The Measure-and-Reprogram Technique 2.0: Multi-Round Fiat-Shamir and More

Jelle Don[1], Serge Fehr[1,2], and Christian Majenz[1,3]

[1] Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands
[2] Mathematical Institute, Leiden University, Netherlands
[3] QuSoft, Amsterdam, Netherlands
jelle.don@cwi.nl, serge.fehr@cwi.nl, c.majenz@uva.nl

Abstract. We revisit recent works by Don, Fehr, Majenz and Schaffner and by Liu and Zhandry on the security of the Fiat-Shamir transformation of Σ-protocols in the quantum random oracle model (QROM). Two natural questions that arise in this context are: (1) whether the results extend to the Fiat-Shamir transformation of *multi-round* interactive proofs, and (2) whether Don et al.'s $O(q^2)$ loss in security is optimal.
Firstly, we answer question (1) in the affirmative. As a byproduct of solving a technical difficulty in proving this result, we slightly improve the result of Don et al., equipping it with a cleaner bound and an even simpler proof. We apply our result to digital signature schemes showing that it can be used to prove strong security for schemes like MQDSS in the QROM. As another application we prove QROM-security of a non-interactive OR proof by Liu, Wei and Wong.
As for question (2), we show via a Grover-search based attack that Don et al.'s quadratic security loss for the Fiat-Shamir transformation of Σ-protocols is optimal up to a small constant factor. This extends to our new multi-round result, proving it tight up to a factor that depends on the number of rounds only, i.e. is constant for any constant-round interactive proof.

1 Introduction

Reprogramming the quantum random oracle. We reconsider the recent work of Don, Fehr, Majenz and Schaffner [DFMS19] on the quantum random oracle model (QROM). On a technical level, they showed how to reprogram the QROM adaptively at *one* input. More precisely, for any oracle quantum algorithm $\mathcal{A}^H$, making q calls to a random oracle H and outputting a pair (x, z) so that some predicate $V(x, H(x), z)$ is satisfied, they showed existence of a "simulator" $\mathcal{S}$ that mimics the random oracle, extracts x from $\mathcal{A}^H$ by measuring one of the oracle queries to H, and then reprograms $H(x)$ to a given value Θ so that z output by $\mathcal{A}^H$ now satisfies $V(x, \Theta, z)$, except with a multiplicative $O(q^2)$ loss in probability (plus a negligible additive loss). We emphasize that the challenging aspect of this problem is that $\mathcal{A}^H$'s queries to H may be in quantum superposition, and thus measuring such a query disturbs the state and thus the behavior of $\mathcal{A}^H$. Still, Don et al. managed to control this disturbance sufficiently. In independent work and using very different techniques, Liu and Zhandry [LZ19] showed a similar kind of result, but with a $O(q^9)$ loss.

As an immediate application of this technique, it is then concluded that the Fiat-Shamir transformation of a Σ-protocol is as secure (in the QROM) as the original Σ-protocol (in the standard model), up to a $O(q^2)$ loss, i.e., any of the typically considered security notions is preserved under the Fiat-Shamir transformation, even in the quantum setting. In combination with prior work on simulating signature queries [Unr17, KLS18], security (in the QROM) of Fiat-Shamir signatures that arise from ordinary Σ-protocols then follows as a corollary.

Given important examples of *multi-round* public-coin interactive proofs, used in, e.g., MQDSS [CHR+16] and for Bulletproofs [BBB+18][1], a natural question that arises is whether these techniques and results extend to the reprogrammability of the QROM at *multiple* inputs and the security of the Fiat-Shamir transformation (in the QROM) of *multi-round* public-coin interactive proofs. Another question is whether the $O(q^2)$ loss (for the original Σ-protocols) is optimal, or whether one might hope for a linear loss as in the classical case.

In this work, we provide answers to both these natural questions — and more.

[1] The security of the original Bulletproofs protocol relies on the hardness of discrete-log; however, work in progress considers post-quantum secure versions [Boo].

A technical hurdle for generalizing [DFMS19] to multi-round Fiat-Shamir. To start with, we observe that the naive approach of applying the original result of [DFMS19] inductively so as to reprogram multiple inputs one by one does not work. This is due to a subtle technical issue that has to do with the precise statement of the original result. In more detail, the statement involves an additive error term $\varepsilon_x \geq 0$ that depends on the particular choice of the point x, which is (adaptively) chosen to be the input on which the random oracle (RO) is reprogrammed. The guarantee provided by [DFMS19] is that this error term stays negligible even *when summed over* all x's, i.e., $\sum_x \varepsilon_x = negl$. The formulation of the result for individual x's with control over $\sum_x \varepsilon_x$ is important for the later applications to the Fiat-Shamir transformation. However, when applying the result twice in a row, with the goal being to reprogram the RO at two inputs x_1, x_2, then we end up with two error terms ε_{x_1} and $\varepsilon^{x_1}_{x_2}$ (with the second one depending on x_1), where the first one stays negligible when summed over x_1 and the second one stays negligible when summed over x_2 (for any x_1); but it is unclear that the sum $\varepsilon_{x_1,x_2} := \varepsilon_{x_1} + \varepsilon^{x_1}_{x_2}$ stays negligible when summed over x_1 *and* x_2, which is what we would need to get the corresponding generalized statement.

Our results As a first contribution, we revise the *original* result from [DFMS19] of reprogramming the QROM at one input by showing an *improved* version that has *no* additive error term, but only the original multiplicative $O(q^2)$ loss. For typical direct cryptographic applications, this improvement makes no big quantitative difference due to the error term being negligible, but: (1) it makes the statement cleaner and easier to formulate, (2) somewhat surprisingly, the proof is simpler than that of the original result in [DFMS19], and (3) most importantly, it removes the technical hurdle to extend to multiple inputs. Indeed, we then get the desired multi-input reprogrammability result by means of a not too difficult, though somewhat tedious, induction argument.

Building on our multi-input reprogrammability result above, our next goal then is to show the security of the Fiat-Shamir transformation (in the QROM) of multi-round public-coin interactive proofs. In contrast to the original result in [DFMS19] for the Fiat-Shamir transformation of Σ-protocols some additional work is needed here, to deal with the order of the messages extracted from the Fiat-Shamir adversary. Thus, as a stepping stone, we consider and analyze a variant of the above multi-input reprogrammability result, which enforces the right order of the extracted messages. As a simple corollary of this, we then obtain the desired security of multi-round Fiat-Shamir. Here, the multiplicative loss becomes $O(q^2 n)$ for a $(2n+1)$-round public-coin interactive proof with constant n.

In the context of digital signatures, the original motivation for the Fiat-Shamir transformation, we extend previous results by Unruh [Unr17] and Don et al. [DFMS19] to show that Fiat-Shamir signature schemes based on a multi-round, honest-verifier zero knowledge public-coin interactive quantum proof of knowledge have standard signature security (existential unforgeability under chosen message attacks, UF-CMA) in the QROM. Assuming the additional collision-resistance-like property of computationally unique responses, they are even strongly unforgeable. We go on to apply this result to the signature scheme MQDSS [CHR+16], a candidate in the ongoing NIST standardization process for post-quantum cryptographic schemes [NIS], providing its first QROM proof.

Another application of our multi-round Fiat-Shamir result would for instance be to Bulletproofs [BBB+18].

As a second application of our multi-input reprogrammability result, we show security (in the QROM) of the non-interactive OR-proof introduced by Liu, Wei and Wong [LWW04], further analyzed by Fischlin, Harasser and Janson [FHJ]. While the well-known (interactive) OR-proof by Cramer, Damgård and Schoenmakers [CDS94] is a Σ-protocol and thus the results from [DFMS19] apply, the inherently non-interactive OR-proof by Liu et al. does *not* follow this blueprint of being obtained as the Fiat-Shamir transformation of a Σ-protocol (though in some sense it is "close" to being of this form). We show here how the 2-input version of our multi-input reprogrammability result implies security of this OR-proof in the QROM.

Our last contribution is a lower bound that shows that the multiplicative $O(q^2)$ loss in the security argument of the Fiat-Shamir transformation of Σ-protocols is tight (up to a factor 4). Thus, the $O(q^2)$ loss is unavoidable in general. Furthermore, we extend this lower bound to the Fiat-Shamir transformation of multi-round interactive proofs as considered in this work, and we show that also here to obtained loss $O(q^{2n})$ is in general optimal, up to a constant that depends on n only.

Related work Before the recently obtained reduction [DFMS19, LZ19] was available, the Fiat-Shamir tranform in the QROM was studied in a number of works [Unr17, DFG13, KLS18], where weaker security properties were shown. In addition, Unruh developed an alternative transform [Unr15] that provided QROM security at the expense of an increased proof size. The Unruh transform was later generalized to apply to 5-round public coin interactive proof systems [CHR+18].

2 Notation

Up to some modifications, we follow closely the notation used in [DFMS19]. We consider a (purified) oracle quantum algorithm $\mathcal{A}$ that makes q queries to an *oracle*, i.e., an unspecified function $H : \mathcal{X} \to \mathcal{Y}$ with finite non-empty sets $\mathcal{X}, \mathcal{Y}$. Formally, $\mathcal{A}$ is described by a sequence of unitaries $A_1, \ldots, A_q$ and an initial state $|\phi_0\rangle$.[2] For technical reasons that will become clear later, we actually allow (some of) the A_i's to be a *projection* followed by a unitary (or vice versa). One can think of such a projection as a measurement performed by the algorithm, with the algorithm aborting except in case of a particular measurement outcome.

For any concrete choice of $H : \mathcal{X} \to \mathcal{Y}$, the algorithm $\mathcal{A}$ computes the state

$$|\phi_q^H\rangle := \mathcal{A}^H|\phi_0\rangle := A_q\mathcal{O}^H \cdots A_1\mathcal{O}^H|\phi_0\rangle\,,$$

where $\mathcal{O}^H$ is the unitary defined by $\mathcal{O}^H : |c\rangle|x\rangle|y\rangle \mapsto |c\rangle|x\rangle|y \oplus c\cdot H(x)\rangle$ for any triple $c \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, with $\mathcal{O}^H$ acting on appropriate registers. We emphasize that we allow *controlled* queries to H. Per se, this gives the algorithm more power, and thus will make our result only stronger, but it is easy to see that such controlled queries to the standard quantum oracle for a function can always be simulated by means of ordinary queries, at the price of one additional query.[3] The final state $\mathcal{A}^H|\phi_0\rangle$ is considered to be a state over registers $\mathsf{X} = \mathsf{X}_1 \ldots \mathsf{X}_n$, Z and E.

Following [DFMS19], we introduce the following notation. For $0 \le i, j \le q$ we set

$$\mathcal{A}^H_{i\to j} := A_j\mathcal{O}^H \cdots A_{i+1}\mathcal{O}^H\,,$$

where, by convention, $\mathcal{A}^H_{i\to j}$ is set to $\mathbb{1}$ if $j \le i$. Furthermore, we let

$$|\phi_i^H\rangle := \left(\mathcal{A}^H_{0\to i}\right)|\phi_0\rangle$$

be the state of $\mathcal{A}$ after the i-th step but right before the $(i+1)$-st query, which is consistent with $|\phi_q^H\rangle$ above.

For a given function $H : \mathcal{X} \to \mathcal{Y}$ and for fixed $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, we define the *reprogrammed* function $H{*}\Theta x : \mathcal{X} \to \mathcal{Y}$ that coincides with H on $\mathcal{X} \setminus \{x\}$ but maps x to Θ. With this notation at hand, we can then write

$$\left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)\left(\mathcal{A}^H_{0\to i}\right)|\phi_0\rangle = \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)|\phi_i^H\rangle$$

for an execution of $\mathcal{A}$ where the oracle is reprogrammed at a given point x after the i-th query. We stress that $(\mathcal{A}^{H*\Theta x}_{i\to q})(\mathcal{A}^H_{0\to i})$ can again be considered to be an oracle quantum algorithm $\mathcal{B}$, which depends on $\Theta \in \mathcal{Y}$, that makes q queries to (the unprogrammed) function H. Indeed, the (controlled) queries to the reprogrammed oracle $H * \Theta x$ can be simulated by means of controlled queries to H (using one additional "work qubit").[4] Exploiting that, in addition to unitaries, we allow projections as elementary operations, we can also understand $(\mathcal{A}^{H*\Theta x}_{i\to q})X(\mathcal{A}^H_{0\to i})$ to be an oracle quantum algorithm again that makes oracle queries to H, where X is the projection $X = |x\rangle\langle x|$, acting on the corresponding query register to the oracle.

[2] Alternatively, we may regard $|\phi_0\rangle$, as an additional input given to $\mathcal{A}$.

[3] Allowing controlled queries to the random oracle is also the more natural model compared to restricting to plain access to the unitary. After all, the motivation for the QROM is that in the real world, an attacker can implement the modeled hash function on their quantum computer, so they can definitely implement the controlled version as well.

[4] Here it is crucial that we allow *controlled* queries to H.

More generally, for any $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ *without duplicate entries*, i.e., $x_i \neq x_j$ for $i \neq j$, and for any $\mathbf{\Theta} \in \mathcal{Y}^n$, we define

$$H * \mathbf{\Theta}\mathbf{x} = H * \Theta_1 x_1 * \cdots * \Theta_n x_n : \mathcal{X} \to \mathcal{Y}$$
$$x \mapsto \begin{cases} \Theta_i & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, n\} \\ H(x) & \text{otherwise.} \end{cases}$$

This will then allow us to consider $(\mathcal{A}^{H * \Theta_1 x_1 * \Theta_2 x_2}_{i_2 \to q}) X_2 (\mathcal{A}^{H * \Theta_1 x_1}_{i_1 \to i_2}) X_1 (\mathcal{A}^{H}_{0 \to i_1})$ as an oracle quantum algorithm with oracle queries to H, etc.

