The Fiat–Shamir Transformation in a Quantum World

Özgür Dagdelen, Marc Fischlin, and 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 [19] 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 [5,33]. In general, finding secure instantiations based on *standard* hash functions is hard for some schemes, as shown in [22,7]. 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., [28,3,23,12,13,35,30,34,25,1,11,2,17]. 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 [33] relies on rewinding the adversary, which is often considered to be critical for quantum adversaries (albeit not impossible [39,38]), 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. [8] 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, [8] 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. [8] 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 [41,40,9] 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 [8] 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 [36], Guillou and Quisquater [24], and Feige, Fiat and Shamir [18] are conceivably of this type (see also [4]). Quantum-immune candidates are, for instance, [31,27,26,30,35,2].

Our negative result does not primarily rely on the rewinding problem for quantum adversaries; our extractor may rewind the adversary (in a black-box way). Instead, our result 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 [8,40,41,9] 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 [8,41,40,9] 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 [8]). 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 [20] 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 [24] 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 [29] via our method. Before, Lyubashevsky only showed security in the classical random-oracle model, despite using an allegedly quantum-immune primitive.

Related work. Since the introduction of the quantum-accessible random-oracle model [8], several works propose cryptographic primitives or revisit their security against quantum algorithms in this stronger model [40,41,9]. In [15], 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 [10] 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 [10] 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. [8]. We also introduce the notion of Σ-protocols to which the Fiat-Shamir transformation applies. In the full version of this paper [14], 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 [32]. Our description follows [8] 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 [32]), 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 $\mathcal{A}_{\mathrm{Q}}^{|O_1(\cdot)\rangle,|O_2(\cdot)\rangle,\dots}(|\psi\rangle)$ 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 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.

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 [38]. 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 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.

For our negative result we assume that the adversary does not perform any measurements before eventually creating the final output, whereas our positive result also works if the adversary measures in between. This is not a restriction, since in the meta-reduction technique we are allowed to choose a specific adversary, without having to consider more general cases. Note that the intrinsic "quantum randomness" of the adversary is fresh for each rewound run but, for our negative result, since measurements of the adversary are postponed till the end, the extractor can re-create the same quantum state as before at every interaction point. 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, possibly reaching the same quantum state as in previous executions when the adversary interacts with external oracles, (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 (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*

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

is not negligible, then

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

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 [8]. 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 [41] we can even replace the assumption about the pseudorandom function and use a q-wise independent function instead.

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 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 [33] 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 Assessment

Before we dive into the technical details of our result let us re-assess the strength and weaknesses of our impossibility result:

1. The extractor has to choose a classical hash function h for answering QRO queries. While this may be considered a restriction in general interactive

quantum proofs, it seems to be inevitable in the QROM; it is rather a consequence of the approach where a quantum adversary mounts attacks in a classical setting. After all, both the honest parties as well as the adversary expect a classical hash function. The adversary is able to check this property easily, even if it treats the hash function otherwise as a black box (and may thus not be able to spot that the hash function uses (pseudo)randomness). We remark again that this approach also complies with previous efforts [8,41,40,9] and our positive result here to answer such hash queries.

2. The extractor *can* rewind the quantum adversary to any point before the final measurement. Recall that for our impossibility result we assume, to the advantage of the extractor, that the adversary does not perform any measurement until the very end. Since the extractor can re-run the adversary from scratch for the same classical randomness, and the "no-cloning restriction" does not apply to our adversary with classical input, the extractor can therefore easily put the adversary in the same (quantum) state as in a previous execution, up to the final measurement. However, because we consider *black-box* extractors, the extractor can only influence the adversary's behavior via the answers it provides to $\mathcal{A}_{\mathrm{Q}}$'s external communication. In this sense, the extractor may always rewind the adversary to such communication points. We also allow the extractor to measure and abort at such communication points.
3. The extraction strategy by Pointcheval and Stern [33] in the purely classical case *can* be cast in our black-box extractor framework. For this the extractor would run the adversary for the same classical randomness twice, providing a lazy-sampling based hash function description, with different replies in the i-th answers in the two runs. The extractor then extracts the witness from two valid signatures. This shows that a different approach than in the classical setting is necessary for extractors in the QROM.

3.2 Prerequisites

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

Definition 3 (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 [21], the Schnorr proof of knowledge [37], or the recent protocol for lattices used in an anonymous credential system [11]. A typical example of non-witness-independent commitment Σ-protocol is the graph 3-coloring ZKPoK scheme [21] where the prover commits to a random permutation of the coloring.

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.

