Tight adaptive reprogramming in the QROM

No Author Given

Abstract. The random oracle model (ROM) enjoys widespread popularity, mostly because it tends to allow for *tight* and *conceptually simple* proofs where provable security in the standard model is elusive or costly. While being the adequate replacement of the ROM in the post-quantum security setting, the quantum-accessible random oracle model (QROM) has thus far failed to provide these advantages in many settings. In this work, we focus on *adaptive reprogrammability*, a feature of the ROM enabling tight and simple proofs in many settings. We show that the straightforward quantum-accessible generalization of adaptive reprogramming is feasible by proving a bound on the adversarial advantage in distinguishing whether a random oracle has been reprogrammed or not. We show that our bound is tight by providing a matching attack. We go on to demonstrate that our technique recovers the mentioned advantages of the ROM in three QROM applications: 1) We give a tighter proof of security of the message compression routine as used by XMSS. 2) We show that the standard ROM proof of chosen-message security for Fiat-Shamir signatures can be lifted to the QROM, straightforwardly, achieving a tighter reduction than previously known. 3) We give the first QROM proof of security against fault injection and nonce attacks for the hedged Fiat-Shamir transform.

Keywords: Post-quantum security, QROM, adaptive reprogramming, digital signature, Fiat-Shamir transform, hedged Fiat-Shamir, XMSS

1 Introduction

Since its introduction, the Random oracle model (ROM) has allowed cryptographers to prove efficient practical cryptosystems secure for which proofs in the standard model have been elusive. In general, the ROM allows for proofs that are conceptually simpler and often tighter than standard model security proofs.

With the advent of post-quantum cryptography, and the introduction of quantum adversaries, the ROM had to be generalized: In this scenario, a quantum adversary interacts with a non-quantum network, meaning that "online" primitives (like signing) stay classical, while the adversary can compute all "offline" primitives (like hash functions) on its own, and hence, in superposition. To account for these stronger capabilities, the quantum-accessible ROM (QROM) was introduced [8]. While successfully fixing the definitional gap, the QROM does not generally come with the advantages of its classical counterpart:

- *Lack of conceptual simplicity.* QROM proofs are extremely complex for various reasons. One reason is that they require some understanding of quantum information theory. More important, however, is the fact that many of the useful properties of the ROM (like preimage awareness and adaptive programmability) are not known to translate directly to the QROM.

- *Tightness.* Many primitives that come with tight security proofs in the ROM are not known to be supported by tight proofs in the QROM. For example, there has been an ongoing effort [33, 24, 25, 7, 27, 21] to give tighter QROM proofs for the well-known Fujisaki-Okamoto transformation [18, 19], which is proven tightly secure in the ROM as long as the underlying scheme fulfills IND-CPA security [20].

In many cases, we expect certain generic attacks to only differ from the ROM counterparts by a square-root factor in the required number of queries if the attack involves a search problem, or no significant factor in the case of guessing. Hence, it was conjectured that it might be sufficient to prove security in the ROM, and then add a square-root factor for search problems. However, recent results [38] demonstrate a separation of ROM and QROM, showing that this conjecture does not hold true in general, as there exist schemes which are provably secure in the ROM and insecure in the QROM. As a consequence, a QROM proof is crucial to establish confidence in a post-quantum cryptosystem.[1]

ADAPTIVE PROGRAMMABILITY. A desirable property of the (classical) ROM is that any oracle value $\mathsf{O}(x)$ can be chosen when O is queried on x for the first time (lazy-sampling). This fact is often exploited by a reduction simulating a security game without knowledge of some secret information. Here, an adversary A will not recognize the reprogramming of $\mathsf{O}(x)$ as long as the new value is uniformly distributed and consistent with the rest of A's view. This property is called *adaptive programmability.*

The ability to query an oracle in superposition renders this formerly simple approach more involved, similar to the difficulties arising from the question how to extract classical preimages from a quantum query (preimage awareness) [35, 4, 7, 27, 39, 16, 28, 10, 14]. Intuitively, a query in superposition can be viewed as a query that might contain all input values at once. Already the first answer of O might hence contain information about every value $\mathsf{O}(x)$ that might need to be reprogrammed as the game proceeds. It hence was not clear whether it is possible to adaptively reprogram a quantum random oracle without causing a change in the adversary's view.

