Random Oracles in a Quantum World

Dan Boneh[1], Özgür Dagdelen[2], Marc Fischlin[2],
Anja Lehmann[3], Christian Schaffner[4], and Mark Zhandry[1]

[1] Stanford University, USA
[2] CASED & Darmstadt University of Technology, Germany
[3] IBM Research Zurich, Switzerland
[4] University of Amsterdam and CWI, The Netherlands

Abstract. The interest in post-quantum cryptography — classical systems that remain secure in the presence of a quantum adversary — has generated elegant proposals for new cryptosystems. Some of these systems are set in the random oracle model and are proven secure relative to adversaries that have classical access to the random oracle. We argue that to prove post-quantum security one needs to prove security in the *quantum-accessible* random oracle model where the adversary can query the random oracle with quantum state.

We begin by separating the classical and quantum-accessible random oracle models by presenting a scheme that is secure when the adversary is given classical access to the random oracle, but is insecure when the adversary can make quantum oracle queries. We then set out to develop generic conditions under which a *classical* random oracle proof implies security in the *quantum-accessible* random oracle model. We introduce the concept of a *history-free reduction* which is a category of classical random oracle reductions that basically determine oracle answers independently of the history of previous queries, and we prove that such reductions imply security in the quantum model. We then show that certain post-quantum proposals, including ones based on lattices, can be proven secure using history-free reductions and are therefore post-quantum secure. We conclude with a rich set of open problems in this area.

Keywords: Quantum, Random Oracle, Signatures, Encryption

1 Introduction

The threat to existing public-key systems posed by quantum computation [Sho97] has generated considerable interest in *post-quantum* cryptosystems, namely systems that remain secure in the presence of a quantum adversary. A promising direction is lattice-based cryptography, where the underlying problems are related to finding short vectors in high dimensional lattices. These problems have so far remained immune to quantum attacks and some evidence suggests that they may be hard for quantum computers [Reg02].

As it is often the case, the most efficient constructions in lattice-based cryptography are set in the random oracle (RO) model [BR93]. For example, Gentry, Peikert, and Vaikuntanathan [GPV08] give elegant random oracle model constructions for existentially unforgeable signatures and for identity-based encryption. Gordon, Katz, and Vaikuntanathan [GKV10] construct a random oracle model group signature scheme. Boneh and Freeman [BF11] give a random oracle homomorphic signature scheme and Cayrel et al. [CLRS10] give a lattice-based signature scheme using the Fiat-Shamir random oracle heuristic. Some of these lattice constructions can now be realized without random oracles, but at a significant cost in performance [CHKP10,ABB10a,Boy10].

Modeling Random Oracles for Quantum Attackers. While quantum resistance is good motivation for lattice-based constructions, most random oracle systems to date are only proven secure relative to an adversary with *classical* access to the random oracle. In this model the adversary is given oracle access to a random hash function $O : \{0,1\}^* \to \{0,1\}^*$ and it can only "learn" a value $O(x)$ by querying the oracle O at the classical state x. However, to obtain a concrete system, the random oracle is eventually replaced by a concrete hash function thereby enabling a quantum attacker to evaluate this hash function on *quantum states*. To capture this issue in the model, we allow the adversary to evaluate the random oracle "in superposition", that is, the adversary can submit quantum states $|\varphi\rangle = \sum \alpha_x |x\rangle$ to the oracle O and receives back the evaluated state $\sum \alpha_x |O(x)\rangle$ (appropriately encoded to make the transformation unitary). We call this the *quantum(-accessible) random oracle model*. It complies with similar efforts from learning theory [BJ99,SG04] and computational complexity [BBBV97] where oracles are quantum-accessible, and from lower bounds for quantum collision finders [AS04]. Still, since we are only interested in classical cryptosystems, *honest* parties and the scheme's algorithms can access O only via classical bit strings.

Proving security in the quantum-accessible RO model is considerably harder than in the classical model. As a simple example, consider the case of digital signatures. A standard proof strategy in the classical settings is to choose randomly one of the adversary's RO queries and embed in the response a given instance of a challenge problem. One then hopes that the adversary uses this response in his signature forgery. If the adversary makes q random oracle queries, then this happens with probability $1/q$ and since q is polynomial this success probability is sufficiently high for the proof of security in the classical setting. Unfortunately, this strategy fails completely in the quantum-accessible random oracle model since *every* random oracle query potentially evaluates the random oracle at exponentially many points. Therefore, embedding the challenge in one response will be of no use to the reduction algorithm. This simple example shows that proving security in the classical RO model does not necessarily prove post-quantum security.

More abstractly, the following common classical proof techniques are not known to carry over to the quantum settings offhand:

- Adaptive Programmability: The classical random oracle model allows a simulator to program the answers of the random oracle for an adversary, often adaptively. Since the quantum adversary can query the random oracle with a state in superposition, the adversary may get some information about all exponentially many values right at the beginning, thereby making it difficult to program the oracle adaptively.
- Extractability/Preimage Awareness: Another application of the random oracle model for classical adversaries is that the simulator learns the pre-images the adversary is interested in. This is, for example, crucial to simulate decryption queries in the security proof for OAEP [FOPS01]. For quantum-accessible oracles the actual query may be hidden in a superposition of exponentially many states, and it is unclear how to extract the right query.
- Efficient Simulation: In the classical world, we can simulate an exponential-size random oracle efficiently via lazy sampling: simply pick random but consistent answers "on the fly". With quantum-accessible random oracles the adversary can evaluate the random oracle on all inputs simultaneously, making it harder to apply the on-demand strategy for classical oracles.
- Rewinding/Partial Consistency: Certain random oracle proofs [PS00] require rewinding the adversary, replaying some hash values but changing at least a single value. Beyond the usual problems of rewinding quantum adversaries, we again encounter the fact that we may not be able to change hash values unnoticed. We note that some form of rewinding is possible for quantum zero-knowledge [Wat09].

We do not claim that these problems are insurmountable. In fact, we show how to resolve the issue of efficient simulation by using (quantum-accessible) pseudorandom functions. These are pseudorandom functions where the quantum distinguisher can submit quantum states to the pseudorandom or random oracle. By this technique, we can efficiently simulate the quantum-accessible random oracle through the (efficient) pseudorandom function. While pseudorandom functions where the distinguisher may use quantum power but only gets classical access to the function can be derived from quantum-immune pseudorandom generators [GGM86], it is an open problem if the stronger quantum-accessible pseudorandom functions exist.

Note, too, that we do not seek to solve the problems related to the random oracle model which appear already in the classical settings [CGH98]. Instead we show that for post-quantum security one should allow for quantum access to the random oracle in order to capture attacks that are available when the hash function is eventually instantiated.

1.1 Our Contributions

Separation. We begin with a separation between the classical and quantum-accessible RO models by presenting a two-party protocol which is:

- secure in the classical random oracle model,

- secure against quantum attackers with classical access to the random oracle model, but insecure under *any* implementation of the hash function, and
- insecure in the quantum-accessible random oracle model.

The protocol itself assumes that (asymptotically) quantum computers are faster than classical (parallel) machines and uses the quadratic gap due to Grover's algorithms and its application to collision search [BHT98] to separate secure from insecure executions.

Constructions. Next, we set out to give general conditions under which a *classical* RO proof implies security for a *quantum* RO. Our goal is to provide generic tools by which authors can simply state that their classical proof has the "right" structure and therefore their proof implies quantum security. We give two flavors of results:

- For signatures, we define a proof structure we call a *history-free reduction* which roughly says that the reduction answers oracle queries independently of the history of queries. We prove that any classical proof that happens to be a history-free reduction implies quantum existential unforgeability for the signature scheme. We then show that the GPV random oracle signature scheme [GPV08] has a history-free reduction and is therefore secure in the quantum settings.

  Next, we consider signature schemes built from claw-free permutations. The first is the Full Domain Hash (FDH) signature system of Bellare and Rogaway [BR93], for which we show that the classical proof technique due to Coron [Cor00] is history-free. We also prove the quantum security of a variant of FDH due to Katz and Wang [KW03] which has a tight security reduction. Lastly, we note that, as observed in [GPV08], claw-free permutations give rise to preimage sampleable trapdoor functions, which gives another FDH-like signature scheme with a tight security reduction. In all three cases the reductions in the quantum-accessible random oracle model achieve essentially the same tightness as their classical analogs.

  Interestingly, we do not know of a history-free reduction for the generic Full Domain Hash of Bellare and Rogaway [BR93]. One reason is that proofs for generic FDH must somehow program the random oracle, as shown in [FLR$^+$10]. We leave the quantum security of generic FDH as an interesting open problem. It is worth noting that at this time the quantum security of FDH is somewhat theoretical since we have no candidate quantum-secure trapdoor permutation to instantiate the FDH scheme, though this may change once a candidate is proposed.
- For encryption we prove the quantum CPA security of an encryption scheme due to Bellare and Rogaway [BR93] and the quantum CCA security of a hybrid encryption variant of [BR93].

Many open problems remain in this space. For signatures, it is still open to prove the quantum security of signatures that result from applying the Fiat-Shamir heuristic to a Σ identification protocol, for example, as suggested in [CLRS10]. Similarly, proving security of generic FDH is still open. For CCA-secure encryption, it is unknown if generic CPA to CCA transformations, such as [FO99], are secure in the quantum settings. Similarly, it is not known if lattice-based identity-based encryption systems secure in the classical RO model (e.g. as in [GPV08,ABB10b]) are also secure in the quantum random oracle model.

Related Work. The quantum random oracle model has been used in a few previous constructions. Aaronson [Aar09] uses quantum random oracles to construct unclonable public-key quantum money. Brassard and Salvail [BS08] give a modified version of Merkle's Puzzles, and show that any quantum attacker must query the random (permutation) oracle asymptotically more times than honest parties. Recently, a modified version was proposed that restores some level of security even in the presence of a quantum adversary [BHK$^+$11]. Quantum random oracles have also been used to prove impossibility results for quantum computation. For example, Bennett et al. [BBBV97] show that relative to a random oracle, a quantum computer cannot solve all of NP.

Some progress toward identifying sufficient conditions under which classical protocols are also quantum immune has been made by Unruh [Unr10] and Hallgren et al. [HSS11]. These results show that, if a cryptographic protocol can be shown to be (computationally [HSS11] resp. statistically [Unr10]) secure in Canetti's universal composition (UC) framework [Can01] against classical adversaries, then the protocol is also resistant against (computationally bounded resp. unbounded) quantum adversaries. This, however, means that the underlying protocol must already provide strong security guarantees in the first place, namely, universal composition security, which is typically more than the aforementioned schemes in the literature satisfy. This also applies to similar results by Hallgren et al. [HSS11] for so-called simulation-based security notions for the starting protocol. Furthermore, all these results do not seem to be applicable immediately to the random oracle model where the quantum adversary now has *quantum* access to the random function (but where the ideal functionality for the random oracle in the UC framework would have only been defined for classical access according to the classical protocol specification), and where the question of instantiation is an integral step which needs to be considered.

2 Preliminaries

A non-negative function $\epsilon = \epsilon(n)$ is negligible if, for all polynomials $p(n)$ we have that $\epsilon(n) < p(n)^{-1}$ for all sufficiently large n. The variational distance between two distributions D_1 and D_2 over Ω is given by

$$|D_1 - D_2| = \sum_{x\in\Omega} |\Pr[x|D_1] - \Pr[x|D_2]|.$$

If the distance between two output distributions is ϵ, the difference in probability of the output satisfying a certain property is at most ϵ.

A classical randomized algorithm A can be thought of in two ways. In the first, A is given an input x, A makes some coin tosses during its computation, and ultimately outputs some value y. We denote this action by $A(x)$ where $A(x)$ is a random variable. Alternatively, we can give A both its input x and randomness r in which case we denote this action as $A(x;r)$. For a classical algorithm, $A(x;r)$ is deterministic. An algorithm A runs is probabilistic polynomial-time (PPT) if it runs in polynomial time in the security parameter (which we often omit from the input for sake of simplicity).

2.1 Quantum Computation

We briefly give some background on quantum computation and refer to [NC00] for a more complete discussion. A quantum system A is associated to a (finite-dimensional) complex Hilbert space $\mathcal{H}_A$ with an inner product $\langle \cdot | \cdot \rangle$. The state of the system is described by a vector $|\varphi\rangle \in \mathcal{H}_A$ such that the Euclidean norm $\| \, |\varphi\rangle \, \| = \sqrt{\langle \varphi | \varphi \rangle}$ is 1. Given quantum systems A and B over spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, we define the joint or composite quantum system through the tensor product $\mathcal{H}_A \otimes \mathcal{H}_B$. The product state of $|\varphi_A\rangle \in \mathcal{H}_A$ and $|\varphi_B\rangle \in \mathcal{H}_B$ is denoted by $|\varphi_A\rangle \otimes |\varphi_B\rangle$ or simply $|\varphi_A\rangle |\varphi_B\rangle$. An n-qubit system lives in the joint quantum system of n two-dimensional Hilbert spaces. The standard orthonormal computational basis $|x\rangle$ for such a system is given by $|x_1\rangle \otimes \cdots \otimes |x_n\rangle$ for $x = x_1 \ldots x_n$. Any (classical) bit string x is encoded into a quantum state as $|x\rangle$. An arbitrary pure n-qubit state $|\varphi\rangle$ can be expressed in the computational basis as $|\varphi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ where α_x are complex amplitudes obeying $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

Transformations. Evolutions of quantum systems are described by unitary transformations with $\mathbb{I}_A$ being the identity transformation on register A. Given a joint quantum system over $\mathcal{H}_A \otimes \mathcal{H}_B$ and a transformation U_A acting only on $\mathcal{H}_A$, it is understood that $U_A |\varphi_A\rangle |\varphi_B\rangle$ refers to $(U_A \otimes \mathbb{I}_B) |\varphi_A\rangle |\varphi_B\rangle$.

