# Random Oracles in a Quantum World

Özgür Dagdelen[1] Marc Fischlin[1] Anja Lehmann[2*] Christian Schaffner[3]

[1]CASED & Darmstadt University of Technology, Germany
[2]IBM Research Zurich, Switzerland
[3]Centrum Wiskunde & Informatica (CWI) Amsterdam, The Netherlands

**Abstract.** Once quantum computers reach maturity most of today's traditional cryptographic schemes based on RSA or discrete logarithms become vulnerable to quantum-based attacks. Hence, schemes which are more likely to resist quantum attacks like lattice-based systems or code-based primitives have recently gained significant attention. Interestingly, a vast number of such schemes also deploy random oracles, which have mainly be analyzed in the classical setting.

Here we revisit the random oracle model in cryptography in light of quantum attackers. We show that there are protocols using quantum-immune primitives and random oracles, such that the protocols are secure in the classical world, but insecure if a quantum attacker can access the random oracle via quantum states. We discuss that most of the proof techniques related to the random oracle model in the classical case cannot be transferred immediately to the quantum case. Yet, we show that "quantum random oracles" can nonetheless be used to show for example that the basic Bellare-Rogaway encryption scheme is quantum-immune against plaintext attacks (assuming quantum-immune primitives).

## 1 Introduction

Lattice-based cryptosystems, as well as coding-based schemes, recently gained a lot of attention as alternatives to traditional schemes based on RSA or the discrete logarithm problem [RSA78, DH76]. Besides broadening the set of cryptographic assumptions, lattice-based systems for example come with three other advantages over traditional schemes:

- They offer average-case to worst-case hardness [Ajt96, MR04].
- They enable new functional properties such as fully homomorphic encryption schemes [Gen09].
- They are likely to be resistant against quantum attackers because —unlike RSA, for example— they are often based on NP-hard problems [Mic98, AR03, AR05]. This is also believed for coding-based schemes.

The latter property becomes tricky when the schemes are combined with random oracles, as in [CGP01, Cou04, Ove09, Rüc08, GPV08, Lyu09, DV09, BM10].

*This work was done while the author was at Darmstadt University of Technology, Germany.

The random oracle model in cryptography has been introduced to ease the design of truly practical schemes withstanding threats from *classical* (i.e., non-quantum) attackers [BR93]. The question is whether it is sufficient to simply replace the number-theoretic primitives in such schemes or if security needs to be re-assessed from scratch when using the random oracle model also as a hedge against quantum attackers.

We note that none of the authors of the above mentioned works claims (or proves) that their random oracle based schemes remain secure against quantum attackers. Still, it would be desirable to show this property and to preserve all advantages over traditional schemes. From a more abstract point of view, taking a closer look at random oracles and quantum attackers is also an intriguing question to prepare for the quantum era: replacing traditional systems like RSA by quantum-immune ones may not be sufficient to save random oracle based constructions.

**Modeling Random Oracles for Quantum Attackers.** In the classical random oracle model the (classical) adversary is given oracle access to a random hash function $H : \{0,1\}^* \to \{0,1\}^*$. As such, the adversary can only "learn" a hash value $H(x)$ by querying the oracle about $x$. We propose two new models for random oracles in the quantum world. In a straightforward attempt to transfer the classical model to the quantum world, we may thus simply lend the adversary "local quantum power" and insist on classical access to $H$.

In a different approach, following the idea behind the random oracle model that the idealized oracle is eventually replaced by a "strong" implementation, the quantum attacker can evaluate the implementation *on quantum states*. To capture such possibilities in the model we allow the adversary to evaluate the oracle "in superposition", i.e. he can submit quantum states $|\varphi\rangle = \sum \alpha_x |x\rangle$ to the oracle $H$ and receives back the evaluated state $\sum \alpha_x |H(x)\rangle$ (appropriately encoded to make the transformation unitary). We call this the *quantum-accessible 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 consider classical algorithms, being able to communicate only classically, *honest* parties and the scheme's algorithms can access $H$ only via classical bit strings.

We remark that the quantum-accessible model provides strictly stronger security guarantees. Yet, none of the two new models is, per se, superior to the other. After all, the (classical) random oracle model already allows for provable secure solutions that lose any security guarantees when translated into the standard model, where the random oracle is instantiated by an efficient hash function [CGH98]. Hence, one may also consider the classical random oracle model to be appropriate to capture a large class of attacks, especially since the only *known provable* speed-up of quantum algorithms that is applicable to hash functions is quadratic, namely, Grover's algorithm [Gro96].[1]

Note, too, that we do not advocate to use either approach in a quantum setting without second thoughts. This is especially true since, at this point, is not even clear what the "right" quantum random oracle model is; we merely suggest two reasonable approaches. One should always seek for random-oracle-free solutions in both the classical and the quantum worlds. The question we are addressing is how much of known results from the classical (random oracle) world we can carry over to the quantum setting instantaneously, given that we adopt the idea of having a strong hash function for which generic attacks are optimal. Since Grover's algorithm —and more generally,

[1] However, special care must be taken with respect to concrete security claims in this case. As an example, finding collisions for random oracles with $n$ bits output would still require $2^{n/2}$ steps on the average for a quantum attacker then, whereas collision search given the hash function's code can be performed in $2^{n/3}$ steps by a quantum computer [BHT98].

any computation on qubits— can be considered to be generic, we should at least take such attack methods into account.

Nonetheless, to motivate our preference for the stronger model we present a two-party protocol which is

- secure in the classical world in the random oracle model, and presumably also when the random oracle is instantiated,
- 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. Note that, without this assumption, quantum computers would not give an advantage over classical (parallel) computers, and the question if the quantum adversaries can be more powerful seems to be moot.