Eventually, we are interested in the probability that after the execution of the original algorithm $\mathcal{A}^H$, and upon measuring register X in the computational basis to obtain $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$, the state of register Z is of a certain form dependent on $\mathbf{x}$ and $H(\mathbf{x}) = (H(x_1), \ldots, H(x_n))$. Such a requirement (for a fixed $\mathbf{x}$) is captured by a projection

$$G^H_{\mathbf{x}} = |\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x}, H(\mathbf{x})} \,,$$

where $\{\Pi_{\mathbf{x},\mathbf{\Theta}}\}_{\mathbf{x},\mathbf{\Theta}}$ is a family of projections with $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{\Theta} \in \mathcal{Y}^n$, and with the understanding that $|\mathbf{x}\rangle\langle\mathbf{x}|$ acts on X and $\Pi_{\mathbf{x},H(\mathbf{x})}$ on register Z. We refer to such a family of projections as a *quantum predicate*. We use $G^{\mathbf{\Theta}}_{\mathbf{x}}$ as a short hand for $G^{H * \mathbf{\Theta}\mathbf{x}}_{\mathbf{x}}$, and we write G^H_x and G^{Θ}_x with $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$ for the case $n = 1$.

For an arbitrary but fixed $\mathbf{x}_\circ \in \mathcal{X}^n$, we are then interested in the probability

$$\Pr\big[\, \mathbf{x} = \mathbf{x}_\circ \wedge V(\mathbf{x}, H(\mathbf{x}), z) : (\mathbf{x}, z) \leftarrow \mathcal{A}^H \,\big] = \big\| G^H_{\mathbf{x}_\circ} |\phi^H_q\rangle \big\|_2^2 .$$

where the left hand side is our notation for this probability, where we understand $\mathcal{A}^H$ to be an algorithm that outputs the measured $\mathbf{x}$ together with the quantum state z in register Z, and V to be the quantum predicate specified by the projections $\Pi_{\mathbf{x},\mathbf{\Theta}}$. Correspondingly, $\Pr\big[x = x_\circ \wedge V(x, H(x), z) : (x, z) \leftarrow \mathcal{A}^H\big] = \|G^H_{x_\circ} |\phi^H_q\rangle\|_2^2$ for the $n = 1$ case.

3 An improved single-input reprogramming result

For the case $n = 1$, Don et al. [DFMS19] show the existence of a black-box *simulator* $\mathcal{S}$ such that for any oracle quantum algorithm $\mathcal{A}$ as considered above with oracle access to a *uniformly random* H, it holds that

$$\begin{aligned} &\Pr_{\Theta}\big[x = x_\circ \wedge V(x, \Theta, z) : (x, z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \Theta\rangle\big] \\ &\quad \geq \frac{1}{2(q+1)(2q+3)} \Pr_H\big[x = x_\circ \wedge V(x, H(x), z) : (x, z) \leftarrow \mathcal{A}^H\big] - \varepsilon_{x_\circ} \,, \end{aligned} \tag{1}$$

for any $x_\circ \in \mathcal{X}$, where the $\varepsilon_{x_\circ}$'s are non-negative and their sum over $x_\circ \in \mathcal{X}$ is bounded by $1/(2q|\mathcal{Y}|)$, i.e., negligible whenever $|\mathcal{Y}|$ is superpolynomial. The notation $(x, z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \Theta\rangle$ is to be understood in that in a first stage $\mathcal{S}^{\mathcal{A}}$ outputs x, and then on input Θ it outputs z. At the core, Equation (1) follows from Lemma 1 of [DFMS19] which shows that

$$\begin{aligned} &\mathop{\mathbb{E}}_{\Theta,i,b}\Big[\big\|(|x\rangle\langle x| \otimes \Pi_{x,\Theta})\big(\mathcal{A}^{H * \Theta x}_{i+b \to q}\big)\big(\mathcal{A}^{H}_{i \to i+b}\big) X |\phi^H_i\rangle\big\|_2^2\Big] \\ &\quad \geq \frac{\mathbb{E}_\Theta\Big[\big\|(|x\rangle\langle x| \otimes \Pi_{x,\Theta})|\phi^{H * \Theta x}_q\rangle\big\|_2^2\Big]}{2(q+1)(2q+3)} - \frac{\big\|X|\phi^H_q\rangle\big\|_2^2}{2(q+1)|\mathcal{Y}|} \,, \end{aligned} \tag{2}$$

and from which the construction of $\mathcal{S}$ can be extracted. The bound (1) on the "success probability" of $\mathcal{S}$ then follows from the observation that $\mathcal{S}$ can simulate the calls to H and to $H * \Theta x$ by means of a $2(q+1)$-wise independent hash function, and that H and $H * \Theta x$ are indistinguishable for random H and Θ.

In this section we show an improved variant of Equation (1), which avoids the additive error term $\varepsilon_{x_\circ}$. While having negligible quantitative effect in typcial situations, it makes the statement simpler. In addition, as explained in the introduction, it circumvents a technical issue one encounters when

trying to extend to the multi-input case. Furthermore, our improved version comes with a simpler proof.[5]

The approach is to avoid the additive error term in Equation (2). We achieve this by slightly tweaking the simulator $\mathcal{S}$. From the technical perspective, while on the left hand side of Equation (2) the expectation is over a random $i \in \{0,\dots,q\}$, selecting one of the $q+1$ queries of $\mathcal{A}$ at random (where the X register of the output state is considered to be a final query), and a random $b \in \{0,1\}$, our new version has syntactically the same left hand side, but with the expectation over a random pair $(i,b) \in (\{0,\dots,q\text{-}1\}\times\{0,1\})\cup\{(q,0)\}$ instead. This allows us to absorb the additive error term into the success probability of the simulator. Furthermore, it holds for any *fixed* choice of Θ (and not only on average for a random choice).

Lemma 1 *Let $\mathcal{A}$ be a q-query oracle quantum algorithm. Then, for any function $H : \mathcal{X} \to \mathcal{Y}$, any $x \in \mathcal{X}$ and $\Theta \in \mathcal{Y}$, and any projection $\Pi_{x,\Theta}$, it holds that*

$$\mathop{\mathbb{E}}_{i,b}\left[\left\|(|x\rangle\langle x|\otimes\Pi_{x,\Theta})\left(\mathcal{A}^{H*\Theta x}_{i+b\to q}\right)\left(\mathcal{A}^{H}_{i\to i+b}\right)X|\phi^H_i\rangle\right\|_2^2\right] \geq \frac{\left\|(|x\rangle\langle x|\otimes\Pi_{x,\Theta})|\phi^{H*\Theta x}_q\rangle\right\|_2^2}{(2q+1)^2}\,,$$

where the expectation is over uniform $(i,b) \in (\{0,\dots,q\text{-}1\}\times\{0,1\})\cup\{(q,0)\}$.

This new version of Equation (2) translates to a simulator $\mathcal{S}$ that works by running $\mathcal{A}$, but with the following modifications. First, one of the $q+1$ queries of $\mathcal{A}$ (also counting the final output in register X) is measured, and the measurement outcome x is output by (the first stage of) $\mathcal{S}$. We emphasize that the crucial difference to [DFMS19] is that each of the q actual queries is picked with probability $\frac{2}{2q+1}$, while the final output is picked with probability $\frac{1}{2q+1}$. Then, very much as in [DFMS19], this very query of $\mathcal{A}$ is answered either using the original H *or* using the reprogrammed oracle $H*\Theta x$, with the choice being made at random[6], while all the remaining queries of $\mathcal{A}$ are answered using oracle $H*\Theta x$. Finally, (the second stage of) $\mathcal{S}$ outputs whatever $\mathcal{A}$ outputs.

In line with Theorem 1 in [DFMS19], i.e. Equation (1) above, we obtain the following result from Lemma 1.

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

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

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

The proof of Lemma 1 follows closely the proof of Equation (1) in [DFMS19], but the streamlined statement and simulator allow to cut some corners.

Proof (of Lemma 1). For any $0 \leq i \leq q$, inserting a resolution of the identity and exploiting that

$$\left(\mathcal{A}^{H*\Theta x}_{i+1\to q}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)(\mathbb{1}-X)|\phi^H_i\rangle = \left(\mathcal{A}^{H*\Theta x}_{i\to q}\right)(\mathbb{1}-X)|\phi^H_i\rangle\,,$$

we can write

$$\begin{aligned}&\left(\mathcal{A}^{H*\Theta x}_{i+1\to q+1}\right)|\phi^H_{i+1}\rangle\\ &\quad= \left(\mathcal{A}^{H*\Theta x}_{i+1\to q+1}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)(\mathbb{1}-X)|\phi^H_i\rangle &&+\left(\mathcal{A}^{H*\Theta x}_{i+1\to q+1}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X|\phi^H_i\rangle\\ &\quad= \left(\mathcal{A}^{H*\Theta x}_{i\to q+1}\right)(\mathbb{1}-X)|\phi^H_i\rangle &&+\left(\mathcal{A}^{H*\Theta x}_{i+1\to q+1}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X|\phi^H_i\rangle\\ &\quad= \left(\mathcal{A}^{H*\Theta x}_{i\to q+1}\right)|\phi^H_i\rangle - \left(\mathcal{A}^{H*\Theta x}_{i\to q+1}\right)X|\phi^H_i\rangle &&+\left(\mathcal{A}^{H*\Theta x}_{i+1\to q+1}\right)\left(\mathcal{A}^{H}_{i\to i+1}\right)X|\phi^H_i\rangle\end{aligned}$$

[5] We thank Dominique Unruh for the idea that it might be possible to avoid the additive error term, and for proposing an argument for achieving that, which inspired us to find the simpler argument we eventually used.

[6] If it is the final output that is measured then there is nothing left to reprogram, so no choice has to be made.

Rearranging terms, applying $G_x^\Theta = (|x\rangle\langle x| \otimes \Pi_{x,\Theta})$ and using the triangle equality, we can thus bound

$$\begin{aligned}\left\|G_x^\Theta\left(\mathcal{A}_{i\to q}^{H*\Theta x}\right)|\phi_i^H\rangle\right\|_2 \leq \left\|G_x^\Theta\left(\mathcal{A}_{i+1\to q}^{H*\Theta x}\right)|\phi_{i+1}^H\rangle\right\|_2 \\ + \left\|G_x^\Theta\left(\mathcal{A}_{i\to q}^{H*\Theta x}\right)X|\phi_i^H\rangle\right\|_2 \\ + \left\|G_x^\Theta\left(\mathcal{A}_{i+1\to q}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+1}^{H}\right)X|\phi_i^H\rangle\right\|_2 .\end{aligned}$$

Summing up the respective sides of the inequality over $i = 0, \ldots, q-1$, we get

$$\left\|G_x^\Theta|\phi_q^{H*\Theta x}\rangle\right\|_2 \leq \left\|G_x^\Theta|\phi_q^H\rangle\right\|_2 + \sum_{\substack{0\leq i<q\\ b\in\{0,1\}}} \left\|G_x^\Theta\left(\mathcal{A}_{i+b\to q}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+b}^{H}\right)X|\phi_i^H\rangle\right\|_2 .$$

By squaring both sides, dividing by $2q+1$ (i.e., the number of terms on the right hand side), and using Jensen's inequality on the right hand side, we obtain

$$\frac{\left\|G_x^\Theta|\phi_q^{H*\Theta x}\rangle\right\|_2^2}{2q+1} \leq \left\|G_x^\Theta|\phi_q^H\rangle\right\|_2^2 + \sum_{\substack{0\leq i<q\\ b\in\{0,1\}}} \left\|G_x^\Theta\left(\mathcal{A}_{i+b\to q}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+b}^{H}\right)X|\phi_i^H\rangle\right\|_2^2$$

and thus, noting that we can write $\left\|G_x^\Theta|\phi_q^H\rangle\right\|_2^2$ as

$$\left\|G_x^\Theta\left(\mathcal{A}_{i+b\to q+1}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+b}^{H}\right)X|\phi_i^H\rangle\right\|_2^2$$

with $i = q$ and $b = 0$,

$$\frac{\left\|G_x^\Theta|\phi_q^{H*\Theta x}\rangle\right\|_2^2}{(2q+1)^2} \leq \mathop{\mathbb{E}}_{i,b}\left[\left\|G_x^\Theta\left(\mathcal{A}_{i+b\to q}^{H*\Theta x}\right)\left(\mathcal{A}_{i\to i+b}^{H}\right)X|\phi_i^H\rangle\right\|_2^2\right] .$$

□

For completeness, let us spell out how Theorem 8 of [DFMS19] on the generic security of the Fiat-Shamir transformation (in the QROM) can now be re-phrased, avoiding the negligible error term present in [DFMS19]. We refer to [DFMS19] or to our later Section 5 for the details on the Fiat-Shamir transformation.

Theorem 3 *There exists a black-box quantum polynomial-time two-stage quantum algorithm $\mathcal{S}$ such that for any adaptive Fiat-Shamir adversary $\mathcal{A}$, making q queries to a uniformly random function H with appropriate domain and range, and for any $x_\circ \in \mathcal{X}$:*

$$\begin{aligned}\Pr\left[x = x_\circ \wedge v = \mathit{accept} : (x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\right] \\ \geq \frac{1}{(2q+1)^2}\Pr_H\left[x = x_\circ \wedge V_{FS}^H(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^H\right] .\end{aligned}$$

4 Multi-input reprogrammability

In this section, we extend our (improved) results on adaptively reprogramming the quantum random oracle at *one* point $x \in \mathcal{X}$ to *multiple* points $x_1, \ldots, x_n \in \mathcal{X}$. This in turn will allow us to extend the results on the security of the Fiat-Shamir transformation to *multi-round* protocols. We point out again that the improvement of Lemma 1 over Lemma 1 in [DFMS19] plays a crucial role here, in that it circumvents the trouble with the negligible error term that occurs when trying to extend the result from [DFMS19] to the setting considered here.

