies $x = \sigma$ (or $H'(m) = \sigma$) and m is never queried to the signing oracle before. Summed over all $m_0 \in \mathcal{M}$, this in particular implies that

$$\begin{aligned}
&\Pr_x[\sigma = x \wedge m \notin Q :\ (m, \sigma) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, x\rangle] \\
&\qquad \geq \frac{1}{(2q+1)^2} \Pr_H[H'(m) = \sigma \wedge m \notin Q :\ (m, \sigma) \leftarrow \mathcal{A}^{H'}],
\end{aligned}$$

where Q is the list of signing queries made by $\mathcal{A}_0$. Recall that, by definition, $H'(m) = \sigma \wedge m \notin Q$ is equivalent to $\mathsf{VrfyFDH}^H_{pk}(m, \sigma) = \mathsf{accept}$. Let $\mathtt{Game}_2$ be $\mathtt{Game}_1$ with the following modifications. During the process, one unique RO query from $\mathcal{A}_0$ is chosen uniformly at random, and measured to hopefully obtain the very m that $\mathcal{A}_0$ will produce in its final forgery. Subsequently, the RO is reprogrammed, so as to answer $H(m)$ with $y = f_{pk}(x)$ for some $x \in \mathcal{X}$, either from this point on or from the following query on, with the binary choice made at random. Since the messages yielded by measuring on these H'-queries cannot pass the MaR predicate V, the reprogram operation on H'-queries that are used for simulating the signing oracle can be removed. Thus, the $\mathcal{A}$ for instantiation of MaR can be transformed into $\mathtt{Game}_2$ with $\mathcal{A}_0$, where the measure-and-reprogram is performed only on the H-queries. Then, the inequality becomes

$$\Pr_x[\sigma = x \wedge m \notin Q : \ (m, \sigma) \leftarrow \langle \mathcal{A}_0, \mathtt{Game}_2 \rangle] \geq \frac{\epsilon}{(2q_H + 1)^2}.$$

Now we are ready to define an algorithm $\mathcal{B}$ that inverts f. Give $\mathcal{B}$ access to the random oracle $H' : \ \mathcal{M} \rightarrow \mathcal{X}$. On input (pk, y), $\mathcal{B}$ works as follows.

- Send pk to $\mathcal{A}_0$, simulate $\mathcal{A}_0$, and play the role of challenger to $\mathcal{A}_0$.
- Choose a uniformly random $i \leftarrow \{1, \ldots, q_H + 1\}$ and $b \leftarrow \{0, 1\}$.
- Construct the (quantum) oracle H such that $H(m) = f_{pk}(H'(m))$. Answer the first $i - 1$ random oracle queries that $\mathcal{A}_0$ makes by H. Measure the i-th query, get m, and answer this query by H for $b = 1$ and by the reprogrammed function $H_{m \rightarrow y}$ for $b = 0$. The remaining queries are answered using $H_{m \rightarrow y}$.
- When $\mathcal{A}_0$ makes a signature query on a message m, return $H'(m)$.
- When $\mathcal{A}_0$ returns a forgery (m, σ), output σ.

Note that the view of $\mathcal{A}_0$ when ran as a subroutine by $\mathcal{B}$ is identical to the view of $\mathcal{A}_0$ in $\mathtt{Game}_2$. We get that the advantage of $\mathcal{B}$ is at least

$$\frac{\epsilon}{(2q_H + 1)^2}.$$

Note that by Lemma 1 the quantum random oracle H' can be efficiently simulated by a family of $2q$-wise independent hash functions.

This completes the proof.

Definition 9 (PFDH Signatures[10]). *Let* $\mathsf{F} = (\mathsf{F.Gen}, f, f^{-1})$ *be a trapdoor permutation with* $f : \ \mathcal{X} \rightarrow \mathcal{Y}$. *As FDH, the scheme uses a hash function* $H : \{0,1\}^* \ \rightarrow \mathcal{Y}$. *The difference is that a random salt of* k_0 *bit is concatenated to the message before hashing it. Specifically, the probabilistic full domain hash (PFDH) signature scheme* $(\mathsf{GenPFDH} = \mathsf{F.Gen}, \mathsf{SignPFDH}^H, \mathsf{VrfyPFDH}^H)$ *works as follows.*

- $\mathsf{SignPFDH}^H_{sk}(m) := (f^{-1}_{sk}(H(m||r)), r)$, *for a uniformly random chosen* $r \leftarrow \{0,1\}^{k_0}$.
- $\mathsf{VrfyPFDH}^H_{pk}(m, \sigma = (s, r)) := \begin{cases} \mathsf{accept} & \textit{if } f_{pk}(s) = H(m||r) \\ \mathsf{reject} & \textit{otherwise.} \end{cases}$

Theorem 3. *Suppose that the trapdoor permutation* F *is quantum one-way. Then the signature scheme PFDH*$[k_0]$ *is UF-CMA-secure in the quantum random oracle model. Detailedly, for any quantum PPT adversary* $\mathcal{A}$ *making at most*

q_H random oracle queries to H and q_S signature queries that breaks PFDH$[k_0]$ with advantage ϵ, there exists a quantum PPT algorithm $\mathcal{B}$ that inverts F *with probability ϵ' such that*

$$\epsilon \leq (2q_H + 1)^2 \epsilon'.$$

Proof. Suppose towards contradiction that there is a quantum adversary $\mathcal{A}_0$ making q_H hash queries, q_S signature queries, that breaks PFDH with probability ϵ.