**Proofs in the Quantum-Accessible ROM.** Proofs in our quantum-accessible ROM become much more involved than in the classical model, mainly because several artefacts of the model do not seem to carry over immediately. Among others, these are:

**Adaptive Programmability:** The classical random oracle model allows a simulator to program the answers of the random oracle for an adversary, often adaptively. A well-known example are simulators in zero-knowledge proofs where challenges are replaced by hash values [BR93] and the simulator adaptively changes the hash value to make it fit. 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.

**Rewinding/Partial Consistency:** The random oracle based proof for Fiat-Shamir type schemes [PS00] requires rewinding of the adversary, replaying some hash values but changing at least a single value. Besides the usual problems of rewinding quantum adversaries, we again encounter the fact that we may not be able to change hash values unnoticed.

**Extractability/Preimage Awareness:** Another advantage of the random oracle model for classical adversaries is that one learns the preimages 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 and it is unclear how to extract the right query.

**Efficient Simulation:** Related to programmability, but requiring a weaker form of simulation, is the plain simulation of random oracles (without injecting appropriate values). In the classical world, we can easily 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 now evaluate the random oracle on all inputs simultaneously, disallowing the on-demand strategy for classical oracles.

We do not claim that any of these problems cannot be solved in the quantum world, e.g., some form of rewinding is possible for quantum zero-knowledge [Wat09]. 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 also 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.

We remark that quantum access to pseudorandom functions seems to be more than one usually needs for security against quantum attackers. Consider for example a symmetric-key identification scheme where the prover, upon receiving a random string $r$ from the verifier, evaluates the pseudorandom function on $r$ and returns the outcome. Then, also a quantum attacker on the verifier's side can only access the pseudorandom function through the prover and therefore only on classical bits. While such pseudorandom functions can be derived from quantum-immune pseudorandom generators [GGM86], it is an open problem if the stronger quantum-accessible pseudorandom functions exist.

**A Positive Example: BR Encryption.** Once we have overcome the simulation problem we show that the extractability property can be ensured for quantum-accessible random oracles, at least in a weak sense. That is, we consider the BR encryption scheme [BR93] where a ciphertext $(f(r), H(r) \oplus m)$ contains the image of a random value $r$ under an injective trapdoor function $f$ (like the lattice-based one described in [GPV08]) and a traditional one-time pad encryption of the message via the quantum-accessible random oracle $H$ evaluated for string $r$.

Here, we show that the adversary, mounting a chosen-plaintext attack against bit strings encrypted by a classical party, cannot submit quantum states to $H$ such that the probability of asking the unknown random value $r$ is large. Otherwise we show how to invert $f$ successfully by measuring such a query, collapsing the state to $r$ with sufficiently large probability. In this sense, we can extract only once, and only with a certain probability, from adversarial quantum queries.

Our reduction, showing how to use the quantum distinguisher against the encryption scheme to build a quantum inverter against $f$, uses the quantum-accessible pseudorandom function to simulate the random oracle. Our proof, however, actually provides a more fine-grained view: In the main step we first devise a quantum inverter which breaks the security of $f$ *relative to a random oracle*. Only then we show that we can simulate the quantum-accessible random oracle via a quantum-accessible pseudorandom function to get an inverter against $f$ *in the plain model*.

Hence, even if quantum-accessible pseudorandom functions do not exist, our result provides a hardness result *relative to a (quantum-accessible) random oracle*. Such results relative to random oracles must be taken with a grain of salt, though, as it is not clear if this provides meaningful implications in the standard world; see [HCKM93, For94] for surveys. We discuss this and further questions arising in the context of random oracles in a quantum world at the end of the paper.

## 2 Random Oracles and Quantum Attackers

We first briefly recall facts about quantum computations and set some notation; for more details, we refer to [NC00].

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 $\alpha_x$ 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 $|\alpha_x|^2$.

Following [BBC+98] we model a quantum attacker $\mathcal{A}_{\mathrm{Q}}$ with access to 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 $m = \mathrm{poly}(n)$ qubits. Here, oracle $O_i : \{0,1\}^n \to \{0,1\}^n$ maps the first $2n$ qubits from basis state $|x\rangle\,|y\rangle$ to basis state $|x\rangle\,|y \oplus O_i(x)\rangle$ for $x, y \in \{0,1\}^n$. 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 (as opposed to oracles that 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.

**Distance Measures.** For two quantum states $|\varphi\rangle = \sum \alpha_x |x\rangle$ and $|\psi\rangle = \sum \beta_x |x\rangle$ in superposition in the basis states $|x\rangle$, the Euclidean distance is given by $\left(\sum_x |\alpha_x - \beta_x|^2\right)^{1/2}$. The total variation distance (aka. statistical difference) of two distributions $\mathcal{D}_0, \mathcal{D}_1$ is defined through $\sum_x |\mathrm{Prob}[\mathcal{D}_0 = x] - \mathrm{Prob}[\mathcal{D}_1 = x]|$. The following fact from [BV93] upperbounds the total variance distance in terms of the Euclidean distance:

**Theorem 2.1 ([BV93])** *Let $|\varphi\rangle, |\psi\rangle$ be quantum states with Euclidean distance at most $\epsilon$. Then, performing the same measurement on $|\varphi\rangle, |\psi\rangle$ yields distributions with statistical distance at most $4\epsilon$.*

# 3 Negative 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 (when using quantum-immune primitives). We then use the quadratic gap between the birthday attack and 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 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 (IS.$\mathsf{KGen}$, $\mathcal{P}, \mathcal{V}$) where IS.$\mathsf{KGen}$ on input $1^n$ returns a key pair $(sk, pk)$. The joint execution of $\mathcal{P}(sk, pk)$ and $\mathcal{V}(pk)$ 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 H.$\mathsf{KGen}$ for input $1^n$ returns a key $k$ (which contains $1^n$), and 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 $1 \leq \ell \leq n$ and $(M, M') \leftarrow \mathcal{A}(k, \ell)$ we have $M \neq M'$ 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 $\ell$ 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 states 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 3.1 (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 assumption 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 3.2 (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 3.3 (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 propose 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. In this stage the verifier checks if the prover is able to produce collisions on a hash function in a particular time. 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.