The starting point is the following generalized version of the problem considered in Section 3. We assume an oracle quantum algorithm $\mathcal{A}^H$ that makes q queries to a random oracle $H : \mathcal{X} \to \mathcal{Y}$ and then produces an output of the form $(x_1, \ldots, x_n, z)$, where z may be quantum, such that a certain (quantum) predicate $V(x_1, H(x_1), \ldots, x_n, H(x_n), z)$ is satisfied with some probability. The goal then is to turn such an $\mathcal{A}^H$ into a multi-stage quantum algorithm $\mathcal{S}$ (the *simulator*) that, stage by stage, outputs the x_i's and takes corresponding Θ_i's as input, and eventually outputs a (possibly quantum) z with the property that $V(x_1, \Theta_1, \ldots, x_n, \Theta_n, z)$ is satisfied with similar probability.

4.1 The general case

Naively, one might hope for an $\mathcal{S}$ that outputs x_1 in the first stage (obtained by measuring one of the queries of $\mathcal{A}^H$), and then on input Θ_1 proceeds by outputting x_2 in the second stage (obtained by measuring one of the subsequent queries of $\mathcal{A}^H$), etc. However, since $\mathcal{A}^H$ may query the hashes of $x_1, \ldots, x_n$ in an arbitrary order, we cannot hope for this to work. Therefore, we have to allow $\mathcal{S}$ to produce $x_1, \ldots, x_n$ in an arbitrary order as well.[7] Formally, we consider $\mathcal{S}$ with the following syntactic behavior: in the first stage it outputs a permutation π together with $x_{\pi(1)}$ and takes as input $\Theta_{\pi(1)}$, and then for every subsequent stage $1 < i \leq n$ it outputs $x_{\pi(i)}$ and takes as input $\Theta_{\pi(i)}$; eventually, in the final stage (labeled by $n+1$) it outputs z. In line with earlier notation, but taking this additional complication into account, we denote such an execution of $\mathcal{S}$ as $(\pi, \pi(\mathbf{x}), z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \pi(\boldsymbol{\Theta}) \rangle$.

A final issue is that if $x_i = x_j$ then $H(x_i) = H(x_j)$ as well, whereas Θ_i and Θ_j may well be different. Thus, we can only expect $\mathcal{S}$ to work well when $x_1, \ldots x_n$ has no duplicates.

For us to be able to mathematically reason about the simulator described above, we introduce some additional notation. For the basic simulator from Lemma 1 we write, using $r_1 = (b_1, i_1)$, as

$$\mathcal{S}^{H,\mathcal{A}}_{\Theta_1,x_1,r_1} := \mathcal{S}^{H,\mathcal{A},\Theta_1,x_1,r_1} := \left(\mathcal{A}^{H*\Theta_1 x_1}_{i_1+b_1\to q}\right)\left(\mathcal{A}^{H}_{i_1\to i_1+b_1}\right) X_1 \left(\mathcal{A}^{H}_{0\to i_1}\right).$$

This can be recursively extended by applying it to $\mathcal{A}^H$ now being $\mathcal{S}^{H,\mathcal{A}}_{\Theta_1,x_1,r_1}$ so as to obtain

$$\mathcal{S}^{H,\mathcal{A}}_{\Theta_{1,2},x_{1,2},r_{1,2}} := \left(\mathcal{S}^{H*\Theta_2 x_2,\mathcal{A},\Theta_1,x_1,r_1}_{i_2+b_2\to q}\right)\left(\mathcal{S}^{H,\mathcal{A},\Theta_1,x_1,r_1}_{i_2\to i_2+b_2}\right) X_2 \left(\mathcal{S}^{H,\mathcal{A},\Theta_1,x_1,r_1}_{0\to i_2}\right).$$

In general, we can consider the following operator, which simulates $\mathcal{A}$ and performs n measurements:

$$\mathcal{S}^{H,\mathcal{A}}_{\boldsymbol{\Theta},\mathbf{x},\mathbf{r}} := \left(\mathcal{S}^{H*\Theta_n x_n,\mathcal{A},\overline{\boldsymbol{\Theta}},\overline{\mathbf{x}},\overline{\mathbf{r}}}_{i_n+b_n\to q}\right)\left(\mathcal{S}^{H,\mathcal{A},\overline{\boldsymbol{\Theta}},\overline{\mathbf{x}},\overline{\mathbf{r}}}_{i_n\to i_n+b_n}\right) X_n \left(\mathcal{S}^{H,\mathcal{A},\overline{\boldsymbol{\Theta}},\overline{\mathbf{x}},\overline{\mathbf{r}}}_{0\to i_n}\right).$$

where, for arbitrary but fixed n and $\boldsymbol{\Theta} = (\Theta_1, \ldots, \Theta_n) \in \mathcal{Y}^n$, the notation $\overline{\boldsymbol{\Theta}}$ is understood as $\overline{\boldsymbol{\Theta}} = (\Theta_1, \ldots, \Theta_{n-1}) \in \mathcal{Y}^{n-1}$, and correspondingly for $\mathbf{x}$ etc. Finally, when considering *fixed* $\boldsymbol{\Theta} \in \mathcal{Y}^n$ and $\mathbf{x} \in \mathcal{X}^n$, we write

$$S^H_{\mathbf{r}}(\mathcal{A}) := \mathcal{S}^{H,\mathcal{A}}_{\boldsymbol{\Theta},\mathbf{x},\mathbf{r}}.$$

At the core of our multi-round result will be the following technical lemma, which generalizes Lemma 1.

Lemma 4 *Let $\mathcal{A}$ be a q-query oracle quantum algorithm. Then, for any function $H : \mathcal{X} \to \mathcal{Y}$, any $\mathbf{x} \in \mathcal{X}^n$ and $\boldsymbol{\Theta}^n \in \mathcal{Y}^n$, and any projection $\Pi_{\mathbf{x},\boldsymbol{\Theta}}$, it holds that*

$$\frac{\left\| \left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\boldsymbol{\Theta}}\right) \mathcal{A}^{H*\boldsymbol{\Theta}\mathbf{x}} |\phi_0\rangle \right\|_2^2}{(2q+1)^{2n}} \leq \mathop{\mathbb{E}}_{\mathbf{r}} \left[\left\| \left(|\mathbf{x}\rangle\langle\mathbf{x}|_A \otimes \Pi_{\mathbf{x},\boldsymbol{\Theta}}\right) \mathcal{S}^H_{\mathbf{r}}(\mathcal{A}) |\phi_0\rangle \right\|_2^2 \right].$$

Proof. The proof is by induction on n, where the base case is given by Lemma 1.

For the induction step we first apply the base case, substituting x_n for x_1, Θ_n for Θ_1, r_n for r_1, $H*\overline{\boldsymbol{\Theta}}\overline{\mathbf{x}}$ for H, and $\hat{\Pi}_{x_n,\Theta_n}$ for Π_{x_1,Θ_1}, where

$$\hat{\Pi}_{x_n,\Theta_n} = |x_1\rangle\langle x_1| \otimes \ldots \otimes |x_{n-1}\rangle\langle x_{n-1}| \otimes \Pi_{\mathbf{x},\boldsymbol{\Theta}}$$

to obtain

$$\begin{aligned} &\frac{\left\| \left(|x_n\rangle\langle x_n| \otimes \hat{\Pi}_{x_n,\Theta_n}\right) \mathcal{A}^{\left(H*\overline{\boldsymbol{\Theta}}\overline{\mathbf{x}}\right)*\Theta_n x_n} |\phi_0\rangle \right\|_2^2}{(2q+1)^2} \\ &\leq \mathop{\mathbb{E}}_{r_n} \left[\left\| \left(|x_n\rangle\langle x_n|_A \otimes \hat{\Pi}_{x_n,\Theta_n}\right) \mathcal{S}^{H*\overline{\boldsymbol{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A}) |\phi_0\rangle \right\|_2^2 \right] \end{aligned}$$

which we can write as

$$\frac{\left\| \left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\boldsymbol{\Theta}}\right) \mathcal{A}^{H*\boldsymbol{\Theta}\mathbf{x}} |\phi_0\rangle \right\|_2^2}{(2q+1)^{2n}} \leq \frac{\mathbb{E}_{r_n} \left[\left\| \left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\boldsymbol{\Theta}}\right) \mathcal{S}^{H*\overline{\boldsymbol{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A}) |\phi_0\rangle \right\|_2^2 \right]}{(2q+1)^{2(n-1)}} \tag{3}$$

[7] Looking ahead, in Section 4.2 we will force $\mathcal{A}^H$ to query, and thus $\mathcal{S}$ to extract, $x_1, \ldots, x_n$ in the *right* order by requiring x_2 to contain $H(x_1)$ as a substring, x_3 to contain $H(x_2)$ as a substring, etc. This will be important for the the multi-round Fiat-Shamir application.

dividing both sides by $(2q+1)^{2(n-1)}$ and swapping registers appropriately (to make sure that the register which contains x_n comes after the others).

Now fix r_n. We define

$$\hat{\Pi}_{\overline{\mathbf{x}},\overline{\mathbf{\Theta}}} := |x_n\rangle\langle x_n| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}.$$

and apply the induction hypothesis for $n-1$, substituting $\mathcal{S}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A})$ for $\mathcal{A}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}$, and $\hat{\Pi}_{\overline{\mathbf{x}},\overline{\mathbf{\Theta}}}$ for $\Pi_{\overline{\mathbf{x}},\overline{\mathbf{\Theta}}}$, in order to derive

$$\begin{aligned}\frac{\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}\right)\mathcal{S}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A})|\phi_0\rangle\right\|_2^2}{(2q+1)^{2(n-1)}} &= \frac{\left\|\left(|\overline{\mathbf{x}}\rangle\langle\overline{\mathbf{x}}| \otimes \hat{\Pi}_{\overline{\mathbf{x}},\overline{\mathbf{\Theta}}}\right)\mathcal{S}^{H*\overline{\mathbf{\Theta}}\overline{\mathbf{x}}}_{r_n}(\mathcal{A})|\phi_0\rangle\right\|_2^2}{(2q+1)^{2(n-1)}} \\ &\leq \mathop{\mathbb{E}}_{\overline{\mathbf{r}}}\left[\left\|\left(|\overline{\mathbf{x}}\rangle\langle\overline{\mathbf{x}}| \otimes \hat{\Pi}_{\overline{\mathbf{x}},\overline{\mathbf{\Theta}}}\right)\mathcal{S}^{H}_{\overline{\mathbf{r}}}(\mathcal{S}_{r_n}(\mathcal{A}))|\phi_0\rangle\right\|_2^2\right] \\ &= \mathop{\mathbb{E}}_{\overline{\mathbf{r}}}\left[\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}\right)\mathcal{S}^{H}_{\mathbf{r}}(\mathcal{A})|\phi_0\rangle\right\|_2^2\right].\end{aligned}$$

Since this inequality holds for any fixed r_n, it also holds in expectation over r_n. Substituting it in Equation 3, we retrieve the statement of the lemma. □

Remark 5 In case of $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X}^n$ *without duplicate entries*, it follows from the resulting mutual orthogonality of the projections X_j and the definition of $\mathcal{S}^H_{\mathbf{r}}(\mathcal{A})$ that the following holds. The term in the expectation $\mathbb{E}_{\mathbf{r}}$ in the inequality of Lemma 4 vanishes for any $\mathbf{r} = (\mathbf{i}, \mathbf{b})$ for which there exist two distinct coordinates $j \neq k$ with $i_j = i_k$. As such, we may well understand this expectation to be over $\mathbf{r} = (\mathbf{i}, \mathbf{b})$ for which $i_j \neq i_k$ whenever $j \neq k$; this only increases the expectation.[8] In other words, we may assume that random *distinct* queries are measured in order to extract $x_1, \ldots, x_n$.

