The Fiat–Shamir Transformation in a Quantum World

Özgür Dagdelen Marc Fischlin Tommaso Gagliardoni

Technische Universität Darmstadt, Germany
www.cryptoplexity.de
oezguer.dagdelen@cased.de marc.fischlin@gmail.com tommaso@gagliardoni.net

Abstract. The Fiat-Shamir transformation is a famous technique to turn identification schemes into signature schemes. The derived scheme is provably secure in the random-oracle model against classical adversaries. Still, the technique has also been suggested to be used in connection with quantum-immune identification schemes, in order to get quantum-immune signature schemes. However, a recent paper by Boneh et al. (Asiacrypt 2011) has raised the issue that results in the random-oracle model may not be immediately applicable to quantum adversaries, because such adversaries should be allowed to query the random oracle in superposition. It has been unclear if the Fiat-Shamir technique is still secure in this quantum oracle model (QROM).

Here, we discuss that giving proofs for the Fiat-Shamir transformation in the QROM is presumably hard. We show that there cannot be black-box extractors, as long as the underlying quantum-immune identification scheme is secure against active adversaries and the first message of the prover is independent of its witness. Most schemes are of this type. We then discuss that for some schemes one may be able to resurrect the Fiat-Shamir result in the QROM by modifying the underlying protocol first. We discuss in particular a version of the Lyubashevsky scheme which is provably secure in the QROM.

1 Introduction

The Fiat-Shamir transformation [FS87] is a well-known method to remove interaction in three-move identification schemes between a prover and verifier, by letting the verifier's challenge ch be determined via a hash function H applied to the prover's first message com. Currently, the only generic, provably secure instantiation is by modeling the hash function H as a random oracle [BR93, PS00]. In general, finding secure instantiations based on *standard* hash functions is hard for some schemes, as shown in [GK03, BDSG$^+$13]. However, these negative results usually rely on peculiar identification schemes, such that for specific schemes, especially more practical ones, such instantiations may still be possible.

The Quantum Random-Oracle model. Recently, the Fiat-Shamir transformation has also been applied to schemes which are advertised as being based on quantum-immune primitives, e.g., [Lyu09, BM10, GKV10, CLRS10, CVA10, SSH11, MGS11, Sak12, GLP12, AFLT12, CNR12, AJLA$^+$12]. Interestingly, the proofs for such schemes still investigate classical adversaries only. It seems unclear if (and how) one can transfer the proofs to the quantum case. Besides the problem that the classical Fiat-Shamir proof [PS00] relies on rewinding the adversary, which is often considered to be critical for quantum adversaries (albeit not impossible [Wat06, Unr12]), a bigger discomfort seems to lie in the usage of the random-oracle model in presence of quantum adversaries.

As pointed out by Boneh et al. [BDF$^+$11] the minimal requirement for random oracles in the quantum world should be *quantum access*. Since the random oracle is eventually replaced by a standard hash function, a quantum adversary could evaluate this hash function in superposition, while still ignoring any advanced attacks exploiting the structure of the actual hash function. To reflect this in the random-oracle model, [BDF$^+$11] argue that the quantum adversary should be also allowed to query the random oracle in superposition. That is, the adversary should be able to query the oracle on a state $|\varphi\rangle = \sum_x \alpha_x |x\rangle |0\rangle$ and in return would get $\sum_x \alpha_x |x\rangle |H(x)\rangle$. This model is called the quantum random-oracle model (QROM).

Boneh et al. [BDF$^+$11] discuss some classical constructions for encryption and signatures which remain secure in the QROM. They do not cover Fiat-Shamir signatures, though. Subsequently, Boneh and Zhandry [Zha12b, Zha12a, BZ12] investigate further primitives with quantum access, such as pseudorandom functions and MACs. Still, the question about the security of the Fiat-Shamir transform in the QROM raised in [BDF$^+$11] remained open.

Fiat-Shamir Transform in the QROM. Here, we give evidence that conducting security proofs for Fiat-Shamir transformed schemes and black-box adversaries is hard, thus yielding a negative result about the provable security of such schemes. More specifically, we use the meta-reduction technique to rule out the existence of quantum extractors with black-box access to a quantum adversary against the converted (classical) scheme. If such extractors would exist then the meta-reduction, together with the extractor, yields a quantum algorithm which breaks the active security of the identification scheme. Our result covers *any* identification scheme, as long as the prover's initial commitment in the scheme is independent of the witness, and if the scheme itself is secure against active quantum attacks where a malicious verifier may first interact with the genuine prover before trying to impersonate or, as we only demand here, to compute a witness afterwards. Albeit not quantum-immune, the classical schemes of Schnorr [Sch90], Guillou and Quisquater [GQ90], and Feige, Fiat and Shamir [FFS88] are conceivably of this type (see also [BP02]). Quantum-immune candidates are, for instance, [MV03, Lyu08, KTX08, MGS11, SSH11, AJLA$^+$12].

Our negative result does not primarily rely on the rewinding problem for quantum adversaries. Instead, it is rather based on the adversary's possibility to hide actual queries to the quantum random oracle in a "superposition cloud", such that the extractor or simulator cannot elicit or implant necessary information for such queries. In fact, our result reveals a technical subtlety in the QROM which previous works [BDF$^+$11, Zha12a, Zha12b, BZ12] have not addressed at all, or at most implicitly. It refers to the question how a simulator or extractor can answer superposition queries $\sum_x \alpha_x |x\rangle |0\rangle$.

A possible option is to allow the simulator to reply with an arbitrary quantum state $|\psi\rangle = \sum_x \beta_x |x\rangle |y_x\rangle$, e.g., by swapping the state from its local registers to the ancilla bits for the answer in order to make this step unitary. This seems to somehow generalize the classical situation where the simulator on input x returns an arbitrary string y for $H(x)$. Yet, the main difference is that returning an arbitrary state $|\psi\rangle$ could also be used to eliminate some of the input values x, i.e., by setting $\beta_x = 0$. This is more than what the simulator is able to do in the classical setting, where the adversary can uniquely identify the preimage x to the answer. In the extreme the simulator in the quantum case, upon receiving a (quantum version of) a classical state $|x\rangle |0\rangle$, could simply reply with an (arbitrary) quantum state $|\psi\rangle$. Since quantum states are in general indistinguishable, in contrast to the classical case the adversary here would potentially continue its execution for inputs which it has not queried for.

In previous works [BDF$^+$11, Zha12b, Zha12a, BZ12] the simulator specifies a classical (possibly probabilistic) function h which maps the adversary query $\sum_x \alpha_x |x\rangle |0\rangle$ to the reply $\sum_x \alpha_x |x\rangle |h(x)\rangle$. Note that the function h is not given explicitly to the adversary, and that it can thus implement keyed functions like a pseudorandom function (as in [BDF$^+$11]). This basically allows the simulator to freely assign values $h(x)$ to each string x, without being able to change the input values. It also corresponds to the idea that, if the random oracle is eventually replaced by an actual hash function, the quantum adversary can check that the hash function is classical, even if the adversary does not aim to exploit any structural weaknesses (such that we still hide h from the adversary).

We thus adopt the approach of letting the simulator determine the quantum answer via a classical probabilistic function h. In fact, our impossibility hinges on this property but which we believe to be rather "natural" for the aforementioned reasons. From a mere technical point of view it at least clearly identifies possible venues to bypass our hardness result. In our case we allow the simulator to specify the (efficient) function h adaptively for each query, still covering techniques like programmability in the classical setting. Albeit this is sometimes considered to be a doubtful property [FLR$^+$10] this strengthens our impossibility result in this regard.

Positive Results. We conclude with some positive result. It remains open if one can "rescue" plain Fiat-Shamir for schemes which are not actively secure, or to prove that alternative but still reasonably efficient approaches work. However, we can show that the Fiat-Shamir technique in general *does* provide a secure signature scheme in the QROM if the protocol allows for oblivious commitments. Roughly, this means that the honest verifier generates the prover's first message com obliviously by sampling a random string and sends com to the prover. In the random oracle transformed scheme the commitment is thus computed via the random oracle, together with the challenge. Such schemes are usually not actively secure against malicious verifiers. Nonetheless, we stress that in order to derive a secure signature scheme via the Fiat-Shamir transform, the underlying identification scheme merely needs to provide passive security and honest-verifier zero-knowledge.

To make the above transformation work, we need that the prover is able to compute the response for commitments chosen obliviously to the prover. For some schemes this is indeed possible if the prover holds some trapdoor information. Albeit not quantum-immune, it is instructive to look at the Guillou-Quisquater RSA-based proof of knowledge [GQ90] where the prover shows knowledge of $w \in \mathbb{Z}_N^*$ with $w^e = y \bmod N$ for $x = (e, N, y)$. For an oblivious commitment the prover would need to compute an e-th root for a given commitment $R \in \mathbb{Z}_N^*$. If the witness would contain the prime factorization of N, instead of the e-th root of y, this would indeed be possible. As a concrete allegedly quantum-immune example we discuss that we can still devise a provably secure signature version of Lyubashevsky's identification scheme [Lyu12] via our method. Before, Lyubashevsky only showed security in the classical random-oracle model, despite using an allegedly quantum-immune primitive.

Our results are summarized in Figure 1. Actively secure identification schemes with witness-independent commitments (lower right area) are hard to prove secure in the quantum random oracle model. Schemes with oblivious and therefore witness-independent commitments can be proven secure (upper right area). Schemes outside of this area may be patched according to our idea exemplified for Lyubashevsky's scheme to turn them into secure signature schemes in the

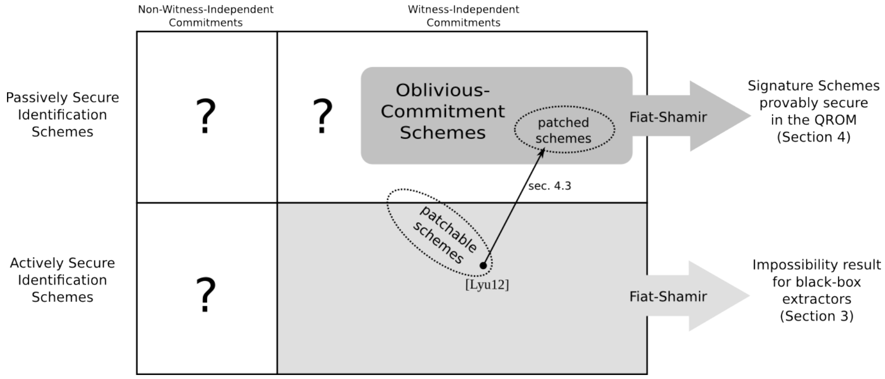

Non-Witness-Independent Commitments
Witness-Independent Commitments
Passively Secure Identification Schemes
Actively Secure Identification Schemes
?
?
?
Oblivious-Commitment Schemes
patched schemes
patchable schemes
sec. 4.3
[Lyu12]
Fiat-Shamir
Fiat-Shamir
Signature Schemes provably secure in the QROM (Section 4)
Impossibility result for black-box extractors (Section 3)

Figure 1: Possibility and impossibility results for the Fiat-Shamir transform of identification schemes in the QROM.

QROM. For any other identification scheme the question remains open.

Related work. Since the introduction of the quantum-accessible random-oracle model [BDF^{+}11], several works propose cryptographic primitives or revisit their security against quantum algorithms in this stronger model [Zha12a, Zha12b, BZ12]. In [DFNS11], Damgård et al. look at the security of cryptographic protocols where the underlying primitives or even parties can be queried by an adversary in a superposition. We here investigate the scenario in which the quantum adversary can only interact classically with the classical honest parties, except for the locally evaluable random oracle.

In a concurrent and independent work, Boneh and Zhandry [BZ13] analyze the security of signature schemes under quantum chosen-message attacks, i.e., the adversary in the unforgeability notion of the signature scheme may query the signing oracle in superposition and, eventually, in the quantum random oracle model. Our negative result carries over to the quantum chosen-message attack model as well, since our impossibility holds even allowing only classical queries to the signing oracle. Moreover, while the authors of [BZ13] show how to obtain signature schemes secure in the quantum-accessible signing oracle model, starting with schemes secure in the classical sense, we focus on signature schemes and proofs of knowledge derived from identification schemes via the Fiat-Shamir paradigm.