Until recently, both properties only had extremely non-tight variants in the QROM. For preimage awareness, it was essentially necessary to randomly guess the right query and measure it (with an unavoidable loss of at least $1/q$ for q queries, and the additional disadvantage of potentially rendering the adversary's output unusable due to measurement disturbance). In a recent breakthrough result, Zhandry developed the compressed oracle technique that provides preimage awareness [39] in many settings. For adaptive reprogramming, variants of Unruh's one-way-to-hiding lemma allowed to prove bounds but only with a square-root loss in the entropy of the reprogramming position [34, 36, 17, 23].

[1] Unless, of course, a standard model proof is available.

In some cases [8, 26, 33, 21], reprogramming could even be avoided by giving a proof that rendered the oracle "a-priori consistent", which is also called a "history-free" proof: In this approach, the oracle is completely redefined in a way such that it is enforced to be *a priori* consistent with the rest of an adversary's view, meaning that it is redefined before execution of the adversary, and on *all* possible input values. Unfortunately, it is not always clear whether it is possible to lift a classical proof to the QROM with this strategy. Even if it is, the "a-priori" approach usually leads to conceptually more complicated proofs. More importantly, it can even lead to reductions that are non-tight with respect to runtime, and may necessitate stronger or additional requirements like, e.g., the statistical counterpart of a property that was only used in its computational variant in the ROM. One example are history-free proofs of CMA security for Fiat-Shamir signatures as e.g. given in [37] and later in [26].

Hence, in this work we are interested in the question:

> **Can we *tightly* prove that adaptive reprogramming can also be done in the quantum random oracle model?**

Our contribution. For common use cases in the context of post-quantum cryptography, this work answers the question above in the affirmative. In more detail, we present a tool for adaptive reprogramming that comes with a tight bound, supposing that the reprogramming positions hold sufficiently large entropy, and reprogramming is triggered by classical queries to an oracle that is provided by the security game (e.g., a signing oracle). These preconditions are usually met in (Q)ROM reductions: The reprogramming is usually triggered by adversarial signature or decryption queries, which remain classical in the post-quantum setting, as the oracles represent honest users.

While we prove a very general lemma, using the simplest variant of the superposition oracle technique [39], we present two corollaries, tailored to cases like a) hash-and-sign with randomized hashing and b) Fiat-Shamir signatures. (Note that we do not have to give a full proof for Fiat-Shamir: We only tend to proving that UF-KOA implies UF-CMA security, as UF-KOA security has already been covered by [37, 26, 16].) In both cases, reprogramming occurs at a position of which one part is an adversarially choosen string. For a), the other part is a random string z, sampled by the reduction (simulating the signer). For b), the other part is a commitment w chosen from a distribution with sufficient min-entropy, together with additional side-information. In both cases, we manage to bound the distinguishing advantage of any adversary that makes q_s signing and q_{H} random oracle queries by

$$1.5 \cdot q_s \sqrt{q_{\mathsf{H}} \cdot 2^{-r}} \ ,$$

where r is the length of z for a), and the min-entropy of w for b). We note that it might be possible to alternatively prove a less general adaptive reprogramming lemma covering the special cases a) and b) above by generalizing the semi-classical O2H lemma from [4].

We then demonstrate the applicability of our tool, by giving

– a tighter proof for hash-and-sign applications leading to a tighter proof for message-compression as used by the hash-based signature scheme XMSS in RFC 8391 [22] as a special case,
– a runtime-tight reduction of unforgeability under adaptive chosen message attacks (UF-CMA) to plain unforgeability ($\mathsf{UF\text{-}CMA}_0$, sometimes denoted UF-KOA or UF-NMA) for Fiat Shamir signatures.
– the first proof of fault resistance for the hedged Fiat-Shamir transform, recently proposed in [5], in the post-quantum setting.

HASH-AND-SIGN. As a first motivating and mostly self-contained application we analyze the hash-and-sign construction that takes a fixed-message-length signature scheme SIG and turns it into a variable-message-length signature scheme SIG′ by first compressing the message using a hash function. We show that if SIG is secure under random message attacks (UF-RMA), SIG′ is secure under adaptively chosen message attacks (UF-CMA). Then we show that along the same lines, we can tighten a recent security proof [9] for message-compression as described for XMSS [11] in RFC 8391. Our new bound shows that one can use random strings of half the length to randomize the message compression in a provably secure way.