More precisely, let $\mathrm{IS} = (\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ be an identification scheme and $\mathrm{H} = (\mathrm{H}.\mathsf{KGen}, \mathrm{H}.\mathsf{Eval})$ be a hash function. We construct an identification scheme $\mathrm{IS}^* = (\mathrm{IS}.\mathsf{KGen}^*, \mathcal{P}^*, \mathcal{V}^*)$ as follows:

The key generation algorithm $\text{IS.}\mathsf{KGen}^*(1^n)$ simply runs $(sk, pk) \leftarrow \text{IS.}\mathsf{KGen}(1^n)$ and returns $sk$ and $(pk, \ell)$ for some value $\ell \leq \log(n)$.

The interactive protocol $\langle \mathcal{P}^*(sk, (pk, \ell)), \mathcal{V}^*(pk, \ell) \rangle$ then consists of two stages: In the first stage the verifier chooses a key $k \leftarrow \text{H.}\mathsf{KGen}$ for the hash function $H$ and sends it to the prover. For timekeeping, the verifier then starts to evaluate the hash function $\text{H.}\mathsf{Eval}(k, \cdot)$ on the messages $\langle c \rangle$ for $c = 1, 2, \ldots, \left\lceil \sqrt[3]{2^\ell} \right\rceil$, 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 $\text{H.}\mathsf{Eval}(k, M) = \text{H.}\mathsf{Eval}(k, M')$ holds for the first $\ell$ bits. One round of the collision-stage ends if $\mathcal{V}^*$ either receives such a collision from $\mathcal{P}^*$ or finishes its $\sqrt[3]{2^\ell}$ hash evaluations. The verifier and the receiver then repeat such a round $r = \text{poly}(n)$ times, sending a fresh key $k$ in each round. Recall that, if assuming a random oracle $H$, then both parties are supposed to evaluate the function $H(k, \cdot)$.

In the second stage of our scheme, the prover and verifier run the underlying identification protocol $\langle \mathcal{P}(sk, pk), \mathcal{V}(pk) \rangle$ leading to a decision bit $b$. Finally, the verifier outputs $b^* = 1$ if $b = 1$ or if the prover was able to provide collisions in at least $r/4$ rounds of the first stage. Else, $\mathcal{V}^*$ rejects with output $b^* = 0$. Note that due to our choice of $\ell \leq \log(n)$, the protocol $\text{IS}^*$ still runs in polynomial time.

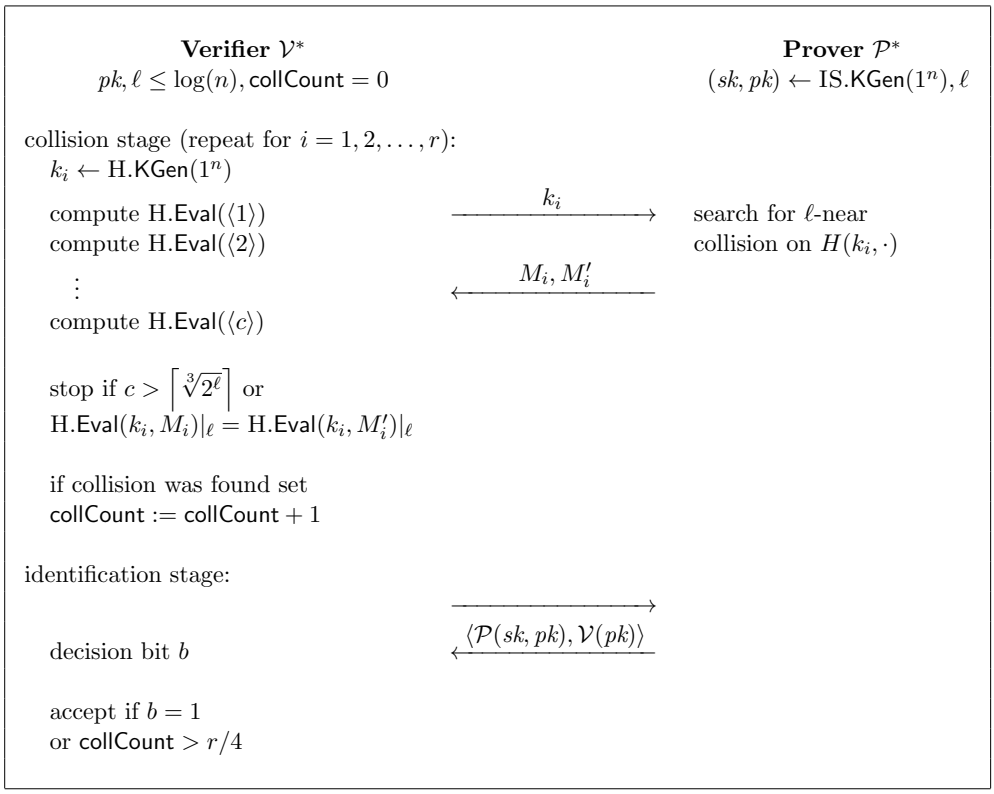


Figure 1: The $\text{IS}^*$-Identification Protocol

Completeness of the $\text{IS}^*$ 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 $\ell$ be such that $\ell > 6\log(\alpha)$ where $\alpha$ is the constant from Assumption 3.1. By assuming that $\mathrm{IS} = (\mathrm{IS}.\mathsf{KGen}, \mathcal{P}, \mathcal{V})$ is a quantum-immune identification scheme, we can show that $\mathrm{IS}^*$ is secure in the standard random oracle model against classical and quantum adversaries.

The main idea is that for oracle models, 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 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 to apply Grover's search algorithm. But 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 A.

# 4 Positive Result

In this section we show that, assuming quantum-accessible pseudorandom functions exist, the Bellare-Rogaway encryption scheme $\mathsf{Enc}(m; r) = (f(r), H(r) \oplus m)$ is indistinguishable against quantum attackers in the quantum-accessible random oracle model.