2 Preliminaries

We first describe (to the level we require it) quantum computations and then recall the quantum random-oracle model of Boneh et al. [BDF^{+}11]. We also introduce the notion of Σ-protocols to which the Fiat-Shamir transformation applies. In the end of this section, we recall the definition of signature schemes and its security.

2.1 Quantum Computations in the QROM

We first briefly recall facts about quantum computations and set some notation; for more details, we refer to [NC00]. Our description follows [BDF^{+}11] closely.

Quantum Systems. A quantum system A is associated to a complex Hilbert space $\mathcal{H}_A$ of finite dimension and with an inner product $\langle \cdot | \cdot \rangle$. The state of the system is given by a (class of) normalized vector $|\varphi\rangle \in \mathcal{H}_A$ with Euclidean norm $\| \, |\varphi\rangle \, \| = \sqrt{\langle \varphi | \varphi \rangle} = 1$. The joint or composite quantum state of two quantum systems A and B over spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively, is given 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 $|\varphi_A\rangle \otimes |\varphi_B\rangle$. We sometimes simply write $|\varphi_A\rangle |\varphi_B\rangle$ or $|\varphi_A, \varphi_B\rangle$. An n-qubit system is associated 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 $|x\rangle = |x_1\rangle \otimes \cdots \otimes |x_n\rangle$ for $x = x_1 \dots x_n \in \{0,1\}^n$. We often assume that any (classical) bit string x is encoded into a quantum state as $|x\rangle$, and vice versa we sometimes view such a state simply as a classical state. Any pure n-qubit state $|\varphi\rangle$ can be expressed as a superposition in the computational basis as $|\varphi\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle$ where α_x are complex amplitudes obeying $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

Quantum Computations. Evolutions of quantum systems are described by unitary transformations with $\mathbb{I}_A$ being the identity transformation on register A. For a composite 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 $U_A |\varphi_A\rangle |\varphi_B\rangle$ is a simplification of $(U_A \otimes \mathbb{I}_B) |\varphi_A\rangle |\varphi_B\rangle$. Note that any unitary operation and, thus, any quantum operation, is invertible.

Information can be extracted from a quantum state $|\varphi\rangle$ by performing a positive-operator valued measurement (POVM) $M = \{M_i\}_i$ 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 $|\alpha_x|^2$. Measurements can refer to a subset of quantum registers and are in general not invertible.

We model a quantum algorithm $\mathcal{A}_{\mathrm{Q}}$ with access to oracles $O_1, O_2, \dots$ by a sequence of unitary transformations

$$U_1, O_1, U_2, \dots, O_{T-1}, U_T$$

over $m = \mathrm{poly}(n)$ qubits. Here, oracle function $O_i : \{0,1\}^a \to \{0,1\}^b$ maps the final $a + b$ qubits from basis state $|x\rangle |y\rangle$ to $|x\rangle |y \oplus O_i(x)\rangle$ for $x \in \{0,1\}^a$ and $y \in \{0,1\}^b$. This mapping is inverse to itself. We can let the oracles share (secret) state by reserving some qubits for the O_i's only, on which the U_j's cannot operate. Note that the algorithm $\mathcal{A}_{\mathrm{Q}}$ may also receive some (quantum) input $|\psi\rangle$. The adversary may also perform measurements. We sometimes write $\mathcal{A}_{\mathrm{Q}}^{|O_1(\cdot)\rangle, |O_2(\cdot)\rangle, \dots}(|\psi\rangle)$ for the output.

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.

Quantum Random Oracles. We can now define the quantum random-oracle model by picking a random function H for a given domain and range, and letting (a subset of) the oracles O_i evaluate H on the input in superposition, namely those O_i's which correspond to hash oracle queries. In this case the quantum adversary can evaluate the hash function in parallel for many inputs by querying the oracle about $\sum_x \alpha_x |x\rangle$ and obtaining $\sum_x \alpha_x |H(x)\rangle$, appropriately encoded as described above. Note that the output distribution $\mathcal{A}_{\mathrm{Q}}^{|O_1(\cdot)\rangle,|O_2(\cdot)\rangle,\ldots}(|\psi\rangle)$ now refers to the $\mathcal{A}_{\mathrm{Q}}$'s measurements and the choice of H (and the random choices for the other oracles, if existing).

2.2 Classical Interactive Proofs of Knowledge

Here, we review the basic definition of Σ-protocols and show the classical Fiat-Shamir transformation which converts the interactive Σ-protocols into non-interactive proof of knowledge (PoK) protocols (in the random-oracle model). Let $\mathcal{L} \in \mathcal{NP}$ be a language with a (polynomially computable) relation $\mathcal{R}$, i.e., $x \in \mathcal{L}$ if and only if there exists some $w \in \{0,1\}^*$ such that $\mathcal{R}(x,w) = 1$ and $|w| = poly(|x|)$ for any x. As usual, w is called a witness for $x \in \mathcal{L}$ (and x is sometimes called a "theorem" or statement). We sometimes use the notation $\mathcal{R}_\lambda$ to denote the set of pairs (x,w) in $\mathcal{R}$ of some complexity related to the security parameter, e.g., if $|x| = \lambda$.

Σ-Protocols. The well-known class of Σ-protocols between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ allows $\mathcal{P}$ to convince $\mathcal{V}$ that it knows a witness w for a public theorem $x \in \mathcal{L}$, without giving $\mathcal{V}$ non-trivially computable information beyond this fact. Informally, a Σ-protocol consists of three messages $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ where the first message com is sent by $\mathcal{P}$ and the challenge ch is sampled uniformly from a challenge space by the verifier. We write $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \langle \mathcal{P}(x,w), \mathcal{V}(x) \rangle$ for the randomized output of an interaction between $\mathcal{P}$ and $\mathcal{V}$. We denote individual messages of the (stateful) prover in such an execution by $\mathsf{com} \leftarrow \mathcal{P}(x,w)$ and $\mathsf{rsp} \leftarrow \mathcal{P}(x,w,\mathsf{com},\mathsf{ch})$, respectively. Analogously, we denote the verifier's steps by $\mathsf{ch} \leftarrow \mathcal{V}(x,\mathsf{com})$ and $d \leftarrow \mathcal{V}(x,\mathsf{com},\mathsf{ch},\mathsf{rsp})$ for the challenge step and the final decision.

Definition 2.1 (Σ-Protocol) *A Σ-protocol $(\mathcal{P},\mathcal{V})$ for an $\mathcal{NP}$-relation $\mathcal{R}$ satisfies the following properties:*

Completeness. *For any security parameter λ, any $(x,w) \in \mathcal{R}_\lambda$, any $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \langle \mathcal{P}(x,w), \mathcal{V}(x) \rangle$ it holds $\mathcal{V}(x,\mathsf{com},\mathsf{ch},\mathsf{rsp}) = 1$.*

Public-Coin. *For any security parameter λ, any $(x,w) \in \mathcal{R}_\lambda$, and any $\mathsf{com} \leftarrow \mathcal{P}(x,w)$, the challenge $\mathsf{ch} \leftarrow \mathcal{V}(x,\mathsf{com})$ is uniform on $\{0,1\}^{\ell(\lambda)}$ where ℓ is some polynomial function.*

Special Soundness. *Given $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ and $(\mathsf{com}, \mathsf{ch}', \mathsf{rsp}')$ for $x \in \mathcal{L}$ (with $\mathsf{ch} \neq \mathsf{ch}'$) where $\mathcal{V}(x,\mathsf{com},\mathsf{ch},\mathsf{rsp}) = \mathcal{V}(x,\mathsf{com},\mathsf{ch}',\mathsf{rsp}') = 1$, there exists a PPT algorithm Ext (the extractor) which for any such input outputs a witness $w \leftarrow \mathsf{Ext}(x,\mathsf{com},\mathsf{ch},\mathsf{rsp},\mathsf{ch}',\mathsf{rsp}')$ for x satisfying $\mathcal{R}(x,w) = 1$.*

Honest-Verifier Zero-Knowledge (HVZK). *There exists a PPT algorithm Sim (the zero-knowledge simulator) which, on input $x \in \mathcal{L}$, outputs a transcript $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ that is computationally indistinguishable from a valid transcript derived in a $\mathcal{P}$-$\mathcal{V}$ interaction. That is, for any polynomial-time quantum algorithm $\mathcal{D} = (\mathcal{D}_0, \mathcal{D}_1)$ the following distributions are indistinguishable:*

- *Let $(x, w, \mathsf{state}) \leftarrow \mathcal{D}_0(1^\lambda)$. If $\mathcal{R}(x,w) = 1$, then $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \langle \mathcal{P}(x,w), \mathcal{V}(x) \rangle$; else, $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \perp$. Output $\mathcal{D}_1(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}, \mathsf{state})$.*

- *Let* $(x, w, \mathsf{state}) \leftarrow \mathcal{D}_0(1^\lambda)$*. If* $\mathcal{R}(x, w) = 1$*, then* $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \mathsf{Sim}(x)$*; else,* $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \perp$*. Output* $\mathcal{D}_1(\mathsf{com}, \mathsf{ch}, \mathsf{rsp}, \mathsf{state})$*.*

Here, state *can be a quantum state.*

FIAT-SHAMIR (FS) TRANSFORMATION. The Fiat-Shamir transformation of a Σ-protocol $(\mathcal{P}, \mathcal{V})$ is the same protocol but where the computation of ch is done as $\mathsf{ch} \leftarrow H(x, \mathsf{com})$ instead of $\leftarrow \mathcal{V}(x, \mathsf{com})$. Here, H is a public hash function which is usually modeled as a random oracle, in which case we speak of the Fiat-Shamir transformation of $(\mathcal{P}, \mathcal{V})$ in the random-oracle model. Note that we include x in the hash computation, but all of our results remain valid if x is omitted from the input. If applying the FS transformation to a (passively-secure) identification protocol one obtains a signature scheme, if the hash computation also includes the message m to be signed. A formal definition of signature schemes and their security can be found in Section 2.4.

2.3 Quantum Extractors and the FS Transform

QUANTUM EXTRACTORS IN THE QROM. Next, we describe a black-box quantum extractor. Roughly, this extractor should be able to output a witness w for a statement x given black-box access to the adversarial prover. There are different possibilities to define this notion, e.g., see the discussion in [Unr12]. Here, we take a simple approach which is geared towards the application of the FS transform to build secure signature schemes. Namely, we assume that, if a quantum adversary $\mathcal{A}_\mathrm{Q}$ on input x and with access to a quantum-accessible random oracle has a non-negligible probability of outputting a valid proof $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$, then there is an extractor $\mathcal{K}_\mathrm{Q}$ which on input x and with black-box access to $\mathcal{A}_\mathrm{Q}$ outputs a valid witness with non-negligible probability, too.

We need to specify how the extractor simulates the quantum-accessible random oracle. This time we view the extractor $\mathcal{K}_\mathrm{Q}$ as a sequence of unitary transformations $U_1, U_2, U_3, \ldots$, interleaved with interactions with the adversary $\mathcal{A}_\mathrm{Q}$, now represented as the sequence of (stateful) oracles $O_1, O_2, \ldots$ to which $\mathcal{K}_\mathrm{Q}$ has access to. Here each O_i corresponds to the local computations of the adversary until the "next interaction with the outside world". In our case this will be basically the hash queries $|\varphi\rangle$ to the quantum-accessible random oracle. We stipulate $\mathcal{K}_\mathrm{Q}$ to write the (circuit description of a) classical function h with the expected input/output length, and which we assume for the moment to be deterministic, in some register before making the next call to an oracle. Before this call is then actually made, the hash function h is first applied to the quantum state $|\varphi\rangle = \sum_x \alpha_x |x\rangle |0\rangle$ of the previous oracle in the sense that the next oracle is called with $\sum_x \alpha_x |x\rangle |h(x)\rangle$. Note that we can enforce this behavior formally by restricting $\mathcal{K}_\mathrm{Q}$'s steps $U_1, U_2, \ldots$ to be of this described form above.