Let $\mathtt{Game}_0$ be the standard attack game for PFDH: the challenger generates (pk, sk) from $\mathsf{GenPFDH}$, and sends pk to the adversary. The adversary can make (quantum) hash queries to the random oracle H, and (classical) signature queries on messages m_i, to which the challenger responds with $\mathsf{SignPFDH}^H_{sk}(m_i)$. $\mathcal{A}_0$ wins if it can produce a pair $(m, \sigma = (s, r))$ such that $m \neq m_i$ for any i, and $\mathsf{VrfyPFDH}^H_{pk}(m, \sigma) = \mathsf{accept}$. The success probability in $\mathtt{Game}_0$ is ϵ.

Let $\mathcal{A}$ be the following algorithm that makes quantum queries to another random oracle $H' : \mathcal{M} \times \{0,1\}^{k_0} \to \mathcal{X}$, and simulates the interaction between $\mathcal{A}_0$ and the challenger: generate (pk, sk) from $\mathsf{GenPFDH}$, send pk to $\mathcal{A}_0$, and run $\mathcal{A}_0$. Further, when $\mathcal{A}_0$ makes a signature query $\mathsf{SignPFDH}^H_{sk}(m)$, $\mathcal{A}$ chooses a random $r \leftarrow \{0,1\}^{k_0}$ and returns $(H'(m||r), r)$. In response to a random oracle query on (m, r), $\mathcal{A}$ first forwards (m, r) to H', gets x, and then returns $f_{pk}(x)$. Finally, $\mathcal{A}$ outputs the forgery $(m, \sigma = (s, r))$ that $\mathcal{A}_0$ outputs, and the total number of queries $\mathcal{A}$ makes to H' is $q = q_S + q_H$. We can now think of $\mathtt{Game}_1$ as follows: run $\mathcal{A}$ with a random oracle to obtain (m, σ). Report that the game is won if and only if $\mathsf{VrfyPFDH}^H_{pk}(m, \sigma) = \mathsf{accept}$ and this happens with the probability ϵ.

Thus, we can apply Lemma 2, with (m, r), s, $\mathcal{M} \times \{0,1\}^{k_0}$, $\mathcal{X}$, H' playing the role of what is referred to as x, z, $\mathcal{X}$, $\mathcal{Y}$, H, respectively, in the theorem statement, to obtain the existence of an algorithm $\mathcal{S}^{\mathcal{A}}$ that produces (m, r) in the first stage, and upon receiving a random $x \in \mathcal{X}$ produces s, such that for any $(m_0, r_0) \in \mathcal{M} \times \{0,1\}^{k_0}$

$$\begin{aligned} &\Pr_x[(m,r) = (m_0, r_0) \wedge V((m,r), x, s) : \ (m,r,s) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, x \rangle] \\ \geq &\frac{1}{(2q+1)^2} \Pr_H[(m,r) = (m_0, r_0) \wedge V((m,r), H'(m||r), s) : \ (m,r,s) \leftarrow \mathcal{A}^{H'}], \end{aligned}$$

where $V((m,r), x, s)$ (or $V((m,r), H'(m||r), s)$) specifies $x = s$ (or $H'(m||r) = s$) and m is never queried to the signing oracle before. Summed over all $(m_0, r_0) \in \mathcal{M} \times \{0,1\}^{k_0}$, this in particular implies that

$$\begin{aligned} &\Pr_x[s = x \wedge m \notin Q : \ (m,r,s) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, x \rangle] \\ &\qquad \geq \frac{1}{(2q+1)^2} \Pr_H[H'(m||r) = s \wedge m \notin Q : \ (m,r,s) \leftarrow \mathcal{A}^{H'}]. \end{aligned}$$

where Q is the list of signing queries made by $\mathcal{A}_0$. Recall that, by definition, $H'(m||r) = s \wedge m \notin Q$ is equivalent to $\mathsf{VrfyPFDH}^H_{pk}(m, \sigma = (s, r)) = \mathsf{accept}$. Let $\mathtt{Game}_2$ be $\mathtt{Game}_1$ with the following modifications. During the process, one unique