Information can be extracted from a quantum state $|\varphi\rangle$ by performing a positive-operator valued measurement (POVM) $M = \{M_i\}$ with positive semi-definite measurement operators M_i that sum to the identity $\sum_i M_i = \mathbb{I}$. Outcome i is obtained with probability $p_i = \langle \varphi | M_i |\varphi\rangle$. A special case are projective measurements such as the measurement in the computational basis of the state $|\varphi\rangle = \sum_x \alpha_x |x\rangle$ which yields outcome x with probability $|\alpha_x|^2$. We can also do a partial measurement on some of the qubits. The probability of the partial measurement resulting in a string x is the same as if we measured the whole state, and ignored the rest of the qubits. In this case, the resulting state will be the same as $|\phi\rangle$, except that all the strings inconsistent with x are removed. This new state will not have a norm of 1, so the actual superposition is obtained by dividing by the norm. For example, if we measure the first n bits of $|\phi\rangle = \sum_{x,y} \alpha_{x,y} |x, y\rangle$, we will obtain the measurement x with probability $\sum_{y'} |\alpha_{x,y'}|^2$, and in this case the resulting state will be

$$|x\rangle \sum_y \frac{\alpha_{x,y}}{\sqrt{\sum_{y'} |\alpha_{x,y'}|^2}} |y\rangle.$$

Following [BBC+98], we model a quantum attacker $\mathcal{A}_{\mathrm{Q}}$ with access to (possibly identical) oracles $O_1, O_2, \ldots$ by a sequence of unitary transformations $U_1, O_1, U_2, \ldots, O_{T-1}, U_T$ over $k = \mathrm{poly}(n)$ qubits. Here, oracle $O_i : \{0,1\}^n \to \{0,1\}^m$ maps the first $n + m$ qubits from basis state $|x\rangle\,|y\rangle$ to basis state $|x\rangle\,|y \oplus O_i(x)\rangle$ for $x \in \{0,1\}^n$ and $y \in \{0,1\}^m$. If we require the access to O_i to be classical instead of quantum, the first n bits of the state are measured before applying the unitary transformation corresponding to O_i. Notice that any quantum-accessible oracle can also be used as a classical oracle. Note that the algorithm $\mathcal{A}_{\mathrm{Q}}$ may also receive some input $|\psi\rangle$. Given an algorithm $\mathcal{A}_{\mathrm{Q}}$ as above, with access to oracles O_i, we sometimes write $\mathcal{A}_{\mathrm{Q}}^{|O_1(\cdot)\rangle, |O_2(\cdot)\rangle, \ldots}$ to indicate that the oracle is quantum-accessible (contrary to oracles which can only process classical bits).

To introduce asymptotics we assume that $\mathcal{A}_{\mathrm{Q}}$ is actually a sequence of such transformation sequences, indexed by parameter n, and that each transformation sequence is composed out of quantum systems for input, output, oracle calls, and work space (of sufficiently many qubits). To measure polynomial running time, we assume that each U_i is approximated (to sufficient precision) by members of a set of universal gates (say, Hadamard, phase, CNOT and $\pi/8$; for sake of concreteness [NC00]), where at most polynomially many gates are used. Furthermore, $T = T(n)$ is assumed to be polynomial, too. Note that T also bounds the number of oracle queries.

We define the Euclidean distance $||\phi\rangle - |\psi\rangle|$ between two states as the value $\left(\sum_x |\alpha_x - \beta_x|^2\right)^{\frac{1}{2}}$ where $|\phi\rangle = \sum_x \alpha_x |x\rangle$ and $|\psi\rangle = \sum_x \beta_x |x\rangle$.

Define $q_r(|\phi_t\rangle)$ to be the magnitude squared of r in the superposition of query t. We call this the query probability of r in query t. If we sum over all t, we get the total query probability of r.

We will be using the following lemmas:

Lemma 1 ([BBBV97] Theorem 3.1). *Let $|\varphi\rangle$ and $|\psi\rangle$ be quantum states with Euclidean distance at most ϵ. Then, performing the same measurement on $|\varphi\rangle$ and $|\psi\rangle$ yields distributions with statistical distance at most 4ϵ.*

Lemma 2 ([BBBV97] Theorem 3.3). *Let A_Q be a quantum algorithm running in time T with oracle access to O. Let $\epsilon > 0$ and let $S \subseteq [1, T] \times \{0,1\}^n$ be a set of time-string pairs such that $\sum_{(t,r) \in S} q_r(|\phi_t\rangle) \leq \epsilon$. If we modify O into an oracle O' which answers each query r at time t by providing the same string R (which has been independently sampled at random), then the Euclidean distance between the final states of A_Q when invoking O and O' is at most $\sqrt{T\epsilon}$.*

2.2 Quantum-Accessible Random Oracles

In the classical random oracle model [BR93] all algorithms used in the system are given access to the same random oracle. In the proof of security, the reduction algorithm answers the adversary's queries with consistent random answers.

In the quantum settings, a quantum attacker issues a random oracle query which is itself a superposition of exponentially many states. The reduction algorithm must evaluate the random oracle at all points in the superposition. To ensure that random oracle queries are answered consistently across queries, it is convenient to assume that quantum-resistant pseudorandom functions exist, and to implement this auxiliary random oracle with such a PRF.

Definition 1 (Pseudorandom Function). *A quantum-accessible pseudorandom function is an efficiently computable function* PRF *where, for all efficient quantum algorithms* D*,*

$$\left|\Pr[D^{\mathsf{PRF}(k,\cdot)}(1^n) = 1] - \Pr[D^{O(\cdot)}(1^n) = 1]\right| < \epsilon$$

where $\epsilon = \epsilon(n)$ *is negligible in* n*, and where* O *is a random oracle, the first probability is over the keys* k *of length* n*, and the second probability is over all random oracles and the sampling of the result of* D*.*

We note that, following Watrous [Wat09], indistinguishability as above should still hold for any auxiliary quantum state σ given as additional input to D (akin to non-uniformity for classical algorithms). We do not include such auxiliary information in our definition in order to simplify.

We say that an oracle O' is computationally indistinguishable from a random oracle if, for all polynomial time quantum algorithms with oracle access, the variational distance of the output distributions when the oracle is O' and when the oracle is a truly random oracle O is negligible. Thus, simulating a random oracle with a quantum-accessible pseudorandom function is computationally indistinguishable from a true random oracle.

We remark that, instead of assuming that quantum-accessible PRFs exist, we can often carry out security reductions relative to a random oracle. Consider, for example, a signature scheme (in the quantum-accessible random oracle model) which we prove to be unforgeable for quantum adversaries, via a reduction to the one-wayness of a trapdoor permutation against quantum inverters. We can then formally first claim that the scheme is unforgeable as long as inverting the trapdoor permutation is infeasible even when having the additional power of a quantum-accessible random oracle; only in the next step we can then conclude that this remains true in the standard model, if we assume that quantum-accessible pseudorandom functions exist and let the inverter simulate the random oracle with such a PRF. We thus still get a potentially reasonable security claim even if such PRFs do not exist. This technique works whenever we can determine the success of the adversary (as in case of inverting a one-way function).

2.3 Hard Problems for Quantum Computers

We will use the following general notion of a hard problem.

Definition 2 (Problem). *A problem is a pair $P = (Game_P, \alpha_P)$ where $Game_P$ specifies a game that a (possibly quantum) adversary plays with a classical challenger. The game works as follows:*

- *On input 1^n, the challenger computes a value x, which it sends to the adversary as its input*
- *The adversary is then run on x, and is allowed to make classical queries to the challenger.*
- *The adversary then outputs a value y, which it sends to the challenger.*
- *The challenger then looks at x, y, and the classical queries made by the adversary, and outputs 1 or 0.*

The value α_P is a real number between 0 (inclusive) and 1 (exclusive). It may also be a function of n, but for this paper, we only need constant α_P, specifically α_P is always 0 or $\frac{1}{2}$.

We say that an adversary A wins the game Game_P if the challenger outputs 1. We define the advantage $\text{Adv}_{A,P}$ of A in problem P as

$$Adv_{A,P} = |\Pr[A \text{ wins in } \text{Game}_P] - \alpha_P|$$

Definition 3 (Hard Problem). *A problem $P = (Game_P, \alpha_P)$ is hard for quantum computers if, for all polynomial time quantum adversaries A, $Adv_{A,P}$ is negligible.*

2.4 Cryptographic Primitives

For this paper, we define the security of standard cryptographic primitives in terms of certain problems being hard for quantum computers. We give a brief sketch here and refer to the appendix for supplementary details.

A trapdoor function $\mathcal{F}$ is secure if $\text{Inv}(\mathcal{F}) = (\text{Game}_{\text{INV}}(\mathcal{F}), 0)$ is a hard problem for quantum computers, where in Game_{INV}, an adversary is given a random element y and public key, and succeeds if it can output an inverse for y relative to the public key. A preimage sampleable trapdoor function, $\mathcal{F}$, is secure if $\text{Inv}(\mathcal{F})$ as described above is hard, and if $\text{Col}(\mathcal{F}) = (\text{Game}_{\text{Col}}(\mathcal{F}), 0)$ is hard for quantum computers, where in Game_{Col}, an adversary is given a public key, succeeds if it can output a collision relative to that public key. A signature scheme $\mathcal{S}$ is secure if the game $\text{Sig-Forge}(\mathcal{S}) = (\text{Game}_{\text{Sig}}(\mathcal{S}), 0)$ is hard, where Game_{Sig} is the standard existential unforgeability under a chosen message attack game. Lastly, a private (resp. public) key encryption scheme $\mathcal{E}$ is secure if $\text{Sym-CCA}(\mathcal{E}) = (\text{Game}_{\text{Sym}}(\mathcal{E}), \frac{1}{2})$ (resp. $\text{Asym-CCA}(\mathcal{E}) = (\text{Game}_{\text{Asym}}(\mathcal{E}), \frac{1}{2})$), where Game_{Sym} is the standard private key CCA attack game, and $\text{Game}_{\text{Asym}}$ is the standard public key attack game.

3 Separation Result

In this section, we discuss a two-party protocol that is provably secure in the random oracle model against both classical and quantum adversaries with classical access to the random oracle (and when using quantum-immune primitives). We then use the polynomial gap between the birthday attack and a collision finder based on Grover's algorithm to show that the protocol remains secure for certain hash functions when only classical adversaries are considered, but becomes insecure for any hash function if quantum adversaries are allowed. Analyzing the protocol in the stronger quantum random oracle model, where we grant the adversary quantum access to the random oracle, yields the same negative result.

3.1 Preliminaries

We start this section by presenting the necessary definitions and assumptions for our construction. For sake of simplicity, we start with a quantum-immune identification scheme to derive our protocol; any other primitive or protocol can be used in a similar fashion.

Identification Schemes. An identification scheme IS consists of three efficient algorithms $(\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ where $\mathrm{IS}.\mathsf{KGen}$ on input 1^n returns a key pair $(\mathsf{sk}, \mathsf{pk})$. The joint execution of $\mathcal{P}(\mathsf{sk}, \mathsf{pk})$ and $\mathcal{V}(\mathsf{pk})$ then defines an interactive protocol between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$. At the end of the protocol $\mathcal{V}$ outputs a decision bit $b \in \{0, 1\}$. We assume completeness in the sense that for any honest prover the verifier accepts the interaction with output $b = 1$. Security of identification schemes is usually defined by considering an adversary $\mathcal{A}$ that first interacts with the honest prover to obtain some information about the secret key. In a second stage, the adversary then plays the role of the prover and has to make a verifier accept the interaction. We say that an identification scheme is *sound* if the adversary can convince the verifier with negligible probability only.

(Near-)Collision-Resistant Hash Functions. A hash function $\mathrm{H} = (\mathrm{H}.\mathsf{KGen}, \mathrm{H}.\mathsf{Eval})$ is a pair of efficient algorithms such that $\mathrm{H}.\mathsf{KGen}$ for input 1^n returns a key k (which contains 1^n), and $\mathrm{H}.\mathsf{Eval}$ for input k and $M \in \{0, 1\}^*$ deterministically outputs a digest $\mathrm{H}.\mathsf{Eval}(k, M)$. For a random oracle H we use k as a "salt" and consider the random function $H(k, \cdot)$. The hash function is called *near-collision-resistant* if for any efficient algorithm $\mathcal{A}$ the probability that for $k \leftarrow \mathrm{H}.\mathsf{KGen}(1^n)$, some constant $1 \le \ell \le n$ and $(M, M') \leftarrow \mathcal{A}(k, \ell)$ we have $M \ne M'$ but $\mathrm{H}.\mathsf{Eval}(k, M)|_\ell = \mathrm{H}.\mathsf{Eval}(k, M')|_\ell$, is negligible (as a function of n). Here we denote by $x|_\ell$ the leading ℓ bits of the string x. Note that for $\ell = n$ the above definition yields the standard notion of collision-resistance.

In the classical setting, (near-)collision-resistance for any hash function is upper bounded by the *birthday attack*. This generic attack state that for any hash function with n bits output, an attacker can find a collision with probability roughly $1/2$ by probing $2^{n/2}$ distinct and random inputs. For random oracles this attack is optimal.

Grover's Algorithm and Quantum Collision Search. Grover's algorithm [Gro96,Gro98] performs a search on an unstructured database with N elements in time $O(\sqrt{N})$ while the best classical algorithm requires $O(N)$ steps. Roughly, this is achieved by using superpositions to examine all entries "at the same time". Brassard et al. [BHT98] showed that this speed-up can also be obtained for solving the collision problem for a hash function $H : \{0,1\}^* \to \{0,1\}^n$. Therefore, one first selects a subset K of the domain $\{0,1\}^*$ and then applies Grover's algorithm on an indicator function f that tests for any input $M \in \{0,1\}^* \backslash K$ if there exists an $M' \in K$ such that $H(M) = H(M')$ holds. By setting $|K| = \sqrt[3]{2^n}$, the algorithm finds a collision after $O(\sqrt[3]{2^n})$ evaluations of H with probability at least 1/2.