At some point the adversary will return some classical proof $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ for x. To allow the extractor to rewind the adversary we may assume that the extractor can invoke another run with the adversary (for the same randomness, or possibly fresh randomness, appropriately encoded in the behavior of oracles). If the reduction asks to keep the same randomness then since the adversary only receives classical input x, this corresponds to a reset to the initial state. Since we do not consider adversaries with auxiliary quantum input, but only with classical input, such resets are admissible. Note that the intrinsic "quantum randomness" is fresh for each run. Also note that the extractor can measure any quantum query of the adversary to the random oracle but then cannot continue the simulation of this instance (unless the adversary chose a classical query in the first place). The latter reflects the fact that the extractor cannot change the quantum input state for answering the adversary's queries to the random oracle.

In summary, the black-box extractor can: (a) run several instances of the adversary from the start for the same or fresh classical randomness, (b) for each query to the QRO either measure and abort this execution, or provide a hash function h, and (c) observe the adversary's final output. The black-box extractor cannot, for instance, interfere with the adversary's program and postpone or perform additional measurements, nor rewind the adversary between interactions with the outside world, nor tamper with the internal state of the adversary. As a consequence, the extractor cannot observe the adversary's queries, but we still allow the extractor to access queries if these are classical. In particular, the extractor may choose h adaptively but not based on quantum queries (only on classical queries). We motivate this model with the observation that, in meaningful scenarios, the extractor should only be able to give a classical description of h, which is then "quantum-implemented" by the adversary $\mathcal{A}_\mathrm{Q}$ through a "quantum programmable oracle gate"; the gate itself will be part of the adversary's circuit, and hence will be outside the extractor's influence. Purification of the adversary is also not allowed, since this would discard those adversaries which perform measurements, and would hence hinder the notion of black-box access.

For an interesting security notion computing a witness from x only should be infeasible, even for a quantum adversary. To this end we assume that there is an efficient instance generator Inst which on input 1^λ outputs a pair $(x, w) \in \mathcal{R}$ such that any polynomial-time quantum algorithm on (classical) input x returns some classical string w' with $(x, w') \in \mathcal{R}$, is negligible (over the random choices of Inst and the quantum algorithm). We say Inst is a *hard instance generator for relation $\mathcal{R}$.*

Definition 2.2 (Black-Box Extractor for Σ-Protocol in the QROM) *Let $(\mathcal{P}, \mathcal{V})$ be a Σ-protocol for an $\mathcal{NP}$-relation $\mathcal{R}$ with hard instance generator Inst. Then a black-box extractor $\mathcal{K}_Q$ is a polynomial-time quantum algorithm (as above) such that for any quantum adversary $\mathcal{A}_Q$ with quantum access to oracle H, it holds that, if*

$$\mathrm{Prob}\Big[\mathcal{V}^H(x, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) = 1 \textit{ for } (x, w) \leftarrow \mathsf{Inst}(1^\lambda); (\mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \mathcal{A}_Q^{|H\rangle}(x)\Big] \not\approx 0$$

is not negligible, then

$$\mathrm{Prob}\Big[(x, w') \in \mathcal{R} \textit{ for } (x, w) \leftarrow \mathsf{Inst}(1^\lambda); w' \leftarrow \mathcal{K}_Q^{\mathcal{A}_Q}(x)\Big] \not\approx 0$$

is also not negligible.

For our negative (and our positive) results we look at special cases of black-box extractors, denoted *input-respecting* extractors. This means that the extractor only runs the adversary on the given input x. All known extractors are of this kind, and in general it is unclear how to take advantage of executions for different x'.

On Probabilistic Hash Functions. We note that we could also allow the extractor to output a description of a *probabilistic* hash function h to answer each random oracle call. This means that, when evaluated for some string x, the reply is $y = h(x; r)$ for some randomness r (which is outside of the extractor's control). In this sense a query $|\varphi\rangle = \sum_x \alpha_x |x\rangle |0\rangle$ in superposition returns $|\varphi\rangle = \sum_x \alpha_x |x\rangle |h(x; r_x)\rangle$ for independently chosen r_x for each x.

We can reduce the case of probabilistic functions h to deterministic ones, if we assume quantum-accessible pseudorandom functions [BDF+11]. These functions are indistinguishable from random functions for quantum adversaries, even if queried in superposition. In our setting, in the deterministic case the extractor incorporates the description of the pseudorandom function for a randomly chosen key κ into the description of the deterministic hash function, $h'(x) = h(x; \mathsf{PRF}_\kappa(x))$. Since the hash function description is not presented to the adversary, using such derandomized hash functions cannot decrease the extractor's success probability significantly. This argument can be carried out formally by a reduction to the quantum-accessible pseudorandom function, i.e., by forwarding each query $|\varphi\rangle$ of the QROM adversary to the random or pseudorandom function oracle, and evaluating h as before on x and the oracle's reply. Using a general technique in [Zha12b] we can even replace the assumption about the pseudorandom function and use a q-wise independent function instead.

2.4 Signature Schemes and Their Security

Here, we recall the definition of signature schemes and their security.

Definition 2.3 (Signature Scheme) *A (digital) signature scheme (in the random-oracle model) consists of three efficient algorithms (SKGen, Sig, SVf) defined as follows.*

KEY GENERATION. *On input the security parameter 1^λ, the probabilistic algorithm SKGen^H with oracle access to H outputs a key pair (sk, pk) where sk (resp. pk) denotes the signing key (resp. public verification key).*

SIGNING. *On input a signing key sk and a message m, the probabilistic algorithm Sig^H outputs a signature σ.*

VERIFICATION. *On input the verification key pk, a message m, and a signature σ, the deterministic algorithm SVf^H outputs either 1 (= valid) or 0 (= invalid).*

We require correctness of the verification, i.e., the verifier will always accept genuine signatures. More formally, for any security parameter λ, any $(sk, pk) \leftarrow \mathsf{SKGen}(1^\lambda)$, for any message m, any signature $\sigma \leftarrow \mathsf{Sig}(sk, m)$, we have $\mathsf{SVf}(pk, m, \sigma) = 1$.

From a signature scheme we require that no outsider should be able to forge signatures. Formally, this property is called unforgeability against adaptively chosen-message attacks (unf-cma) and is defined as follows.

Definition 2.4 (UNF-CMA Security) *A (digital) signature scheme $\mathcal{S} = (\mathsf{SKGen}, \mathsf{Sig}, \mathsf{SVf})$ in the random-oracle model is (t, Q, ε)-unforgeable against adaptively chosen-message attacks with $Q = (q_H, q_S)$ if for any algorithm $\mathcal{A}$ with runtime t and making at most q_H (resp. q_S) queries to the random oracle (resp. its signing oracle), the probability that the following experiment returns 1 is at most ε.*

```
pick random function $H$
$(sk, pk) \xleftarrow{\$} \mathsf{SKGen}^H(1^\lambda)$
$(m^*, \sigma^*) \xleftarrow{\$} \mathcal{A}^{H, \mathsf{Sig}^H(sk,\cdot)}(pk)$
Return $1$ iff $\mathsf{SVf}^H(pk, m^*, \sigma^*) = 1$ and $m^* \notin \mathsf{M}$.
    Here, $\mathsf{M}$ is the set of message queried to $\mathsf{Sig}^H(sk, \cdot)$.
```

The probability is taken over all coin tosses of SKGen, Sig, and $\mathcal{A}$, and the choice of H.

We call a signature scheme *existentially unforgeable under chosen message attacks* in the (quantum) random-oracle model if for any PPT (quantum) algorithm making at most polynomial number of (superposition) queries to the (quantum) random oracle and classical queries to the signature scheme, the probability for the above experiment is negligible in the security parameter.

3 Impossibility Result for Quantum-Fiat-Shamir

We use meta-reductions techniques to show that, if the Fiat-Shamir transformation applied to the identification protocol would support a knowledge extractor, then we would obtain a contradiction to the active security. That is, we first build an all-powerful quantum adversary $\mathcal{A}_{\mathrm{Q}}$ successfully generating accepted proofs. Coming up with such an adversary is necessary to ensure that a black-box extractor $\mathcal{K}_{\mathrm{Q}}$ exists in the first place; Definition 2.2 only requires $\mathcal{K}_{\mathrm{Q}}$ to succeed *if* there is some successful adversary $\mathcal{A}_{\mathrm{Q}}$. The adversary $\mathcal{A}_{\mathrm{Q}}$ uses its unbounded power to find a witness w to its input x, and then uses the quantum access to the random oracle model to "hide" its actual query in a superposition. The former ensures that that our adversary is trivially able to construct a valid proof by emulating the prover for w, the latter prevents the extractor to apply the rewinding techniques of Pointcheval and Stern [PS00] in the classical setting. Once we have designed our adversary $\mathcal{A}_{\mathrm{Q}}$ and ensured the existence of $\mathcal{K}_{\mathrm{Q}}$, we wrap $\mathcal{K}_{\mathrm{Q}}$ into a reduction $\mathcal{M}_{\mathrm{Q}}$ which takes the role of $\mathcal{A}_{\mathrm{Q}}$ and breaks active security. The (quantum) meta-reduction now plays against the honest prover of the identification scheme "on the outside", using the extractor "on the inside". In this inner interaction $\mathcal{M}_{\mathrm{Q}}$ needs to emulate our all-powerful adversary $\mathcal{A}_{\mathrm{Q}}$ towards the extractor, but this needs to be done efficiently in order to make sure that the meta-reduction (with its inner interactions) is efficient.

In the argument below we assume that the extractor is input-respecting (i.e., forwards x faithfully to the adversary). In this case we can easily derandomize the adversary (with respect to classical randomness) by "hardwiring" a key of a random function into it, which it initially applies to its input x to recover the same classical randomness for each run. Since the extractor has to work for all adversaries, it in particular needs to succeed for those where we pick the function randomly but fix it from thereon.

3.1 Prerequisites

Witness-Independent Commitments. We first identify a special subclass of Σ-protocols which our result relies upon:

Definition 3.1 (Σ-protocols with witness-independent commitment) *A Σ-protocol has* witness-independent commitments *if the prover's commitment* com *does not depend on the witness w. That is, we assume that there is a PPT algorithm* Com *which, on input x and some randomness r, produces the same distribution as the prover's first message for input (x, w).*

Examples of such Σ-protocols are the well known graph-isomorphism proof [GMW87], the Schnorr proof of knowledge [Sch91], or the recent protocol for lattices used in an anonymous credential system [CNR12]. A typical example of non-witness-independent commitment Σ-protocol is the graph 3-coloring ZKPoK scheme [GMW87] where the prover commits to a random permutation of the coloring.

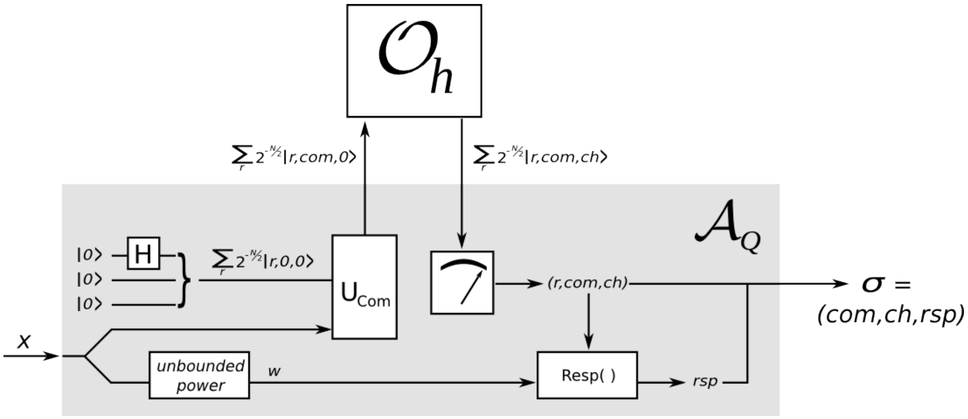


Figure 2: The canonical adversary

We note that perfectly hiding commitments do not suffice for our negative result. We need to be able to generate (the superposition of) all commitments without knowledge of the witness.

Weak Security Against Active Quantum Adversaries. We next describe the underlying security of (non-transformed) Σ-protocols against a weak form of active attacks where the adversary may use quantum power but needs to eventually compute a witness. That is, we let $\mathcal{A}_{\mathrm{Q}}^{\mathcal{P}(x,w)}(x)$ be a quantum adversary which can interact classically with several prover instances. The prover instances can be invoked in sequential order, each time the prover starts by computing a fresh commitment $\mathsf{com} \leftarrow \mathcal{P}(x,w)$, and upon receiving a challenge $\mathsf{ch} \in \{0,1\}^\ell$ it computes the response rsp. Only if it has returned this response $\mathcal{P}$ can be invoked on a new session again. We say that the adversary *succeeds in an active attack* if it eventually returns some w' such that $(x,w') \in \mathcal{R}$.