Theorem 6 (Measure-and-reprogram, multiple inputs) *Let n be a positive integer, and let $\mathcal{X}, \mathcal{Y}$ be finite non-empty sets. There exists a black-box polynomial-time $(n+1)$-stage quantum algorithm $\mathcal{S}$ with the syntax as outlined at the start of this section, satisfying the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : \mathcal{X} \to \mathcal{Y}$ and that outputs a tuple $\mathbf{x} \in \mathcal{X}^n$ and a (possibly quantum) output z. Then, for any $\mathbf{x}^\circ \in X^n$* without duplicate entries *and for any predicate V:*

$$\begin{aligned}&\Pr_{\mathbf{\Theta}}\left[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, \mathbf{\Theta}, z) : (\pi, \pi(\mathbf{x}), z) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \pi(\mathbf{\Theta})\rangle\right] \\ &\qquad \geq \frac{1}{(q+1)^{2n}} \Pr_{H}\left[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, H(\mathbf{x}), z) : (\mathbf{x}, z) \leftarrow \mathcal{A}^H\right].\end{aligned}$$

Proof. We consider the inequality of Lemma 4 with the expectation over $\mathbf{r}$ understood as in Remark 5. Additionally taking the expectation over H and $\mathbf{\Theta}$ on both sides, we obtain

$$\mathop{\mathbb{E}}_{H,\mathbf{\Theta}}\left[\frac{\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}\right)\mathcal{A}^{H*\mathbf{\Theta}\mathbf{x}}|\phi_0\rangle\right\|_2^2}{(2q+1)^{2n}}\right] \leq \mathop{\mathbb{E}}_{H,\mathbf{\Theta},\mathbf{r}}\left[\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}\right)\mathcal{S}^{H}_{\mathbf{r}}(\mathcal{A})|\phi_0\rangle\right\|_2^2\right]$$

and note that this is equivalent to

$$\mathop{\mathbb{E}}_{H}\left[\frac{\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},H(\mathbf{x})}\right)\mathcal{A}^{H}|\phi_0\rangle\right\|_2^2}{(2q+1)^{2n}}\right] \leq \mathop{\mathbb{E}}_{H,\mathbf{\Theta},\mathbf{r}}\left[\left\|\left(|\mathbf{x}\rangle\langle\mathbf{x}| \otimes \Pi_{\mathbf{x},\mathbf{\Theta}}\right)\mathcal{S}^{H}_{\mathbf{r}}(\mathcal{A})|\phi_0\rangle\right\|_2^2\right].$$

since all values Θ_j and $H(x_j)$ have the same distribution. The term $\mathcal{S}^H_{\mathbf{r}}(\mathcal{A})|\phi_0\rangle = \mathcal{S}^{H,\mathcal{A}}_{\mathbf{\Theta},\mathbf{x},\mathbf{r}}|\phi_0\rangle$ corresponds to the output of the simulator that uses oracle access to H to run $\mathcal{A}$ on an initial state $|\phi_0\rangle$, while measuring queries i_j (finding x_j as the outcome) and reprogramming the oracle at x_j to Θ_j from the $(i_j + b_j)$-th query onwards, with $(i_j, b_j) = r_j$.

Next, we note that the value of the right hand side does not change [Zha12] when instead of giving $\mathcal{S}$ oracle access to H, we let it choose a random instance from a family of $2q$-wise[9]

[8] One might try to exploit this actual improvement in the bound; however, for typical choices of parameters, with n a small constant and q large, this is insignificant.

[9] It is easy to see that the result of [Zha12] also holds for controlled-query algorithms. Alternatively, the q controlled queries can be simulated using $q+1$ plain queries, and a $2(q+1)$-wise independent function can be used.

independent hash functions to simulate $\mathcal{A}$ on. The choice of $\mathbf{r}$ uniquely determines the permutation π with the property $i_{\pi(1)} < \cdots < i_{\pi(n)}$; by definition of $\mathcal{S}^{H,\mathcal{A}}_{\mathbf{\Theta},\mathbf{x},\mathbf{r}}$, the values $\mathbf{x} = (x_1, \ldots, x_n)$ are then extracted from the adversary's queries in the order $\pi(\mathbf{x}) = (x_{\pi(1)}, \ldots, x_{\pi(n)})$. Since $\mathcal{S}$ chooses this $\mathbf{r}$ itself, we can assume that it includes π in its output. Likewise, the simulator takes as input to every stage — from the second to the $(n+1)$-st — a fresh random value, in the order given by $\pi(\mathbf{\Theta})$. However, by definition of $\Pi_{\mathbf{x},\mathbf{\Theta}}$ the final output of the simulator satisfies the predicate V with respect to the given order (without π), i.e. such that $V(\mathbf{x}, \mathbf{\Theta}, z) = 1$, as is the claim of the theorem. □

4.2 The time-ordered case

In some applications, like the multi-round version of the Fiat-Shamir transformation, we need that the simulator extracts the messages in the right order. This can be achieved by replacing the hash *list* $H(\mathbf{x}) = \big(H(x_1), \ldots, H(x_n)\big)$, consisting of individual hashes, by a hash *chain*, where subsequent hashes depend on previous hashes. Intuitively, this enforces $\mathcal{A}$ to query the oracle in the given order.

Formally, considering a function $H : (\mathcal{X}_0 \cup \mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$ and given a tuple $\mathbf{x} = (x_0, x_1, \ldots, x_n)$ in $\mathcal{X}_0 \times \mathcal{X}^n$, we define the *hash chain* $\mathbf{h}^{H,\mathbf{x}} = \big(h_1^{H,\mathbf{x}}, \ldots, h_n^{H,\mathbf{x}}\big)$ given by

$$h_1^{H,\mathbf{x}} = H(x_0, x_1) \qquad \text{and} \qquad h_i^{H,\mathbf{x}} := H\big(h_{i-1}^{H,\mathbf{x}}, x_i\big)$$

for $2 \le i \le n$.

Theorem 7 (Measure-and-reprogram, enforced extraction order) *Let n be a positive integer, and let $\mathcal{X}_0, \mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box polynomial-time $(n+1)$-stage quantum algorithm $\mathcal{S}$, satisfying the following property. Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm that makes q queries to a uniformly random $H : (\mathcal{X}_0 \cup \mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$ and that outputs a tuple $\mathbf{x} = (x_0, x_1, \ldots, x_n) \in (\mathcal{X}_0 \times \mathcal{X}^n)$ and a (possibly quantum) output z. Then, for any $\mathbf{x}^\circ \in (\mathcal{X}_0 \times \mathcal{X}^n)$ without duplicate entries and for any predicate V:*

$$\begin{aligned}&\Pr_{\mathbf{\Theta}}\big[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, \mathbf{\Theta}, z) : (\mathbf{x}, z) \leftarrow \langle \mathcal{S}^A, \mathbf{\Theta}\rangle\big] \\ &\ge \frac{n!}{(q+n+1)^{2n}} \Pr_H\big[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, \mathbf{h}^{H,\mathbf{x}}, z) : (\mathbf{x}, z) \leftarrow \mathcal{A}^H\big] - \epsilon_{\mathbf{x}^\circ}\,.\end{aligned}$$

where $\epsilon_{\mathbf{x}^\circ}$ is equal to $\frac{n!}{|\mathcal{Y}|}$ when summed over all $\mathbf{x}^\circ$.

Remark 8 The additive error term $n!/|\mathcal{Y}|$ stems from the fact that the extraction in the right order fails if $\mathcal{A}$ succeeds in guessing one (or more) of the hashes in the hash chain. The claimed term can be improved to $(n-1)^2/|\mathcal{Y}| + n!/|\mathcal{Y}|^2$ by doing a more fine-grained analysis, distinguishing between permutations $\pi \neq \mathrm{id}$ that bring 2 elements "out of order" or more. In any case, it can be made arbitrary small by extending the range $\mathcal{Y}$ of H for computing the hash chain.

Proof. First, we note that $V(\mathbf{x}, \mathbf{h}^{H,\mathbf{x}}, z) = V'(\mathbf{v}, H(\mathbf{v}), z)$ for $\mathbf{v} = (v_1, \ldots, v_n)$ given by $v_1 = (x_0, x_1)$ and $v_i = \big(h_{i-1}^{H,\mathbf{x}}, x_i\big) = \big(H(v_{i-1}), x_i\big)$ for $i \ge 2$, and $V'(\mathbf{v}, \mathbf{h}, z) := \big[V(\mathbf{x}, \mathbf{h}, z) \wedge h'_i = h_{i-1} \forall i \ge 2\big]$ for any $\mathbf{v}$ of the form $v_1 = (x_0, x_1)$ and $v_i = \big(h'_i, x_i\big)$ for $i \ge 2$. Next, at the cost of n additional queries, we can extend $\mathcal{A}$ to an algorithm $\mathcal{A}_+$ that actually outputs $(\mathbf{v}, z)$, since $\mathcal{A}_+$ can easily obtain the $H(v_i)$'s by making n queries to H. These observations together give

$$\Pr_H\big[\mathbf{x}{=}\mathbf{x}^\circ \wedge V(\mathbf{x}, \mathbf{h}^{H,\mathbf{x}}, z) : (\mathbf{x}, z) \leftarrow \mathcal{A}^H\big] = \Pr_H\big[\mathbf{x}{=}\mathbf{x}^\circ \wedge V'(\mathbf{v}, H(\mathbf{v}), z) : (\mathbf{v}, z) \leftarrow \mathcal{A}_+^H\big]\,.$$

Let $\mathbf{v}^\circ = (v_1^\circ, \ldots, v_n^\circ)$ with $v_i^\circ := (h_i^\circ, x_i^\circ)$, where $h_1^\circ = x_0^\circ$ and $h_i^\circ \in \mathcal{Y}$ is arbitrary but fixed for $i \ge 2$. Let $\mathbf{\Theta}$ be uniformly random in $\mathcal{Y}^n$. An application of Theorem 6 yields a simulator $\hat{\mathcal{S}}$ with

$$\begin{aligned}&\Pr_{\mathbf{\Theta}}\big[\mathbf{v}{=}\mathbf{v}^\circ \wedge V'(\mathbf{v}, \mathbf{\Theta}, z) : (\pi, \pi(\mathbf{v}), z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\mathbf{\Theta})\rangle\big] \\ &\qquad \ge \frac{1}{(q+n+1)^{2n}} \Pr_H\big[\mathbf{v}{=}\mathbf{v}^\circ \wedge V'(\mathbf{v}, H(\mathbf{v}), z) : (\mathbf{v}, z) \leftarrow \mathcal{A}_+^H\big]\,.\end{aligned}$$

Summing both sides of the inequality over h_i° for $i \geq 2$ yields

$$\begin{aligned}
&\Pr_{\boldsymbol{\Theta}}\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z) : (\pi,\pi(\mathbf{v}),z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle\right] \\
&\qquad \geq \frac{1}{(q+n+1)^{2n}} \Pr_H\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},H(\mathbf{v}),z) : (\mathbf{v},z) \leftarrow \mathcal{A}_+^H\right] \\
&\qquad = \frac{1}{(q+n+1)^{2n}} \Pr_H\left[\mathbf{x}=\mathbf{x}^\circ \wedge V(\mathbf{x},\mathbf{h}^{H,\mathbf{x}},z) : (\mathbf{x},z) \leftarrow \mathcal{A}^H\right].
\end{aligned} \tag{4}$$

Recalling its construction, the simulator $\hat{\mathcal{S}}^{\mathcal{A}_+}$ begins by sampling a uniformly random permutation π, so we can write

$$\begin{aligned}
&\Pr_{\boldsymbol{\Theta}}\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z) : (\pi,\pi(\mathbf{v}),z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle\right] \\
&\quad = \frac{1}{n!}\sum_{\sigma\in S_n} \Pr_{\boldsymbol{\Theta}}\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z) : (\pi,\pi(\mathbf{v}),z) \leftarrow \langle \hat{\mathcal{S}}^{\mathcal{A}_+}, \pi(\boldsymbol{\Theta})\rangle \middle| \pi=\sigma\right].
\end{aligned} \tag{5}$$

By definition, the predicate $V'(\mathbf{v},\boldsymbol{\Theta},z)$ (with $\mathbf{v}$ of the form as explained above) is false whenever there exists an $i \geq 2$ such that $h_i \neq \Theta_{i-1}$. Now suppose that $\pi \neq \mathrm{id}$, then there must be some j such that $\pi(j) < \pi(j-1)$. This implies that the first $\pi(j)$ stages of $\hat{\mathcal{S}}^{\mathcal{A}_+}$ which together (in the $\pi(j)$-th stage) produce $v_j = (h_j, x_j)$ are independent of Θ_{j-1}, since Θ_{j-1} is given as input only at the *later* stage $\pi(j-1)$. We thus have the following, taking it as understood, here and in the sequel, that the random variables $\pi, \mathbf{v}, \boldsymbol{\Theta}$ and z are as in (5).

$$\Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z)\middle|\pi\neq\mathrm{id}\right] \leq \Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge h_j=\Theta_{j-1}|\pi\neq\mathrm{id}\right] = \frac{\Pr\left[\mathbf{x}=\mathbf{x}^\circ|\pi\neq\mathrm{id}\right]}{|\mathcal{Y}|}.$$

Using Equation (5), we can bound

$$\frac{1}{n!}\sum_{\sigma\in S_n}\Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z)\middle|\pi=\sigma\right] \leq \frac{1}{n!}\Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z)\middle|\pi=\mathrm{id}\right] + \frac{\Pr\left[\mathbf{x}=\mathbf{x}^\circ|\pi\neq\mathrm{id}\right]}{|\mathcal{Y}|}.$$

We note that by definition of V',

$$\Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge V(\mathbf{x},\boldsymbol{\Theta},z)\middle|\pi=\mathrm{id}\right] \geq \Pr\left[\mathbf{x}=\mathbf{x}^\circ \wedge V'(\mathbf{v},\boldsymbol{\Theta},z)\middle|\pi=\mathrm{id}\right].$$

Furthermore, we may define a new simulator $\mathcal{S}$ which takes oracle access to $\mathcal{A}$ and turns it into $\mathcal{A}_+$, and always chooses $\pi = \mathrm{id}$ instead of a random permutation. Where $\hat{\mathcal{S}}$ would output $(\mathbf{v}, z)$, $\mathcal{S}$ ignores the $\mathbf{h}$-part of $\mathbf{v}$ and simply outputs $(\mathbf{x}, z)$. We then have

$$\begin{aligned}
&\Pr_{\boldsymbol{\Theta}}\left[\mathbf{x}=\mathbf{x}^\circ \wedge V(\mathbf{x},\boldsymbol{\Theta},z) : (\mathbf{x},z) \leftarrow \langle \mathcal{S}^{A}, \boldsymbol{\Theta}\rangle\right] \\
&\qquad \geq \frac{n!}{(q+n+1)^{2n}} \Pr_H\left[\mathbf{x}=\mathbf{x}^\circ \wedge V(\mathbf{x},\mathbf{h}^{H,\mathbf{x}},z) : (\mathbf{x},z) \leftarrow \mathcal{A}^H\right] - \epsilon_{\mathbf{x}^\circ}.
\end{aligned}$$

with $\epsilon_{\mathbf{x}^\circ}$ given by $\epsilon_{\mathbf{x}^\circ} := n! \cdot \Pr_{\boldsymbol{\Theta}}\left[\mathbf{x} = \mathbf{x}^\circ|\pi \neq \mathrm{id}\right]/|\mathcal{Y}|$. □

5 The multi-round Fiat-Shamir transformation

A straightforward generalization of the Fiat-Shamir transformation can be applied to arbitrary (i.e., multi-round) public-coin interactive proof systems (PCIP). We show here security of this multi-round Fiat-Shamir transformation in the QROM.