RO query from $\mathcal{A}_0$ is chosen uniformly at random, and measured to hopefully obtain the very (m, r) that $\mathcal{A}_0$ will produce in its final forgery. Subsequently, the RO is reprogrammed, so as to answer $H(m||r)$ with $y = f_{pk}(x)$ for some $x \in \mathcal{X}$, either from this point on or from the following query on, with the binary choice made at random. Since the messages yielded by measuring on these H'-queries cannot pass the MaR predicate V, the reprogram operation on H'-queries that are used for simulating the signing oracle can be removed. Thus, the $\mathcal{A}$ for instantiation of MaR can be transformed into $\texttt{Game}_2$ with $\mathcal{A}_0$, where the measure-and-reprogram is performed only on the H-queries. Then, the inequality becomes

$$\Pr_x[s = x \wedge m \notin Q : \ (m, \sigma = (r, s)) \leftarrow \langle \mathcal{A}_0, \texttt{Game}_2 \rangle] \geq \frac{\epsilon}{(2q_H + 1)^2}.$$

Now we are ready to define an algorithm $\mathcal{B}$ that inverts f. Give $\mathcal{B}$ access to the random oracle $H' : \ \mathcal{M} \times \{0,1\}^{k_0} \to \mathcal{X}$. On input (pk, y), $\mathcal{B}$ works as follows.

- Send pk to $\mathcal{A}_0$, simulate $\mathcal{A}_0$, and play the role of challenger to $\mathcal{A}_0$.
- Choose a uniformly random $i \leftarrow \{1, \ldots, q_H + 1\}$ and $b \leftarrow \{0, 1\}$.
- Construct the (quantum) oracle H such that $H(m||r) = f_{pk}(H'(m||r))$. Answer the first $i-1$ random oracle queries that $\mathcal{A}_0$ makes by H. Measure the i-th query, get (m, r), and answer this query by H for $b = 1$ and by the reprogrammed function $H_{(m,r)\to y}$ for $b = 0$. The remaining queries are answered using $H_{(m,r)\to y}$.
- When $\mathcal{A}_0$ makes a signature query on a message m, choose a random $r \in \{0,1\}^{k_0}$, and return $(H'(m||r), r)$.
- When $\mathcal{A}_0$ returns a forgery $(m, \sigma = (s, r))$, output s.

Note that the view of $\mathcal{A}_0$ when ran as a subroutine by $\mathcal{B}$ is identical to the view of $\mathcal{A}_0$ in $\texttt{Game}_2$. We get that the advantage of $\mathcal{B}$ is at least

$$\frac{\epsilon}{(2q_H + 1)^2}.$$

Note that by Lemma 1 the quantum random oracle H' can be efficiently simulated by a family of $2q$-wise independent hash functions.

This completes the proof.

Remark 2. We note that if the trapdoor permutation has some sort of homomorphic property, the security bound can be further tightened with $O(q_H^2)$ being replaced by $O(q_S)$, which is a significantly better result in practice since q_S is usually much smaller than q_H. The basic idea is similar to Theorem 2 in [9].

We say that the trapdoor permutation $\mathsf{F} = (\mathsf{F.Gen}, f, f^{-1})$ is homomorphic with respect to two group operations $+$ and $\odot$ if for any pk from $\mathsf{F.Gen}$, it holds that $f_{pk}(a + b) = f_{pk}(a) \odot f_{pk}(b), \ \forall a, b$. We give the following result regarding FDH-TDP with homomorphic property.

Theorem 4. *Suppose that the trapdoor permutation* F *is quantum one-way and homomorphic with respect to two group operations* $+$ *and* $\odot$*. Then the signature scheme FDH is UF-CMA-secure in the quantum random oracle model. Detailedly, for any quantum PPT adversary* $\mathcal{A}$ *making at most* q_H *random oracle queries to* H *and* q_S *signature queries that breaks FDH with advantage* ϵ*, there exists a quantum PPT algorithm* $\mathcal{B}$ *that inverts* F *with probability* ϵ' *such that*

$$\epsilon \leq 4q_S\epsilon',$$

Proof. Suppose towards contradiction that there is a quantum adversary $\mathcal{A}$ making q_H hash queries, q_S signature queries, that breaks FDH with probability ϵ.

Let $p \in (0,1)$ to be chosen later. The inverter $\mathcal{B}$ is given (pk, y) as input, and has quantum access to two random oracles $O_1 : \mathcal{M} \to \mathcal{X}$ and $O_2 : \mathcal{M} \to \{0,1\}$, outputting 1 with probability p. These oracles can be efficiently simulated according to Lemma 1. $\mathcal{B}$ works as follows.