For an interesting security notion computing a witness from x only should be infeasible, even for a quantum adversary. To this end we assume that there is an efficient instance generator Inst which on input 1^λ outputs a pair $(x,w) \in \mathcal{R}$ such that any polynomial-time quantum algorithm on (classical) input x returns some classical string w' with $(x,w') \in \mathcal{R}$, is negligible (over the random choices of Inst and the quantum algorithm). We say Inst is a *hard instance generator for relation* $\mathcal{R}$.

Definition 3.2 (Weakly Secure Σ-Protocol Against Active Quantum Adversaries) *A Σ-protocol $(\mathcal{P},\mathcal{V})$ for an $\mathcal{NP}$-relation $\mathcal{R}$ with hard instance generator Inst is weakly secure against active quantum adversaries if for any polynomial-time quantum adversaries $\mathcal{A}_Q$ the probability that $\mathcal{A}_Q^{\mathcal{P}(x,w)}(x)$ succeeds in an active attack for $(x,w) \leftarrow \mathsf{Inst}(1^\lambda)$ is negligible (as a function of λ).*

We call this property weak security because it demands the adversary to compute a witness w', instead of passing only an impersonation attempt. If the adversary finds such a witness, then completeness of the scheme implies that it can successfully impersonate. In this sense we put more restrictions on the adversary and, thus, weaken the security guarantees.

3.2 The Adversary and the Meta-Reduction

Adversary. Our (unbounded) adversary works roughly as follows (see Figure 2). It receives as input a value x and first uses its unbounded computational power to compute a random witness

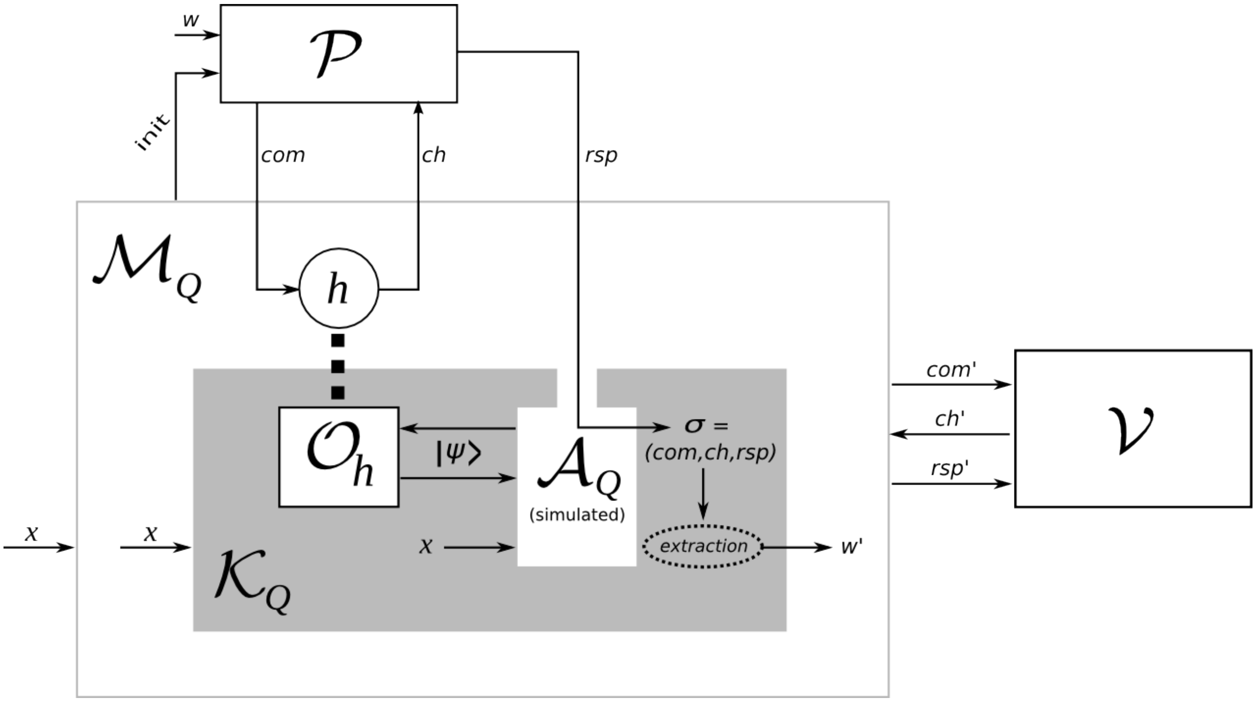


Figure 3: An overview of our meta-reduction

w (according to uniform distributions of coin tosses ω subject to $\mathsf{Inst}(1^n;\omega) = (x, w)$, but where ω is a random function of x). Then it prepares all possible random strings $r \in \{0,1\}^N$ (where $N = \mathrm{poly}(n)$) for the prover's algorithm in superposition. It then evaluates (a unitary version of) the classical function Com() for computing the prover's commitment on this superposition (and on x) to get a superposition of all $|r\rangle\,|\mathsf{com}_{x,r}\rangle$. It evaluates the random oracle H on the com-part, i.e., to be precise, the hash values are stored in ancilla bits such that the result is a superposition of states $|r\rangle\,|\mathsf{com}_{x,r}\rangle\,|H(x, \mathsf{com}_{x,r})\rangle$. The adversary measures in the computational basis, yielding a sample $(r, \mathsf{com}_{x,r}, \mathsf{ch})$ for $\mathsf{ch} = H(x, \mathsf{com}_{x,r})$ where r is uniform over all random strings. Finally, the adversary completes the protocol by computing a response $\mathsf{rsp}_{x,w,r}$ for x, w, and r; it outputs the transcript $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$.

The Meta-Reduction. We illustrate the meta-reduction in Figure 3. Assume that there exists a (quantum) black-box extractor $\mathcal{K}_{\mathrm{Q}}$ which on input x, sampled according to Inst, and which is also given to $\mathcal{A}_{\mathrm{Q}}$, is able to extract a witness w to x by running several resetting executions of $\mathcal{A}_{\mathrm{Q}}$, each time answering $\mathcal{A}_{\mathrm{Q}}$'s (only) random oracle query $|\varphi\rangle$ by supplying a classical, possibly probabilistic function h. We then build a (quantum) meta-reduction $\mathcal{M}_{\mathrm{Q}}$ which breaks the weak security of the identification scheme in an active attack when communicating with the classical prover.

The quantum meta-reduction $\mathcal{M}_{\mathrm{Q}}$ receives as input the public statement x. It forwards it to $\mathcal{K}_{\mathrm{Q}}$ and waits until $\mathcal{K}_{\mathrm{Q}}$ invokes $\mathcal{A}_{\mathrm{Q}}(x)$, which is now simulated by $\mathcal{M}_{\mathrm{Q}}$. For each (reset) execution the meta-reduction skips the step where the adversary would compute the witness, and instead immediately computes the same superposition query $|r\rangle\,|\mathsf{com}_{x,r}\rangle$ as $\mathcal{A}_{\mathrm{Q}}$ and outputs it to $\mathcal{K}_{\mathrm{Q}}$. When $\mathcal{K}_{\mathrm{Q}}$ creates (a description of) the possibly probabilistic function h we let $\mathcal{M}_{\mathrm{Q}}$ initiate an interaction with the prover to receive a classical sample $\mathsf{com}_{x,r}$, on which it evaluates h to get a challenge ch. Note that $\mathcal{M}_{\mathrm{Q}}$ in principle does not need a description of h for this, but only a possibility to compute h once. The meta-reduction forwards the challenge to the prover to get a response rsp. It outputs $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ to the reduction. If the reduction eventually outputs a potential witness w' then $\mathcal{M}_{\mathrm{Q}}$ uses this value w' to break the weak security.

3.3 Analysis

For the analysis note that the extractor's perspective in each execution is identical in both cases, when interacting with the actual adversary $\mathcal{A}_{\mathrm{Q}}$, or when interacting with the meta-reduction $\mathcal{M}_{\mathrm{Q}}$. The reason is that the commitments are witness-independent such that the adversary (using its computational power to first compute a witness) and the meta-reduction computing the commitments without knowledge of a witness, create the same distribution on the query to the random oracle. Since up to this point the extractor's view is identical in both runs, its distribution on h is also the same in both cases. But then the quantum adversary internally computes, in superposition over all possible random strings r, the challenge $\mathsf{ch} \leftarrow h(x, \mathsf{com}_{x,r})$ and the response $\mathsf{rsp}_{x,w,r}$ for x, w, and ch. It then measures r in the computational basis, such that the state collapses to a classical tuple $(\mathsf{com}_{x,r}, \mathsf{ch}, \mathsf{rsp}_{x,w,r})$ over uniformly distributed r. Analogously, the meta-reduction, upon receiving h (with the same distribution as in $\mathcal{A}_{\mathrm{Q}}$'s attack), receives from the prover a commitment $\mathsf{com}_{x,r}$ for a uniformly distributed r. It then computes $\mathsf{ch} \leftarrow h(x, \mathsf{com}_{x,r})$ and obtains $\mathsf{rsp}_{x,w,r}$ from the prover, which is determined by x, w, r and ch. It returns $(\mathsf{com}_{x,r}, \mathsf{ch}, \mathsf{rsp}_{x,w,r})$ for such a uniform r.

In other words, $\mathcal{M}_{\mathrm{Q}}$ considers only a single classical execution (with r sampled at the outset), whereas $\mathcal{A}_{\mathrm{Q}}$ basically first runs everything in superposition and only samples r at the very end. Since all the other computations in between are classical, the final results are identically distributed. Furthermore, since the extractor is input-respecting, the meta-reduction can indeed answer all runs for the very same x with the help of the external prover (which only works for x). Analogously, the fact that the adversary always chooses, and uses, the same witness w in all runs, implies that the meta-reduction can again rely on the external prover with the single witness w.

Since the all-powerful adversary succeeds with probability 1 in the original experiment, to output a valid proof given x and access to a quantum random oracle only, the extractor must also succeed with non-negligible probability in extracting a witness. Hence, $\mathcal{M}_{\mathrm{Q}}$, too, succeeds with non-negligible probability in an active attack against weak security. Furthermore, since $\mathcal{K}_{\mathrm{Q}}$ runs in polynomial time, $\mathcal{M}_{\mathrm{Q}}$ invokes at most a polynomial number of interactions with the external prover. Altogether, we thus obtain the following theorem:

Theorem 3.3 (Impossibility Result) *For any Σ-protocol $(\mathcal{P}, \mathcal{V})$ with witness-independent commitments, and which is weakly secure against active quantum adversaries, there does not exist an input-preserving black-box quantum knowledge extractor for $(\mathcal{P}, \mathcal{V})$.*

We note that our impossibility result is cast in terms of proofs of knowledge, but can be easily adapted for the case of signatures. In fact, the adversary $\mathcal{A}_{\mathrm{Q}}$ would be able to compute a valid proof (i.e., a signature) for any given message m which it receives as additional input to x.

Our Meta-Reduction and Classical Queries to the Random Oracle. One might ask why the meta-reduction does not apply to the Fiat-Shamir transform when adversaries have only classical access to the random oracle. The reason is the following: if the adversary made a classical query about a single commitment (and so would the meta-reduction), then one could apply the rewinding technique of Pointcheval and Stern [PS00] changing the random oracle answers, and extract the underlying witness via special soundness of the identification scheme. The quantum adversary here, however, queries the random oracle in a superposition. In this scenario, as we explained above, the extractor is not allowed to "read" the query of the adversary unless it makes the adversary stop. In other words, the extractor cannot measure the query and then keep running the adversary until a valid witness is output. This intrinsic property of black-box quantum extractors, hence, makes "quantum" rewinding impossible. Note that rewinding in the classical sense —as described by Pointcheval and Stern [PS00]— is still possible, as this essentially means to start the adversary with the same random coins. One may argue that it might be possible to measure the query state without disturbing $\mathcal{A}_{\mathrm{Q}}$'s behavior significantly, but as we already pointed out, this would lead to a non-black-box approach —vastly more powerful than the classical read-only access.