5.1 Public coin interactive proofs and multi-round Fiat-Shamir

We begin by defining PCIPs, mainly to fix notation, and the corresponding multi-round Fiat-Shamir transformation.

Definition 9 (Public coin interactive proof system (PCIP)) *A $(2n+1)$-round public coin interactive proof system (PCIP) $\Pi = (\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L}$ is a $(2n+1)$-round two-party interactive protocol of the form, with $\mathcal{C}$ being a finite non-empty set, and V a predicate:*

Prover $\mathcal{P}(x)$		*Verifier $\mathcal{V}(x)$*
	$\xrightarrow{a_1}$	
	$\xleftarrow{c_1}$	$c_1 \xleftarrow{\$} \mathcal{C}$
	$\vdots$	
	$\xrightarrow{a_n}$	
	$\xleftarrow{c_n}$	$c_n \xleftarrow{\$} \mathcal{C}$
	$\xrightarrow{z}$	Accept iff $V(x, a_1, c_1, ..., a_n, c_n, z) = 1$

Remark 10 If the language $\mathcal{L}$ is definied by means of an (efficiently verifiable) witness relation $R \subseteq \mathcal{X} \times \mathcal{W}$, then the prover typcially gets a witness w for x as an additional input. We then also say that Π is a PCIP *for the relation* R. In case of a $(2n+1)$-round PCIP Π for a witness relation R that is *hard on average*, meaning that there exists an instance generator Gen with the property that for $(w, x) \leftarrow \mathsf{Gen}$ it holds that $(w, x) \in R$, but given x alone it is computationally hard to find w with $(w, x) \in R$, Π is also called an *identification scheme*.

Just as in the ordinary Fiat-Shamir transformation, the interaction used to enforce the time order between the prover committing to the message a_i and receiving the challenge c_i can be replaced by means of a hash function. In addition, we can include the previous challenge (i.e. the previous hash value) in the hash determining the next challenge to enforce the ordering of the n pairs (a_i, c_i) according to increasing i. We thus obtain the following non-interactive proof system.

Definition 11 (Fiat-Shamir transformation for general PCIP (mFS))
Given an $(2n+1)$-round PCIP $\Pi = (\mathcal{P}, \mathcal{V})$ for a language $\mathcal{L}$ and a hash function H with appropriate domain, and range equal to $\mathcal{C}$, we define the non-interactive proof system $\mathsf{FS}[\Pi] = (\mathcal{P}_{FS}^H, \mathcal{V}_{FS}^H)$ as follows. The prover $\mathcal{P}$ outputs

$$(x, a_1, ..., a_n, z) \leftarrow \mathcal{P}_{FS}^H$$

where z and a_i for $i = 1, ..., n$ are computed using $\mathcal{P}$, and the challenges are computed as

$$\begin{aligned} c_1 &= H(0, x, a_1) \text{ and} \\ c_i &= H(i-1, c_{i-1}, a_i) \text{ for } i = 2, ..., n\,, \end{aligned}$$

The verifier outputs 'accept' iff $V(x, a_1, c_1, ..., a_n, c_n, z) = 1$ for $c_1 = H(0, x, a_1)$ and $c_i = H(i-1, c_{i-1}, a_i)$, $i = 2, ..., n$, denoted by $V_{FS}(x, a_1, c_1, ..., a_n, c_n, z) = 1$.

Remark 12 The challenge number i (minus 1) is included in the hash input to ensure that the challenges are generated using distinct inputs to H with probability 1. This is to enable us to apply Theorem 7, which only holds for duplicate-free lists of hash inputs. In fact, any additional strings can be included in the argument when computing c_i using H, without influencing the security properties of the non-interactive proof system in a detrimental way. In the literature one sometimes sees that the entire previous transcript is hashed (in which case the counter number i may then be omitted).

5.2 General security of multi-round Fiat-Shamir in the QROM

When constructing a reduction for mFS, this reduction is participating as a prover in the underlying PCIP, and is hence only provided with random challenges one at a time. We thus need the special simulator from Theorem 7, which always outputs the corresponding messages in the right order. The success of this simulator is based on the very essence of the Fiat-Shamir transformation,

namely the fact that the intractability of the hash function takes the role of the interaction in enforcing a time order in the transcript of the PCIP.

The security of the multi-round Fiat-Shamir transformation follows as a simple Corollary of Theorem 7.

Corollary 13 *There exists a black-box quantum polynomial-time $(n+1)$-stage quantum algorithm $\mathcal{S}$ such that for any adaptive adversary $\mathcal{A}$ against the multi-round Fiat-Shamir transformed version $\mathsf{FS}[\Pi]$ of a $(2n+1)$-round PCIP Π, making q queries to a uniformly random function H with appropriate domain and range equal $\mathcal{C}$, and for any $x^\circ \in \mathcal{X}$:*

$$\begin{aligned}&\Pr\left[x = x^\circ \wedge v = \mathit{accept} : (x,v) \leftarrow \langle \mathcal{S}^{\mathcal{A}}, \mathcal{V}\rangle\right] \\ &\quad \geq \frac{n!}{(2q+n+1)^{2n}} \Pr_H\left[x = x^\circ \wedge V_{FS}^H(x,\pi) : (x,\pi) \leftarrow \mathcal{A}^H\right] - \epsilon_{x^\circ}\,.\end{aligned}$$

where the additive error term ϵ_{x° is equal to $\frac{n!}{|\mathcal{C}|}$ when summed over all x°.

Proof. We may simply set $\mathbf{x}^\circ = (x^\circ, (0,a_1), \ldots, (n-1,a_n))$ for arbitrary $a_1,\ldots,a_n$, apply Theorem 7 and then sum over all choices of $a_1,\ldots,a_n$ to obtain the claimed inequality. Note that the round indices ensure that every such $\mathbf{x}^\circ$ is duplicate free, satisfying the corresponding requirement of Theorem 7.

Note that the additive error terms reflect the fact that the random oracle only *approximately* succeeds in enforcing the original time order in the transcript of the PCIP. However, it can be made arbitrarily small, as discussed below.

Remark 14 There exist PCIPs with soundness error much smaller than $1/|\mathcal{C}|$. As an example, consider the sequential repetition of a Σ-protocol with special soundness. Here, the soundness error is $1/|\mathcal{C}|^n$. In this case, the term proportional to $1/|\mathcal{C}|$ renders the bound from the above theorem trivial. Note however, that (i) this situation is extremely artificial, as there is absolutely no reason to repeat sequentially instead of in parallel, and (ii) the additive error term can be made arbitrarily small by considering a variant Π' of Π where the random challenges are enlarged with a certain number of bits that are ignored otherwise, see Remark 8.

In fact, we suspect that the observation from (i) is true in a much broader sense: if a PCIP still has negligible soundness error when allowing the adversary to learn one of the challenges c_i in advance of sending the corresponding commitment-type message a_i, it seems like the number of rounds can be reduced and the loss in soundness error can be won back by parallel repetition.

As for the case of the Fiat-Shamir transformation for Σ-protocols, the general reduction implies that security properties that protect against dishonest provers carry over from the interactive to the non-interactive proof system. For a definition of the properties considered in the following theorem, see, e.g. [DFMS19]. The quantum proof-of-knowledge-property was intoduced in [Unr12].

Corollary 15 (Preservation of Soundness/PoK) *Let Π be a constant-round PCIP that has (statistical/computational) soundness, and/or the (statistical/computational) quantum proof-of-knowledge-property, respectively. Then, in the QROM, $\mathsf{FS}[\Pi]$ has (statistical/computational) soundness, and/or the (statistical/computational) quantum proof-of-knowledge-property, too.*

Proof. Corollary 13 turns any dishonest prover $\mathcal{A}_{\mathsf{FS}[\Pi]}$ for $\mathsf{FS}[\Pi]$ with success probability ϵ into a dishonest prover $\mathcal{A}_\Pi$ for Π, with success probability $\epsilon \cdot (2q+1)^{-2n}$, where $2n+1$ is the number of rounds in Π. Since n is constant and q is polynomial in the security parameter, the success probabilities of the respective provers are polynomially related. The claimed implications follow now using the same arguments as in Corollaries 13 and 16 in [DFMS19]. □

6 Tightness of the reductions

Here, we show tightness of our results. We start with proving tightness of Theorems 2 and 3 (up to essentially a factor 4). This implies that a $O(q^2)$-loss is unavoidable in general. Indeed, the following result shows that for a large and natural class of Σ-protocols Σ, there exists an attack

against $\mathsf{FS}[\Sigma]$ that succeeds with a probability q^2 times larger than the best attack against Σ. The attack is based on an application of Grover's quantum algorithm for unstructured search.

To our surprise, we could not find an analysis of Grover's algorithm in the regime we require in the literature. Grover search has been analyzed in the case of an unknown number of solutions [BBHT98], but the focus of that work is on analyzing the expected number of queries required to find a solution, while we analyze the probability with which the Grover search algorithm succeeds for a *fixed but arbitrary* number of queries.

Theorem 16 *Let $\mathcal{L}$ be a language, and let Σ be a Σ-protocol for $\mathcal{L}$ with challenge set $\mathcal{C}$, special soundness and perfect honest-verifier zero-knowledge. Furthermore, we assume that the triples (a, c, z) produced by the simulator $\mathcal{S}_{\mathrm{ZK}}(x)$ are always accepted by the verifier even for instances $x \notin \mathcal{L}$, and that a has min-entropy γ.*[10] *Then for any q such that $(q^2+1)\cdot e^2 \cdot (5q)^6 < |\mathcal{C}|$ and $2^\gamma/(5q)^3 > 2$, there exists a q-query dishonest prover that succeeds with probability $q^2/|\mathcal{C}|$ in producing a valid $\mathsf{FS}[\Sigma]$-proof for an instance $x \notin \mathcal{L}$.*

The idea of the attack against $\mathsf{FS}[\Sigma]$ is quite simple. For a Σ-protocol that is *special* honest-verifier zero-knowledge, meaning that the simulation works by first sampling the challenge c and the repsonse z and then computing a fitting answer a as a function $a(c, z)$, one simply does a Grover search to find a pair (c, z) for which $H\big(x, a(c, z)\big) = c$. For a typical H, this will give a quadratic improvement over the classical search, which, for a random H, succeeds with probability $q/|\mathcal{C}|$ (due to the special soundness). A subtle issue is that, for some (unlikely) choices of H, there are actually *many* (c, z) for which $H\big(x, a(c, z)\big) = c$, in which case the Grover search "overshoots". In the formal proof below, this is dealt with by controlling the probability of H having this (unlikely) property. Also, it removes the *special* honest-verifier zero-knowledge property by doing the Grover search over the randomness of the simulator, which requires some additional caution.

Remark 17 It is not hard to see that Theorem 16 still holds in the following two variations of the statement. (1) $H(x, a)$ is random and independent for different choices of a, but is *not* necessarily independent for different choices of x. (2) The Σ-protocol Σ is replaced by Σ', which has its challenge enlarged with a certain number of bits that are ignored otherwise, in line with Remark 14, and $\mathsf{FS}[\Sigma']$ then uses an H with a correspondingly enlarged range.[11]

Proof. Let $\mathcal{S}_{\mathrm{ZK}}$ be the zero-knowledge simulator given by the perfect honest-verifier zero-knowledge property of Σ. Consider an adversary $\mathcal{A}_{FS}$ against $\mathsf{FS}[\Sigma]$, that works as follows for an arbitrary instance $x \notin \mathcal{L}$:

- Define the function $f^H : R \to \{0, 1\}$ (where R is the set of random coins for $\mathcal{S}_{\mathrm{ZK}}$) as

$$f^H(\rho) = \begin{cases} 1 & \text{for } \mathcal{S}_{\mathrm{ZK}}(x;\rho) \to (a, c, z) \wedge H(x||a) = c \\ 0 & \text{otherwise.} \end{cases}$$

- Use Grover's algorithm for q steps, to try and find ρ s.t. $f(\rho) = 1$
- Run $\mathcal{S}_{\mathrm{ZK}}(x;\rho) \to (a, c, z)$ and output $(x, a||z)$.

Let p_1^H be the fraction of random coins from R that map to 1 under f^H. Note that by the special soundness of Σ, in any accepting triple a determines c and we thus have $\mathbb{E}_H[p_1^H] = \frac{1}{|\mathcal{C}|}$. By the way Grover works, after q iterations (requiring q queries to H) the probability p_2^H of finding such an input is $\sin^2((2q+1)\Theta^H)$, where $0 \le \Theta^H \le \pi/2$ is such that $\sin^2(\Theta^H) = p_1^H$. Now as long as Θ is not too large to begin with (i.e. as long as the Grover search will not 'overshoot'), p_2^H is approximately a factor q^2 larger than p_1^H. Our goal will be to show that also on average over H,

[10] These additional assumptions on the simulator could be avoided, but they simplify the proof. Furthermore, for typical Σ-protocols they are satisfied. In particular, the simulated transcripts for hard instances are accepted by the verifier with high probability. Otherwise, the two polynomial-time algorithms could otherwise be used to solve the hard instances, a contradiction.