THE FIAT-SHAMIR TRANSFORM. In Section 4.1, we show that if an identification scheme ID is Honest-Verifier Zero-Knowledge (HVZK), and if the resulting Fiat-Shamir signature scheme $\mathsf{SIG} := \mathsf{FS}[\mathsf{ID}, \mathsf{H}]$ furthermore possesses $\mathsf{UF\text{-}CMA}_0$ security, then SIG is also UF-CMA secure, in the quantum random oracle model. Here, $\mathsf{UF\text{-}CMA}_0$ denotes the security notion in which the adversary only obtains the public key and has to forge a valid signature without access to a signing oracle. While this statement was already proven in [26], we want to point out several advantages of our proof strategy and the resulting bounds.

Conceptual simplicity. A well-known proof strategy for $\mathsf{HVZK}, \mathsf{UF\text{-}CMA}_0 \Rightarrow \mathsf{UF\text{-}CMA}$ in the random oracle model (implicitly contained in [1]) is to replace honest transcripts with simulated ones, and to render H *a-posteriori* consistent with the signing oracle during the proceedings of the game. I.e., $\mathsf{H}(w, m)$ is patched *after* oracle SIGN was queried on m. Applying our lemma, we observe that this approach actually works in the quantum setting as well. We obtain a very simple QROM proof that is congruent with its ROM counterpart.

In [26], the issue of reprogramming quantum random oracle H was circumvented by giving a history-free proof: In the proof, messages are tied to potential transcripts by generating the latter with message-dependent randomness, *a priori*, and H is patched accordingly, right from the beginning of the game. During each computation of $\mathsf{H}(w, m)$, the reduction therefore has to keep H a-priori consistent by going over all transcript candidates (w_i, c_i, z_i) belonging to m, and returning c_i if $w = w_i$.

Applicability to a broader class of signature schemes. To achieve a-priori consistency, [26] crucially relies on *statistical* HVZK. Furthermore, they require that the HVZK simulator outputs transcripts such that the challenge c is uniformly distributed. We are able to drop the requirement on c altogether, and to only require *computational* HVZK. As a practical example, alternate NIST candidate Picnic [12] satisfies only *computational* HVZK: here, we give the first QROM reduction from chosen-message security, i.e. UF-CMA, to plain unforgeability, i.e. $\mathsf{UF\text{-}CMA}_0$.[2]

Tightness with regards to running time. Our reduction B has about the running time of the adversary A, as it can simply sample simulated transcripts and reprogram H, accordingly. The reduction in [26] suffers from a quadratic blow-up in its running time: They have running time $\text{Time}(\mathsf{B}) \approx \text{Time}(\mathsf{A}) + q_{\mathsf{H}} q_S$, as the reduction has to execute q_S computations upon each query to H in order to keep it a-priori consistent. As they observe, this quadratic blow-up renders the reduction non-tight in all practical aspects. On the other hand, our upper bound comes with a bigger disruption in terms of commitment entropy (the min-entropy of the first message (the *commitment*) in the identification scheme). While the source of non-tightness in [26] can not be balanced out, however, we offer a trade-off: If needed, the commitment entropy can be increased by appending a random string to the commitment.[3]

Robustness of the hedged Fiat-Shamir transform against fault attacks. When it comes to real-world implementations, the assessment of a signature scheme will not solely take into consideration whether an adversary could forge a fresh signature as formalized by the UF-CMA game, as UF-CMA does not capture all avenues of real-world attacks. For instance, an adversary interacting with hardware that realizes a cryptosystem can try to induce a hardware malfunction, also called fault injection, in order to derail the key generation or signing process. Although it might not always be straightforward to predict where exactly a triggered malfunction will affect the execution, it is well understood that even a low-precision malfunction can seriously injure a schemes' security. In the context of the ongoing effort to standardize post-quantum secure primitives [31], it hence made sense to affirm [32] that desirable additional security features include, amongst others, resistance against fault attacks and randomness generation that has some bias.