3.4 On the Necessity of Active Security

We briefly discuss that active security is basically necessary for an impossibility result as above. That is, we outline a three-move protocol for any $\mathcal{NP}$ language which, when applying the FS transformation supports a straight-line extractor, and is honest-verifier zero-knowledge, but not actively secure. This holds as long as there are quantum-immune dense encryption, and quantum-immune non-interactive zero-knowledge proofs. The latter are classical non-interactive zero-knowledge proofs (in the common random string model) for which simulated and genuine proofs are indistinguishable, even for *quantum* distinguishers. The former are encryption schemes which are IND-CPA against quantum adversaries (see, for example, [BDF$^+$11]) but where, in addition, honestly generated public keys are quantum-indistinguishable from random strings.

The construction is based on the (classical) non-interactive zero-knowledge proofs of knowledge of De Santis and Persiano [DP92] and works as follows: The first message is irrelevant, e.g., we let the prover simply send the constant 0 (potentially padded with redundant randomness). In the second message the verifier sends a random string which the prover interprets as a public key *pk* of the dense encryption scheme and a common random string crs for the NIZK. The prover encrypts the witness under *pk* and gives a NIZK that the encrypted value forms a valid witness for the public value x. The verifier only checks the NIZK proof.

The protocol is clearly not secure against active (classical) adversaries because such an adversary can create a public key *pk* via the key generation algorithm, thus, knowing the secret key and allowing the adversary to recover the witness from a proof by the prover. It is, however, honest-verifier zero-knowledge against quantum distinguishers because of the IND-CPA security and the simulatability of the NIZK hide the witness and allow for a simulation. We omit a more formal argument here, as it will be covered as a special case from our general result in the next section.

4 Positive Results for Quantum-Fiat-Shamir

In Section 3.4 we have sketched a generic construction of a Σ-protocol based on NIZKPoKs [DP92] which can be converted to a secure NIZK-PoK against quantum adversaries in the QROM via the Fiat-Shamir (FS) paradigm. While the construction is rather inefficient and relies on additional primitives and assumptions, it shows the path to a rather efficient solution: drop the requirement on active security and let the (honest) verifier choose the commitment obliviously, i.e., such that it does not know the pre-image, together with the challenge. If the prover is able to use a trapdoor to compute the commitment's pre-image then it can complete the protocol as before.

4.1 Σ-protocols with Oblivious Commitments

The following definition captures the notion of Σ-protocols with oblivious commitments formally.

Definition 4.1 (Σ-protocols with Oblivious Commitments) *A Σ-protocol $(\mathcal{P}, \mathcal{V})$ has* oblivious commitments *if there are PPT algorithms* $\textsc{Com}$ *and* $\textsc{SmplRnd}$ *such that for any $(x, w) \in \mathcal{R}$ the following distributions are statistically close:*

- *Let* $\mathsf{com} = \textsc{Com}(x; \rho)$ *for* $\rho \leftarrow \{0,1\}^{\lambda}$, $\mathsf{ch} \leftarrow \mathcal{V}(x, \mathsf{com})$, *and* $\mathsf{rsp} \leftarrow \mathcal{P}(x, w, \mathsf{com}, \mathsf{ch})$. *Output* $(x, w, \rho, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$.
- *Let* $(x, w, \rho, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ *be a transcript of a protocol run between* $\mathcal{P}(x, w)$ *and* $\mathcal{V}(x)$, *where* $\rho \leftarrow \textsc{SmplRnd}(x, \mathsf{com})$.

Note that the prover is able to compute a response from the given commitment com without knowing the randomness used to compute the commitment. This is usually achieved by placing some extra trapdoor into the witness w. For example, for the Guillou-Quisquater RSA based proof of knowledge [GQ90] where the prover shows knowledge of $w \in \mathbb{Z}_N^*$ with $w^e = y \bmod N$ for $x = (e, N, y)$, the prover would need to compute an e-th root for a given commitment $R \in \mathbb{Z}_N^*$. If the witness would contain the prime factorization of N, instead of the e-th root of y, this would indeed be possible.

Σ-protocols with oblivious commitments allow to move the generation of the commitment from the prover to the honest verifier. For most schemes this infringes with active security, because a malicious verifier could generate the commitment "non-obliviously". However, the scheme remains honest-verifier zero-knowledge, and this suffices for deriving secure signature schemes. In particular, using random oracles one can hash into commitments by computing the random output of the hash function and running $\textsc{Com}(x; \rho)$ on this random string ρ to sample a commitment obliviously.

In the sequel we therefore often identify ρ with $\textsc{Com}(x; \rho)$ in the sense that we assume that the hash function maps to $\textsc{Com}(x; \rho)$ directly. The existence of $\textsc{SmplRnd}$ guarantees that we could "bend" this value back to the actual pre-image ρ. In fact, for our positive result it would suffice that the distributions are computationally indistinguishable for random $(x, w) \leftarrow \mathsf{Inst}(1^n)$ against quantum distinguishers.

4.2 FS Transformation for Σ-protocols with Oblivious Commitments

We explain the FS transformation for schemes with oblivious commitments for signatures only; the case of (simulation-sound) NIZK-PoKs is similar, the difference is that for signatures the message is included in the hash computation for signature schemes. For sake of concreteness let us give the full description of the transformed signature scheme. We note that for the transformation we also include a random string r in the hash computation (chosen by the signer). Jumping ahead, we note that this source of entropy ensures simulatability of signatures; for classical Σ-protocols this is usually given by the entropy of the initial commitment but which has been moved to the verifier here. Recall from the previous section that we simply assume that we can hash into commitments directly, instead of going through the mapping via $\textsc{Com}$ and $\textsc{SmplRnd}$.

Construction 4.2 *Let $(\mathcal{P},\mathcal{V})$ be a Σ-protocol for relation $\mathcal{R}$ with oblivious commitments and instance generator* Inst. *Then construct the following signature scheme $\mathcal{S} = (\mathit{SKGen}, \mathit{Sig}, \mathit{SVf})$ in the (quantum) random-oracle model:*

Key Generation. *$\mathit{SKGen}(1^\lambda)$ runs $(x,w) \leftarrow \mathsf{Inst}(1^\lambda)$ and returns $sk = (x,w)$ and $pk = x$.*

Signing. *For message $m \in \{0,1\}^*$ the signing algorithm Sig^H on input sk, picks random $r \xleftarrow{\$}$* Rnd *from some superpolynomial space, computes $(\mathsf{com},\mathsf{ch}) = H(pk,m,r)$, and obtains $\mathsf{rsp} \leftarrow \mathcal{P}(pk, sk, \mathsf{com}, \mathsf{ch})$. The output is the signature $\sigma = (r, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$.*

Verification. *On input pk,m, and $\sigma = (r, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ the verification algorithm Vf^H outputs 1 iff $\mathcal{V}(pk, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) = 1$ and $(\mathsf{com},\mathsf{ch}) = H(pk,m,r)$; else, it returns 0.*

Note that one can shorten the signature size by simply outputting $\sigma = (r, \mathsf{rsp})$. The remaining components $(\mathsf{com},\mathsf{ch})$ are obtained by hashing the tuple (pk,m,r). Next, we give the main result of this section saying that the Fiat-Shamir transform on Σ-protocols with oblivious commitments yield a quantum-secure signature scheme.

Theorem 4.3 *If* Inst *is a hard instance generator for the relation $\mathcal{R}$ and the Σ-protocol $(\mathcal{P},\mathcal{V})$ has oblivious commitments, then the signature scheme in Construction 4.2 is existentially unforgeable under chosen message attacks against quantum adversaries in the quantum-accessible random-oracle model.*

The idea is roughly as follows. Assume for the moment that we are only interested in key-only attacks and would like to extract the secret key from an adversary $\mathcal{A}_\mathrm{Q}$ against the signature scheme. For given x we first run the honest-verifier zero-knowledge simulator of the Σ-protocol to create a transcript $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$. We choose another random challenge $\mathsf{ch}' \leftarrow \{0,1\}^\ell$. Then, we run the adversary, injecting $(\mathsf{com}^\star, \mathsf{ch}')$ into the hash replies. This appropriate insertion will be based on techniques developed by Zhandry [Zha12b] to make sure that superposition queries to the random oracle are harmless. With sufficiently large probability the adversary will then output a proof $(\mathsf{com}^\star, \mathsf{ch}', \mathsf{rsp}')$ from which we can, together with $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$ extract a witness due to the special-soundness property. Note that, if this extraction fails because the transcript $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$ is only simulated, we could distinguish simulated signatures from genuine ones. We can extend this argument to chosen-message attacks by simulating signatures as in the classical case. This is the step where we take advantage of the extra random string r in order to make sure that the previous adversary's quantum hash queries have a negligible amplitude in this value (x,m,r). Using techniques from [BBBV97] we can show that changing the oracle in this case does not change the adversary's success probability significantly.

4.3 Technical Results for the Security Proof

We start by recalling two results from Bernstein and Vazirani [BV97] and Bennett et al. [BBBV97] which we make use of in the proof of Theorem 4.3. Before so, we introduce distance measures.

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 [BV97] upperbounds the total variance distance in terms of the Euclidean distance:

Lemma 4.4 ([BV97, Lemma 3.6]) *Let $|\varphi\rangle, |\psi\rangle$ be quantum states with Euclidean distance at most ϵ. Then, performing the same measurement on $|\varphi\rangle, |\psi\rangle$ yields distributions with statistical distance at most 4ϵ.*

Let $q_\rho(|\phi_t\rangle)$ be the magnitude squared of ρ in the superposition of query t which we call the query probability of r in query t. If we sum over all queries t, we get an upper bound on the total query probability of r. The following is a result from Bennett et al [BBBV97].

Lemma 4.5 ([BBBV97, Theorem 3.3]) *Let $\mathcal{A}_Q$ be a quantum algorithm running in time T with oracle access to H. Let $\epsilon > 0$ and let $S \subseteq [1, T] \times \{0, 1\}^n$ be a set of time-string pairs such that $\sum_{(t,\rho)\in S} q_\rho(|\phi_t\rangle) \leq \epsilon$. If we modify H into an oracle H' which answers each query ρ at time t by providing the same string R (which has been sampled independently form H), then the Euclidean distance between the final states of $\mathcal{A}_Q$ when invoking H and H' is at most $\sqrt{T\epsilon}$.*

Injecting Values into Oracles. Let us now introduce some definitions and results including so-called semi-constant distributions SC_δ introduced by Zhandry [Zha12b].

Definition 4.6 (Semi-Constant Distributions) *Let $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}} = \{ H : \mathcal{X} \to \mathcal{Y} \}$ be a family of functions for sets $\mathcal{X}$ and $\mathcal{Y}$ and let $\delta \in [0, 1]$. We define the* semi-constant distribution *SC_δ as the distribution over $\mathcal{H}_{\mathcal{X}\times\mathcal{Y}}$ resulting from the following process:*

- *first, pick a random element $y \in \mathcal{Y}$;*
- *then, for each $x \in \mathcal{X}$ do the following:*
  - *with probability δ, set $H(x) = y$;*
  - *otherwise, set $H(x)$ to be a (uniformly) randomly chosen element in $\mathcal{Y}$.*

Notice that SC_0 is the uniform distribution, while SC_1 is a constant distribution. Also note that the distribution, when used within an oracle, is consistent in the sense that the settings are chosen once at the outset. We will use this definition to describe a quantum random oracle which has been "reprogrammed" on a fraction δ of its possible inputs.