[11] While (1) follows by inspecting the proof, (2) holds more generically: the dishonest prover attacking $\mathsf{FS}[\Sigma']$ simply runs the prover attacking $\mathsf{FS}[\Sigma]$ but enlarges the output register of the hash queries, with the corresponding state being set to be the fully mixed state in each query, and then dismisses these additional qubits again.

the improvement is at least q^2. To this end we define $H_{\text{bad}} := \{H : p_1^H > \sin^2(\frac{\pi}{6q+3})\}$ and H_{good} its complement. Then,

$$\begin{aligned}\mathop{\mathbb{E}}_H[p_2^H] &= (1-\alpha)\cdot \mathop{\mathbb{E}}_H\left[p_2^H | H \in H_{\text{good}}\right] + \alpha \cdot \mathop{\mathbb{E}}_H\left[p_2^H | H \in H_{\text{bad}}\right] \\ &\geq (1-\alpha)\cdot \mathop{\mathbb{E}}_H\left[p_2^H | H \in H_{\text{good}}\right]\end{aligned}$$

where $\alpha = \Pr_H[H \in H_{\text{bad}}]$ and $1-\alpha = \Pr_H[H \in H_{\text{good}}]$.

We first compute $\mathbb{E}_{H_{\text{good}}}\left[p_2^H\right]$. Let $H \in H_{\text{good}}$. We have $(2q+1)\Theta^H \leq \frac{\pi}{3}$. Since $\frac{\mathrm{d}}{\mathrm{d}\Theta}\sin(\Theta) = \cos(\Theta) \geq 1/2$ for $\Theta \in [0, \frac{\pi}{3}]$, and $\Theta \geq \sin(\Theta)$, it follows that

$$\sin((2q+1)\cdot\Theta^H) \quad \geq \quad \sin(\Theta^H) + \frac{2q\cdot\Theta^H}{2} \quad \geq \quad (q+1)\cdot\sin(\Theta^H).$$

Using $\sin(\Theta) \geq 0$ for $\Theta \in [0, \frac{\pi}{3}]$, we obtain

$$p_2^H = \sin^2((2q+1)\cdot\Theta^H) \geq (q+1)^2\cdot\sin^2(\Theta^H) = (q+1)^2\cdot p_1^H.$$

Therefore,

$$\begin{aligned}\mathop{\mathbb{E}}_H[p_2^H] \quad &\geq \quad \mathop{\mathbb{E}}_H\left[p_2^H | H \in H_{\text{good}}\right]\cdot \Pr_H[H \in H_{\text{good}}] \\ &\geq \quad (q+1)^2\cdot \mathop{\mathbb{E}}_H\left[p_1^H | H \in H_{\text{good}}\right]\cdot \Pr_H[H \in H_{\text{good}}] \\ &\geq \quad (q+1)^2\cdot\left(\mathop{\mathbb{E}}_H[p_1^H] - \Pr_H[H \in H_{\text{bad}}]\right).\end{aligned} \tag{6}$$

Next we bound $\alpha = \Pr_H[H \in H_{\text{bad}}] = \Pr_H[p_1^H > \sin^2(\frac{\pi}{6q+3})]$. Note that for p_1^H to be large, we need that for many first messages a, $H(a)$ must be the unique challenge c for which there exist an accepting response. For a random H this is unlikely to happen. Formally, we argue as follows, using the Chernoff bound eventually.

We first define the following equivalence relation:

$$\rho \sim \rho' \text{ iff } \mathcal{S}_{\text{ZK}}(\rho) = (a, c, z) \wedge \mathcal{S}_{\text{ZK}}(\rho') = (a, c', z') \text{ for } \rho, \rho' \in R.$$

$R/_\sim$ then denotes the set of equivalence classes $[\rho] = \{\rho' \in R \,|\, \rho \sim \rho'\}$. By the perfect special soundness property and the assumptions on $\mathcal{S}_{\text{ZK}}$, we have that a determines c (remember that $x \notin \mathcal{L}$), and therefore f^H is constant on elements within a given equivalence class. Thus, $f^H : R/_\sim \to \{0,1\}$. For two distinct equivalence classes $[\rho] \neq [\rho']$, we have

$$\Pr_H[f^H([\rho]) = 1 \wedge f^H([\rho']) = 1] = \Pr_H[f^H([\rho]) = 1]\cdot\Pr_H[f^H([\rho']) = 1],$$

since $H(x||a)$ is chosen independently for different a. Finally, taking $X^H := \sum_{[\rho]} f^H([\rho])$ we have

$$\begin{aligned}p_1^H &= \Pr_\rho[f^H(\rho) = 1] = \frac{\sum_\rho f(\rho)}{|R|} \\ &= \frac{\sum_{[\rho]}\left(f^H([\rho])\cdot|[\rho]|\right)}{|R|} \leq \frac{|[\rho_{\max}]|\cdot\sum_{[\rho]} f^H([\rho])}{|R|} = X^H\cdot 2^{-\gamma}\end{aligned}$$

where $[\rho_{\max}]$ is the $[\rho]$ that maximizes $|[\rho]|$. It follows that

$$\begin{aligned}\alpha &= \Pr_H[p_1^H > \sin^2\left(\frac{\pi}{6q+3}\right)] \\ &\leq \Pr_H\left[X^H > \sin^2\left(\frac{\pi}{6q+3}\right)\cdot 2^\gamma\right] \leq \Pr_H\left[X^H > \frac{2^\gamma}{|\mathcal{C}|} + \frac{2^\gamma}{(5q)^3}\right]\end{aligned}$$

where we used $\sin^2(x) > x^3$ for $0 \leq x \leq 0.80$ and $\frac{\pi}{6q+3} > \frac{1}{5q} + \sqrt[3]{\frac{1}{|\mathcal{C}|}}$ for $|\mathcal{C}| > (5q)^3$ in the last inequality. By definition of f, for any $[\rho]$ we have $\Pr_H[f(\rho) = 1] = \frac{1}{|\mathcal{C}|}$, hence

$$\mathop{\mathbb{E}}_H[X] = \sum_{[\rho]}\mathop{\mathbb{E}}_H[f^H([\rho])] = \sum_{[\rho]}\Pr_H[f^H([\rho]) = 1] = \frac{|R/_\sim|}{|\mathcal{C}|} \geq \frac{2^\gamma}{|\mathcal{C}|}.$$

We use the following Chernoff bound:

$$\Pr_H\left[X^H > (1+\delta)\cdot \mathop{\mathbb{E}}_H\left[X^H\right]\right] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mathbb{E}_H\left[X^H\right]} < \left(\frac{e^{1+\delta}}{\delta^{1+\delta}}\right)^{\mathbb{E}_H\left[X^H\right]} = \left(\frac{e}{\delta}\right)^{\mathbb{E}_H\left[X^H\right]\cdot(1+\delta)}.$$

Setting $\delta := \frac{|\mathcal{C}|}{(5q)^3}$, together with the inequalities derived above this leads to

$$\alpha \leq \left(\frac{e\cdot(5q)^3}{|\mathcal{C}|}\right)^{\frac{2^\gamma}{|\mathcal{C}|}+\frac{2^\gamma}{(5q)^3}} < \frac{e^2\cdot(5q)^6}{|\mathcal{C}|^2} < \frac{1}{|\mathcal{C}|\cdot(q^2+1)}$$

where we used $\frac{2^\gamma}{(5q)^3} > 2$ in the second to last, and $|\mathcal{C}| > (q^2+1)\cdot e^2\cdot(5q)^6$ in the last inequality. Plugging this bound into Equation 6, we get

$$\mathop{\mathbb{E}}_H[p_2^H] \geq (q^2+1)\cdot\left(p_1 - \frac{1}{|\mathcal{C}|\cdot(q^2+1)}\right) = \frac{q^2}{|\mathcal{C}|} + \frac{1}{|\mathcal{C}|} - \frac{1}{|\mathcal{C}|} = \frac{q^2}{|\mathcal{C}|}.$$

Thus, the success probability of our adversary $\mathcal{A}_{FS}$ after making q queries to H is at least $\frac{q^2}{|\mathcal{C}|}$. □

The tightness of Corollary 13 follows from the above tightness result for the case of Σ-protocols in a fairly straightforward manner.

Theorem 18 *For every positive integer n, there exists a $(2n+1)$-round PCIP $\mathsf{\Pi}$ with soundness error ϵ and challenge space $\mathcal{C}$ such that $|\mathcal{C}| \geq 1/\epsilon$ and such that there exists a q-query dishonest prover $\mathcal{A}$ on $\mathsf{FS}(\mathsf{\Pi})$ with success probability $n^{-2n}q^{2n}\epsilon$.*

Before proving the theorem, we show how it implies the tightness of Theorem 13.

Corollary 19 *The security loss in the bound in Corollary 13 is optimal, up to a multiplicative factor that depends on n only.*

Proof. Let $\mathsf{\Pi}$ be a PCIP as shown to exist in Theorem 18. Let ϵ_{Π}, and $\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q)$, be the soundness error of $\mathsf{\Pi}$, and the one of its Fiat Shamir transformation against q-query adversaries, respectively. By Theorem 18,

$$\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) \geq n^{-2n}q^{2n}\epsilon_{\mathsf{\Pi}}. \tag{7}$$

Theorem 13, on the other hand, yields

$$\epsilon_{\mathsf{\Pi}} \geq \frac{n!}{(2q+n+1)^{2n}}\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) - \frac{n!}{|\mathcal{C}|} \tag{8}$$

$$\geq \frac{n!}{(2q+n+1)^{2n}}\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) - n!\epsilon_{\mathsf{\Pi}}, \tag{9}$$

where we used the condition on the challenge space size from Theorem 18 in the last line. Rearranging terms we obtain

$$\epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) \leq (2q+n+1)^{2n}\left(1+\frac{1}{n!}\right)\epsilon_{\mathsf{\Pi}}(q) \tag{10}$$

$$\leq 2(n+3)^2q^{2n}\epsilon_{\mathsf{\Pi}}(q), \tag{11}$$

where we have used $1 \leq q$ in the last line. In summary, we have constants $c_1 = n^{-2n}$ and $c_2 = 2(n+3)^{2n}$ such that

$$c_1q^{2n}\epsilon_{\mathsf{\Pi}} \leq \epsilon_{\mathsf{FS}(\mathsf{\Pi})}(q) \leq c_2q^{2n}\epsilon_{\mathsf{\Pi}}. \tag{12}$$

□

Proof (of Theorem 18). Let $\hat{\Sigma}$ be a Σ-protocol for a language $\mathcal{L}$ fulfilling the requirements of Theorem 16. Let the challenge space be denoted by $\hat{\mathcal{C}}$. Given an arbitrary positive integer, we define an $(2n+1)$-round PCIP Π for the same language $\mathcal{L}$ by means of n sequential independent executions of $\hat{\Sigma}$. Concretely, the $2n+1$ messages of Π are given in terms of the messages $\hat{a}_i, \hat{c}_i$ and $\hat{z}_i$ of the i-th repetition of $\hat{\Sigma}$ as

$$\begin{aligned} a_1 &= \hat{a}_1 \\ c_i &= (\hat{c}_i, r_i) \text{ for } i = 1, ..., n \\ a_i &= (\hat{a}_i, \hat{z}_{i-1}) \text{ for } i = 2, ..., n, \text{ and} \\ z &= \hat{z}_n, \end{aligned}$$

where r_i is an independent random string of arbitrary (but fixed) length, which is ignored otherwise (in line with Remark 14). The purpose of r_i is to make the challenge space $\mathcal{C}$ of Π arbitrary large, as required. The verification procedure of Π simply checks if all the triples $(\hat{a}_i, \hat{c}_i, \hat{z}_i)$ are accepted by $\hat{\Sigma}$. By the special soundness property of $\hat{\Sigma}$, the soundness error of this PCIP is $\epsilon = |\hat{\mathcal{C}}|^{-n}$.

Using Theorem 16, we can attack the Fiat-Shamir transformation of $\hat{\Sigma}$ repeatedly to devise an attack agains $\mathsf{FS}(\Pi)$: first use Theorem 16 to find $\hat{a}_1$ and $\hat{z}_1$, then use it again to find $\hat{a}_2$ and $\hat{z}_2$, etc., having the property that with the correctly computed challenges these form valid triples for an instance $x \notin \mathcal{L}$. In each invocation of Theorem 16 we use a q'-query attack, which then succeeds with probability $q'^2/|\hat{\mathcal{C}}|$. Thus, using in total $q = nq'$ queries, we succeed in breaking $\mathsf{FS}[\Pi]$ with probability $q'^{2n}/|\hat{\mathcal{C}}|^n = n^{-2n}q^{2n}\epsilon$, as claimed.

There are two issues we neglected in the above argument. First, we actually employ Theorem 16 for attacking a *variant* of $\hat{\Sigma}$ that has its challenge enlarged (and thus is not special sound); and, second, the challenge c_i is computed as

$$c_i = H(i-1, ..., H(1, H(0, x, \hat{a}_1), \hat{a}_2), ..., \hat{a}_i)\,,$$

which is *not* a uniformly random function of x and $\hat{a}_i$ (but only of $\hat{a}_i$). However, by Remark 17, the attack from Theorem 16 still applies. □

7 Applications

7.1 Digital signature schemes from multi-round Fiat-Shamir

One of the prime applications of the Fiat-Shamir transformation is the construction of digital signature schemes from interactive identification schemes. In this context, multi-round variants have also been used. An example where a QROM reduction is especially desirable is MQDSS [CHR+16], a candidate digital signature scheme in the ongoing NIST standardization process for post-quantum cryptographic schemes [NIS]. This digital signature scheme is constructed by applying the multi-round Fiat-Shamir transformation to the 5-round identification scheme by Sakumoto, Shirai, and Hiwatari [SSH11] based on the hardness of solving systems of multivariate quadratic equations.