## 4.1 Quantum-Immune Encryption

A public-key encryption scheme $\mathrm{E} = (\mathrm{E}.\mathsf{KGen}, \mathrm{E}.\mathsf{Enc}, \mathrm{E}.\mathsf{Dec})$ consists of efficient classical algorithms to generate a key pair $(sk, pk) \leftarrow \mathrm{E}.\mathsf{KGen}(1^n)$, to encrypt messages $C \leftarrow \mathrm{E}.\mathsf{Enc}(pk, m)$ and to decrypt ciphertexts $m \leftarrow \mathrm{E}.\mathsf{Dec}(sk, C)$ such that $m = \mathrm{E}.\mathsf{Dec}(sk, \mathrm{E}.\mathsf{Enc}(pk, m))$ for all $m, (sk, pk) \leftarrow \mathrm{E}.\mathsf{KGen}(1^n)$.

A well-known encryption scheme in the random oracle model is the Bellare-Rogaway (BR) scheme [BR93], allowing also to deploy quantum-immune primitives like the injective trapdoor function of Gentry et al. [GPV08]. Note that a quantum-immune injective trapdoor function $\mathrm{F} = (\mathrm{F}.\mathsf{KGen}, \mathrm{F}.\mathsf{Eval}, \mathrm{F}.\mathsf{Inv})$ consists of efficient algorithms where $\mathrm{F}.\mathsf{KGen}(1^n)$ generates descriptions of functions $f, f^{-1}$ such that $y = \mathrm{F}.\mathsf{Eval}(f, x)$ and $x = \mathrm{F}.\mathsf{Inv}(f^{-1}, y)$ for all $x \in \{0,1\}^n$, and all $f, f^{-1}$ output by $\mathrm{F}.\mathsf{KGen}$. We often identify $\mathrm{F}.\mathsf{Eval}(f, \cdot)$ with $f(\cdot)$ and $\mathrm{F}.\mathsf{Inv}(f^{-1}, \cdot)$ with $f^{-1}(\cdot)$. Furthermore, for all efficient quantum algorithms $\mathcal{I}_\mathrm{Q}$ the probability $\mathrm{Prob}[\mathcal{I}_\mathrm{Q}(f, f(x)) = x]$ that $\mathcal{I}_\mathrm{Q}$ breaks one-wayness is negligible (in $n$), where the probability is over the choice of $f$ and $x$ and $\mathcal{I}_\mathrm{Q}$'s random choices.

**Construction 4.1 (BR-encryption scheme)** *Let $F$ be an injective trapdoor function and $H$ be a hash function. Then define the BR-encryption scheme as follows:*

- *Key generation:* $\mathrm{E.KGen}$ *on input* $1^n$ *runs* $\mathrm{F.KGen}(1^n)$ *to get* $(f, f^{-1})$ *and returns* $(sk, pk) = (f^{-1}, f)$.
- *Encryption:* $\mathrm{E.Enc}(pk, m)$ *encrypts a message* $m \in \{0,1\}^n$ *by picking* $r \leftarrow \{0,1\}^n$ *at random and returning* $(f(r), H(r) \oplus m)$.
- *Decryption:* $\mathrm{E.Dec}(sk, C)$ *parses* $C$ *as* $(y, z)$ *and returns* $H(f^{-1}(y)) \oplus z$ *(or* $\perp$ *if inverting* $y$ *fails).*

We define chosen-plaintext security of an encryption scheme against quantum adversaries analogously to the classical case. That is, the adversary $\mathcal{A}_{\mathrm{Q}}$ receives as input the public key $pk$ of the encryption scheme, generated as $(sk, pk) \leftarrow \mathrm{E.KGen}(1^n)$, and can query an external oracle once. This oracle accepts as input two messages $m_0, m_1 \in \{0,1\}^n$, picks a random bit $b$ and returns $\mathrm{E.Enc}(pk, m_b)$, denoted as the challenge ciphertext. We call this oracle the *challenge oracle* $\mathsf{Ch}_b$ for bit $b$. Algorithm $\mathcal{A}_{\mathrm{Q}}$ eventually outputs a bit $b'$ trying to predict $b$. Note that we are only considering the case where the attacker uses quantum power, whereas the other parties rely on classical communication and thus require confidentiality for regular bit strings only.

**Definition 4.2** *An encryption scheme* E *is qIND-CPA (quantum indistinguishable under chosen-plaintext attacks) if for any efficient quantum adversary* $\mathcal{A}_Q$ *the probability* $\mathrm{Prob}\left[\mathcal{A}_Q^{\mathsf{Ch}_b}(pk) = b\right]$ *for the challenge oracle* $\mathsf{Ch}_b$ *is negligibly close to* $1/2$ *for random bit* $b$.

## 4.2 Security Proof Relative to Quantum-Accessible Random Oracles

We now show that the BR scheme is qIND-CPA in the quantum-accessible random oracle model. Recall that our proof is in two steps: First we show that, if one can break qIND-CPA, then one can break the trapdoor function relative to a random oracle where we give the inverter algorithm $\mathcal{I}_{\mathrm{Q}}$ quantum access to a random oracle $H$ and measure its probability to invert $f(r)$ also with respect to the choice of this oracle. In the second step we show that we can simulate the random oracle via quantum-accessible pseudorandom functions to break one-wayness of the trapdoor function in the plain model. The following theorem provides the first step:

**Theorem 4.3** *Assume that* F *is a quantum-immune injective trapdoor function (in the quantum-accessible random oracle model). Then the BR-encryption scheme is qIND-CPA in the quantum-accessible random oracle model.*

The proof follows the classical case in spirit. We assume that a successful attacker of the encryption scheme exists and show that this assumption leads to a contradiction. We can define the aggregated probability that the attacker queries an element $r$ from the oracle by summing up all the squared amplitudes of his quantum states before the oracle calls are made. We distinguish the two cases where the attacker queries the preimage $r$ with negligible or non-negligible total probability. In the latter case, we can use the attacker to invert F which contradicts the one-wayness of F (relative to a random oracle).