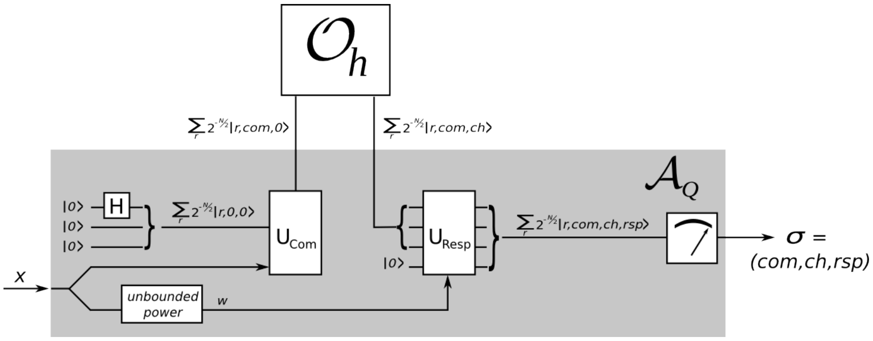

Fig. 1. The canonical adversary

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 4 (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.

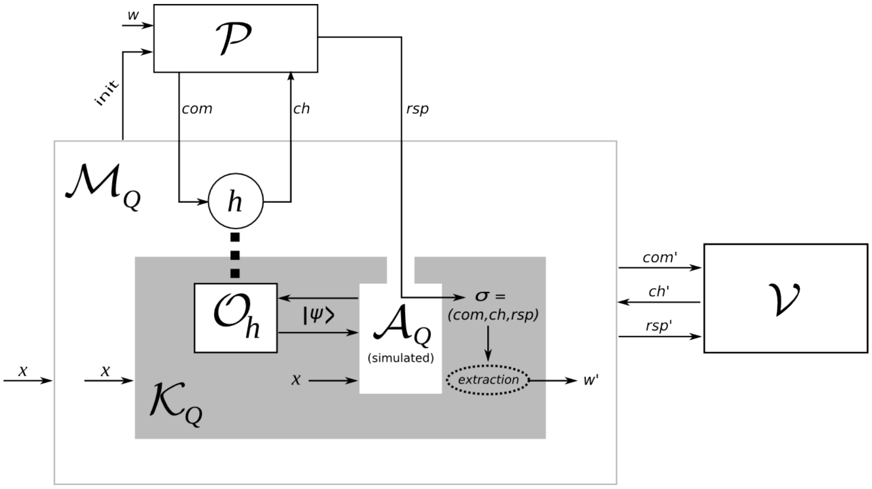

Fig. 2. An overview of our meta-reduction

3.3 The Adversary and the Meta-Reduction

Adversary. Our (unbounded) adversary works roughly as follows (see Figure 1). It receives as input a value x and first uses its unbounded computational power to compute a random witness 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 $N = \text{poly}(n)$) 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 computes, in superposition, responses for all values and finally measures in the computational basis, yielding a sample $(r, \mathsf{com}_{x,r}, \mathsf{ch}, \mathsf{rsp}_{x,w,r})$ for $\mathsf{ch} = H(x, \mathsf{com}_{x,r})$ where r is uniform over all random strings; it outputs the transcript $(\mathsf{com}, \mathsf{ch}, \mathsf{rsp})$.

The Meta-Reduction. We illustrate the meta-reduction in Figure 2. 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.

3.4 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 1 (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 [33] 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 [33]— 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.

On the Necessity of Active Security. If we drop the requirement on active security we can indeed devise a solution based on quantum-immune primitives. Namely, we use the (classical) non-interactive zero-knowledge proofs of knowledge of De Santis and Persiano [16] to build the following three-move scheme: The first message is irrelevant, e.g., we let the prover simply send the constant 0 (potentially padded with redundant randomness), making the commitment also witness-independent. In the second message the verifier sends a random string which the prover interprets as a public key *pk* of a 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, even against quantum distinguishers if the primitives are quantum-secure, because then 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 [16] 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 5 (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 [24] 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 preimage 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 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 $\mathcal{S} = (\mathsf{SKGen}, \mathsf{Sig}, \mathsf{SVf})$ in the (quantum) random-oracle model:*

Key Generation. *$\mathsf{SKGen}(1^\lambda)$ 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 $r \xleftarrow{\$} \textsc{Rnd}$ 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 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 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 [41] 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 [6] we can show that changing the oracle in this case does not change the adversary's success probability significantly.

The full proof with preliminary results appears in the full version [14].

Moreover, we also discuss a concrete instantiation based on Lyubashevsky's lattice-based scheme [29] in the full version [14] to show that one can use our technique in principle, and how it could be used for other schemes.