The following lemma by Zhandry [Zha12b] gives an upper bound on the probability that a quantum algorithm's output behavior changes when switching from a truly random oracle to an oracle drawn from SC_δ in terms of statistical distance:

Lemma 4.7 ([Zha12b, Corollary 4.3]) *Let $\mathcal{A}_Q^{|H\rangle}$ be a quantum algorithm making at most q_H queries to the quantum-accessible random oracle H. Let $\delta \in (0, 1)$ and let H' be the oracle obtained by reprogramming H on a fraction δ of its possible inputs, i.e., let H' be described by distribution SC_δ. Then,*

$$\left| \mathcal{A}_Q^{|H\rangle} - \mathcal{A}_Q^{|H'\rangle} \right| \leq \frac{8}{3} \cdot q_H^4 \delta^2 \, .$$

Recall our quantum adversary $\mathcal{A}_\mathrm{Q}$ against the unforgeability property of the signature scheme from Construction 4.2. It works by performing at most $q_H = \mathrm{poly}(\lambda)$ queries to the quantum-accessible random oracle. This means that the statistical distance in the two cases, and in particular the probability ϵ' that $\mathcal{A}_{\mathrm{Q}}^{|H'\rangle}$ successfully forges, is at least $\epsilon - \frac{8}{3} \cdot q_H^4 \delta^2$. Hence, we can make the probabilities arbitrarily small while still keeping δ noticeable (in the order of q_H^{-2}). This is important in order to extract the secret key successfully. Specifically, the following two (seemingly contradictory) conditions have to be fulfilled:

- We need to ensure that $\mathcal{A}_{\mathrm{Q}}$ eventually outputs a valid signature $(r, \mathsf{com}^\star, \mathsf{ch}', \mathsf{rsp}')$ for some message m for the commitment $\mathsf{com}^\star$ of our choice (the one we obtained from the zero-knowledge simulator of the Σ-protocol which we inject into H's responses). This requires that $\mathsf{com}^\star$ appears with sufficiently large probability in the responses for oracle queries.

- Secondly, we still require that $\mathcal{A}_{\mathrm{Q}}$ has a small probability of distinguishing a true random oracle H from the re-programmed one. Otherwise, the adversary may refuse to give a valid signature at all.

The following lemma shows that both conditions can be satisfied simultaneously.

Lemma 4.8 *Let $\mathcal{A}_Q^{|H\rangle}$ as in Lemma 4.7, and let H' be the oracle obtained by reprogramming H on a fraction δ of its possible inputs (pk, m, r) such that $H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}')$ for values $\mathsf{com}^\star$ and ch'. Let m, $\sigma = (r, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ be the output of $\mathcal{A}_Q^{|H'\rangle}$ on input pk. Then,*

$$\Pr\left[\mathit{Vf}^{H'}(pk, m, \sigma) = 1 \ \wedge\ (\mathsf{com}, \mathsf{ch}) = H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}') \right] \geq \delta \cdot \epsilon - \frac{8}{3} \cdot q_H^4 \delta^2 \ .$$

Proof. Consider the probability that we first run the adversary on the original oracle H and check if it successfully forges a signature for message m, and then we also verify that its output (pk, m, r) is thrown to $(\mathsf{com}^\star, \mathsf{ch}')$ under H'. We claim that

$$\Pr\left[\mathsf{Vf}^H(pk, m, \sigma) = 1 \ \wedge\ H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}') \right] \geq \epsilon \cdot \delta.$$

This follows from the independence of the events: the oracle H' re-programs the output with probability δ, independently of $\mathcal{A}$'s behavior when interacting with oracle H. Next, we argue that

$$\Pr\left[\mathsf{Vf}^{H'}(pk, m, \sigma) = 1 \ \wedge\ H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}') \right] \geq \delta \cdot \epsilon - \frac{8}{3} \cdot q_H^4 \delta^2.$$

Note that the difference is now that the adversary interacts with oracle H', and that we also verify the adversary's success with respect to H'. Instructively, the reader may imagine that, after the adversary's attack ends, we also check that $H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}')$; the equation $H'(pk, m, r) = (\mathsf{com}, \mathsf{ch})$, as in the lemma's claim, trivially follows already if verification holds.

According to the previous lemma, switching to oracle H' can change the distance of the output distribution of $\mathcal{A}$ when playing against H' instead of H (including the final verification) by at most $\frac{8}{3} \cdot q_H^4 \delta^2$. Hence, since the subsequent computation and check $H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}')$ cannot increase this distance, we conclude that the probability for event

$$\mathsf{Vf}^{H'}(pk, m, \sigma) = 1 \ \wedge\ H'(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}') = (\mathsf{com}, \mathsf{ch})$$

cannot be smaller than the claimed bound. □

The previous lemma informally tell us that, in order to succeed, we have to balance between a large δ to increase the chances of the adversary outputting a signature containing our desired $\mathsf{com}^\star$, and a small δ to avoid that the adversary detects the reprogrammed oracle.

4.4 Security Proof

We are now ready to prove the main theorem.

Proof (of Theorem 4.3). We assume towards contradiction the existence of an efficient quantum adversary $\mathcal{A}_{\mathrm{Q}}$ which, on input a public key pk, outputs a valid forgery (m, σ) under pk with non-negligible probability ϵ, hence breaking the existential unforgeability of the signature scheme. This adversary has access to a quantum-accessible random oracle H with $H(pk, m_i, r_j) = (\mathsf{com}_{i,j}, \mathsf{ch}_{i,j})$, and to a signing oracle $\mathcal{S}$ for the key sk producing, on input a classical strings m, (classical) signatures $\sigma = (r, \mathsf{com}, \mathsf{ch}, \mathsf{rsp}) \leftarrow \mathsf{Sig}^H(sk, m)$.

The adversary $\mathcal{A}_{\mathrm{Q}}$ gets pk as an input, and is then allowed to perform up to $q_H = \mathrm{poly}(\lambda)$ queries to H in superposition, and up to $q_S = \mathrm{poly}(\lambda)$ classical queries to $\mathcal{S}$. Recall that the signer still operates on classical bits. Then, after running for $\mathrm{poly}(\lambda)$ time, adversary $\mathcal{A}_{\mathrm{Q}}$ produces (with probability ϵ) a valid forgery (m, σ) under pk such that m has never been asked to the signing oracle $\mathcal{S}$ throughout $\mathcal{A}_{\mathrm{Q}}$'s execution (i.e., m is a fresh message). We assume that q_H also covers a classical query of the verifier to check the signature.

Under these assumptions we show how to build an efficient quantum adversary $\mathcal{B}_{\mathrm{Q}}$, with access to $\mathcal{A}_{\mathrm{Q}}$ as a subroutine, and which is able to break the scheme's underlying hard mathematical problem with non-negligible probability. That is, $\mathcal{B}_{\mathrm{Q}}$ on input x generated according to $\mathsf{Inst}(1^\lambda)$, is able to output a valid witness w' to statement x, i.e., $(x, w') \in \mathcal{R}$. The adversary $\mathcal{B}_{\mathrm{Q}}$ works as follows:

- On input statement x, it first runs a simulation of the underlying Σ-protocol to obtain a valid transcript $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$. This is possible because of the honest-verifier zero-knowledge property. Note also that this does not require access to the random oracle. Also note that we assume for simplicity that the oblivious commitment is a random string; else we would need to run SmplRnd on $\mathsf{com}^\star$ now to derive ρ, and use ρ in the hash reply (and argue that this is indistinguishable).

- Then, $\mathcal{B}_{\mathrm{Q}}$ simulates an oracle H_0 which is obtained by reprogramming a (simulated) quantum random oracle H over a fraction δ of its possible inputs (pk, m, r) with the value $(\mathsf{com}^\star, \mathsf{ch}')$. Here, δ is some non-negligible probability in the security parameter, and ch' is a fix, arbitrarily chosen challenge different from $\mathsf{ch}^\star$. That is, $H_0(pk, m, r) = (\mathsf{com}^\star, \mathsf{ch}')$ with probability δ, and random elsewhere.

- Next, $\mathcal{B}_{\mathrm{Q}}$ invokes $\mathcal{A}_{\mathrm{Q}}$ on input $pk = x$.

- Whenever $\mathcal{A}_{\mathrm{Q}}$ performs the i-th query to $\mathcal{S}$ for signing a message m_i, adversary $\mathcal{B}_{\mathrm{Q}}$ does the following:
  - choose a random value $r_i \xleftarrow{\$} \textsc{Rnd}$;
  - execute the honest-verifier zero-knowledge simulator Sim of the identification scheme, obtaining a valid (simulated) transcript $(\mathsf{com}_i, \mathsf{ch}_i, \mathsf{rsp}_i)$;
  - reprogram H_{i-1} with value $(\mathsf{com}_i, \mathsf{ch}_i)$ for the input (pk, m_i, r_i). We denote by H_i the reprogrammed oracle after the i-th query to the signing oracle;
  - then output the signature $\sigma_i = (r_i, \mathsf{com}_i, \mathsf{ch}_i, \mathsf{rsp}_i)$ as $\mathcal{S}$'s reply to $\mathcal{A}_{\mathrm{Q}}$.

- Finally, when $\mathcal{A}_\mathrm{Q}$ outputs a (hopefully valid) forgery (m, σ), where $\sigma = (r, \mathsf{com}, \mathsf{ch}, \mathsf{rsp})$, algorithm $\mathcal{B}_\mathrm{Q}$ aborts if $\mathsf{com} \neq \mathsf{com}^\star$ or $\mathsf{ch} = \mathsf{ch}^\star$. Otherwise, it uses the special soundness extractor Ext of the underlying Σ-protocol on input $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$ and $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$ to obtain a valid witness w' for x.

Note that we can formally let $\mathcal{B}_\mathrm{Q}$ implement the hash evaluations by a classical algorithm with access to a random oracle, basically hardwiring all changes due to re-programming into the code of the algorithm. In a second step we can eliminate the random oracle, either via quantum-accessible pseudorandom functions [BDF+11], or without any assumptions by using q-wise independent function as shown in [Zha12b, Theorem 6.1]. These functions can be implemented by classical algorithms.

We next show that the success probability of our extraction procedure $\mathcal{B}_\mathrm{Q}$ is non-negligible given a successful $\mathcal{A}_\mathrm{Q}$. The proof follows the common game-hopping technique where we gradually deprive the adversary a (negligible amount) of its success probability. We start with $\textsc{Game}_1$ where the adversary attacks the original scheme.

Game$_1$. This is $\mathcal{A}_\mathrm{Q}$'s original attack on the signature scheme as constructed according to Construction 4.2 initialized by public key pk. By assumption we have

$$\Pr\left[\mathcal{A}_\mathrm{Q} \text{ wins } \textsc{Game}_1\right] = \epsilon$$

for some non-negligible value ϵ.

Game$_2$. This game is identical to $\textsc{Game}_1$, except that we abort if $\mathcal{A}_\mathrm{Q}$ outputs a valid forgery (m, σ) where σ *does not* contain the pre-selected commitment $\mathsf{com}^\star$ and challenge ch'. Furthermore, we replace the quantum-accessible random oracle H with the oracle H_0 drawn from a semi-constant distribution SC_δ. Recall that H_0 is obtained by reprogramming H on a fraction δ of its entries with the value $(\mathsf{com}^\star, \mathsf{ch}')$, where $\mathsf{com}^\star$ was obtained by a run of the honest-verifier zero-knowledge simulator Sim on input x and ch' was picked as in $\mathcal{B}_\mathrm{Q}$'s simulation. By Lemma 4.8 we have

$$\Pr\left[\mathcal{A}_\mathrm{Q} \text{ wins } \textsc{Game}_2\right] \geq \delta\epsilon - \frac{8}{3} q_H^4 \delta^2 \; .$$

Game$_3^{(1)}$. As $\textsc{Game}_2$, but this time H_0 is reprogrammed to H_1 (on the single point (pk, m_1, r_1)) as soon as $\mathcal{A}_\mathrm{Q}$ performs its 1^{st} classical query m_1 to $\mathcal{S}$. From then on, the oracle H_1 always answers consistently with this value. We need to show that this switching does not change the winning probability significantly. For this we basically need to show that, so far, the amplitudes of this value (pk, m_1, r_1) in the queries to the quantum oracle are small, else the adversary may be able to spot some inconsistency.

Let $|\textsc{Rnd}| = 2^n = \exp(\lambda)$. We define the value (pk, m_i', r_j') to have *high amplitude* if there exists at least one of the quantum queries $|\phi_1\rangle, |\phi_2\rangle, \ldots$ to the quantum-accessible oracle H_0 *before the signing query*, where the amplitude $\alpha_{i,j}$ associated to the corresponding basis element is such that $|\alpha_{i,j}|^2 \geq 2^{\frac{-n}{2}}$. Otherwise, the tuple is said to have *low amplitude*. Note that each query to the quantum oracle can have at most $2^{\frac{n}{2}}$ tuples with high amplitude, because the (square of the) amplitudes need to sum up to 1.