If $r$ is only queried with negligible probability, Theorem 4.4 below due to Bennett et al. [BBBV97] allows us to replace the answer to the $r$-query of the random oracle with an independent random string without changing the attacker's success probability significantly. In this new experiment however, the response from the challenge oracle does not depend on $b$ anymore and the maximal

success probability of any attacker must be 1/2, a contradiction to the assumed non-negligible advantage of our attacker.

In order to apply Theorem 4.4, we will argue that the quantum registers of $\mathcal{A}_{\mathrm{Q}}$ are always in a pure state, also after querying the classical challenge oracle. To this end, we model a quantum attacker $\mathcal{A}_{\mathrm{Q}}$ as follows: His quantum state consists of $m$ qubits initialized in the state $|0\rangle^{\otimes m}$ and he is given a classical description of $f$. He then performs a unitary transformation $U_1$ to obtain the state $|\phi_1\rangle$. Then, the random oracle $O$ acts on the first $2n$ registers of $|\phi_1\rangle$ ($n$ qubits for the in- and output, respectively) and the next unitary $U_2$ is performed (on all $m$ qubits) yielding $|\phi_2\rangle$, etc. At some point $1 \leq t_c \leq T$, the classical challenge oracle is queried. Formally, after the unitary $U_{t_c}$, the first $2n$ registers of $|\phi_{t_c}\rangle$ are measured in the computational basis yielding outcomes $m_0, m_1$ which are given to the challenge oracle. The oracle $\mathsf{Ch}_b$ samples uniformly $b \in \{0,1\}$ and $r \in \{0,1\}^n$ and returns $(f(r), H(r) \oplus m_b)$. Let us abbreviate all the classical information learned at this challenge step by $k := (m_0, m_1, f(r), H(r) \oplus m_b)$. The subsequent unitary operations $U_{t_c+1}, \ldots, U_T$ by $\mathcal{A}_{\mathrm{Q}}$ can now depend on the classical information $k$. We note that the quantum state of $\mathcal{A}_{\mathrm{Q}}$ remains pure also after the challenge phase, because the interaction with $\mathsf{Ch}_b$ is entirely classical. After the last unitary $U_T$, the first qubit of $|\phi_T\rangle$ is measured in the computational basis and the outcome is interpreted as $\mathcal{A}_{\mathrm{Q}}$'s guess of the bit $b$.

As we have seen, $|\phi_t\rangle$ denotes the quantum state of the adversary after performing the $t$-th unitary $U_t$ and before making the $t$-th oracle call. For $r \in \{0,1\}^n$, let $q_r(|\phi_t\rangle)$ be the query probability of string $r$ at time $t$, i.e., if we write out the state in the computational basis $|\phi_t\rangle = \sum_x \alpha_x |x\rangle$ and the input to the oracle $O$ is given in the first $n$ registers, then $q_r(|\phi_t\rangle) = \sum_{v \in \{0,1\}^{m-n}} |\alpha_{rv}|^2$.

**Theorem 4.4 (Theorem 3.3 in [BBBV97])** *Let $\mathcal{A}_Q$ 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 $\sum_{(t,r)\in S} q_r(|\phi_t\rangle) \leq \epsilon^2/T$. 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), and we consider the state $|\phi'_T\rangle$ when $\mathcal{A}_Q$ is invoking $O'$ instead of $O$, then the Euclidean distance between $|\phi_T\rangle$ and $|\phi'_T\rangle$ is at most $\epsilon$.*

*Proof (of Theorem 4.3).* Suppose that there exists a quantum attacker $\mathcal{A}_{\mathrm{Q}}$ distinguishing the two cases in the encryption scheme. Assume that this adversary has non-negligible advantage $\delta = \delta(n)$. Since $\mathcal{A}_{\mathrm{Q}}$ receives either $(f(r), H(r) \oplus m_0)$ or $(f(r), H(r) \oplus m_1)$ from the challenge oracle it must have learned the value $H(r)$, otherwise $H(r) \oplus m_b$ is from $\mathcal{A}_{\mathrm{Q}}$'s point of view indistinguishable from random. Note that the interaction with the challenge oracle is purely classical, as we are interested in security in the scenario where only the adversary is granted quantum power and he intercepts classical ciphertexts.

Suppose that we change the experiment of $\mathcal{A}_{\mathrm{Q}}$ such that instead of computing $H(r) \oplus m_b$ in the challenge ciphertext, we pick a random value $R$ and return $R \oplus m_b$ (together with $f(r)$). Denote this experiment by $\mathsf{Random}$. It is clear that the success probability of any attacker in $\mathsf{Random}$ is 1/2, because all the information seen is completely independent of the random bit $b$.

For $r \in \{0,1\}^n$, let $Q_r = \sum_{t=1}^{T-1} q_r(|\phi_t^r\rangle)$ be the aggregated probability that $\mathcal{A}_{\mathrm{Q}}$ in the game $\mathsf{Random}$ asks query $r$ to the random oracle. Call $r$ *good* if $Q_r$ is negligible. We distinguish the following two cases:

1. There exists a non-negligible fraction of bad $r$'s. In this case, we show how $\mathcal{A}_{\mathrm{Q}}$ can be used to invert $f$, contradicting the one-wayness of $f$.

2. A random $r$ is good with overwhelming probability. In this case, Theorem 4.4 will allow us to argue that $\mathcal{A}_{\mathrm{Q}}$'s success probability only negligibly differs from 1/2 in contradiction to our assumption.