Computational and Timing Assumptions. To allow reasonable statements about the security of our protocol we need to formalize assumptions concerning the computational power of the adversary and the time that elapses on quantum and classical computers. We first demand that the speed-up one can gain by using a parallel machine with many processors, is bounded by a fixed term. This basically resembles the fact that in the real world there is only a concrete and finite amount of equipment available that can contribute to such a performance gain.

Assumption 1 (Parallel Speed-Up) *Let $T(C)$ denote the time that is required to solve a problem C on a classical computer, and $T_P(C)$ is the required time that elapses on a parallel system. Then, there exist a constant $\alpha \geq 1$, such that for any problem C it holds that $T_P(C) \geq T(C)/\alpha$.*

We also introduce two assumptions regarding the time that is needed to evaluate a hash function or to send a message between two parties. Note that both assumptions are merely for the sake of convenience, as one could patch the idea by relating the timings more rigorously. The first assumption states that the time that is required to evaluate a hash function H is independent of the input and the computational environment.

Assumption 2 (Unit Time) *For any hash function H and any input message M (resp. M_Q for quantum-state inputs) the evaluation of $H(M)$ requires a constant time $T(H(M)) = T_P(H(M)) = T_Q(H(M_Q))$ (where T_Q denotes the time that elapses on a quantum computer).*

Furthermore, we do not charge any extra time for sending and receiving messages, or for any computation other than evaluating a hash function (e.g., maintaining lists of values).

Assumption 3 (Zero Time) *Any computation or action that does not require the evaluation of a hash function, costs zero time.*

The latter assumption implicitly states that the computational overhead that quantum algorithms may create to obtain a speed-up is negligible when compared to the costs of a hash evaluation. This might be too optimistic in the near future, as indicated by Bernstein [Ber09]. That is, Bernstein discussed that the overall costs of a quantum computation can be higher than of massive parallel computation. However, as our work addresses conceptional issues that arise when *efficient* quantum computers exist, this assumption is somewhat inherent in our scenario.

3.2 Construction

We now present our identification scheme between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. The main idea is to augment a secure identification scheme IS by a collision-finding stage for some hash function H. In this first stage, the verifier checks if the prover is able to produce collisions on a hash function in a particular time. More precisely, the verifier starts for timekeeping to evaluate the hash function $\mathrm{H}.\mathsf{Eval}(k,\cdot)$ on the messages $\langle c\rangle$ for $c = 1, 2, \ldots, \left\lceil \sqrt[3]{2^\ell} \right\rceil$ for a key k chosen by the verifier and where $\langle c\rangle$ stands for the binary representation of c with $\log \left\lceil \sqrt[3]{2^\ell} \right\rceil$ bits. The prover has now to respond with a near-collision $M \neq M'$ such that $\mathrm{H}.\mathsf{Eval}(k, M) = \mathrm{H}.\mathsf{Eval}(k, M')$ holds for the first ℓ bits. One round of the collision-stage ends if the verifier either receives such a collision or finishes its $\sqrt[3]{2^\ell}$ hash evaluations. The verifier and the receiver then repeat such a round $r = \mathrm{poly}(n)$ times, sending a fresh key k in each round.

Subsequently, both parties run the standard identification scheme. At the end, the verifier accepts if the prover was able to find enough collisions in the first stage or identifies correctly in the second stage. Thus, as long as the prover is not able to produce collisions in the required time, the protocol mainly resembles the IS protocol.

Completeness of the IS^* protocol follows easily from the completeness of the underlying IS scheme.

Security against Classical and Quantum Adversaries. To prove security of our protocol, we need to show that an adversary $\mathcal{A}$ after interacting with an honest prover $\mathcal{P}^*$, can subsequently not impersonate $\mathcal{P}^*$ such that $\mathcal{V}^*$ will accept the identification. Let ℓ be such that $\ell > 6\log(\alpha)$ where α is the constant reflecting the bounded speed-up in parallel computing from Assumption (1). By assuming that $\mathrm{IS} = (\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ is a quantum-immune identification scheme, we can show that IS^* is secure in the standard random oracle model against classical and quantum adversaries.

The main idea is that for the standard random oracle model, the ability of finding collisions is bounded by the birthday attack. Due to the constraint of granting only time $O(\sqrt[3]{2^\ell})$ for the collision search and setting $\ell > 6\log(\alpha)$, even an adversary with quantum or parallel power is not able to make at least $\sqrt{2^\ell}$ random oracle queries. Thus, $\mathcal{A}$ has only negligible probability to respond in more than $1/4$ of r rounds with a collision.

When considering only classical adversaries, we can also securely instantiate the random oracle by a hash function H that provides near-collision-resistance

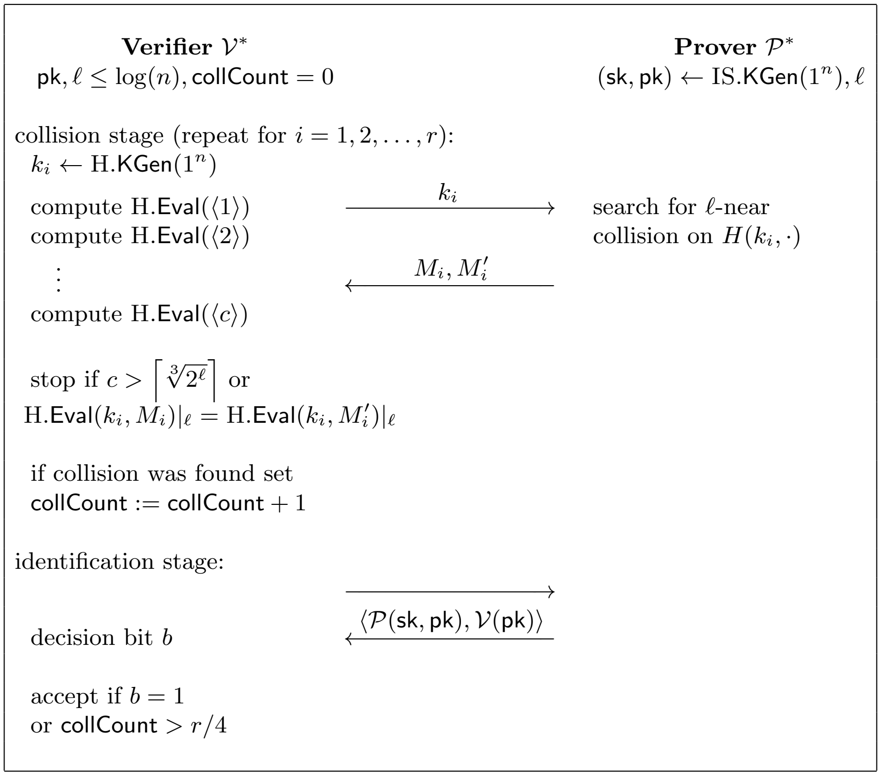
Verifier $\mathcal{V}^*$
$\mathsf{pk}, \ell \leq \log(n), \mathsf{collCount} = 0$
Prover $\mathcal{P}^*$
$(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathrm{IS}.\mathsf{KGen}(1^n), \ell$
collision stage (repeat for $i = 1, 2, \dots, r$):
$k_i \leftarrow \mathrm{H}.\mathsf{KGen}(1^n)$
compute $\mathrm{H}.\mathsf{Eval}(\langle 1 \rangle)$
compute $\mathrm{H}.\mathsf{Eval}(\langle 2 \rangle)$
$\vdots$
compute $\mathrm{H}.\mathsf{Eval}(\langle c \rangle)$
$\xrightarrow{k_i}$
search for ℓ-near
collision on $H(k_i, \cdot)$
$\xleftarrow{M_i, M_i'}$
stop if $c > \left\lceil \sqrt[3]{2^\ell} \right\rceil$ or
$\mathrm{H}.\mathsf{Eval}(k_i, M_i)|_\ell = \mathrm{H}.\mathsf{Eval}(k_i, M_i')|_\ell$
if collision was found set
$\mathsf{collCount} := \mathsf{collCount} + 1$
identification stage:
$\langle \mathcal{P}(\mathsf{sk}, \mathsf{pk}), \mathcal{V}(\mathsf{pk}) \rangle$
decision bit b
accept if $b = 1$
or $\mathsf{collCount} > r/4$

Fig. 1. The IS*-Identification Protocol

close to the birthday bound. Note that this property is particularly required from the SHA-3 candidates [NIS07].

However, for adversaries $\mathcal{A}_{\mathrm{Q}}$ with quantum power, such an instantiation is not possible for *any* hash function. This stems from the fact that $\mathcal{A}_{\mathrm{Q}}$ can locally evaluate a hash function on quantum states which in turns allows it to apply Grover's search algorithm. Then an adversary will find a collision in time $\sqrt[3]{2^\ell}$ with probability at least $1/2$, and thus will be able to provide $r/4$ collisions with noticeable probability. The same result holds in the quantum-accessible random oracle model, since Grover's algorithm only requires (quantum) black-box access to the hash function.

Formal proofs of all statements are given in Appendix B.

4 Signature Schemes in the Quantum-Accessible Random Oracle Model

We now turn to proving security in the quantum-accessible random oracle model. We present general conditions for when a proof of security in the classical random oracle model implies security in the quantum-accessible random oracle model. The result in this section applies to signatures whose classical proof of security is a *history-free reduction* as defined next. Roughly speaking, history-freeness means that the classical proof of security simulates the random oracle and signature oracle in a history-free fashion. That is, its responses to queries do not depend on responses to previous queries or the query number. We then show that a number of classical signature schemes have a history-free reduction thereby proving their security in the quantum-accessible random oracle model.

Definition 4 (History-free Reduction). *A random oracle model signature scheme $\mathcal{S} = (G, S^O, V^O)$ has a history-free reduction from a hard problem $P = (Game_P, 0)$ if there is a proof of security that uses a classical PPT adversary A for $\mathcal{S}$ to construct a classical PPT algorithm B for problem P such that:*

- *Algorithm B for P contains four explicit classical algorithms:* START, RAND^{O_c}, SIGN^{O_c}, *and* FINISH^{O_c}. *The latter three algorithms have access to a shared classical random oracle O_c. These algorithms, except for* RAND^{O_c}, *may also make queries to the challenger for problem P. The algorithms are used as follows:*
  - *(1) Given an instance x for problem P as input, algorithm B first runs* $\mathrm{START}(x)$ *to obtain (pk, z) where pk is a signature public key and z is private state to be used by B. Algorithm B sends pk to A and plays the role of challenger to A.*
  - *(2) When A makes a classical random oracle query to $O(r)$, algorithm B responds with* $\mathrm{RAND}^{O_c}(r, z)$. *Note that* RAND *is given the current query as input, but is unaware of previous queries and responses.*
  - *(3) When A makes a classical signature query $S(\mathsf{sk}, m)$, algorithm B responds with* $\mathrm{SIGN}^{O_c}(m, z)$.
  - *(4) When A outputs a signature forgery candidate (m, σ), algorithm B outputs* $\mathrm{FINISH}^{O_c}(m, \sigma, z)$.
- *There is an efficiently computable function* $\mathrm{INSTANCE}(\mathsf{pk})$ *which produces an instance x of problem P such that* $\mathrm{START}(x) = (\mathsf{pk}, z)$ *for some z. Consider the process of first generating $(\mathsf{sk}, \mathsf{pk})$ from $G(1^n)$, and then computing* $x = \mathrm{INSTANCE}(\mathsf{pk})$. *The distribution of x generated in this way is negligibly close to the distribution of x generated in $Game_P$.*
- *For fixed z, consider the classical random oracle* $O(r) = \mathrm{RAND}^{O_c}(r, z)$. *Define a quantum oracle O_{quant}, which transforms a basis element $|x, y\rangle$ into $|x, y \oplus O(x)\rangle$. We require that O_{quant} is quantum computationally indistinguishable from a random oracle.*

- SIGN^{O_c} *either aborts (and hence B aborts) or it generates a valid signature relative to the oracle $O(r) = \mathrm{RAND}^{O_c}(r, z)$ with a distribution negligibly close to the correct signing algorithm. The probability that none of the signature queries abort is non-negligible.*
- *If (m, σ) is a valid signature forgery relative to the public key* pk *and oracle $O(r) = \mathrm{RAND}^{O_c}(r, z)$ then the output of B (i.e. $\mathrm{FINISH}^{O_c}(m, \sigma, z)$) causes the challenger for problem P to output 1 with non-negligible probability.* □

We now show that history-free reductions imply security in the quantum settings.

Theorem 1. *Let $\mathcal{S} = (G, S, V)$ be a signature scheme. Suppose that there is a history-free reduction that uses a classical PPT adversary A for $\mathcal{S}$ to construct a PPT algorithm B for a problem P. Further, assume that P is hard for polynomial-time quantum computers, and that quantum-accessible pseudorandom functions exist. Then $\mathcal{S}$ is secure in the quantum-accessible random oracle model.*

Proof. The history-free reduction includes five (classical) algorithms START, RAND, SIGN, FINISH, and INSTANCE, as in Definition 4. We prove the quantum security of $\mathcal{S}$ using a sequence of games, where the first game is the standard quantum signature game with respect to $\mathcal{S}$.

Game 0. Define Game_0 as the game a quantum adversary A_Q plays for problem Sig-Forge($\mathcal{S}$). Assume towards contradiction that A_Q has a non-negligible advantage.

Game 1. Define Game_1 as the following modification to Game_0: after the challenger generates $(\mathsf{sk}, \mathsf{pk})$, it computes $x \leftarrow \mathrm{INSTANCE}(\mathsf{pk})$ as well as $(\mathsf{pk}, z) \leftarrow \mathrm{START}(x)$. Further, instead of answering A_Q's quantum random oracle queries with a truly random oracle, the challenger simulates for A_Q a quantum-accessible random oracle O_{quant} as an oracle that maps a basis element $|x, y\rangle$ into the element $\left|x, y \oplus \mathrm{RAND}^{O_q}(x, z)\right\rangle$, where O_q is a truly random quantum-accessible oracle. The history-free guarantee on RAND ensures that O_{quant} is computationally indistinguishable from random for quantum adversaries. Therefore, the success probability of A_Q in Game_1 is negligibly close to its success probability in Game_0, and hence is non-negligible.