In this section, we present a generic construction of a digital signature scheme based on multi-round FS, and give a proof sketch of its strong unforgeability under chosen message attacks. We refrain from giving a full, self-contained proof here so as to not distract from our main technical result and its implications. Many, though not all, parts of the argument are very similar to the ones made elsewhere for the 3-round case.

The following construction is a straightforward generalization of the original construction of Fiat and Shamir.

Definition 20 (Fiat-Shamir signatures from a general PCIP) *Given an $(2n+1)$-round public coin identification scheme $\Pi = (\mathsf{Gen}, \mathcal{P}, \mathcal{V})$ for a witness relation R and a hash function H with appropriate domain and range equal to $\mathcal{C}$, we define the digital signature scheme $\mathsf{Sig}[\Pi] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ as follows. The key generation algorithm Gen is just the one from Π. The signing algorithm Sign, on input a secret key sk and a message m, outputs*

$$\sigma = (a_1, ..., a_n, z) \leftarrow \mathsf{Sign}_{sk}(m)$$

where z and a_i for $i = 1, ..., n$ are computed using $\mathcal{P}(pk)$, and the challenges are computed as

$$c_1 = H(0, pk, m, a_1) \text{ and}$$
$$c_i = H(i-1, c_{i-1}, a_i) \text{ for } i = 2, ..., n\,.$$

The verification algorithm Verify*, on input a public key pk, a message m and a signature $\sigma = (a_1, ..., a_n, z)$, computes c_i as specified above, outputs 'accept' iff $\mathcal{V}_{pk}(a_1, c_1, ..., a_n, c_n, z) = 1$, denoted by $\mathsf{Verify}_{pk}(m, \sigma) = 1$.*

We note that the above definition is equivalent to the following, alternative formulation: Let $\mathsf{Sign}_{sk}(m)$ produce σ by running $P^H_{FS}(x||m)$, and let $\mathsf{Verify}(m, \sigma)$ be equal to the outcome of $V^H_{FS}(x||m)$, where $(P^H_{FS}, V^H_{FS}) = \mathsf{FS}[\Pi^*]$ and $\Pi^* = (\mathcal{P}^*, \mathcal{V}^*)$ is the identification scheme obtained from Π by setting $\mathcal{P}^*(x||m) = \mathcal{P}(x)$ and $\mathcal{V}^*(x||m) = \mathcal{V}(x)$ for any m. This alternative formulation will be convenient in the proof of Theorem 23.

Remark 21 As in the case of the plain multi-round Fiat-Shamir transformation, one can include arbitrary additional strings in the argument when computing the challenges c_i. Examples where this is done include the MQDSS signature scheme [CHR+16], where the message m and the first commitment a_1 are also included in the argument for computing the second challenge, and Bulletproofs, where the challenges are computed by hashing the entire transcript up to that point [BBB+18].

As an identification scheme is an interactive honest-verifier zero knowledge proof of knowledge of a secret key, the above signature scheme is a a non-interactive zero knowledge proof of knowledge of a secret key according to Corollary 13. For a digital signature scheme, however, the stronger security notion of (strong) unforgeability against chosen message ((s)UF-CMA) attacks is required.

In the following, we give a proof sketch for the fact that the above signature scheme is (s)UF-CMA. This fact follows immediately once we have convinced ourselves that a certain result by Unruh about the Fiat-Shamir transformation holds for the multi-round case as well: For the Fiat-Shamir transformation of Σ-protocols, extractability implies a stronger notion of extractability enabling a proof of (s)UF-CMA [Unr17]. Here, we just patch the parts of the proof from [Unr17] that make use of the fact that the underlying PCIP has only three rounds.

For the following we need the notion of a PCIP having computationally unique responses.

Definition 22 (Computationally unique responses - PCIP) *A $(2n+1)$-round PCIP $\Pi = (\mathcal{P}, \mathcal{V})$ is said to have* computationally unique responses *if given a partial transcript $(x, a_1, c_1, \ldots a_i, c_i)$ it is computationally hard to find two accepting conversations that both extend the partial transcript but differ in (at least) a_{i+1} (here we consider z to be equal to a_{n+1}), i.e. for $con_i = x, a_1, c_1, \ldots a_i, c_i, a^{(j)}_{i+1}, c^{(j)}_{i+1} \ldots, a^{(j)}_n, c^{(j)}_n, z^{(j)}$, $j = 1, 2$ we have that*

$$\Pr\left[\mathcal{V}(con_1) = 1 \wedge \mathcal{V}(con_2) = 1 : (con_1, con_2) \leftarrow \mathcal{A}\right]$$

is negligible for computationally bounded (quantum) $\mathcal{A}$, where $a^{(1)}_{i+1} \neq a^{(2)}_{i+1}$.

Equipped with this definition, we can state the main result of this section.

Theorem 23 ((s)UF-CMA of multi-round FS signatures) *Let Π be a PCIP for some hard relation R, which is a quantum proof of knowledge and satisfies completeness, HVZK, and has unpredictable commitments[12] as well as a superpolynomially large challenge space. Then $\mathsf{Sig}[\Pi]$ is existentially unforgeable under chosen message attack (UF-CMA). If Π in addition has computationally unique responses, $\mathsf{Sig}[\Pi]$ is* strongly *existentially unforgeable under chosen message attack (sUF-CMA).*

In [Unr17] (Theorem 24, and 25, respectively), it is proven that an extractable FS proof system (of an HVZK Σ-protocol, and of an HVZK Σ-protocol with computationally unique responses, respectively) satisfies the stronger notion of *(strong) simulation-sound extractability*. In addition, it

[12] We take unpredictable commitments for PCIP's to be exactly the same as for Σ-protocols, with the first message playing the role of the commitment.

is shown that such a FS proof system gives rise to a (s)UF-CMA signature scheme if the underlying relation is hard. Corollary 15 implies that $\mathsf{FS}[\Pi^*]$ is indeed extractable if Π is extractable. Below we rely on the proof in [Unr17] to argue simulation-sound extractability, only pointing out a particular difference for the multi-round case.

Proof (sketch). Since Π is a quantum proof of knowledge, so is Π^*. By Corollary 15, $\mathsf{FS}[\Pi^*]$ is a quantum proof of knowledge (extractable), and by Theorem 20 in [Unr17] (which easily generalizes to the multi-round setting), completeness, unpredictable commitments[13] and HVZK of Π^* together imply ZK for $\mathsf{FS}[\Pi^*]$. For the proof that $\mathsf{FS}[\Pi^*]$ is also simulation-sound extractable, we refer to the proof of Theorem 24 in [Unr17], noting only that in the hop from Game 1 to Game 2 we have to adjust the argument as follows: Let $\mathcal{S}_{ZK}$ be the zero-knowledge simulator that runs the HVZK simulator from Π^* and reprograms the oracle as necessary. We write H_f for the oracle H after it has been reprogrammed by $\mathcal{S}_{ZK}$, at the end of the run of $\mathcal{A}$. We have to show that $V_{FS}^{H_f}(x, a_1, \ldots, a_n, z) = 1$ implies $V_{FS}^{H}(x, a_1, \ldots, a_n, z) = 1$, where $(x, a_1, \ldots, a_n, z)$ is the final output of $\mathcal{A}$. Suppose the implication does not hold. Then either (i) $H_f(0, x, a_1) \neq H(0, x, a_1)$ or (ii) $H_f(i-1, c_{i-1}, a_i) \neq H(i-1, c'_{i-1}, a_i)$ for some i, where c_{i-1} is the $(i-1)$-st challenge as recomputed by $V_{FS}^{H_f}$ and c'_{i-1} is the one computed by V_{FS}^{H}. In case (i) holds, $\mathcal{A}$ has queried x and the corresponding forged proof that was output by $\mathcal{S}_{ZK}$ starts with a_1. In case (ii), assume that $H_f(j-1, c_{j-1}, a_j) = H(j-1, c_{j-1}, a_j)$ for all $j < i$, so that $c_{i-1} = c'_{i-1}$. Then,

$$H_f(i-1, ..., H(1, H(0, x, a_1), a_2), ..., a_i) \neq H(i-1, ..., H(1, H(0, x, a_1), a_2), ..., a_i)$$

which means that $\mathcal{A}$ either queried x and the corresponding forged proof that was output by $\mathcal{S}_{ZK}$ starts with a_1, or else $\mathcal{A}$ has queried some x' such that

$$\begin{aligned} &H(i-2, \ldots, H(1, H(0, x', a'_1), a'_2), \ldots a'_{i-1}) \\ &\qquad = H(i-2, \ldots, H(1, H(0, x, a_1), a_2), \ldots, a_{i-1}) \end{aligned}$$

and $a_i = a'_i$, where $(a'_1, \ldots, a'_i)$ is part of the $\mathcal{S}_{ZK}$ proof resulting from the query x'. By the fact that H is a random oracle, it is infeasible for $\mathcal{A}$ to find such an x'.

In the context of weak simulation-sound extractability, the fact that $\mathcal{A}$ has queried x is enough to derive a contradiction. For the strong variant, we now have that $\mathcal{S}_{ZK}$ has output $(x, a_1, a'_2, \ldots, a'_n, z')$ such that

$$\mathcal{V}(x, a_1, H_f(0, x, a_1), a'_2, c'_2 \ldots, a'_n, c'_n, z') = 1$$

and $\mathcal{A}$ has output $(x, a_1, a_2, \ldots, a_n, z)$ such that

$$\mathcal{V}(x, a_1, H_f(0, x, a_1), a_2, c_2, \ldots, a_n, c_n, z) = 1$$

(and $\mathcal{A}$ knows both since it interacted with $\mathcal{S}_{ZK}$). By the computationally unique responses property of Π, it must be that $a_2 = a'_2$. But then it follows that

$$c_2 = H_f(1, H_f(0, x, a_1), a_2) = H_f(1, H_f(0, x, a_1), a'_2) = c'_2$$

(remember that both proofs are accepting with respect to H_f) which in turn implies that $a_3 = a'_3$, etc. Thus, we obtain that $\mathcal{A}$ has output a proof that was produced by $\mathcal{S}_{ZK}$, yielding a contradiction. We conclude that

$$V_{FS}^{H_f}(x, a_1, \ldots, a_n, z) = 1 \text{ implies } V_{FS}^{H}(x, a_1, \ldots, a_n, z) = 1$$

except with negligible probability.

In the rest of the proof of Theorems 24 and 25 in [Unr17], no properties specific to a three-round scheme are used, and so the results extend to the PCIP context, that is, $\mathsf{FS}[\Pi^*]$ is (strongly) simulation-sound extractable. Now applying Theorem 31 from [Unr17], we obtain that $\mathsf{Sig}[\Pi]$ is (s)UF-CMA. □

[13] This property is required to have sufficient entropy on the inputs to the oracle that are reprogrammed by the zero-knowledge simulator $\mathcal{S}_{ZK}$. While $\mathcal{S}_{ZK}$ may reprogram the oracle on inputs $(i-1, c_{i-1}, a_i)$ for $i > 1$, it is enough to require the first message a_1 to have sufficient entropy, since with c_{i-1}, these later inputs all include a uniformly random element from the superpolynomially large challenge space.

Together with the fact that commit-and-open PCIPs can easily be made quantum extractable in the right sense by using standard hash-based commitments based on a collapsing hash function, we obtain the security of the MQDSS signature scheme. Recall that the standard hash-based commitment scheme works as follows. On input s, the commitment algorithm samples a random opening string u and outputs it together with the commitment $c = H(s, u)$. Opening just works by recomputing the hash and comparing it with c . Note that, while this commitment scheme is collapse-binding [Unr16], we need the stronger property of collapsingness of the function defined by the commitment algorithm that, on input a string and some randomness, outputs a commitment (collapse-binding only requires the collapsingness with respect to the committed string, not the opening information).

Corollary 24 (sUF-CMA of MQDSS) *Let Π_{SSH} be the 5-round identification scheme from [SSH11] repeated in parallel a suitable number of times and instantiated with the standard hash-based commitment scheme using a collapsing hash function. Then the Fiat-Shamir signature scheme constructed from Π_{SSH} is sUF-CMA.*

Proof (sketch). In Π_{SSH}, the honest prover's first message consists of two commitments, and the second and final messages contain functions of the strings committed to in the first message. This structure, together with the computational binding property (implied by the collapse binding property) of the commitments, immediately implies that Π_{SSH} has computationally unique responses. According to Corollary 30 in the appendix, Π_{SSH} is a quantum proof of knowledge. It also has HVZK according to [SSH11]. Finally, the first message of Π_{SSH} is clearly unpredictable. An application of Theorem 23 finishes the proof. □

7.2 Sequential Or Proofs

A second application of our multi-input version of the measure-and-reprogram result is to the OR-proof as introduced by Liu, Wei and Wong [LWW04] and further analyzed by Fischlin, Harasser and Janson [FHJ]. This is an alternative (non-interactive) proof for proving existence/knowledge of (at least) one of two witnesses without revealing which one, compared to the well known technique by Cramer, Damgård and Schoenmakers [CDS94].