For the first case, we assume that there exists a non-negligible fraction of *bad* $r$'s for which $Q_r$ is non-negligible. In particular, since $T$ is polynomial this means that there exists a $t$ for which $q_r(|\phi_t\rangle)$ is non-negligible for such $r$'s. We show that we can then invert the trapdoor function $f$ with non-negligible probability by a quantum attacker $\mathcal{I}_{\mathrm{Q}}$. The idea is to guess the time $t$ as follows.

Algorithm $\mathcal{I}_{\mathrm{Q}}$ is given $f$ and $f(r)$ for random $r$ as input. It also picks a random index $i$ between 1 and $T$, a random bit $b$ and a random value $R$. It then executes algorithm $\mathcal{A}_{\mathrm{Q}}$ against the encryption scheme in game $\mathsf{Random}$. For the first $i-1$ times if adversary $\mathcal{A}_{\mathrm{Q}}$ would make a call to the random oracle algorithm then $\mathcal{I}_{\mathrm{Q}}$ forwards the query to his own quantum-accessible random oracle. If the adversary asks for the challenge ciphertext then $\mathcal{I}_{\mathrm{Q}}$ returns $(f(r), R \oplus m_b)$. Before executing the $i$-th query, algorithm $\mathcal{I}_{\mathrm{Q}}$ measures the first $n$ registers in the computational basis. If the registers contain the preimage $r$ of $f(r)$ then stop and output $r$; else stop with output $\perp$. It now follows from our assumptions that the inverter $\mathcal{I}_{\mathrm{Q}}$ succeeds with non-negligible probability.

For the second case, let $r$ be the randomness used by the challenge oracle in the game $\mathsf{Random}$. As $r$ is chosen independently at random, it follows from our assumption that $Q_r$ is negligible with overwhelming probability. We define $S = \{(t, r) | 1 \leq t \leq T\}$ to be the set of pairs where the time-part $t$ is arbitrary and the string equals $r$. Hence, with overwhelming probability, the assumption of Theorem 4.4 is fulfilled and the Euclidean distance of the final state $|\phi_T\rangle$ of the game $\mathsf{Random}$ and the final state $|\phi'_T\rangle$ of an experiment where we changed the answers of the random oracle on query $r$ to $R$ is negligibly close. Note that the modified experiment is identical to a run of the original experiment in which $\mathcal{A}_{\mathrm{Q}}$ has non-negligible advantage $\delta(n)$. By Theorem 2.1, the variational distance of the distribution of the output bits in the two experiments is negligible and therefore, the advantage of $\mathcal{A}_{\mathrm{Q}}$ in $\mathsf{Random}$ should be non-negligible, contradicting our previous observation that no attacker in $\mathsf{Random}$ can have an advantage over random guessing. □

### 4.3 Replacing Random Oracles by Pseudorandom Functions

Note that in the above proof we grant the adversary $\mathcal{I}_{\mathrm{Q}}$ also access to a random oracle, such that we formally have reduced the qIND-CPA security of the BR-encryption scheme to a quantum-immune injective trapdoor function which is secure in the (quantum-accessible) random oracle model. However, we can obtain the same result for F in the standard model, if we assume that quantum-accessible pseudorandom functions exist and let $\mathcal{I}_{\mathrm{Q}}$ simulate the random oracle with such a PRF.

**Quantum-Accessible Pseudorandom Functions.** Recall that there are two ways to define quantum-immune pseudorandom functions. In both cases the distinguisher is an efficient quantum algorithm but in one case it can access the random or pseudorandom oracle only classically, and in the other case it gets quantum-access to the oracle (i.e., the oracles operate on quantum states). We need the latter for our application and drop the term "quantum-immune" as it follows from the fact that it is quantum accessible:

**Definition 4.5** *A quantum-accessible pseudorandom function is a pair of efficient classical algorithms,* $(\mathrm{PRF}.\mathsf{KGen}, \mathrm{PRF}.\mathsf{PRF})$*, such that for any efficient quantum algorithm* $\mathcal{D}_Q$ *the difference*

$$\left|\mathrm{Prob}\left[\mathcal{D}_Q^{|\mathrm{PRF}.\mathsf{PRF}(\kappa,\cdot)\rangle}(1^n) = 1\right] - \mathrm{Prob}\left[\mathcal{D}_Q^{|R(\cdot)\rangle}(1^n) = 1\right]\right|$$

*is negligible (in $n$), where the probability in the first case is over the choice of $\kappa \leftarrow \mathrm{PRF}.\mathsf{KGen}(1^n)$ and $\mathcal{D}_Q$'s random choices (e.g., measurements), and in the second case over the random function $R$ (with the same domain and range as $\mathrm{PRF}.\mathsf{PRF}$) and $\mathcal{D}_Q$'s random choices.*

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

As mentioned in the introduction, we are not aware if such pseudorandom functions exist.

**Security Proof.** We next show that, starting with our successful inverter in the random oracle model, we can apply quantum-accessible pseudorandom functions to get security without reference to random oracles:

**Lemma 4.6** *If quantum-accessible pseudorandom functions exist then any quantum-immune injective trapdoor function in the quantum-accessible random oracle model is also quantum-immune in the standard model (i.e., without random oracles).*

*Proof.* Let $\mathcal{I}_\mathrm{Q}$ be a successful quantum inverter against the trapdoor function $F$ in the quantum-accessible random oracle model. We construct now adversary $\mathcal{I}'_\mathrm{Q}$ (without random oracle access) which too inverts the trapdoor function sufficiently often. This algorithm $\mathcal{I}'_\mathrm{Q}$ works as $\mathcal{I}_\mathrm{Q}$ but replaces the queries that $\mathcal{I}_\mathrm{Q}$ would issue to its quantum-accessible random oracle by executions of the circuit for computing $\mathrm{PRF}(\kappa, \cdot)$ (on quantum states) for a self chosen key $\kappa$. Inverter $\mathcal{I}'_\mathrm{Q}$ eventually copies the (classical) output of $\mathcal{I}_\mathrm{Q}$ and stops.