Game 2. Modify the challenger from Game_1 as follows: instead of generating $(\mathsf{sk}, \mathsf{pk})$ and computing $x = \mathrm{INSTANCE}(\mathsf{pk})$, start off by running the challenger for problem P. When that challenger sends x, then start the challenger from Game_1 using this x. Also, when A_Q asks for a signature on m, answer with $\mathrm{SIGN}^{O_q}(m, z)$. First, since INSTANCE is part of a history-free reduction, this change in how we compute x only negligibly affects the distribution of x, and hence the behavior of A_Q. Second, as long as all signing algorithms succeed, changing how we answer signing queries only negligibly affects the behavior of A_Q. Thus, the probability that A_Q succeeds is the product of the following two probabilities:

- The probability that all of the signing queries are answered without aborting.

- The probability that A_Q produces a valid forgery given that the signing queries were answered successfully.

The first probability is non-negligible by assumption, and the second is negligibly close to the success probability of A_Q in Game$_1$, which is also non-negligible. This means that the success probability of A_Q in Game$_3$ is non-negligible.

Game 3. Define Game$_3$ as in Game$_2$, except that for two modifications to the challenger: First, it generates a key k for the quantum-accessible PRF. Then, to answer a random oracle query $O_q(|\phi\rangle)$, the challenger applies the unitary transformation that takes a basis element $|x, y\rangle$ into $|x, y \oplus \mathsf{PRF}(k, x)\rangle$. If the success probability in Game$_3$ was non-negligibly different from that of Game$_2$, we could construct a distinguisher for PRF which plays both the role of A_Q and the challenger. Hence, the success probability in Game$_3$ is negligibly close to that of Game$_2$, and hence is also non-negligible.

Given a quantum adversary that has non-negligible advantage in Game 3 we construct a quantum algorithm B_Q that breaks problem P. When B_Q receives instance x from the challenger for problem P, it computes $(\mathsf{pk}, z) \leftarrow \mathrm{START}(x)$ and generates a key k for PRF. Then, it simulates A_Q on pk. B_Q answers random oracle queries using a quantum-accessible function built from $\mathrm{RAND}^{\mathsf{PRF}(k,\cdot)}(\cdot, z)$ as in Game 1. It answers signing queries using $\mathrm{SIGN}^{\mathsf{PRF}(k,\cdot)}(\cdot, z)$. Then, when A_Q outputs a forgery candidate (m, σ), B_Q computes $\mathrm{FINISH}^{\mathsf{PRF}(k,\cdot)}(m, \sigma, z)$, and returns the result to the challenger for problem P.

Observe that the behavior of A_Q in Game$_3$ is identical to that as a subroutine of B_Q. Hence, A_Q as a subroutine of B_Q will output a valid forgery (m, σ) with non-negligible probability. If (m, σ) is a valid forgery, then since FINISH is part of a history-free reduction, $\mathrm{FINISH}^{\mathsf{PRF}(k,\cdot)}(m, \sigma, z)$ will cause the challenger for problem P to accept with non-negligible probability. Thus, the probability that P accepts is also non-negligible, contradicting our assumption that P is hard for quantum computers.

Hence we have shown that any polynomial quantum algorithm has negligible advantage against problem Sig-Forge($\mathcal{S}$) which completes the proof. □

We note that, in every step of the algorithm, the adversary A_Q remains in a pure state. This is because, in each game, A_Q's state is initially pure (since it is classical), and every step of the game either involves a unitary transformation, a partial measurement, or classical communication. In all three cases, if the state is pure before, it is also pure after.

We also note that we could have stopped at Game$_2$ and assumed that the cryptographic problem P is hard relative to a (quantum-accessible) random oracle. Assuming the existence of quantum-accessible pseudorandom functions allows us to draw the same conclusion in the standard (i.e., non-relativized) model at the expense of an extra assumption.

4.1 Secure Signatures From Preimage Sampleable Trapdoor Functions (PSF)

We now use Theorem 1 to prove the security of the Full Domain Hash signature scheme when instantiated with a preimage sampleable trapdoor function (PSF), such as the one proposed in [GPV08]. Loosely speaking, a PSF $\mathcal{F}$ is a tuple of PPT algorithms $(G, \text{Sample}, f, f^{-1})$ where $G(\cdot)$ generates a key pair $(\mathsf{pk}, \mathsf{sk})$, $f(\mathsf{pk}, \cdot)$ defines an efficiently computable function, $f^{-1}(\mathsf{sk}, y)$ samples from the set of pre-images of y, and $\text{Sample}(\mathsf{pk})$ samples x from the domain of $f(\mathsf{pk}, \cdot)$ such that $f(\mathsf{pk}, x)$ is statistically close to uniform in the range of $f(\mathsf{pk}, \cdot)$. The PSF of [GPV08] is not only one-way, but is also collision resistant.

Recall that the full domain hash (FDH) signature scheme [BR93] is defined as follows:

Definition 5 (Full Domain Hash). *Let $\mathcal{F} = (G, f, f^{-1})$ be a trapdoor permutation, and O a hash function whose range is the same as the range of f. The full domain hash signature scheme is $\mathcal{S} = (G, T, V)$ where:*

- $G = G_0$
- $S^O(\mathsf{sk}, m) = f^{-1}(\mathsf{sk}, O(m))$
- $V^O(\mathsf{pk}, m, \sigma) = \begin{cases} 1 & \text{if } O(m) = f(\mathsf{pk}, \sigma) \\ 0 & \text{otherwise} \end{cases}$

Gentry et al. [GPV08] show that the FDH signature scheme can be instantiated with a PSF $\mathcal{F} = (G, \text{Sample}, f, f^{-1})$ instead of a trapdoor permutation. Call the resulting system FDH-PSF. They prove that FDH-PSF is secure against classical adversaries, provided that the pre-image sampling algorithm used during signing is derandomized (e.g. by using a classical PRF to generate its random bits). Their reduction is not quite history-free, but we show that it can be made history-free.

Consider the following reduction from a classical adversary A for the FDH-PSF scheme $\mathcal{S}$ to a classical collision finder B for $\mathcal{F}$:

- On input pk, B computes $\text{START}(\mathsf{pk}) := (\mathsf{pk}, \mathsf{pk})$, and simulates A on pk.
- When A queries $O(r)$, B responds with
$$\text{RAND}^{O_c}(r, \mathsf{pk}) := f(\mathsf{pk}, \text{Sample}(1^n; O_c(r))).$$
- When A queries $S(\mathsf{sk}, m)$, B responds with
$$\text{SIGN}^{O_c}(m, \mathsf{pk}) := \mathit{Sample}(1^n; O_c(m)).$$
- When A outputs (m, σ), B outputs
$$\text{FINISH}^{O_c}(m, \sigma, \mathsf{pk}) := \big(\mathit{Sample}\big(1^n; O_c(m)\big), \sigma\big).$$

In addition, we define $\text{INSTANCE}(\mathsf{pk}) := \mathsf{pk}$. Algorithms INSTANCE and START trivially satisfy the requirements of history-freeness (Definition 4). Before showing that the above reduction is in history-free form, we need the following technical lemma whose proof is given in the appendix .

Lemma 3. *Say A is a quantum algorithm that makes q quantum oracle queries. Suppose further that we draw the oracle O from two distributions. The first is the random oracle distribution. The second is the distribution of oracles where the value of the oracle at each input x is identically and independently distributed by some distribution D whose variational distance is within ϵ from uniform. Then the variational distance between the distributions of outputs of A with each oracle is at most $4q^2\sqrt{\epsilon}$.*

Proof Sketch. We show that there is a way of moving from O to O_D such that the oracle is only changed on inputs in a set K where the sum of the amplitudes squared of all $k \in K$, over all queries made by A, is small. Thus, we can use Lemma 2 to show that the expected behavior of any algorithm making polynomially many quantum queries to O is only changed by a small amount. □

Lemma 3 shows that we can replace a truly random oracle O with an oracle O_D distributed according to distribution D without impacting A, provided D is close to uniform. Note, however, that while this change only affects the output of A negligibly, the effects are larger than in the classical setting. If A only made classical queries to O, a simple hybrid argument shows that changing to O_D affects the distribution of the output of A by at most $q\epsilon$, as opposed to $4q^2\sqrt{\epsilon}$ in the quantum case. Thus, quantum security reductions that use Lemma 3 will not be as tight as their classical counterparts.

We now show that the reduction above is history-free.

Theorem 2. *The reduction above applied to FDH-PSF is history-free.*

Proof. The definition of a PSF implies that the distribution of $f(\mathsf{pk}, \text{Sample}(1^n))$ is within ϵ_{sample} of uniform, for some negligible ϵ_{sample}. Now, since $O(r) = \text{RAND}^{O_c}(r, \mathsf{pk}) = f(\mathsf{pk}, \text{Sample}(1^n; O_c(r)))$ and O_c is a true random oracle, the quantity $O(r)$ is distributed independently according to a distribution that is ϵ_{sample} away from uniform. Define a quantum oracle O_{quant} which transforms the basis state $|x, y\rangle$ into $|x, y \oplus O(x)\rangle$. Using Lemma 3, for any algorithm B making q random oracle queries, the variational distance between the probability distributions of the outputs of B using a truly random oracle and the "not-quite" random oracle O_{quant} is at most $4q^2\sqrt{\epsilon_{\text{sample}}}$, which is still negligible. Hence, O_q is computationally indistinguishable from random.

Gentry et al. [GPV08] also show that $\text{SIGN}^{O_c}(m, \mathsf{pk})$ is consistent with $\text{RAND}^{O_c}(\cdot, \mathsf{pk})$ for all queries, and that if A outputs a valid forgery (m, σ), $\text{FINISH}^{O_c}(m, \sigma, \mathsf{pk})$ produces a collision for $\mathcal{F}$ with probability $1 - 2^{-E}$, where E is the minimum over all y in the range of $f(\mathsf{pk}, \cdot)$ of the min-entropy of the distribution on σ given $f(\mathsf{pk}, \sigma) = y$. The PSF of Gentry et al. [GPV08] has super-logarithmic min-entropy, so $1 - 2^{-E}$ is negligibly close to 1, though any constant non-zero min-entropy will suffice to make the quantity a non-negligible fraction of 1. □

We note that the security proof of Gentry et al. [GPV08] is a tight reduction in the following sense: if the advantage of an adversary A for $\mathcal{S}$ is ϵ, the reduction

gives a collision finding adversary B for $\mathcal{F}$ with advantage negligibly close to ϵ, provided that the lower bound over y in the range of $f(\mathsf{pk}, \cdot)$ of the min-entropy of σ given $f(\mathsf{pk}, \sigma) = y$ is super-logarithmic. If the PSF has a min-entropy of 1, the advantage of B is still $\epsilon/2$.
The following corollary, which is the main result of this section, follows from Theorems (1) and (2).

Corollary 1. *If quantum-accessible pseudorandom functions exist, and $\mathcal{F}$ is a secure PSF against quantum adversaries, then the FDH-PSF signature scheme is secure in the quantum-accessible random oracle model.*

4.2 Secure Signatures from Claw-Free Permutations

In this section, we show how to use claw-free permutations to construct three signature schemes that have history-free reductions and are therefore secure in the quantum-accessible random oracle model. The first is the standard FDH from Definition 5, but when the underlying permutation is a claw-free permutation. We adapt the proof of Coron [Cor00] to give a history-free reduction. The second is the Katz and Wang [KW03] signature scheme, and we also modify their proof to get a history-free reduction. Lastly, following Gentry et al. [GPV08], we note that claw-free permutations give rise to a pre-image sampleable trapdoor function (PSF), which can then be used in FDH to get a secure signature scheme as in Section 4.1. The Katz-Wang and FDH-PSF schemes from claw-free permutations give a tight reduction, whereas the Coron-based proof loses a factor of q_s in the security reduction, where q_s is the number of signing queries.

Recall that a claw-free pair of permutations [GMR88] is a pair of trapdoor permutations $(\mathcal{F}_1, \mathcal{F}_2)$, where $\mathcal{F}_i = (G_i, f_i, f_i^{-1})$, with the following properties:

- $G_1 = G_2$. Define $G = G_1 = G_2$.
- For any key pk, $f_1(\mathsf{pk}, \cdot)$ and $f_2(\mathsf{pk}, \cdot)$ have the same domain and range.
- Given only pk, the probability that any PPT adversary can find a pair (x_1, x_2) such that $f_1(\mathsf{pk}, x_1) = f_2(\mathsf{pk}, x_2)$ is negligible. Such a pair is called a claw.

Dodis and Reyzin [DR03] note that claw-free permutations are a generalization of trapdoor permutations with a random self-reduction. A random self-reduction is a way of taking a worst-case instance x of a problem, and converting it into a random instance y of the same problem, such that a solution to y gives a solution to x. Dodis and Reyzin [DR03] show that any trapdoor permutation with a random self reduction (e.g. RSA) gives a claw-free pair of permutations.

We note that currently there are no candidate pairs of claw-free permutations that are secure against quantum adversaries, but this may change in time.

FDH Signatures from Claw-Free Permutations Coron [Cor00] shows that the Full Domain Hash signature scheme, when instantiated with the RSA trapdoor permutation, has a tighter security reduction than the general Full Domain

Hash scheme, in the classical world. That is, Coron's reduction loses a factor of approximately q_s, the number of signing queries, as apposed to q_h, the number of hash queries. Of course, the RSA trapdoor permutation is not secure against quantum adversaries, but his reduction can be applied to any claw-free permutation and is equivalent to a history-free reduction with similar tightness.

To construct a FDH signature scheme from a pair of claw-free permutations $(\mathcal{F}_1, \mathcal{F}_2)$, we simply instantiate FDH with $\mathcal{F}_1$, and ignore the second permutation $\mathcal{F}_2$, to yield the following signature scheme

- G is the generator for the pair of claw-free permutations.
- $S^O(\mathsf{sk}, m) = f_1^{-1}(\mathsf{sk}, O(m))$
- $V^O(\mathsf{pk}, m, \sigma) = 1$ if and only if $f_1(\mathsf{pk}, \sigma) = O(m)$.