Formally, given two Σ-protocols Σ_0, and Σ_1, for languages $\mathcal{L}_0$, and $\mathcal{L}_1$, respectively, [LWW04] proposes as a non-interactive proof for the OR-language $\mathcal{L}_\vee = \{(x_0, x_1) \,:\, x_0 \in \mathcal{L}_0 \vee x_1 \in \mathcal{L}_1\}$ a quadruple $\pi_\vee = (a_0, a_1, z_0, z_1)$ such that

$$V_\vee^H(x_0, x_1, \pi_\vee) := \big[V_0\big(x_0, a_0, H(1, x_0, x_1, a_1), z_0\big) \wedge V_1\big(x_1, a_1, H(0, x_0, x_1, a_0), z_1\big)\big]$$

is satisfied. Fischlin et al. call this construction *sequential OR proof*. We emphasize that the two challenges c_0 and c_1 are computed "over cross", i.e., the challence c_0 for the execution of Σ_0 is computed by hashing a_1, and vice versa. It is straightforward to verify that if Σ_0 and Σ_1 are special honest-verifier zero-knowledge, meaning that for any challenge c and response z one can efficiently compute a first message a such that (a, c, z) is accepted, then it is sufficient to be able to succeed in *one* of the two *interactive* protocols Σ_0 and Σ_1 in order to honestly produce such an OR-proof $\pi_\vee$. Thus, depending on the context, it is sufficient that one instance is in the corresponding language, or that the prover knows one of the two witnesses, to produce $\pi_\vee$. Indeed, if, say, $x_0 \in \mathcal{L}_0$ (and a witness w_0 is available), then $\pi_\vee$ can be produced as follows. Prepare a_0 according to Σ_0, compute $c_1 := H(0, x_0, x_1, a_0)$ and simulate z_1 and a_1 using the special honest-verifier zero-knowledge property of Σ_1 so that $V_1(x_1, a_1, c_1, z_1)$ is satisfied, and then compute the response z_0 for the challenge $c_0 := H(1, x_0, x_1, a_1)$ according to Σ_0.

On the other hand, intuitively one expects that one of the two instances must be true in order to be able to successfully produce a proof. Indeed, [LWW04] shows security of the sequential OR in the (classical) ROM. [FHJ] go a step further and show security in the (classical) *non-programmable* ROM. Here we show that our multi-input version of the measure-and-reprogram result (as a matter of fact the 2-input version) implies security in the QROM.

Theorem 25 *There exists a black-box quantum polynomial-time interactive algorithm $\hat{\mathcal{P}}$, which first outputs a bit b and two instances x_0, x_1, and in a second stage acts as an interactive prover*

that runs Σ_b on instance x_b, such that for any adversary $\mathcal{A}$ making q queries to a uniformly random function H and for any x_0°, x_1°:

$$\Pr\left[x_0 = x_0^\circ \wedge x_1 = x_1^\circ \wedge v_b = \mathit{accept} : (b, x_0, x_1, v_b) \leftarrow \langle \hat{\mathcal{P}}^{\mathcal{A}}, \mathcal{V}_b \rangle\right]$$
$$\geq \frac{1}{(2q+1)^4} \Pr_H\left[x_0 = x_0^\circ \wedge x_1 = x_1^\circ \wedge V_\vee^H(x_0, x_1, \pi_\vee) : (x_0, x_1, \pi_\vee) \leftarrow \mathcal{A}^H\right].$$

As explained above, the execution $(b, x_0, x_1, v_b) \leftarrow \langle \hat{\mathcal{P}}^{\mathcal{A}}, \mathcal{V}_b \rangle$ should be understood in that $\hat{\mathcal{P}}^{\mathcal{A}}$ first outputs x_0, x_1 and b, and then it engages with $\mathcal{V}_b$ to execute Σ_b on instance x_b. Thus, the statement ensures that if $\mathcal{A}^H$ succeeds to produce a convincing proof $\pi_\vee$ then $\hat{\mathcal{P}}^{\mathcal{A}}$ succeeds to convincingly run Σ_0 *or* Σ_1 (with similar success probability), where it is up to $\hat{\mathcal{P}}^{\mathcal{A}}$ to choose which one it wants to do.

Of course, the statement translates to the *static* setting where the two instances x_0 and x_1 are *fixed* and not produced by the dishonest prover.

Proof. The algorithm $\mathcal{A}$ fits well into the statement of Theorem 6 with the two extractable inputs $\tilde{x}_0 = (0, x_0, x_1, a_0)$ and $\tilde{x}_1 = (1, x_0, x_1, a_1)$. Thus, we can consider the 3-stage algorithm $\mathcal{S}$ ensured by Theorem 6, which behaves as follows with at least the probability given by the right hand side of the claimed inequality. In the first stage, it outputs a permutation on the set $\{0, 1\}$, which we represent by a bit $b \in \{0, 1\}$ with $b = 0$ corresponding to the identity permutation, as well as $\tilde{x}_b = (b, x_0, x_1, a_b)$. On input a random $\Theta_b = c_{1-b}$ ("locally" chosen by $\hat{\mathcal{P}}$), $\mathcal{S}$ then outputs $\tilde{x}_{1-b} = (1-b, x_0, x_1, a_{1-b})$. Finally, on input a random $\Theta_{1-b} = c_b$ (provided by $\mathcal{V}_b$ as the challenge upon the first message a_b), $\mathcal{S}$ outputs z_0, z_1 so that $V_\vee$ is satisfied with the challenges c_b and c_{1-b}, and thus in particular $V_b\big(x_b, a_b, c_b, z_b\big)$ is satisfied. This directly shows the existence of $\hat{\mathcal{P}}$ as claimed.

□

8 Acknowledgement

We thank Dominque Unruh for hinting towards the possibility of the improved Theorem 2 (compared to [DFMS19]), see also Footnote 8, and Andreas Hülsing for helpful discussions. CM was funded by a NWO VENI grant (Project No. VI.Veni.192.159). SF was partly supported by the EU Horizon 2020 Research and Innovation Program Grant 780701 (PROMETHEUS).

References

BBB+18. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, May 2018.

BBHT98. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998.

Boo. Jonathant Bootle. Recursive techniques for lattice-based zero-knowledge. https://www.youtube.com/watch?v=NEayIq_k4ks. Accessed: 06.02.2020.

CDS94. Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology — CRYPTO '94*, pages 174–187, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

CHR+16. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass mq-based identification to mq-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016*, pages 135–165, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

CHR+18. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. Sofia: Mq-based signatures in the qrom. In Michel Abdalla and Ricardo Dahab, editors, *Public-Key Cryptography – PKC 2018*, pages 3–33, Cham, 2018. Springer International Publishing.

DFG13. Özgür Dagdelen, Marc Fischlin, and Tommaso Gagliardoni. The fiat–shamir transformation in a quantum world. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, pages 62–81, Berlin, Heidelberg, 2013. Springer.

DFMS19. Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 356–383, Cham, 2019. Springer International Publishing.

FHJ. Marc Fischlin, Patrick Harasser, and Christian Janson. Signatures from sequential-or proofs. Unpublished Manuscript.

KLS18. Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 552–586, Cham, 2018. Springer.

LWW04. Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In Huaxiong Wang, Josef Pieprzyk, and Vijay Varadharajan, editors, *Information Security and Privacy*, pages 325–335, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

LZ19. Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 326–355, Cham, 2019. Springer International Publishing.

NIS. Nist post-quantum cryptography standardization. https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions,.

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*, pages 706–723, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

Unr12. Dominique Unruh. Quantum proofs of knowledge. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 135–152, Berlin, Heidelberg, 2012. Springer.

Unr15. Dominique Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 755–784, Berlin, Heidelberg, 2015. Springer.

Unr16. Dominique Unruh. Computationally binding quantum commitments. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016*, pages 497–527, Berlin, Heidelberg, 2016. Springer.

Unr17. Dominique Unruh. Post-quantum security of Fiat-Shamir. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 65–95, Cham, 2017. Springer.

Zha12. Mark Zhandry. How to construct quantum random functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, oct 2012.

A Quantum extractability of q2 identification schemes

A class of identification schemes that is of particular interest are so-called q2-identification schemes. The NIST candidate signature scheme MQDSS, for example, is obtained from such an identification scheme via the multi-round Fiat-Shamir transformation from Definition 20 (with some additional strings included in the hash arguments). In this section, we will prove that a PCIP with a so-called "q2 extractor" [CHR+16, Definition 4.6] is a quantum proof of knowledge if it has an additional collapsingness property. This is necessary for its Fiat-Shamir transformation to fulfill (s)UF-CMA in the QROM (for (s)UF-CMA in the ROM, the q2-extractor alone is sufficient [CHR+16]).

We begin by defining q2 identification schemes and their extractors.

Definition 26 *A 5-round identification scheme is a q2 identification scheme, if the second challenge is a single bit. A q2 identification scheme is called q2-extractable if there exists a polynomial-time algorithm that, on input four transcripts* $t^{(i)} = (a_1^{(i)}, c_1^{(i)}, a_2^{(i)}, c_2^{(i)}, z^{(i)})$, $i = 1, 2, 3, 4$, *such that*

$$\begin{aligned} c_1^{(1)} = c_1^{(2)} \neq c_1^{(3)} = c_1^{(4)} \text{ and} \\ c_2^{(1)} = c_3^{(3)} \neq c_2^{(2)} = c_2^{(4)}, \end{aligned} \tag{13}$$

outputs the secret key with non-negligible probability.

For ease of exposition we have assumed that the different challenges of a single PCIP come all from the same challenge space. A q2 identification scheme can be brought into this form by having the prover compute the second challenge by selecting the first bit of an augmented second challenge that is as large as the first one. For classical provers, four transcripts as required by the above definition can be obtained by straightforward rewinding. In the following, we show that, if the q2 identification scheme has an additional property similar to the quantum-computationally unique responses property introduced in [DFMS19, LZ19], then the existence of a q2 extractor implies

that there exists a quantum extractor. This makes the scheme a quantum proof of knowledge. The argument follows the same lines as the one given in [DFMS19] to prove that t-soundness and quantum-computationally unique responses imply the quantum proof-of-knowledge-property, which in turn is an extension of the result by Unruh for Σ-protocols with perfect unique responses [Unr12].

Recall the definition of a collapsing relation, [DFMS19, Definition 23], a generalization of the notion of a collapsing hash function [Unr16]. We define the notion of collapsingness for interactive proof systems as follows:

Definition 27 *A $(2n+1)$-round interactive proof system Π is called collapsing, if the relation $R_\Pi : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ with $\mathcal{X} = \mathcal{C}^n \times \mathcal{A}_1$ and $\mathcal{Y} = \mathcal{A}_2 \times ... \times \mathcal{A}_n \times \mathcal{Z}$ given by the verification predicate V_Π of Π is collapsing from $\mathcal{X}$ to $\mathcal{Y}$.*

Note that for $n = 1$, this notion of collapsingness coincides with the notion of quantum-computationally unique responses from [DFMS19].

Given a q2-identification scheme Π, consider the following straightforward (first stage of a) quantum extractor $\mathcal{E}_\Pi^{\mathcal{A}}$. The extractor runs the prover $\mathcal{A}$ using honestly sampled challenges to obtain a first transcript $t^{(1)}$. Now it rewinds three times and reruns $\mathcal{A}$, each time with a fresh pair of challenges, chosen such as to obtain $t^{(i)}$, $i = 2, 3, 4$ such that the four transcripts fulfill the conditions (13). For this extractor, we obtain the following

Theorem 28 *Let Π a q2-extractable q2-identification scheme that is also collapsing. Then the success probability of the extractor $\mathcal{E}_\Pi^{\mathcal{A}}$ is lower-bounded in terms of the success probability of the prover $\mathcal{A}$ as*

$$\Pr[\mathcal{E}_\Pi^{\mathcal{A}} \text{ extracts}] \geq \left(\Pr\left[v = \mathit{accept} : (x, v) \leftarrow \langle \mathcal{A}, \mathcal{V}_\Pi \rangle\right]\right)^7 \tag{14}$$

The proof of this theorem is essentially the same as for Theorem 25 in [DFMS19], which is a slight modification of an argument from [Unr12].

As a corollary, we obtain the fact that for q2 identification schemes, q2-extractability and collapsingness imply the quantum proof of knowledge property as defined in [Unr12].

Corollary 29 *Let Π a q2-extractable q2-identification scheme that is also collapsing. Then it is a quantum proof of knowledge.*

In particular, the 5-round identification scheme Π_{SSH} from [SSH11] which is used to construct the post-quantum digital signature scheme MQDSS has these properties under plausible assumptions, namely that it is instantiated with the standard hash-based commitment scheme using a collapsing hash function [Unr16] (see discussion towards the end of Section 7.1). For MQDSS, this is no additional assumption, as the Fiat-Shamir transformation uses the QROM anyway, and a quantum accessible random oracle is collapsing by [Unr16].

Corollary 30 *If the 5-round identification scheme from [SSH11] is instantiated with the standard hash-based commitment scheme using a collapsing hash function, it is a quantum proof of knowledge.*

Proof (sketch). According to [CHR+16], Π_{SSH} is a q2-extractable q2 identification scheme. In Π_{SSH}, the honest prover's first message consists of two commitments, and the second and final messages contain functions of the strings commited to in the first message, and some opening information, respectively. Measuring a function of a register is equivalent to a partial computational basis measurement of that register. According to the the collapsing property of the hash function, no efficient algorithm can distinguish whether the the committed string and the opening information are measured or not. This clearly implies the same indistinguishability for partial measurements of the string register, which implies that Π_{SSH} is collapsing. □

Note that the above proof works for any multi-round PCIP that has a similar commit-and-open structure.