Recently [5], the hedged Fiat-Shamir construction was proven secure against biased nonces and several types of fault injections, in the ROM. This result

[2] As a matter of fact, the inapplicability of the history-free reduction from [26], that was used in [16] to give a full reduction for Fiat Shamir signatures (starting with a quantum-extractable identification scheme) was initially overlooked by the Picnic Team. The Picnic team has acknowledged that, and is working on a revision of the Picnic submission to the NIST standardization process for post-quantum cryptographic schemes that will use our reduction.

[3] While this increases the signature size, the increase is mild in typical post-quantum Fiat-Shamir based digital signature schemes. As an example, suppose Dilithium-1024x768, which has a signature size of 2044 bytes, had zero commitment entropy (it actually has quite some, see remarks in [26]). To ensure that about 2^{128} hash queries are necessary to make the term in our security bound that depends on the commitment entropy equal 1, about 32 bytes would need to be added, an increase of about 1.6% (assuming 2^{64} signing queries).

can for example be used to argue that alternate NIST candidate Picnic [12] is robust against many types of fault injections. We revisit the hedged Fiat-Shamir construction in Section 4.2 and lift the result of [5] to the QROM. In particular, we thereby obtain that Picnic is resistant against many fault types, even when attacked by an adversary with quantum capabilities.

We considered to generalize the result further by replacing the standard Fiat-Shamir transform with the Fiat–Shamir with aborts transform [29, 26]. While our security statements can be extended in a straightforward manner, we decided not to further complicate our proof with the required modifications. For Dilithium, the implications are limited anyway, as several types of faults are only proven ineffective if the underlying scheme is subset-revealing, which Dilithium is not.[4]

Optimality of our bound. We also show that our lower bound is tight for the given setting, presenting a quantum attack that matches our bound, up to a constant factor. Let us restrict our attention to the simple case where $H : \{0,1\}^n \to \{0,1\}^k$ is a random function, which is potentially reprogrammed at a random position x^* resulting in a new oracle H'. Consider an attacker that is allowed $2q$ queries to the random oracle.

A classical attack that matches the classical bound for the success probability, $O(q \cdot 2^{-n})$, is the following: pick values $x_1, ..., x_q$ and compute the XOR of the outputs $H(x_i)$. After the oracle is potentially reprogrammed, the attacker outputs 0 iff the checksum computed before is unchanged.

In order to match the quantum lower bound, we use the same attack, but on a superposition of tuples of inputs: the attacker queries H with the superposition of all possible inputs, and then applies a cyclic permutation σ on the input register. This process is repeated $q-1$ times (on the same state). After the potential reprogramming, we repeat the same process, but now applying the permutation σ^{-1} and querying H'. Using techniques from [2], we show how to distinguish the two cases with advantage $\Omega\left(\sqrt{\frac{q}{2^n}}\right)$ in time $\mathsf{poly}(q,n)$.

2 Adaptive reprogramming: the toolbox

Before we describe our adaptive reprogramming theorem, let us quickly recall how we usually model adversaries with quantum access to a random oracle: As established in [8, 6], we model quantum access to a random oracle $\mathsf{O} : X \times Y$ via oracle access to a unitary U_{O}, which is defined as the linear completion of $|x\rangle_X |y\rangle_Y \mapsto |x\rangle_X |y \oplus \mathsf{O}(x)\rangle_Y$, and adversaries A with quantum access to O as a sequence of unitaries, interleaved with applications of U_{O}. We write $\mathsf{A}^{|\mathsf{O}\rangle}$ to indicate that O is quantum-accessible.

As a warm-up, we will first present our reprogramming lemma in the simplest setting. Say we reprogram an oracle R many times, where the position is partially controlled by the adversary, and partially picked at random. More formally, let

[4] Intuitively, an identification scheme is called subset-revealing if its responses do not depend on the secret key. Dilithium computes its responses as $z := y + c \cdot s_1$, where s_1 is part of the secret key.