We claim that the transition from the random oracle to the quantum-accessible pseudorandom function does not have a noticeable impact on the success probability of $\mathcal{I}_\mathrm{Q}$. That is, if this was not the case and the success probability would drop significantly, then we can construct a distinguisher $\mathcal{D}_\mathrm{Q}$ against the quantum-accessible pseudorandom function as follows.

Algorithm $\mathcal{D}_\mathrm{Q}$ initially generates $(f, f^{-1}) \leftarrow \mathrm{F}.\mathsf{KGen}(1^n)$, picks a random $r$ and computes $f(r)$. It invokes $\mathcal{I}_\mathrm{Q}$ on $f$, meaning that it gradually performs the computation steps of $\mathcal{I}_\mathrm{Q}$ and for each oracle call, algorithm $\mathcal{D}_\mathrm{Q}$ calls its own oracle. If $\mathcal{I}_\mathrm{Q}$ eventually stops with some output, algorithm $\mathcal{D}_\mathrm{Q}$ verifies if the output matches $r$ and returns 1 if so, and 0 otherwise.

In case $\mathcal{D}_\mathrm{Q}$'s oracle implements a truly random function, $\mathcal{D}_\mathrm{Q}$'s experiment corresponds exactly to the attack of $\mathcal{I}_\mathrm{Q}$ in the quantum-accessible random oracle model. In this case, $\mathcal{D}_\mathrm{Q}$ outputs 1 with non-negligible probability by assumption. If $\mathcal{D}_\mathrm{Q}$'s oracle implements a pseudorandom function then the experiment corresponds exactly to the attack of $\mathcal{I}'_\mathrm{Q}$ and, by assumption, $\mathcal{D}_\mathrm{Q}$ outputs 1 with negligible probability only. In summary, algorithm $\mathcal{D}_\mathrm{Q}$ would distinguish the two cases significantly. □

# 5 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.

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 the example of the Bellare-Rogaway scheme where some ideas can indeed be re-used for the quantum world, if 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.

Alternatively to assuming the existence of quantum-accessible pseudorandom functions, one may —as we did in a first step of our proof— consider reductions to primitives *relative to a random oracle*. However, the hardness of the primitives in a relativized world may not hold in the standard world, i.e., such reductions may not provide meaningful security claims. That is, Bennett and Gill [BG81] originally conjectured that hardness results in complexity theory, which hold relative to a random oracle with probability 1, should also be true in the standard setting (Random Oracle Hypothesis). This conjecture, though, turned out to be false, mainly for complexity classes related to interactive proof systems. See again [HCKM93, For94] for overviews.

As explained in the introduction, there are several strategies to use random oracles in proofs, besides the question on how to simulate the random oracle. It is currently unclear whether similar security proofs can be obtained for cryptographic schemes that require stronger intervention with the random oracle, such as OAEP [BR95, FOPS01] where the reduction needs to observe *all* queries to the random oracles, or FDH [BR96] where one even has to "program" parts of the output. Thus, it remains an interesting open problem to revisit the security guarantees for those schemes in the quantum-accessible model.

## Acknowledgments

Marc Fischlin and Anja Lehmann were supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG). This work was also supported by CASED (http://www.cased.de) and an NWO VICI grant.

## References

[Ajt96] Miklós Ajtai. *Generating Hard Instances of Lattice Problems (Extended Abstract).* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1996, pages 99–108. ACM, 1996.

[AR03] Dorit Aharonov and Oded Regev. *A Lattice Problem in Quantum NP.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2003, pages 210–219. IEEE Computer Society Press, 2003.

[AR05] Aharonov and Regev. *Lattice Problems in NP intersect coNP. JACM: Journal of the ACM*, 52, 2005.

[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 Vazirani. *Strengths and Weaknesses of Quantum Computing. SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BBC+98] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. *Quantum Lower Bounds by Polynomials*. 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?* Workshop Record of SHARCS'09: Special-purpose Hardware for Attacking Cryptographic Systems, 2009.

[BG81] Charles H. Bennett and John Gill. *Relative to a Random Oracle A, $P^A$ != $NP^A$ != co-$NP^A$ with Probability 1. SIAM Journal on Computing*, 10(1):96–113, 1981.

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. *Quantum Cryptanalysis of Hash and Claw-Free Functions*. 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.

[BKR94] M. Bellare, J. Kilian, and P. Rogaway. *The Security of Cipher Block Chaining Message Authentication Code*. Advances in Cryptology — Crypto'94, Volume 839 of Lecture Notes in Computer Science, pages 341–358. Springer-Verlag, 1994.

[BM10] Paulo S. L. M. Barreto and Rafael Misoczki. *A new one-time signature scheme from syndrome decoding*. Number 2010/017 in Cryptology eprint archive. `eprint.iacr.org`, 2010.

[BR93] Mihir Bellare and Pil Rogaway. *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*. Proceedings of the Annual Conference on Computer and Communications Security (CCS). ACM Press, 1993.

[BR95] Mihir Bellare and Phillip Rogaway. *Optimal Asymmetric Encryption — How to Encrypt with RSA*. Advances in Cryptology — Eurocrypt'94, Volume 950 of Lecture Notes in Computer Science, pages 92–111. Springer-Verlag, 1995.

[BR96] Mihir Bellare and Phillip Rogaway. *The exact security of digital signatures — How to sign with RSA and Rabin*. Advances in Cryptology — Eurocrypt'96, Volume 1070 of Lecture Notes in Computer Science, pages 399–416. Springer-Verlag, 1996.

[BV93] Ethan Bernstein and Umesh V. Vazirani. *Quantum complexity theory*. Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1993, pages 11–20. ACM Press, 1993.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. *The Random Oracle Methodology, Revisited*. Proceedings of the Annual Symposium on the Theory of Computing (STOC) 1998, pages 209–218. ACM Press, 1998.