We now present a history-free reduction for this scheme. The random oracle for this reduction, $O_c(r)$, returns a random pair (a, b), where a is a random element from the domain of $\mathcal{F}_1$ and $\mathcal{F}_2$, and b is a random element from $\{1, ..., p\}$ for some p to be chosen later.

We construct history-free reduction from a classical adversary A for $\mathcal{S}$ to a classical adversary B for $(\mathcal{F}_1, \mathcal{F}_2)$. Algorithm B, on input pk, works as follows:

- Compute $\text{START}(\mathsf{pk}, y) = (\mathsf{pk}, \mathsf{pk})$, and simulate A on pk. Notice that $z = \mathsf{pk}$ is the state saved by B.
- When A queries $O(r)$, compute $\text{RAND}^{O_c}(r, \mathsf{pk})$. For each string r, RAND works as follows: compute $(a, b) \leftarrow O_c(r)$. If $b = 1$, return $f_2(\mathsf{pk}, a)$. Otherwise, return $f_1(\mathsf{pk}, a)$
- When A queries $S(\mathsf{sk}, m)$, compute $\text{SIGN}^{O_c}(m, \mathsf{pk})$. SIGN works as follows: compute $(a, b) \leftarrow O_c(m)$ and return a if $b \neq 1$. Otherwise, fail.
- When A returns (m, σ), compute $\text{FINISH}^{O_c}(m, \sigma, \mathsf{pk})$. FINISH works as follows: compute $(a, b) \leftarrow O_c(m)$ and output (σ, a).

In addition, we have $\text{INSTANCE}(\mathsf{pk}) = \mathsf{pk}$ and $\text{START}(\text{INSTANCE}(\mathsf{pk})) = (\mathsf{pk}, \mathsf{pk})$, so INSTANCE and START satisfy the required properties.

Theorem 3. *The reduction above is in history-free form.*

Proof. $\text{RAND}^{O_c}(r, \mathsf{pk})$ is completely random and independently distributed, as $f_1(\mathsf{pk}, a)$ and $f_2(\mathsf{pk}, a)$ are both random ($f_b(\mathsf{pk}, \cdot)$ is a permutation and a is truly random). As long as $b \neq 1$, where $(a, b) = O_c(m)$, $\text{SIGN}^{O_c}(m, \mathsf{pk})$ will be consistent with RAND. This is because because $V^{\text{RAND}^{O_c}(\cdot, \mathsf{pk})}(\mathsf{pk}, m, \text{SIGN}^{O_c}(m, \mathsf{pk}))$ outputs 1 if $\text{RAND}^{O_c}(m, \mathsf{pk}) = f_1(\mathsf{pk}, \text{SIGN}^{O_c}(m, \mathsf{pk}))$. But $\text{RAND}^{O_c}(m, \mathsf{pk}) = f_1(\mathsf{pk}, a)$ (since $b \neq 1$), and $\text{SIGN}^{O_c}(m, \mathsf{pk})) = a$. Thus, the equality holds. The probability over all signature queries of no failure is $(1 - 1/p)^{q_{\text{SIGN}}}$. If we chose $p = q_{\text{SIGN}}$, this quantity is at least $e^{-1} - o(1)$, which is non-negligible.

Suppose A returns a valid forgery (m, σ), meaning A never asked for a forgery on m and $f_1(\mathsf{sk}, \sigma) = \text{RAND}^{O_c}(m, \mathsf{pk})$. If $b = 1$ (where $(a, b) = O_c(m)$), then we have $f_1(\mathsf{sk}, \sigma) = \text{RAND}^{O_c}(m, \mathsf{pk}) = f_2(\mathsf{pk}, a)$, meaning that (σ, a) is a claw. Since A never asked for a signature on m, there is no way A could have figured out a, so the case where $b = 1$ and a is the preimage of $O(m)$ under f_2, and the case where $b \neq 1$ and a is the preimage of $O(m)$ under f_1 are indistinguishable. Thus, $b = 1$ with probability $1/p$. Thus, B converts a valid signature into a claw with non-negligible probability. □

Corollary 2. *If quantum-accessible pseudorandom functions exists, and $(\mathcal{F}_1, \mathcal{F}_2)$ is a pair claw-free trapdoor permutations, then the FDH scheme instantiated with $\mathcal{F}_1$ is secure against quantum adversaries.*

Note that in this reduction, our simulated random oracle is truly random, so we do not need to rely on Lemma 3. Hence, the tightness of the reduction will be the same as the classical setting. Namely, if the quantum adversary A has advantage ϵ when making q_{SIGN} signature queries, B will have advantage approximately $\epsilon/q_{\mathrm{SIGN}}$.

The Katz-Wang Signature Scheme In this section, we consider a variant of FDH due to Katz and Wang [KW03]. This scheme admits an almost tight security reduction in the classical world. That is, if an adversary has advantage ϵ, the reduction gives a claw finder with advantage $\epsilon/2$. Their proof of security is not in history-free form, but it can be modified so that it is in history-free form. Given a pair of trapdoor permutation $(\mathcal{F}_1, \mathcal{F}_2)$, the construction is as follows:

- G is the key generator for $\mathcal{F}$.
- $S^O(\mathsf{sk}, m) = f_1^{-1}(\mathsf{sk}, O(b, m))$ for a random bit b.
- $V^O(\mathsf{pk}, m, \sigma)$ is 1 if either $f_1(\mathsf{pk}, \sigma) = O(0, m)$ or $f_1(\mathsf{pk}, \sigma) = O(1, m)$

We construct a history-free reduction from an adversary A for $\mathcal{S}$ to an adversary B for $(\mathcal{F}_1, \mathcal{F}_2)$. The random oracle for this reduction, $O_c(r)$, generates a random pair (a, b), where a is a random element from the domain of $\mathcal{F}_1$ and $\mathcal{F}_2$, and b is a random bit. On input pk, B works as follows:

- Compute $\mathrm{START}(\mathsf{pk}, y) = (\mathsf{pk}, \mathsf{pk})$, and simulate A on pk. Notice that $z = \mathsf{pk}$ is the state saved by B.
- When A queries $O(b, r)$, compute $\mathrm{RAND}^{O_c}(b, r, \mathsf{pk})$. For each string (b, r), RAND works as follows: compute $(a, b') = O_c(r)$. If $b = b'$, return $f_1(\mathsf{pk}, a)$. Otherwise, return $f_2(\mathsf{pk}, a)$.
- When A queries $S(\mathsf{sk}, m)$, compute $\mathrm{SIGN}^{O_c}(m, \mathsf{pk})$. SIGN works as follows: compute $(a, b) = O_c(m)$ and return a.
- When A returns (m, σ), compute $\mathrm{FINISH}^{O_c}(m, \sigma, \mathsf{pk})$. FINISH works as follows: compute $(a, b) = O_c(m)$. If $\sigma = a$, abort. Otherwise, output (σ, a).

In addition, we have $\mathrm{INSTANCE}(\mathsf{pk}) = \mathsf{pk}$ and $\mathrm{START}(\mathrm{INSTANCE}(\mathsf{pk})) = (\mathsf{pk}, \mathsf{pk})$, so INSTANCE and START satisfy the required properties.

Theorem 4. *The reduction above is in history-free form.*

Proof. $\mathrm{RAND}^{O_c}(b, r, \mathsf{pk})$ is completely random and independently distributed, as $f_1(\mathsf{pk}, a)$ and $f_2(\mathsf{pk}, a)$ are both random (f_b is a permutation and a is truly random). Observe that $f_1(\mathsf{pk}, \mathrm{SIGN}^{O_c}(m, \mathsf{pk})) = f_1(\mathsf{pk}, a) = O(b, m)$ where $(a, b) = O_c(m)$. Thus, signing queries are always answered with a valid signature, and the distribution of signatures is identical to that of the correct signing algorithm since b is chosen uniformly.

Suppose A returns a valid forgery (m, σ). Let $(a, b) = O_c(m)$. There are two cases, corresponding to whether σ corresponds to a signature using b or $1 - b$. In the first case, we have $f_1(\mathsf{pk}, \sigma) = O(b, m) = f_1(\mathsf{pk}, a)$, meaning $\sigma = a$, so we abort. Otherwise, $f_1(\mathsf{pk}, \sigma) = O(1 - b, m) = f_2(\mathsf{pk}, a)$, so (σ, a) form a claw. Since the adversary never asked for a signing query on m, these two cases are indistinguishable by the same logic as the proof for FDH. Thus, the probability of failure is at most a half, which is non-negligible. □

Corollary 3. *If quantum-accessible pseudorandom functions exists, and $(\mathcal{F}_1, \mathcal{F}_2)$ is a pair claw-free trapdoor permutations, then the Katz-Wang signature scheme instantiated with $\mathcal{F}_1$ is secure against quantum adversaries.*

As in the case of FDH, our simulated quantum-accessible random oracle is truly random, so we do not need to rely on Lemma 3. Thus, the tightness of our reduction is the same as the classical case. In particular, if the quantum adversary A_Q has advantage ϵ then B will have advantage $\epsilon/2$.

PSF Signatures from Claw-Free Permutations Gentry et al. [GPV08] note that Claw-Free Permutations give rise to pre-image sampleable trapdoor functions (PSFs). These PSFs can then be used to construct an FDH signature scheme as in Section 4.1.

Given a pair of claw-free permutations $(\mathcal{F}_1, \mathcal{F}_2)$, define the following PSF: G is just the generator for the pair of permutations. Sample(pk) generates a random bit b and random x in the domain of f_b, and returns (x, b). $f(\mathsf{pk}, x, b) = f_b(\mathsf{pk}, x)$, and $f^{-1}(\mathsf{sk}, y) = (f_b^{-1}(\mathsf{sk}, y), b)$ for a random b. Suppose we have a collision $((x_1, b_1), (x_2, b_2))$ for this PSF. Then

$$f_{b_1}(\mathsf{pk}, x_1) = f(\mathsf{pk}, x_1, b_1) = f(\mathsf{pk}, x_2, b_2) = f_{b_2}(\mathsf{pk}, x_2)$$

If $b_1 = b_2$, then $x_1 = x_2$ since f_{b_1} is a permutation. But this is impossible since $(x_1, b_1) \neq (x_2, b_2)$. Thus, $b_1 \neq b_2$, so one of (x_1, x_2) or (x_2, x_1) is a claw for $(\mathcal{F}_1, \mathcal{F}_2)$.

Hence, we can instantiate FDH with this PSF to get the following signature scheme:

- G is the generator for the permutations.
- $S^O(\mathsf{sk}, m) = (f_b^{-1}(\mathsf{sk}, O(m)), b)$ for a random bit b.
- $V^O(\mathsf{pk}, m, (\sigma, b)) = 1$ if and only if $f_b(\mathsf{pk}, \sigma) = O(m)$.

The security of this scheme follows from Corollary 1, with a similar tightness guarantee (this PSF has only a pre-image min-entropy of 1, which results in a loss of a factor of two in the tightness of the reduction). In particular, if we have a quantum adversary A_Q for $\mathcal{E}$ with advantage ϵ, we get a quantum algorithm B_Q for the PSF with advantage $\epsilon/2$, which gives us a quantum algorithm C_Q that finds claws of $(\mathcal{F}_1, \mathcal{F}_2)$ with probability $\epsilon/2$.

5 Encryption Schemes in the Quantum-Accessible Random Oracle Model

In this section, we prove the security of two encryption schemes. The first is the BR encryption scheme due to Bellare and Rogaway [BR93], which we show is CPA secure. The second is a hybrid generalization of the BR scheme, which we show is CCA secure.

Ideally, we could define a general type of classical reduction like we did for signatures, and show that such a reduction implies quantum security. Unfortunately, defining a history-free reduction for encryption is considerably more complicated than for signatures. We therefore directly prove the security of two random oracle schemes in the quantum setting.

5.1 CPA Security of BR Encryption

In this section, we prove the security of the BR encryption scheme [BR93] against quantum adversaries:

Definition 6 (BR Encryption Scheme). *Let $\mathcal{F} = (G_0, f, f^{-1})$ be an injective trapdoor function, and O a hash function with the same domain as $f(\mathsf{pk}, \cdot)$. We define the following encryption scheme, $\mathcal{E} = (G, E, D)$ where:*

- $G = G_0$
- $E^O(\mathsf{pk}, m) = (f(\mathsf{pk}, r), O(r) \oplus m)$ *for a randomly chosen r.*
- $D^O(\mathsf{sk}, (y, c)) = c \oplus f^{-1}(\mathsf{sk}, y)$

A candidate quantum-immune injective trapdoor function can be built from hard problems on lattices [PW08].

Theorem 5. *If quantum-accessible pseudorandom functions exists and $\mathcal{F}$ is a quantum-immune injective trapdoor function, then $\mathcal{E}$ is quantum CPA secure.*

We omit the proof of Theorem 5 because the CPA security of the BR encryption scheme is a special case of the CCA security of the hybrid encryption scheme in the next section.

5.2 CCA Security of Hybrid Encryption

We now prove the CCA security of the following standard hybrid encryption, a generalization of the BR encryption scheme scheme [BR93], built from an injective trapdoor function and symmetric key encryption scheme.

Definition 7 (Hybrid Encryption Scheme). *Let $\mathcal{F} = (G_0, f, f^{-1})$ be an injective trapdoor function, and $\mathcal{E_S} = (E_S, D_S)$ be a CCA secure symmetric key encryption scheme, and O a hash function. We define the following encryption scheme, $\mathcal{E} = (G, E, D)$ where:*

- $G = G_0$
- $E^O(\mathsf{pk}, m) = (f(\mathsf{pk}, r), E_S(O(r), m))$ *for a randomly chosen* r.
- $D^O(\mathsf{sk}, (y, c)) = D_S(O(r'), c)$ *where* $r' = f^{-1}(\mathsf{sk}, y)$

We note that the BR encryption scheme from the previous section is a special case of this hybrid encryption scheme where $\mathcal{E_S}$ is the one-time pad. That is, $E_S(k, m) = k \oplus m$ and $D_S(k, c) = k \oplus c$.