When H_0 is reprogrammed to H_1, the choice of m_1 is fixed (i.e., determined by the 1^{st} query of $\mathcal{A}_\mathrm{Q}$ to S), but r_1 is still chosen uniformly at random in $\textsc{Rnd}$. Since $\mathcal{A}_\mathrm{Q}$ performs at most q_H queries to the quantum-accessible oracle according to H_0 before the signing query, we have thus at most $q_H \cdot 2^{\frac{n}{2}}$ tuples with high amplitude before this query. The probability of hitting such a tuple is then given by:

$$\Pr\left[(pk, m_1, r_1) \text{ has high amplitude}\right] \leq q_H \cdot 2^{\frac{-n}{2}}. \tag{1}$$

Moreover, provided (pk, m_1, r_1) has *low* amplitude, and since there are at most $q_H + q_S$ query steps, using Lemma 4.4 and Lemma 4.5 we obtain:

$$\left| \mathcal{A}_{\mathrm{Q}}^{|H_0\rangle} - \mathcal{A}_{\mathrm{Q}}^{|H_1\rangle} \right| \leq 4\sqrt{(q_H + q_S) \cdot 2^{\frac{-n}{2}}}. \tag{2}$$

Let us assume, on behalf of the adversary, that $\mathcal{A}_{\mathrm{Q}}$ fails whenever (pk, m_1, r_1) has high amplitude. Still, from equations (1) and (2), we have:

$$\begin{aligned} \Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_3^{(1)}\right] &\geq \Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_2\right] - 4\sqrt{(q_H + q_S) \cdot 2^{\frac{-n}{2}}} - q_H \cdot 2^{\frac{-n}{2}} \\ &= \delta\epsilon - \frac{8}{3} q_H^4 \delta^2 - \mathrm{negl}(\lambda) \ . \end{aligned}$$

Here, we use the fact that reprogramming the oracle for (pk, m_1, r_1) does not change the adversary's success probability for a forgery *for a fresh message m*. That is, since the adversary's forgery is for $m \neq m_1, m_2, \ldots$ it cannot simply copy a signature query as a forgery, but must still forge on the original oracle H_0. Hence the argument about the winning probability applies as it did for H_0.

We now repeat at most q_S times the game hopping, from $\textsc{Game}_3^{(1)}$ to $\textsc{Game}_3^{(q_S)}$, every time repeating the previous game but switching from H_{i-1} to H_i during the i^{th} query to $\mathcal{S}$, each time losing at most a negligible factor in the winning probability. Note that the probability of hitting a high amplitude with the signature generation in each hop increases to at most $q_H \cdot 2^{\frac{-n}{2}} + q_S \cdot 2^{-n}$ when taking into account the at most q_S hash queries in the previous signature requests, but this remains negligible. After q_S steps we reach the following game.

Game$_3^{(q_S)}$. As $\textsc{Game}_2$, but now H_0 is dynamically reprogrammed as a sequence $H_1, \ldots, H_{q_S}$ throughout all of the $\mathcal{A}_{\mathrm{Q}}$'s queries to $\mathcal{S}$. We have

$$\Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_3^{(q_S)}\right] \geq \delta\epsilon - \frac{8}{3} q_H^4 \delta^2 - \mathrm{negl}(\lambda) \ .$$

Game$_4$. As before, but now $\mathcal{S}$ is just simulated through the zero-knowledge simulator Sim of the underlying Σ-protocol. If, by contradiction, $\mathcal{A}_{\mathrm{Q}}$'s winning probability is affected by more than a negligible amount in so doing, then we could use $\mathcal{A}_{\mathrm{Q}}$ to build an efficient distinguisher between 'real' and 'simulated' transcripts of the Σ-protocol. This would require a distinguisher with access to a random oracle, in order to simulate the game. According to [Zha12b, Theorem 6.1], however, we can simulate the oracle via q-wise independent functions (which exists without requiring cryptographic assumptions). Furthermore, a hybrid argument can be used to reduce the case of q_S proofs to a single proof.

$$\Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_4\right] \geq \delta\epsilon - \frac{8}{3} q_H^4 \delta^2 - \mathrm{negl}(\lambda) \ .$$

Game$_5$. Finally, in this game the special soundness extractor Ext is run on the transcript obtained from $\mathcal{A}_{\mathrm{Q}}$'s output from the previous game. Change the winning condition of $\mathcal{A}_{\mathrm{Q}}$ such that the adversary wins if this extraction yields a valid witness w' for x. If the winning probability in this game is more than negligibly far from the winning probability of $\mathcal{A}_{\mathrm{Q}}$ in the previous game then this can only be due to the fact that the simulated proof with $(\mathsf{com}^\star, \mathsf{ch}^\star, \mathsf{rsp}^\star)$ cannot be accepted by the verifier; else the extractor would be be guaranteed to work for this proof and the (accepted) signature. But this would allow an easy distinguisher against the zero-knowledge property, similar to the previous games. Hence:

$$\Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_5\right] \geq \delta\epsilon - \frac{8}{3} q_H^4 \delta^2 - \mathrm{negl}(\lambda) \ .$$

Note that $\mathcal{A}_{\mathrm{Q}}$'s winning condition in the final game corresponds exactly to the probability of $\mathcal{B}_{\mathrm{Q}}$ successfully deriving a witness w' for its input x. This winning probability can be maximized (by zeroing the first derivative in δ) by choosing:

$$\delta = \frac{3\epsilon}{16 q_H^4} \ .$$

This yields:

$$\Pr\left[\mathcal{A}_{\mathrm{Q}} \text{ wins } \textsc{Game}_5\right] \geq \frac{3\epsilon^2}{16 q_H^4} - \mathrm{negl}(\lambda) \ ,$$

which is non-negligible. This concludes the proof of the main theorem. □

4.5 Example Instantiation

In this subsection, we present an instantiation of Σ-protocols with oblivious commitments which is secure against quantum adversaries. We look at the lattice-based signature scheme by Lyubashevsky [Lyu12] which is obtained by applying the FS transformation. The security of this signature scheme is reduced to the hardness of the Small Integer Solution (SIS) problem, which is believed to be hard even for quantum algorithms.

Similarly, other works using the FS transformation and relying on the quantum hardness of the underlying primitives, are not known to be necessarily secure against quantum adversaries, e.g., [Lyu09, BM10, GKV10, SSH11, Sak12, GLP12, AFLT12, CNR12, AJLA$^+$12]. This holds also for signature schemes obtained from the FS extension [ADV$^+$12] for multi-pass identification protocols (e.g., [CLRS10, CVA10, SSH11, Sak12]). Furthermore, the FS transform can be applied on the identification and zero-knowledge protocols [MV03, Lyu08, KTX08, MGS11] as they are secure against quantum adversaries. Still, the converted signature scheme is not necessarily quantum-secure anymore. A similar patch approach, as we describe for [Lyu12], can also applied to most of the aforementioned schemes.

Trapdoors to SIS instances. We are going to illustrate how a Σ-protocol with oblivious commitments can be obtained through our patch. Basically, we need to provide the prover with a trapdoor to extract a candidate preimage to a given commitment. In the scheme from [Lyu12] the prover has to solve an SIS instance. Roughly speaking, the prover has to find preimages for functions $f_{\mathbf{A}}(\mathbf{v}) := \mathbf{A}\mathbf{v}$ for $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$ where v is distributed according to the discrete Gaussian distribution D_s over $\mathbb{Z}^m$ with standard deviation s. The parameters q, n, m as well as s determine the hardness of the SIS instance.

From [Ajt99, GPV08, AP09, Pei10, MP12] we know the existence of trapdoors allowing to sample such preimages. The most efficient construction from [MP12] finds preimages of length $\beta \approx s\sqrt{m}$ for lattice dimension $m \approx 2n \log q$ with (at least) $s \approx 16\sqrt{n \log q}$. Let T denote the trapdoor for function $f_{\mathbf{A}}$ which is generated together with matrix $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}$ by algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$. Then, the function $\mathsf{SampleD}(\mathbf{T}, \mathbf{A}, \mathbf{X}, s)$ samples an element $\mathbf{x}$ from the distribution within negligibly close (in n) statistical distance of D_s^m such that $\mathbf{Ax} = \mathbf{X}$ (see, e.g., Algorithm 3 of [MP12]).

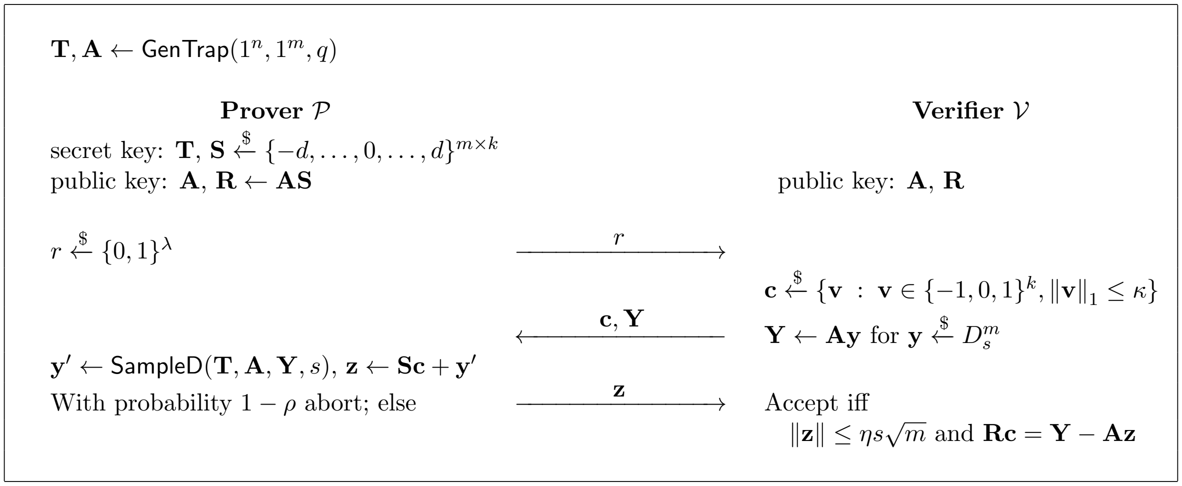
Our Patch on the ID Scheme within [Lyu12]. We take as input the identification (ID) scheme from which the signature scheme in [Lyu12] is derived from. Now, we provide the prover with necessary trapdoor information in order to enable the prover to respond to a challenge for oblivious commitments. The scheme is parameterized by security parameter n, q, d, k, and η. Moreover, $m \approx 2n \log q$[1], κ is chosen such that $2^\kappa \cdot \binom{k}{\kappa} \geq 2^{100}$, and $s \approx 12d\kappa\sqrt{m}$.

The prover first runs $(\mathbf{A} \overset{\$}{\leftarrow} \mathbb{Z}_q^{n\times m}, \mathbf{T}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$. The prover's secret is a matrix $\mathbf{S} \overset{\$}{\leftarrow} \{-d, \ldots, 0, \ldots, d\}^{m\times k}$ and the trapdoor T. The corresponding public key consists of the matrices $\mathbf{A}$ and $\mathbf{R} = \mathbf{AS}$. The prover $\mathcal{P}$ picks first a random string $r \overset{\$}{\leftarrow} \{0,1\}^\lambda$, and sends it over to the verifier. The verifier $\mathcal{V}$ randomly picks a challenge $\mathbf{c} \overset{\$}{\leftarrow} V = \{\mathbf{v} \ : \ \mathbf{v} \in \{-1, 0, 1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$ and computes the commitment $\mathbf{Y} \leftarrow \mathbf{Ay}$ for random $\mathbf{y} \in \mathbb{Z}^m$ sampled according to D_s^m. The verifier forwards both c and $\mathbf{Y}$ to $\mathcal{P}$. Now, $\mathcal{P}$ samples first a valid preimage $\mathbf{y}'$ of $\mathbf{Y}$ under function $f_{\mathbf{A}}$ through algorithm $\mathsf{SampleD}$, and then computes $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}'$. With a certain probability ρ (depending on the public parameters, $\mathbf{z}$, and $\mathbf{Sc}$) the pair $(\mathbf{z}, \mathbf{c})$ is handed over to $\mathcal{V}$. Upon receiving $\mathbf{z}$, $\mathcal{V}$ accepts iff $\|\mathbf{z}\| \leq \eta s\sqrt{m}$ and $\mathbf{Rc} = \mathbf{Y} - \mathbf{Az}$. The underlying interactive scheme is also given in Figure 4. Note that we can assume that ρ is sufficiently large such that we simply let the signer occasionally fail; to get a valid signature repeatedly call the signer about the same message m.