Acknowledgments

We thank the anonymous reviewers for some 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). Özgür Dagdelen and Tommaso Gagliardoni are 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

1. Abdalla, M., Fouque, P.A., Lyubashevsky, V., Tibouchi, M.: Tightly-secure signatures from lossy identification schemes. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 572–590. Springer (Apr 2012)
2. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer (Apr 2012)
3. Barreto, P.S.L.M., Misoczki, R.: A new one-time signature scheme from syndrome decoding. Cryptology ePrint Archive, Report 2010/017 (2010), http://eprint.iacr.org/

4. Bellare, M., Palacio, A.: GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 162–177. Springer (Aug 2002)
5. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 93. pp. 62–73. ACM Press (Nov 1993)
6. Bennett, C.H., Bernstein, E., Brassard, G., Vazirani, U.V.: Strengths and weaknesses of quantum computing. SIAM J. Comput. 26(5), 1510–1523 (1997)
7. Bitansky, N., Dachman-Soled, D., Garg, S., Jain, A., Kalai, Y.T., Lopez-Alt, A., Wichs, D.: Why fiat-shamir for proofs lacks a proof. In: TCC. LNCS, Springer (2013)
8. Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., Zhandry, M.: Random oracles in a quantum world. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 41–69. Springer (Dec 2011)
9. Boneh, D., Zhandry, M.: Quantum-secure message authentication codes. Cryptology ePrint Archive, Report 2012/606 (2012), http://eprint.iacr.org/
10. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. Cryptology ePrint Archive, Report 2013/088 (2013), http://eprint.iacr.org/
11. Camenisch, J., Neven, G., Rückert, M.: Fully anonymous attribute tokens from lattices. In: Visconti, I., Prisco, R.D. (eds.) SCN 12. LNCS, vol. 7485, pp. 57–75. Springer (Sep 2012)
12. Cayrel, P.L., Lindner, R., Rückert, M., Silva, R.: Improved zero-knowledge identification with lattices. In: Heng, S.H., Kurosawa, K. (eds.) ProvSec 2010. LNCS, vol. 6402, pp. 1–17. Springer (Oct 2010)
13. Cayrel, P.L., Véron, P., Alaoui, S.M.E.Y.: A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 171–186. Springer (Aug 2010)
14. Dagdelen, Ö., Fischlin, M., Gagliardoni, T.: The fiat-shamir transformation in a quantum world. Cryptology ePrint Archive, Report 2013/245 (2013), http://eprint.iacr.org/
15. Damgård, I., Funder, J., Nielsen, J.B., Salvail, L.: Superposition attacks on cryptographic protocols. Cryptology ePrint Archive, Report 2011/421 (2011), http://eprint.iacr.org/
16. De Santis, A., Persiano, G.: Zero-knowledge proofs of knowledge without interaction (extended abstract). In: FOCS. pp. 427–436. IEEE Computer Society (1992)
17. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8042, pp. 40–56. Springer, Santa Barbara, CA, USA (Aug 2013)
18. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology 1(2), 77–94 (1988)
19. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer (Aug 1987)
20. Fischlin, M., Lehmann, A., Ristenpart, T., Shrimpton, T., Stam, M., Tessaro, S.: Random oracles with(out) programmability. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 303–320. Springer (Dec 2010)
21. Goldreich, O., Micali, S., Wigderson, A.: How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 171–185. Springer (Aug 1987)

22. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th FOCS. pp. 102–115. IEEE Computer Society Press (Oct 2003)
23. Gordon, S.D., Katz, J., Vaikuntanathan, V.: A group signature scheme from lattice assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 395–412. Springer (Dec 2010)
24. Guillou, L.C., Quisquater, J.J.: A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 216–231. Springer (Aug 1990)
25. Güneysu, T., Lyubashevsky, V., Pöppelmann, T.: Practical lattice-based cryptography: A signature scheme for embedded systems. In: Prouff, E., Schaumont, P. (eds.) CHES 2012. LNCS, vol. 7428, pp. 530–547. Springer (Sep 2012)
26. Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 372–389. Springer (Dec 2008)
27. Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Cramer, R. (ed.) PKC 2008. LNCS, vol. 4939, pp. 162–179. Springer (Mar 2008)
28. Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer (Dec 2009)
29. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer (Apr 2012)
30. Melchor, C.A., Gaborit, P., Schrek, J.: A new zero-knowledge code based identification scheme with reduced communication. CoRR abs/1111.1644 (2011)
31. Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 282–298. Springer (Aug 2003)
32. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
33. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. Journal of Cryptology 13(3), 361–396 (2000)
34. Sakumoto, K.: Public-key identification schemes based on multivariate cubic polynomials. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 172–189. Springer (May 2012)
35. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-key identification schemes based on multivariate quadratic polynomials. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 706–723. Springer (Aug 2011)
36. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 239–252. Springer (Aug 1990)
37. Schnorr, C.P.: Efficient signature generation by smart cards. Journal of Cryptology 4(3), 161–174 (1991)
38. Unruh, D.: Quantum proofs of knowledge. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 135–152. Springer (Apr 2012)
39. Watrous, J.: Zero-knowledge against quantum attacks. In: Kleinberg, J.M. (ed.) 38th ACM STOC. pp. 296–305. ACM Press (May 2006)
40. Zhandry, M.: How to construct quantum random functions. In: IEEE Annual Symposium on Foundations of Computer Science. pp. 679–687. IEEE Computer Society (2012)
41. Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer (Aug 2012)