Theorem 6. *If quantum-accessible pseudorandom functions exists, $\mathcal{F}$ is a quantum-immune injective trapdoor function, and $\mathcal{E}_S$ is a quantum CCA secure symmetric key encryption scheme, then $\mathcal{E}$ is quantum CCA secure.*

Proof. Suppose we have an adversary A_Q that breaks $\mathcal{E}$. We start with the standard security game for CCA secure encryption:

Game 0. Define Game$_0$ as the game a quantum adversary A_Q plays for problem Asym-CCA($\mathcal{E}$).

Game 1. Define Game$_1$ as the following game: the challenger generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow G(1^n)$, a random r in the domain of $\mathcal{F}$, a random k in the key space of $\mathcal{E_S}$, and computes $y = f(\mathsf{pk}, r)$. The challenger has access to a quantum-accessible random oracle O_q whose range is the key space of $\mathcal{E_S}$. It then sends pk to A_Q. The challenger answers queries as follows:

- Random oracle queries are answered with the random oracle O_{quant}, which takes a basis element $|x, y\rangle$ into $|x, y \oplus O_q(f(\mathsf{pk}, x))\rangle$.
- Decryption queries on (y', c') are answered as follows:

  Case 1: If $y = y'$, respond with $D_S(k, c')$.

  Case 2: If $y \neq y'$, respond with $D_S(O_q(y'), c')$.
- The challenge query on (m_0, m_1) is answered as follows: choose a random b. Then, respond with $(y, E_S(k, m_b))$.

When A_Q responds with b', we say that A_Q won if $b = b'$.

Observe that, because f is injective and O_q is random, the oracle O_{quant} is a truly random oracle with the same range as O_q. The challenge ciphertext (y, c) seen by A_Q is distributed identically to that of Game$_0$. Further, it is a valid encryption of m_b relative to the random oracle being O_{quant} if $O_q(y) = k$. For $y' \neq y$, the decryption of (y', c') is

$$D_S(O_q(y'), c') = D_S(O_{\text{quant}}(f^{-1}(\mathsf{sk}, y')), c') = D^{O_{\text{quant}}}(\mathsf{sk}, (y', c'))$$

Which is correct. Likewise, if $O_q(y) = k$, the decryption of (y, c') is also correct. Thus, the view of A_Q in Game$_1$ is identical to that in Game$_0$ if $O_q(y) = k$. We now make the following observations:

- The challenge query and decryption query answering algorithms never query O_q on y.
- Each quantum random oracle query from the adversary to O_{quant} leads to a quantum random oracle query from the challenger to O_q. The query magnitude of y in the challenger's query to O_q is the same as the query magnitude of r in the adversary's query O_{quant}.

Let ϵ be the sum of the square magnitudes of y over all queries made to O_q (i.e. the total query probability of y). This is identical to the total query probability of r over all queries A_Q makes to O_{quant}.

We now construct a quantum algorithm $B_{\mathcal{F}}^{O_q}$ that uses a quantum-accessible random oracle O_q, and inverts f with probability ϵ/q, where q is the number of random oracle queries made by A_Q. $B_{\mathcal{F}}^{O_q}$ takes as input (pk, y), and its goal is to output $r = f^{-1}(\mathsf{sk}, y)$. $B_{\mathcal{F}}^{O_q}$ works as follows:

- Generate a random k in the key space of $\mathcal{E}_{\mathcal{S}}$. Also, generate a random $i \in \{1, ..., q\}$. Now, send pk to A_Q and play the role of challenger to A_Q.
- Answer random oracle queries with the random oracle O_{quant}, which takes a basis element $|x, y\rangle$ into $|x, y \oplus O_q(f(\mathsf{pk}, x))\rangle$.
- Answer decryption queries on (y', c') as follows:

  Case 1: If $y = y'$, respond with $D_S(k, c')$.

  Case 2: If $y \neq y'$, respond with $D_S(O_q(y'), c')$.
- Answer the challenge query on (m_0, m_1) as follows: choose a random b. Then, respond with $(y, E_S(k, m_b))$.
- At the ith random oracle query, sample the query to get r', and output r' and terminate.

Comparing our definition of $B_{\mathcal{F}}^{O_q}$ to Game$_1$, we can conclude that the view seen by A_Q in both cases is identical. Thus, the total query probability that A_Q makes to O_{quant} at the point r is ϵ. Hence, the probability that $B_{\mathcal{F}}^{O_q}$ outputs r is ϵ/q. If we assume that $\mathcal{F}$ is secure against quantum adversaries that use a quantum-accessible random oracle, then this quantity, and hence ϵ, must be negligible. As in the case of signatures (Section 4), we can replace this assumption with the assumption that $\mathcal{F}$ is secure against quantum adversaries (i.e. with no access to a quantum random oracle) and that pseudorandom functions exists to reach the same conclusion.

Since ϵ is negligible, we can change $O_q(y) = k$ in Game$_1$, thus getting a game identical to Game$_0$ from the adversary's point of view. Notice that in Game$_0$ and Game$_1$, A_Q is in a pure state because we are only applying unitary transformations, performing measurements, or performing classical communication. We are only changing the oracle at a point with negligible total query probability, so Lemma 2 tells us that making this change only affects the distribution of the outcome of Game$_1$ negligibly. This allows us to conclude that the success probability of A_Q in Game$_1$ is negligibly close to that in Game$_0$.

Now, assume that the success probability of A_Q in Game$_1$ is non-negligible. We now define a quantum algorithm $B_{\mathcal{E_S}}^{O_q}$ that uses a quantum-accessible random oracle O_q to break the CCA security of $\mathcal{E_S}$. $B_{\mathcal{E_S}}^{O_q}$ works as follows:

- On input 1^n, generate $(\mathsf{sk}, \mathsf{pk}) \leftarrow G(1^n)$. Also, generate a random r, and compute $y = f(\mathsf{pk}, r)$. Now send pk to A_Q and play the role of challenger to A_Q.
- Answer random oracle queries with the random oracle O_{quant}, which takes a basis element $|x, y\rangle$ into $|x, y \oplus O_q(f(\mathsf{pk}, x))\rangle$.
- Answer decryption queries on (y', c') as follows:

  Case 1: If $y = y'$, ask the $\mathcal{E_S}$ challenger for a decryption $D_S(k, c')$ to obtain m'. Return m' to A_Q.

  Case 2: If $y \neq y'$, respond with $D_S(O_q(y'), c')$.
- Answer the challenge query on (m_0, m_1) by forwarding the pair $\mathcal{E_S}$. When the challenger responds with c (which equals $E_S(k, m_b)$ for some b), return (y, c) to A_Q.
- When A_Q outputs b', output b' and halt.

Comparing our definition of $B_{\mathcal{E_S}}^{O_q}$ to that of Game$_1$, we can conclude that the view of A_Q in both cases is identical. Thus, A_Q succeeds with non-negligible probability. If A_Q succeeds, it means it returned b, meaning $B_{\mathcal{E_S}}^{O_q}$ also succeeded. Thus, we have an algorithm with a quantum random oracle that breaks $\mathcal{E_S}$. This is a contradiction if $\mathcal{E_S}$ is CCA secure against quantum adversaries with access to a quantum random oracle, which holds since $\mathcal{E_S}$ is CCA secure against quantum adversaries and quantum-accessible pseudorandom functions exist, by assumption.

Thus, the success probability of A_Q in Game$_1$ is negligible, so the success probability of A_Q in Game$_0$ is also negligible. Hence, we have shown that all polynomial time quantum adversaries have negligible advantage in breaking in breaking the CCA security of $\mathcal{E}$, so $\mathcal{E}$ is CCA secure. □

We briefly explain why Theorem 5 is a special case of Theorem 6. Notice that, in the above proof, $B_{\mathcal{E_S}}$ only queries its decryption oracle when answering decryption queries made by A_Q, and that it never makes encryption queries. Hence, if A_Q makes no decryption queries, $B_{\mathcal{E_S}}$ makes no queries at all except the challenge query. If we are only concerned with the CPA security of $\mathcal{E}$, we then only need E_S to be secure against adversaries that can only make the challenge query. Further, if we only let A_Q make a challenge query with messages of length n, then E_S only has to be secure against adversaries making challenges of a specific length. But this is exactly the model in which the one-time pad is unconditionally secure. Hence, the BR encryption scheme is secure, and we have proved Theorem 5.

6 Conclusion

We have shown that great care must be taken if using the random oracle model when arguing security against quantum attackers. Proofs in the classical case should be reconsidered, especially in case the quantum adversary can access the random oracle with quantum states. We also developed conditions for translating security proofs in the classical random oracle model to the quantum random oracle model. We applied these tools to certain signature and encryption schemes.

The foremost question raised by our results is in how far techniques for "classical random oracles" can be applied in the quantum case. This stems from the fact that manipulating or even observing the interaction with the quantum-accessible random oracle would require measurements of the quantum states. That, however, prevents further processing of the query in a quantum manner. We gave several examples of schemes that remain secure in the quantum setting, provided quantum-accessible pseudorandom functions exist. The latter primitive seems to be fundamental to simulate random oracles in the quantum world. Showing or disproving the existence of such pseudorandom functions is thus an important step.

Many classical random oracle results remain open in the quantum random oracle settings. It is not known how to prove security of generic FDH signatures as well as signatures derived from the Fiat-Shamir heuristic in the quantum random oracle model. Similarly, a secure generic transformation from CPA to CCA security in the quantum RO model is still open.

Acknowledgments

We thank Chris Peikert for helpful comments. Dan Boneh was supported by NSF, the Air Force Office of Scientific Research (AFO SR) under a MURI award, and by the Packard Foundation. Marc Fischlin and Anja Lehmann were supported by grants Fi 940/2-1 and Fi 940/3-1 of the German Research Foundation (DFG). Özgür Dagdelen and Marc Fischlin were also supported by CASED (`www.cased.de`). Christian Schaffner is supported by a NWO VENI grant.

References

Aar09. Scott Aaronson. Quantum copy-protection and quantum money. In *Structure in Complexity Theory Conference*, pages 229–242, 2009.

ABB10a. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of Eurocrypt'10*, volume 6110 of *LNCS*, pages 553–572, 2010.

ABB10b. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter ciphertext hierarchical IBE. In *Proc. of Crypto'10*, volume 6223 of *LNCS*, pages 98–115, 2010.

AS04. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

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

BBC$^+$98. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 1998*, pages 352–361. IEEE Computer Society Press, 1998.

Ber09. Daniel J. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? In *SHARCS'09: Special-purpose Hardware for Attacking Cryptographic Systems*, 2009.

BF11. Dan Boneh and David Freeman. Homomorphic signatures for polynomial functions. In *Advances in Cryptology — EUROCRYPT 2011*, volume 6632, pages 149–168, 2011.

BHK$^+$11. Gilles Brassard, Peter Hoyer, Kassem Kalach, Marc Kaplan, Sophie Laplante, and Louis Salvail. Merkle puzzles in a quantum world. In *Advances in Cryptology — Crypto 2011*, Lecture Notes in Computer Science. Springer-Verlag, 2011.

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

BJ99. Nader H. Bshouty and Jeffrey C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1999.

Boy10. Xavier Boyen. Lattice mixing and vanishing trapdoors : A framework for fully secure short signatures and more. In *Proc. of PKC 2010*, LNCS, 2010.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of ACM Conference on Computers and Communication Security*, pages 62–73, 1993.

BS08. Gilles Brassard and Louis Salvail. Quantum Merkle Puzzles. *Second International Conference on Quantum, Nano and Micro Technologies (ICQNM 2008)*, pages 76–79, February 2008.

Can01. Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2001*. IEEE Computer Society Press, for an updated version see `eprint.iacr.org`, 2001.

CGH98. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1998*, pages 209–218. ACM Press, 1998.

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of Eurocrypt'10*, pages 523–552, 2010.

CLRS10. Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. Improved zero-knowledge identification with lattices. *Provable Security*, pages 1–17, 2010.

Cor00. Jean-Sébastien Coron. On the exact security of full domain hash. In *Advances in Cryptology — CRYPTO 2000*, pages 229–235. Springer, 2000.

DR03. Yevgeniy Dodis and Leonid Reyzin. On the power of claw-free permutations. *Security in Communication Networks*, pages 55–73, 2003.

FLR$^+$10. Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 303–320. Springer, 2010.

FO99. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology – Crypto '99*, volume 1666 of *LNCS*, pages 537–554, 1999.

FOPS01. Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. RSA—OAEP is secure under the RSA assumption. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO ' 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 260–274. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 2001.

GGM86. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.

GKV10. Samuel D. Gordon, Jonathan Katz, and Vaikuntanathan Vaikuntanathan. A Group Signature Scheme from Lattice Assumptions. *Advances in Cryptology — ASIACRYPT 2010*, pages 395–412, 2010.

GMR88. Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing*, 17(2):281, 1988.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *Proceedings of the fourtieth annual ACM symposium on Theory of computing - STOC '08*, page 197, 2008.

Gro96. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1996*, pages 212–219. ACM, 1996.

Gro98. Lov K. Grover. Quantum search on structured problems. In *Quantum Computing and Quantum Communications (QCQC) 1998*, volume 1509 of *Lecture Notes in Computer Science*, pages 126–139. Springer-Verlag, 1998.

HSS11. Sean Hallgren, Adam Smith, and Fang Song. Classical cryptographic protocols in a quantum world. In *Advances in Cryptology — Crypto 2011*, Lecture Notes in Computer Science. Springer-Verlag, 2011.

KW03. Jonathan Katz and Nan Wang. Efficiency improvements for signature schemes with tight security reductions. *Proceedings of the 10th ACM conference on Computer and communication security - CCS '03*, page 155, 2003.

NC00. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

NIS07. NIST. National institute of standards and technology: Sha-3 competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/`, 2007.

PS00. David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

PW08. Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the 14th annual ACM symposium on Theory of computing - STOC '08*, page 187, 2008.