On the Quantum Security. We stress that the resulting (identification) scheme has now oblivious commitments, i.e., it satisfies Definition 4.1. Note that the security as an identification scheme does not depend on the first message sent by the prover. However, if one converts the ID protocol to a signature scheme, this message serves as the randomness input to the hash function together with the message to be signed. As such, a signature on a message m consists of (randomized) $\sigma = (r, \mathsf{rsp})$ where $(\mathsf{com}, \mathsf{ch}) = (\mathbf{Y}, \mathbf{c}) \leftarrow H(pk, r, m)$. Hence, the signature scheme obtained by the FS transformation (see Construction 4.2) on the above identification scheme is secure against quantum adversaries in the quantum-accessible random-oracle model following from the result of Theorem 4.3.

Notice that our resulting signature scheme is close to a variant of the hash-and-sign signature scheme (GPV) by Gentry, Peikert, and Vaikuntanathan [GPV08]. The GPV signature scheme uses a pre-images sampleable trapdoor function (PSF), and signing a message here is basically providing a preimage to the hashed message. If in our construction, the commitment and challenge is merely the hash of the message, both signatures coincide. In a concurrent work [BZ13], the GPV signature scheme is proven secure in the QROM. Interestingly, this gives us two different

[1] In the original paper, the author sets $m \approx 64 + n \log q / \log(2d + 1)$. We slightly increase m in order to obtain a trapdoor. This merely strengthens the underlying hardness.



$\mathbf{T}, \mathbf{A} \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$

Prover $\mathcal{P}$ | **Verifier $\mathcal{V}$**

secret key: $\mathbf{T}$, $\mathbf{S} \overset{\$}{\leftarrow} \{-d, \ldots, 0, \ldots, d\}^{m \times k}$

public key: $\mathbf{A}$, $\mathbf{R} \leftarrow \mathbf{AS}$ | public key: $\mathbf{A}$, $\mathbf{R}$

$r \overset{\$}{\leftarrow} \{0,1\}^\lambda$ $\xrightarrow{\quad r \quad}$

$\mathbf{c} \overset{\$}{\leftarrow} \{\mathbf{v} \ : \ \mathbf{v} \in \{-1,0,1\}^k, \|\mathbf{v}\|_1 \leq \kappa\}$

$\xleftarrow{\quad \mathbf{c}, \mathbf{Y} \quad}$ $\mathbf{Y} \leftarrow \mathbf{Ay}$ for $\mathbf{y} \overset{\$}{\leftarrow} D_s^m$

$\mathbf{y}' \leftarrow \mathsf{SampleD}(\mathbf{T}, \mathbf{A}, \mathbf{Y}, s)$, $\mathbf{z} \leftarrow \mathbf{Sc} + \mathbf{y}'$

With probability $1 - \rho$ abort; else $\xrightarrow{\quad \mathbf{z} \quad}$ Accept iff

$\|\mathbf{z}\| \leq \eta s \sqrt{m}$ and $\mathbf{Rc} = \mathbf{Y} - \mathbf{Az}$

Figure 4: Patched Σ-Protocol from [Lyu12]

proof approaches for similar schemes. While Boneh and Zhandry [BZ13] give security results for the hash-and-sign paradigm and thereby show the security of the GPV signature, our security proof follows immediately from Theorem 4.3 once the scheme by Lyubashevsky is patched to have oblivious commitments.

5 Conclusion

Our impossibility result indicates that the Fiat-Shamir paradigm should be taken with great caution when used to argue quantum resistance. A proof for a scheme in the classical random-oracle model, even if the underlying problem is quantum-resistant, may not yield a protocol which is also secure in the QROM. For some schemes, however, a formal proof in the quantum random oracle is possible after a minor modification. Interestingly, this modification may first weaken the scheme, e.g., remove active security.

It remains open to bypass our black-box separation result by other means, e.g., by using witness-*dependent* commitments, to extend the class of admissible protocols for which the transformation yields a secure scheme in the QROM. Alternatively, one may try to give a "Fiat-Shamir-like" transformation which also yields secure signature schemes in the QROM. Natural candidates would be the constructions with online extractors by Pass [Pas03] and by Fischlin [Fis05], which potentially also circumvent the rewinding problem. We leave this as an interesting open question.

Acknowledgments

We thank the anonymous reviewers for valuable comments. We also thank Dominique Unruh for useful discussions on black-box extractors. Marc Fischlin is supported by the Heisenberg grant Fi 940/3-1 of the German Research Foundation (DFG). Tommaso Gagliardoni is supported by the German Federal Ministry of Education and Research (BMBF) within EC-SPRIDE. This work was also supported by CASED (`www.cased.de`).

References

[ADV$^+$12] Sidi Mohamed El Yousfi Alaoui, Özgür Dagdelen, Pascal Véron, David Galindo, and Pierre-Louis Cayrel. Extended security arguments for signature schemes. In Aikaterini Mitrokotsa and Serge Vaudenay, editors, *AFRICACRYPT 12: 5th International Conference on Cryptology in Africa*, volume 7374 of *Lecture Notes in Computer Science*, pages 19–34, Ifrance, Morocco, July 10–12, 2012. Springer, Berlin, Germany. (Cited on page 22.)

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. Tightly-secure signatures from lossy identification schemes. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 572–590, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[AJLA$^+$12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 483–501, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on pages 1, 2, and 22.)

[Ajt99] Mikls Ajtai. Generating hard instances of the short basis problem. In Ji Wiedermann, Peter Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin Heidelberg, 1999. (Cited on page 23.)

[AP09] Joël Alwen and Chris Peikert. Generating shorter bases for hard random lattices. In *STACS*, pages 75–86, 2009. (Cited on page 23.)

[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. (Cited on pages 16 and 17.)

[BDF$^+$11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69, Seoul, South Korea, December 4–8, 2011. Springer, Berlin, Germany. (Cited on pages 2, 3, 4, 8, 14, and 20.)

[BDSG$^+$13] Nir Bitansky, Dana Dachman-Soled, Sanjam Garg, Abhishek Jain, Yael Tauman Kalai, Adriana Lopez-Alt, and Daniel Wichs. Why fiat-shamir for proofs lacks a proof. In *TCC*, Lecture Notes in Computer Science. Springer, Berlin, Germany, 2013. (Cited on page 1.)

[BM10] Paulo S. L. M. Barreto and Rafael Misoczki. A new one-time signature scheme from syndrome decoding. Cryptology ePrint Archive, Report 2010/017, 2010. `http://eprint.iacr.org/`. (Cited on pages 1 and 22.)

[BP02] Mihir Bellare and Adriana Palacio. GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 162–177, Santa Barbara, CA, USA, August 18–22, 2002. Springer, Berlin, Germany. (Cited on page 2.)

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press. (Cited on page 1.)

[BV97] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. (Cited on pages 16 and 17.)

[BZ12] Dan Boneh and Mark Zhandry. Quantum-secure message authentication codes. Cryptology ePrint Archive, Report 2012/606, 2012. `http://eprint.iacr.org/`. (Cited on pages 2, 3, and 4.)

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a post- quantum world. Cryptology ePrint Archive, Report 2013/088, 2013. `http://eprint.iacr.org/`. (Cited on pages 4, 23, and 24.)

[CLRS10] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. Improved zero-knowledge identification with lattices. In Swee-Huay Heng and Kaoru Kurosawa, editors, *ProvSec 2010: 4th International Conference on Provable Security*, volume 6402 of *Lecture Notes in Computer Science*, pages 1–17, Malacca, Malaysia, October 13–15, 2010. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[CNR12] Jan Camenisch, Gregory Neven, and Markus Rückert. Fully anonymous attribute tokens from lattices. In Ivan Visconti and Roberto De Prisco, editors, *SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 57–75, Amalfi, Italy, September 5–7, 2012. Springer, Berlin, Germany. (Cited on pages 1, 10, and 22.)

[CVA10] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010: 17th Annual International Workshop on Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 171–186, Waterloo, Ontario, Canada, August 12–13, 2010. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[DFNS11] Ivan Damgård, Jakob Funder, Jesper Buus Nielsen, and Louis Salvail. Superposition attacks on cryptographic protocols. Cryptology ePrint Archive, Report 2011/421, 2011. `http://eprint.iacr.org/`. (Cited on page 4.)

[DP92] Alfredo De Santis and Giuseppe Persiano. Zero-knowledge proofs of knowledge without interaction (extended abstract). In *FOCS*, pages 427–436. IEEE Computer Society, 1992. (Cited on page 14.)

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of Cryptology*, 1(2):77–94, 1988. (Cited on page 2.)

[Fis05] Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 152–168, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany. (Cited on page 24.)

[FLR$^+$10] Marc Fischlin, Anja Lehmann, Thomas Ristenpart, Thomas Shrimpton, Martijn Stam, and Stefano Tessaro. Random oracles with(out) programmability. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 303–320, Singapore, December 5–9, 2010. Springer, Berlin, Germany. (Cited on page 3.)

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany. (Cited on page 1.)

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th Annual Symposium on Foundations of Computer Science*, pages 102–115, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press. (Cited on page 1.)

[GKV10] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A group signature scheme from lattice assumptions. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412, Singapore, December 5–9, 2010. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 530–547, Leuven, Belgium, September 9–12, 2012. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 171–185, Santa Barbara, CA, USA, August 1987. Springer, Berlin, Germany. (Cited on page 10.)

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th Annual ACM Symposium on Theory of Computing*, pages 197–206, Victoria, British Columbia, Canada, May 17–20, 2008. ACM Press. (Cited on page 23.)

[GQ90] Louis C. Guillou and Jean-Jacques Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In Shafi Goldwasser, editor, *Advances* *in Cryptology – CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231, Santa Barbara, CA, USA, August 21–25, 1990. Springer, Berlin, Germany. (Cited on pages 2, 3, and 15.)

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389, Melbourne, Australia, December 7–11, 2008. Springer, Berlin, Germany. (Cited on pages 2 and 22.)

[Lyu08] Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In Ronald Cramer, editor, *PKC 2008: 11th International Conference on Theory and Practice of Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179, Barcelona, Spain, March 9–12, 2008. Springer, Berlin, Germany. (Cited on pages 2 and 22.)

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on pages 3, 22, 23, and 24.)

[MGS11] Carlos Aguilar Melchor, Philippe Gaborit, and Julien Schrek. A new zero-knowledge code based identification scheme with reduced communication. *CoRR*, abs/1111.1644, 2011. (Cited on pages 1, 2, and 22.)

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on page 23.)

[MV03] Daniele Micciancio and Salil P. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany. (Cited on pages 2 and 22.)

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000. (Cited on pages 4 and 5.)

[Pas03] Rafael Pass. On deniability in the common reference string and random oracle model. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 316–337, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany. (Cited on page 24.)

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany. (Cited on page 23.)

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000. (Cited on pages 1, 10, and 14.)

[Sak12] Koichi Sakumoto. Public-key identification schemes based on multivariate cubic polynomials. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *PKC 2012: 15th International Workshop on Theory and Practice in Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 172–189, Darmstadt, Germany, May 21–23, 2012. Springer, Berlin, Germany. (Cited on pages 1 and 22.)

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Berlin, Germany. (Cited on page 2.)

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991. (Cited on page 10.)

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 706–723, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Berlin, Germany. (Cited on pages 1, 2, and 22.)

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 135–152, Cambridge, UK, April 15–19, 2012. Springer, Berlin, Germany. (Cited on pages 2 and 7.)

[Wat06] John Watrous. Zero-knowledge against quantum attacks. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 296–305, Seattle, Washington, USA, May 21–23, 2006. ACM Press. (Cited on page 2.)

[Zha12a] Mark Zhandry. How to construct quantum random functions. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE Computer Society, 2012. (Cited on pages 2, 3, and 4.)

[Zha12b] Mark Zhandry. Secure identity-based encryption in the quantum random oracle model. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 758–775, Santa Barbara, CA, USA, August 19–23, 2012. Springer, Berlin, Germany. (Cited on pages 2, 3, 4, 9, 16, 17, 20, and 21.)