[CGP01] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *FLASH, a Fast Multivariate Signature Algorithm*. Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2001, Volume 2020 of Lecture Notes in Computer Science, pages 298–307. Springer-Verlag, 2001.

[Cou04] Nicolas T. Courtois. *Short Signatures, Provable Security, Generic Attacks and Computational Security of Multivariate Polynomial Schemes such as HFE, Quartz and Sflash.* Number 2004/143 in Cryptology eprint archive. `eprint.iacr.org`, 2004.

[DH76] W. Diffie and M. E. Hellman. *New Directions in Cryptography. IEEE Transactions on Information Theory*, 22(5):644–654, November 1976.

[DV09] Léonard Dallot and Damien Vergnaud. *Provably Secure Code-Based Threshold Ring Signatures.* IMA International Conference, Volume 5921 of Lecture Notes in Computer Science, pages 222–235. Springer-Verlag, 2009.

[FOPS01] Eiichiro Fujisaki, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern. *RSA—OAEP is secure under the RSA Assumption.* Advances in Cryptology — Crypto 2001, Volume 2139 of Lecture Notes in Computer Science, pages 260–274. Springer-Verlag, 2001.

[For94] Lance Fortnow. *The Role of Relativization in Complexity Theory. Bulletin of the European Association for Theoretical Computer Science*, 52:52–229, 1994.

[Gen09] Craig Gentry. *Fully homomorphic encryption using ideal lattices.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2009, pages 169–178. ACM, 2009.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. *How to Construct Random Functions. Journal of the ACM*, 33:792–807, 1986.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. *Trapdoors for hard lattices and new cryptographic constructions.* Proceedings of the Annual Symposium on the Theory of Computing (STOC) 2008, pages 197–206. ACM, 2008.

[Gro96] Lov K. Grover. *A fast quantum mechanical algorithm for database search.* 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.* Quantum Computing and Quantum Communications (QCQC) 1998, Volume 1509 of Lecture Notes in Computer Science, pages 126–139. Springer-Verlag, 1998.

[HCKM93] Juris Hartmanis, Richard Chang, Jim Kadin, and Stephen G. Mitchell. *Relativization: a Revisionistic Retrospective. Current Trends in Theoretical Computer Science*, pages 537–545, 1993.

[Lyu09] Vadim Lyubashevsky. *Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures.* Advances in Cryptology — Asiacrypt 2009, Volume 5912 of Lecture Notes in Computer Science, pages 598–616. Springer-Verlag, 2009.

[Mic98] Daniele Micciancio. *On the hardness of the shortest vector problem.* PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 1998.

[MR04] Micciancio and Regev. *Worst-Case to Average-Case Reductions Based on Gaussian Measures*. Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 2004. IEEE Computer Society Press, 2004.

[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.

[Ove09] Raphael Overbeck. *A Step Towards QC Blind Signatures*. Number 2009/102 in Cryptology eprint archive. `eprint.iacr.org`, 2009.

[PS00] David Pointcheval and Jacques Stern. *Security Arguments for Digital Signatures and Blind Signatures. Journal of Cryptology*, 13(3):361–396, 2000.

[RSA78] R. L. Rivest, A. Shamir, and L. Adelman. *A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM*, 21(2):120–126, 1978.

[Rüc08] Markus Rückert. *Lattice-based Blind Signatures*. Number 2008/322 in Cryptology eprint archive. `eprint.iacr.org`, 2008.

[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.

[Wat09] John Watrous. *Zero-Knowledge against Quantum Attacks. SIAM Journal on Computing*, 39(1):25–58, 2009.

# A 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 A.1** *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 $(pk, \ell)$ 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

$$\mathrm{Prob}[\,\mathcal{A} \text{ “breaks” } \mathrm{IS}^*] \leq \mathrm{Prob}[\,\mathsf{collCount} > r/4] + \mathrm{Prob}[\,\mathcal{A} \text{ “breaks” } \mathrm{IS}]\,.$$

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[2], 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 3.1). 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. [BKR94]). 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:

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

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 can not 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 $\text{Prob}[\,\mathsf{Coll}]$ in each round.

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

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

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

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 A.1 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 not possible 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 is in contrast to the security that can still be achieved in the (classical) random oracle model:

[2]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 $\ell$ bits.

**Lemma A.2** *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* $\mathrm{IS}^*$ *is secure in the random oracle model.*

*Proof.* By assuming that $\mathrm{IS}_Q$ is a quantum-immune identification scheme, an adversary $\mathcal{A}_Q$ trying to convince a verifier $\mathcal{V}^*$ in the $\mathrm{IS}^*$ 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 A.1 for details). □

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

**Lemma A.3** *There exist an efficient* quantum *adversary* $\mathcal{A}_Q$ *such that for any hash function* $\mathrm{H} = (\mathrm{H}.\mathsf{KGen}, \mathrm{H}.\mathit{Eval})$ *the protocol* $\mathrm{IS}^*$ *is* not *secure.*

*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 $\mathrm{H}_Q$. For the transformation we use the fact that any classical computation can done on a quantum computer as well [NC00]. The ability to evaluate $\mathrm{H}_Q$ on superpositions then allows to apply Grover's algorithm in a straightforward manner: for any key $k_i$ that is send 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 $\mathrm{H}_Q$ the algorithm outputs a collision $M_i, M_i'$ with probability $> 1/2$. As we assume that a quantum evaluation of $\mathrm{H}_Q$ 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:

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

which is roughly $\mathrm{Prob}[\,\mathsf{Coll} < r/4] \;\leq\; 0.94^r$ 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 $1 - \mathrm{Prob}[\,\mathsf{collCount} < r/4]$. □

As Grover's algorithm only requires (quantum-accessible) black-box access to the hash function, the approach described in the proof of Lemma A.3 directly applies to the quantum-accessible random oracle model, as well:

**Lemma A.4** *The protocol* $\mathrm{IS}^*$ *is* not *secure in the quantum-accessible random oracle model.*