Reg02. Oded Regev. Quantum computation and lattice problems. In *FOCS*, pages 520–529, 2002.

SG04. Rocco A. Servedio and Steven J. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004.

Sho97. Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

Unr10. Dominique Unruh. Universally composable quantum multi-party computation. In *Advances in Cryptology — Eurocrypt*, volume 6110 of *Lecture Notes in Computer Science*, pages 486–505. Springer-Verlag, 2010.

Wat09. John Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

A Definitions

Definition 8 (Trapdoor Permutation). *A trapdoor permutation is a triple of functions* $\mathcal{F} = (G, f, f^{-1})$ *where:*

- $G(1^n)$ *generates a private/public key pair* $(\mathsf{sk}, \mathsf{pk})$.
- $f(\mathsf{pk}, \cdot)$ *is a permutation for all* pk.
- $f^{-1}(\mathsf{sk}, \cdot)$ *is the inverse of* $f(\mathsf{pk}, \cdot)$ *for all* $(\mathsf{pk}, \mathsf{sk})$ *generated by* G*. That is,* $f^{-1}(\mathsf{sk}, f(\mathsf{pk}, x)) = x$ *and* $f(\mathsf{pk}, f^{-1}(\mathsf{sk}, y)) = y$.

For a trapdoor permutation $\mathcal{F}$, we define the problem $Inv(\mathcal{F}) = (Game(\mathcal{F}), 0)$ where $Game(\mathcal{F})$ is the following game between a quantum adversary A and the challenger Ch: Ch, on input n, runs $G(1^n)$ to obtain $(\mathsf{sk}, \mathsf{pk})$ and generates a random y in the range of $f(\mathsf{pk}, \cdot)$. It sends (pk, y) to A. A is allowed to make quantum random oracle queries $O(\cdot)$. When A outputs x, Ch outputs 1 if and only if $f(\mathsf{pk}, x) = y$.

Definition 9. *A trapdoor permutation* $\mathcal{F}$ *is secure against quantum adversaries if* $Inv(\mathcal{F})$ *is hard for quantum computers*

The following definition is due to [GPV08]:

Definition 10 (Preimage Sampleable Trapdoor Function). *A quadruple of functions* $\mathcal{F} = (G, Sample, f, f^{-1})$ *is a trapdoor collision-resistant hash function with preimage sampling (PSF) if:*

- $G(1^n)$ *generates secret and public keys* $(\mathsf{sk}, \mathsf{pk})$.
- $f(\mathsf{pk}, \cdot)$ *has domain* D *and range* R.
- $Sample(1^n)$ *samples from a distribution on* D *such that for all* pk *the distribution* $f\big(\mathsf{pk},\ Sample(1^n)\big)$ *is within* ϵ_{sample} *of uniform.*
- $f^{-1}(\mathsf{sk}, y)$ *generates an* x *such that* $f(\mathsf{pk}, x) = y$*. The distribution is within* ϵ_{pre} *of the conditional distribution of* $Sample()$ *given* $f(\mathsf{pk}, x) = y$*, where* ϵ_{pre} *is negligible.*
- *Pre-image Min-entropy: For all* $y \in R$*, the probability of any element in the conditional distribution of* $Sample(1^n)$ *given* $f(\mathsf{pk}, x) = y$ *is less than* ϵ_{prob}*, where* ϵ_{prob} *is negligible.*

For a PSF, we define two problems: $Inv(\mathcal{F}$ is identical to the problem with the same name for trapdoor permutations, and $Col(\mathcal{F}) = (Game(\mathcal{F}), 0)$ where $Game(\mathcal{F})$ is the following game between a quantum adversary A and the challenger Ch: Ch, on input n, runs $G(1^n)$ to obtain $(\mathsf{sk}, \mathsf{pk})$, and sends pk to A. A

is allowed to make quantum random oracle queries $O(\cdot)$. When A outputs a pair (x_1, x_2), Ch outputs 1 if and only if both $x_1 \neq x_2$ and $f(\mathsf{pk}, x_1) = f(\mathsf{pk}, x_2)$.

Definition 11. *A PSF $\mathcal{F}$ is secure against quantum adversaries if $Inv(\mathcal{F})$ and $Col(\mathcal{F})$ are both hard for quantum computers*

[GPV08] construct a PSF whose security is based on the hardness of lattice problems.

Signature schemes. We next review signature schemes using our unified notation.

Definition 12 (Signature Scheme). *A random oracle signature scheme is a triple of functions $\mathcal{S} = (G, S^O, V^O)$ where:*

- *O is a random oracle.*
- *$G(1^n)$ generates a pair $(\mathsf{sk}, \mathsf{pk})$ where sk is the signer's private key, and pk is the public key.*
- *$S^O(\mathsf{sk}, m)$ generates a signature σ.*
- *$V^O(\mathsf{pk}, m, \sigma)$ returns 1 if and only if σ is a valid signature on m.*
  We require that $V^O(\mathsf{pk}, m, S^O(\mathsf{sk}, m)) = 1$ for all m and $(\mathsf{sk}, \mathsf{pk})$ generated by G.

For a signature scheme $\mathcal{S}$, we define the problem $Sig{-}Forge(\mathcal{S}) = (Game(\mathcal{S}), 0)$ where $Game(\mathcal{S})$ is the following game between a quantum adversary A and the challenger Ch^O: Ch^O, on input n, runs $G(1^n)$ to obtain $(\mathsf{sk}, \mathsf{pk})$. It then sends pk to A. A is allowed to make quantum random oracle queries $O(\cdot)$ and classical signature queries $S^O(\mathsf{sk}, \cdot)$ to Ch^O. When A outputs a forgery candidate (m, σ), Ch^O outputs 1 if and only if A never asked for a signature on m and σ is a valid signature for m ($V^O(\mathsf{pk}, m, \sigma) = 1$).

Definition 13. *A signature scheme $\mathcal{S}$ is secure against quantum adversaries if $Sig - Forge(\mathcal{S})$ is hard for quantum computers*

Encryption. We next review encryption systems using our notation.

Definition 14 (Symmetric Key Encryption Scheme). *A symmetric key random oracle encryption scheme is a pair of functions $\mathcal{E} = (E^O, D^O)$ where:*

- *$E^O(k, m)$ generates a ciphertext c.*
- *$D^O(k, c)$ computes the plaintext m corresponding to ciphertext c. We require that $D^O(k, E^O(k, m)) = m$.*

For a symmetric key encryption scheme $\mathcal{E}$, we define the problem $Sym{-}CCA(\mathcal{E}) = (Game(\mathcal{E}), \frac{1}{2})$ where $Game(\mathcal{E})$ is the following game between a quantum adversary A and the challenger Ch^O: Ch^O, on input n, generates a key k of length n at random, and sends k to A. A is allowed to make quantum random oracle queries $O(\cdot)$, classical encryption queries $E^O(k, \cdot)$, and classical decryption queries $D^O(k, \cdot)$. A is also allowed one classical challenge query, where it sends Ch^O a pair (m_0, m_1). Ch^O chooses a random bit b, and computes $c = E^O(k, m_b)$, which it sends to A. When A returns a bit b', Ch^O outputs 1 if and only if both $b = b'$ and there was no decryption query $D^O(k, c)$ after the challenge query.

Definition 15 (Symmetric Key CCA Security). *A symmetric key encryption scheme $\mathcal{E}$ is Chosen Ciphertext Attack (CCA) secure against quantum adversaries if $Sym - CCA(\mathcal{E})$ is hard for quantum computers.*

Definition 16 (Asymmetric Key Encryption Scheme). *An Asymmetric key encryption scheme is a triple of functions $\mathcal{E} = (G, E^O, D^O)$ where:*

- *$G(1^n)$ generates a private/public key pair* $(\mathsf{sk}, \mathsf{pk})$
- *$E^O(\mathsf{pk}, m)$ generates a ciphertext c.*
- *$D^O(\mathsf{sk}, c)$ computes the plaintext m corresponding to ciphertext c. We require that $D^O(\mathsf{sk}, E^O(\mathsf{pk}, m)) = m$.*

For a symmetric key encryption scheme $\mathcal{E}$, we define the problem $Asym - CCA(\mathcal{E}) = (Game(\mathcal{E}), \frac{1}{2})$ where $Game(\mathcal{E})$ is the following game between a quantum adversary A and the challenger Ch^O: Ch^O, on input n, uses $G(1^n)$ to generate $(\mathsf{sk}, \mathsf{pk})$, and sends pk to A. A is allowed to make quantum random oracle queries $O(\cdot)$ and classical decryption queries $D^O(\mathsf{sk}, \cdot)$. A if also allowed to make one classical challenge query, where it sends Ch^O a pair (m_0, m_1). Ch^O chooses a random bit b, and computes $c = E^O(\mathsf{pk}, m_b)$, which it sends to A. When A returns a bit b', Ch^O outputs 1 if and only if both $b = b'$ and there was no decryption query $D^O(\mathsf{sk}, c)$ after the challenge query.

Definition 17 (Asymmetric Key CCA Security). *An Asymmetric key encryption scheme $\mathcal{E}$ is Chosen Ciphertext Attack (CCA) secure against quantum adversaries if $Asym - CCA(\mathcal{E})$ is hard for quantum computers.*

B Security of the IS* Protocol

To prove security of our protocol we need to show that an adversary $\mathcal{A}$ after interacting with an honest prover $\mathcal{P}^*$, can subsequently not impersonate $\mathcal{P}^*$ such that $\mathcal{V}^*$ accepts the identification.

Security against Classical Adversaries. We first show that the IS* protocol is secure in the (standard) random oracle model against classical adversaries and then discuss that there exist hash functions, which can securely replace the random oracle.

Lemma 4. *Let* $\mathrm{IS} = (\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ *be a secure identification scheme. Then for any efficient* classical *adversary $\mathcal{A}$ and $\ell > 6\log(\alpha)$ the protocol* IS^* *is secure in the random oracle.*

Proof. Assume towards contradiction that a verifier $\mathcal{V}^*$ after interacting with an adversary $\mathcal{A}$, both given (pk, ℓ) as input, accepts with output $b^* = 1$. Thus, $\mathcal{A}$ must have convinced $\mathcal{V}^*$ in the evaluation of the IS-protocol or provided at least $r/4$ collisions. Due to the independence of the two stages of our protocol (in particular, sk is not used during the collision search) we have

$$\Pr \mathcal{A} \text{ “breaks” IS}^* \leq \Pr \mathsf{collCount} > r/4 + \Pr \mathcal{A} \text{ “breaks” IS.}$$

Since we assume that the underlying identification scheme is secure, the latter probability is negligible. Thus, it remains to show that an adversary $\mathcal{A}$ with access to a random oracle H finds $r/4$ near-collisions on $H(k_i, \cdot)$ for given k_i in time $O(\sqrt[3]{2^\ell})$ with negligible probability only. In the random oracle model, the ability of finding collisions is bounded by the birthday attack, which states that after sending $\sqrt{2^\ell}$ random input values[1], at least one pair will collide with probability $\geq 1/2$. Taking possible parallel power of the adversary into account, the protocol allows $\mathcal{A}$ to make at most $\alpha \cdot \sqrt[3]{2^\ell}$ queries for some constant $\alpha \geq 1$ (Assumption 1). Since $\ell > 6\log(\alpha)$ we have $\alpha \cdot \sqrt[3]{2^\ell} < \sqrt{2^\ell}$ and thus $\mathcal{A}$'s success probability for finding a collision in each round is $< 1/2$ which vanishes when repeating the collision search r times.

More concretely, the upper bound on the birthday probability for q queries and a function with range size N is given by $\frac{q(q-1)}{2N}$ (see e.g. [**?**]). Thus, when considering an adversary making at most $q = \alpha\sqrt[3]{2^\ell}$ queries to a random oracle with range $\{0,1\}^\ell$ we obtain:

$$\Pr \mathsf{Coll} \;\leq\; \frac{\alpha^2}{2\sqrt[3]{2^\ell}} \;\leq\; \frac{\alpha^2}{2\sqrt[3]{n}}$$

due to the choice of $\ell \leq \log n$. The repetition of such a constrained collision search does not increase the success probability of the adversary, since the verifier sends a fresh “key” k_i in each round. Thus, the adversary cannot reuse already learned values from the random oracle, but has to start the collision search from scratch for each new key. That is, the probability of $\mathcal{A}$ finding a collision is at most $\Pr \mathsf{Coll}$ in each round.

Applying the Chernoff-bound yields the probability for finding at least $r/4$ collision in r independent rounds:

$$\Pr \mathsf{collCount} > r/4 \;\leq\; \exp\left(-\frac{r\alpha^2}{2\sqrt[3]{n}} \cdot \left(\frac{\sqrt[3]{n} - 2\alpha^2}{2\alpha^2}\right)^2 \cdot \frac{1}{4}\right) \;\leq\; \exp\left(-\frac{r\sqrt[3]{n}}{32\alpha^2}\right)$$

Thus, for a constant α, and setting $r = \mathrm{poly}(n)$ the above term is negligible in n. However, then, the overall success probability of $\mathcal{A}$ is negligible as well.

[1] Note that we give all statements for a random oracle outputting directly $\ell \leq \log(n)$ bits, as we are interested in near-collisions. Such an oracle can be obtained from a random oracle with range $\{0,1\}^n$ by simply truncating the output to the first ℓ bits.

When considering classical adversaries only, we can securely instantiate the random oracle in the IS^* scheme by a hash function H that provides near-collision-resistance close to the birthday bound. Under this assumption, the security proof of our identification scheme carries over to the standard model, as well. (We omit a formal proof, as it follows the argumentation of Lemma 4 closely.) Note that it is a particular requirement of the SHA-3 competition [NIS07], that the hash function candidates achieve collision-resistance approximately up to the birthday bound and provide this property also for any fixed subset of the hash functions' output bits. Thus, all remaining SHA-3 candidates (or at least the winner of the competition) is supposed to be quasi-optimal near-collision-resistant.