- Send pk to $\mathcal{A}$, simulate $\mathcal{A}$, and play the role of challenger to $\mathcal{A}$.
- Construct a quantum oracle H such that

$$H(m) := \begin{cases} y \odot f_{pk}(O_1(m)) & \text{if } O_2(m) = 1 \\ f_{pk}(O_1(m)) & \text{otherwise.} \end{cases}$$

- When $\mathcal{A}$ makes a signature query on m, abort if $O_2(m) = 1$, and otherwise returns $O_1(m)$.
- When $\mathcal{A}$ produces a forgery (m, σ), output $\sigma + O_1(m)^{-1}$ if $O_2(m) = 1$, and otherwise abort.

Then, if $\mathcal{A}$ produces a valid forgery (m, σ) such that $O_2(m) = 1$, we have $f_{pk}(\sigma + O_1(m)^{-1}) = H(m) \odot (y \odot H(m))^{-1} = y$, and thus $\mathcal{B}$ outputs the invert of y for f_{pk}. So with probability at least $p(1 - pq_S)$, no abortion occurs and take $p = 1/(2q_S)$, $\mathcal{B}$ wins with probability at least $\epsilon/(4q_S)$.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) Advances in Cryptology - CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 98–115. Springer (2010), https://doi.org/10.1007/978-3-642-14623-7_6
2. Beals, R., Buhrman, H., Cleve, R., Mosca, M., de Wolf, R.: Quantum lower bounds by polynomials. In: Proceedings 39th Annual Symposium on Foundations of Computer Science. pp. 352–361 (1998). https://doi.org/10.1109/SFCS.1998.743485
3. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Denning, D.E., Pyle, R., Ganesan, R., Sandhu, R.S., Ashby, V. (eds.) CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993. pp. 62–73. ACM (1993), https://doi.org/10.1145/168588.168596
4. Bellare, M., Rogaway, P.: The exact security of digital signatures - how to sign with RSA and rabin. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96. Lecture Notes in Computer Science, vol. 1070, pp. 399–416. Springer (1996), https://doi.org/10.1007/3-540-68339-9_34

5. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 41–69. Springer (2011), https://doi.org/10.1007/978-3-642-25385-0_3
6. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) Advances in Cryptology - CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer (2001), https://doi.org/10.1007/3-540-44647-8_13
7. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007). pp. 647–657. IEEE Computer Society (2007), https://doi.org/10.1109/FOCS.2007.64
8. Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai trees, or how to delegate a lattice basis. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 523–552. Springer (2010), https://doi.org/10.1007/978-3-642-13190-5_27
9. Coron, J.: On the exact security of full domain hash. In: Bellare, M. (ed.) Advances in Cryptology - CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 229–235. Springer (2000), https://doi.org/10.1007/3-540-44598-6_14
10. Coron, J.: Optimal security proofs for PSS and other signature schemes. In: Knudsen, L.R. (ed.) Advances in Cryptology - EUROCRYPT 2002. Lecture Notes in Computer Science, vol. 2332, pp. 272–287. Springer (2002), https://doi.org/10.1007/3-540-46035-7_18
11. Don, J., Fehr, S., Majenz, C.: The measure-and-reprogram technique 2.0: Multiround fiat-shamir and more. In: Micciancio, D., Ristenpart, T. (eds.) Advances in Cryptology - CRYPTO 2020. Lecture Notes in Computer Science, vol. 12172, pp. 602–631. Springer (2020), https://doi.org/10.1007/978-3-030-56877-1_21
12. Don, J., Fehr, S., Majenz, C., Schaffner, C.: Security of the fiat-shamir transformation in the quantum random-oracle model. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology - CRYPTO 2019. Lecture Notes in Computer Science, vol. 11693, pp. 356–383. Springer (2019), https://doi.org/10.1007/978-3-030-26951-7_13
13. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Dwork, C. (ed.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing. pp. 197–206. ACM (2008), https://doi.org/10.1145/1374376.1374407
14. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (1988), https://doi.org/10.1137/0217017
15. Katsumata, S., Yamada, S., Yamakawa, T.: Tighter security proofs for GPV-IBE in the quantum random oracle model. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018. Lecture Notes in Computer Science, vol. 11273, pp. 253–282. Springer (2018), https://doi.org/10.1007/978-3-030-03329-3_9
16. Kosuge, H., Xagawa, K.: Probabilistic hash-and-sign with retry in the quantum random oracle model. Cryptology ePrint Archive, Paper 2022/1359 (2022), https://eprint.iacr.org/2022/1359
17. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
18. NIST: Call for additional digital signature schemes for the post-quantum cryptography standardization process, https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) Advances in Cryptology, Proceedings of CRYPTO '84. Lecture Notes in Computer Science, vol. 196, pp. 47–53. Springer (1984), https://doi.org/10.1007/3-540-39568-7_5
20. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) Advances in Cryptology - CRYPTO 2012. Lecture Notes in Computer Science, vol. 7417, pp. 758–775. Springer (2012), https://doi.org/10.1007/978-3-642-32009-5_44