Security against Quantum Adversaries. We now show that such a result is impossible in the quantum world, i.e., for any hash function H there exists a quantum-adversary $\mathcal{A}_Q$ that breaks the IS^* protocol (regardless of the security of the underlying identification scheme). This contrasts with the security that can still be achieved in the (classical) random oracle model:

Lemma 5. *Let* $\mathrm{IS}_Q = (\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ *be a secure quantum-immune identification scheme. Then for any efficient* quantum *adversary* $\mathcal{A}_Q$ *and* $\ell > 6\log(\alpha)$ *the protocol* IS^* *is secure in the random oracle model.*

Proof. By assuming that IS_Q is a quantum-immune identification scheme, an adversary $\mathcal{A}_Q$ trying to convince a verifier $\mathcal{V}^*$ in the IS^* protocol must provide at least $r/4$ many collisions in the first stage of the protocol. Thus, we have to show that a quantum adversary $\mathcal{A}_Q$ can succeed in the collision-search with negligible probability only.

Note that in order to gain advantage of the quantum speed-up (e.g., by applying Grover's search algorithm) the random oracle H, resp. the indicator function based on H, has to be evaluated on quantum states, i.e., on superpositions of many input strings. However, by granting $\mathcal{A}_Q$ only classical access to the random oracle, it is not able to exploit its additional quantum power to find collisions on H. Thus, $\mathcal{A}_Q$ has to stick to the classical collision-search on a random oracle, which we have proven to succeed in $r/4$ of r rounds with negligible probability, due to the constraint of making at most $\alpha \cdot \sqrt[3]{2^\ell}$ oracle queries per round (see proof of Lemma 4 for details).

We now show that our IS^* scheme becomes totally insecure for any instantiation of the random oracle by a hash function H.

Lemma 6. *There exist an efficient* quantum *adversary* $\mathcal{A}_Q$ *such that for any hash function* $\mathrm{H} = (\mathsf{H.KGen}, \mathrm{H}.\mathsf{\mathit{Eval}})$ *the protocol* IS^* *is* insecure*.*

Proof. For the proof, we show that a quantum-adversary $\mathcal{A}_Q$ can find collisions on H in at least $r/4$ rounds with non-negligible probability. To this end, we first transform the classical hash function H into a quantum-accessible function

H_Q. For the transformation, we use the fact that any classical computation can be done on a quantum computer as well [NC00]. The ability to evaluate H_Q on superpositions then allows to apply Grover's algorithm in a straightforward manner: for any key k_i that is sent by the verifier $\mathcal{V}^*$, the adversary invokes Grover's search on an indicator function testing whether $\mathrm{H}_Q.\mathsf{Eval}(k_i, x)|_\ell = \mathrm{H}_Q.\mathsf{Eval}(k_i, x')|_\ell$ for distinct $x \neq x'$ holds. After $\sqrt[3]{2^\ell}$ evaluations of H_Q the algorithm outputs a collision M_i, M_i' with probability $> 1/2$. As we assume that a quantum evaluation of H_Q requires roughly the same time than an evaluation of the corresponding classical function H, and we do not charge $\mathcal{A}_Q$ for any other computation, the collision search of $\mathcal{A}_Q$ terminates before $\mathcal{V}^*$ stops a round of the collision-finding stage.

Hence, $\mathcal{A}_Q$ provides a collision with probability $> 1/2$ in each of the r rounds. Using the Chernoff bound, we can now upper bound the probability that $\mathcal{A}_Q$ finds *less* than $r/4$ collision as:

$$\Pr \mathsf{collCount} < r/4 \;\leq\; \exp\left(-\frac{r}{2} \cdot \left(\frac{1}{2}\right)^2 \cdot \frac{1}{2}\right) \;\leq\; \exp\left(-\frac{r}{16}\right)$$

which is roughly $\Pr \mathsf{Coll} < r/4 \;\leq\; 0.94^r$ and thus negligible as a function of r. That is, the adversary $\mathcal{A}_Q$ can make $\mathcal{V}^*$ accept the interaction with noticeable probability at least $1 - \Pr \mathsf{collCount} < r/4$.

As Grover's algorithm only requires (quantum-accessible) black-box access to the hash function, the approach described in the proof of Lemma 6 directly applies to the quantum-accessible random oracle model, as well:

Lemma 7. *The protocol* IS^* *is* not *secure in the quantum-accessible random oracle model.*

C Proof of Lemma 3

Before we prove Lemma 3, we need to prove the following two technical lemmas:

Lemma 8. *Let* $|\phi\rangle$ *and* $|\phi'\rangle$ *be superpositions with* $|\phi - \phi'| \leq \gamma$. *Let* P *be some property on strings. Suppose measuring* $|\phi\rangle$ *gives a string that satisfies* P *with probability* ϵ. *Then measuring* $|\phi'\rangle$ *gives a string that satisfies* P *with probability* ϵ' *where*

$$\sqrt{\epsilon} - \gamma \leq \sqrt{\epsilon'} \leq \sqrt{\epsilon} + \gamma$$

Proof. We will prove this lemma geometrically. We can think of a state $|\phi\rangle$ as a vector $\boldsymbol{\phi}$ in $\mathbb{C}^n$. Then the basis elements $|x\rangle$ as elements of the standard basis for $\mathbb{C}^n$. We are given that $|\phi - \phi'| \leq \gamma$, meaning that $\boldsymbol{\phi}$ and $\boldsymbol{\phi'}$ have Euclidean distance of at most γ.

For a bit string x, the probability that sampling $|\phi\rangle$ results in x is $|\langle \boldsymbol{x}, \boldsymbol{\phi}\rangle|^2$. Let S_P be the set of basis elements $\boldsymbol{x}$ such that x satisfied P. The probability that sampling $|\phi\rangle$ results in a string satisfying P is then given by

$$\sum_{\boldsymbol{x}\in S_P} |\langle \boldsymbol{x}, \boldsymbol{\phi}\rangle|^2$$

This also is the square of the length of the projection of $\boldsymbol{\phi}$ onto the subspace spanned by S_P. So, let $\boldsymbol{\phi_P}$ and $\boldsymbol{\phi'_P}$ be the projections of $\boldsymbol{\phi}$ and $\boldsymbol{\phi'}$ onto the space spanned by S_P. The probability that sampling $|\phi\rangle$ (resp. $|\phi'\rangle$) results in a string satisfying P is simply $\epsilon = |\boldsymbol{\phi_P}|^2$ (resp. $\epsilon' = |\boldsymbol{\phi'_P}|^2$). Projections only decrease distance, so by the triangle inequality,

$$\sqrt{\epsilon'} = |\boldsymbol{\phi'_P}| \leq |\boldsymbol{\phi_P}| + |\boldsymbol{\phi_P} - \boldsymbol{\phi'_P}| \leq |\boldsymbol{\phi_P}| + |\boldsymbol{\phi} - \boldsymbol{\phi'}| \leq \sqrt{\epsilon} + \gamma$$

Reversing the roles of $|\phi\rangle$ and $|\phi'\rangle$ gives us the other inequality.

Lemma 9. *Let A be an quantum algorithm that makes at most q queries to quantum random oracle O. Fix a y in the co-domain of O. The expected value of the total query probability of all x such that $O(x) = y$ is at most $\frac{2q^3}{2^m}$.*

Proof. Suppose we have an oracle O' for which the output on every input is distributed identically and independently, with a uniform distribution over $\{0,1\}^m \setminus \{y\}$. We now modify the oracle as follows: for each input x, with probability 2^{-m}, replace the output with y. This oracle is now a random oracle, so its distribution is identical to O.

Let σ_i be the total query magnitude over the first $i-1$ queries of x such that we change $O'(x)$. Let δ_i be the query magnitude of those x in the ith query. Let γ_i be the Euclidean distance between the state of A at the ith query when using oracle O' and the modified oracle O. By (2), $\gamma_i \leq \sqrt{(i-1)\sigma_i}$. Let ρ_i be the query magnitude of x such that $O(x) = y$ (which is the same as the query probability of x such that we changed $O'(x)$). By the above lemma,

$$\begin{aligned}\rho_i &\leq (\sqrt{\delta_i} + \gamma_i)^2 \\ &= \delta_i + \gamma_i^2 + 2\sqrt{\delta}\gamma_i \\ &\leq \delta_i + (i-1)\sigma_i + 2\sqrt{(i-1)\delta_i\sigma_i} \\ &\leq \delta_i + (i-1)\sigma_i + 2\sqrt{i-1}(\delta_i + \sigma_i)\end{aligned}$$

Now, observe that since we are deciding whether to change the output of a query point at random and independently, the expected query probability of the points that we changed in each query is exactly 2^{-m}. Thus, $\mathbb{E}[\delta_i] = 2^{-m}$ and $\mathbb{E}[\sigma_i] = (i-1)2^{-m}$. Thus,

$$\mathbb{E}[\rho_i] \leq 2^{-m}(1 + (i-1)^2 + 2\sqrt{i-1}(1 + (i-1))) \leq 2^{-m}2i^2$$

This result is not surprising, as it implies that any quantum algorithm which is to output a preimage of y with overwhelming probability must make $O(\sqrt{2^{-m}})$

quantum oracle queries, which is well known lower bound for the unstructured search problem (see Bennett et al. [BBBV97] for more). Summing over all q queries gives the expected query probability of x such that $O(x) = y$ to be at most $2 \times 2^{-m}q^3$.

Proof of Lemma 3. We are given a random oracle O and a distribution D that is ϵ-close to uniform. Observe that:

$$\begin{aligned}
\epsilon &= \sum_y \left|\Pr[y|D] - 2^{-m}\right| \\
&= \sum_{y:\Pr[y|D]\geq 2^{-m}} \left(\Pr[y|D] - 2^{-m}\right) + \sum_{y:\Pr[y|D]<2^{-m}} \left(2^{-m} - \Pr[y|D]\right) \\
0 &= \sum_{y:\Pr[y|D]\geq 2^{-m}} \left(\Pr[y|D] - 2^{-m}\right) - \sum_{y:\Pr[y|D]<2^{-m}} \left(2^{-m} - \Pr[y|D]\right)
\end{aligned}$$

Thus,

$$\frac{\epsilon}{2} = \sum_{y:\Pr[y|D]\geq 2^{-m}} \left(\Pr[y|D] - 2^{-m}\right) = \sum_{y:\Pr[y|D]<2^{-m}} \left(2^{-m} - \Pr[y|D]\right)$$

Define a distribution D' as follows:

- If $\Pr[y|D] < 2^{-m}$, $\Pr[y|D'] = 0$.
- If $\Pr[y|D] \geq 2^{-m}$, $\Pr[y|D'] = (\Pr[y|D] - 2^{-m})2/\epsilon$

All the probabilities are clearly non-negative. For this to be a probability distribution, the probabilities need to um to 1:

$$\sum_y \Pr[y|D'] = \sum_{y:\Pr[y|D]\geq 2^{-m}} (\Pr[y|D] - 2^{-m})\frac{2}{\epsilon} = \frac{\epsilon}{2}\frac{2}{\epsilon} = 1$$

Now, we can create another distribution D'' as follows: first, generate y uniformly at random. Then,

- If $\Pr[y|D] \geq 2^{-m}$, output y.
- If $\Pr[y|D] < 2^{-m}$, then with probability $2^m \Pr[y|D]$, output y. Otherwise, pick a y' from D' and output y'.

If $\Pr[y|D] < 2^{-m}$, $\Pr[y|D''] = 2^{-m} \times (2^m \Pr[y|D]) = \Pr[y|D]$. Otherwise,

$$\begin{aligned}
\Pr[y|D''] &= 2^{-m} + \sum_{y':\Pr[y'|D]<2^{-m}} 2^{-m}(1 - 2^m \Pr[y'|D]) \Pr[y|D'] && \text{(C.1)} \\
&= 2^{-m} + \sum_{y':\Pr[y'|D]<2^{-m}} (2^{-m} - \Pr[y'|D])(\Pr[y|D] - 2^{-m})\frac{2}{\epsilon} && \text{(C.2)} \\
&= 2^{-m} + \frac{\epsilon}{2}(\Pr[y|D] - 2^{-m})\frac{2}{\epsilon} = \Pr[y|D] && \text{(C.3)}
\end{aligned}$$

Thus $D'' = D$. This demonstrates that we can construct the oracle O' whose elements are distributed according to D as follows: Start with the random oracle O, and for each input x, if $\Pr[O(x)|D] < 2^{-m}$, then with probability $1 - 2^m \Pr[O(x)|D]$, replace the output with a y' drawn from D'_X. Otherwise leave the oracle unchanged at that point.

Now we bound the expected query magnitude of x such that the oracle changed. By the above lemma, the expected total query probability of any x such that $O(x) = y$ is $2q^3 2^{-m}$. Let σ be the query magnitude of points x at which we changed the oracle:

$$\begin{aligned}
\mathbb{E}[\sigma] &= \mathbb{E}\left[\sum_{x:\Pr[O(x)|D]<2^{-m}} (1 - 2^{-m}\Pr[O(x)|D]) \times (\text{total query magnitude of } x)\right] \\
&= \sum_{y:\Pr[y|D]<2^{-m}} (1 - 2^m \Pr[y|D])\, \mathbb{E}[\text{total query magnitude of } x \text{ such that } O(x) = y] \\
&\leq \sum_{y:\Pr[y|D]<2^{-m}} (1 - 2^m \Pr[y|D]) 2q^3 2^{-m} \\
&= 2q^3 \sum_{y:\Pr[y|D]<2^{-m}} (2^{-m} - \Pr[y|D]) = \frac{2q^3\epsilon}{2} = q^3\epsilon
\end{aligned}$$

Thus the expected Euclidean distance is

$$\mathbb{E}[\sqrt{q\sigma}] \leq \sqrt{q\,\mathbb{E}[\sigma]} \leq \sqrt{q \times q^3\epsilon} = q^2\sqrt{\epsilon}$$

This means the expected variational distance of the output distributions is at most $4q^2\sqrt{\epsilon}$. Thus, the distribution of outputs when the oracle values are distributed according to D is at most $4q^2\sqrt{\epsilon}$ away from the distribution of outputs when the oracle is truly random. □