X_1 and X_2 be two finite sets, where X_1 specifies the domain from which the random portions are picked, and X_2 specifies the domain of the adversarially controlled portions. We will now formalize what it means to distinguish a random oracle $\mathsf{O_0} : X_1 \times X_2 \to Y$ from its reprogrammed version $\mathsf{O_1}$. Consider the two Repro games, given in Fig. 1: In games Repro_b, the distinguisher has quantum access to oracle $\mathsf{O_b}$ (see line 03) that is either the original random oracle $\mathsf{O_0}$ (if $b = 0$), or the oracle $\mathsf{O_1}$ which gets reprogrammed adaptively ($b = 1$). To model the actual reprogramming, we endow the distinguisher with (classical) access to a reprogramming oracle Reprogram. Given a value $x_2 \in X_2$, oracle Reprogram samples random values x_1 and y, and programs the random oracle to map $x_1 \| x_2$ to y (see line 06). Note that apart from already knowing x_2, the adversary even learns the part x_1 of the position at which $\mathsf{O_1}$ was reprogrammed.

GAME Repro_b	$\text{Reprogram}(x_2)$
01 $\mathsf{O_0} \leftarrow_\$ Y^{X_1 \times X_2}$	05 $(x_1, y) \leftarrow_\$ X_1 \times Y$
02 $\mathsf{O_1} := \mathsf{O_0}$	06 $\mathsf{O_1} := \mathsf{O_1}^{(x_1 \| x_2) \mapsto y}$
03 $b' \leftarrow \mathsf{A}^{\|\mathsf{O_b}\rangle, \text{Reprogram}}$	07 **return** x_1
04 **return** b'	

Fig. 1. Adaptive reprogramming games Repro_b for bit $b \in \{0, 1\}$ in the most basic setting.

Proposition 1. *Let X_1, X_2 and Y be finite sets, and let A be any algorithm issuing R many calls to* Reprogram *and q many (quantum) queries to $\mathsf{O_b}$ as defined in Fig. 1. Then the distinguishing advantage of A is bounded by*

$$|\Pr[\text{Repro}_1^{\mathsf{A}} \Rightarrow 1] - \Pr[\text{Repro}_0^{\mathsf{A}} \Rightarrow 1]| \leq \frac{3R}{2}\sqrt{\frac{q}{|X_1|}}. \tag{1}$$

The above theorem constitutes a significant improvement over previous bounds. In [34] and [17], a bound proportional to $q|X_1|^{-1/2}$ for the distinguishing advantage in similar settings, but for $R = 1$, was given. In [23], a bound proportional to $q^2|X_1|^{-1}$ is claimed, but that seems to have resulted from a "translation mistake" from [17] and should be similar to the bounds from [34, 17]. What is more, we show in Section 6 that the above bound, and therefore also its generalizations, are tight, by presenting a distinguisher that achieves an advantage equal to the right hand side of Eq. (1) for trivial X_1, up to a constant factor.

In fact, we prove something more general than Proposition 1: We prove that an adversary will not behave significantly different, even if

- the adversary does not only control a portion x_2, but instead it even controls the distributions according to which the whole positions $x := (x_1, x_2)$ are sampled at which $\mathsf{O_1}$ is reprogrammed,
- it can additionally pick different distributions, adaptively, and
- the distributions produce some additional side information x' which the adversary also obtains,

as long as the reprogramming positions x hold enough entropy.

Overloading notation, we formalize this generalization by games Repro, given in Fig. 2: Reprogramming oracle Reprogram now takes as input the description of a distribution p that generates a whole reprogramming position x, together with side information x'. Reprogram samples x and x' according to p, programs the random oracle to map x to a random value y, and returns (x, x').

GAME $\textsc{Repro}_b$	$\textsc{Reprogram}(p)$
$\mathsf{O}_0 \leftarrow_\$ Y^X$	$(x, x') \leftarrow p$
$\mathsf{O}_1 := \mathsf{O}_0$	$y \leftarrow_\$ Y$
$b' \leftarrow \mathsf{D}^{\vert\mathsf{O}_b\rangle, \textsc{Reprogram}}$	$\mathsf{O}_1 := \mathsf{O}_1^{x \mapsto y}$
return b'	**return** (x, x')

Fig. 2. Adaptive reprogramming games $\textsc{Repro}_b$ for bit $b \in \{0, 1\}$.

We are now ready to present our main Theorem 1. On a high level, the only difference between the statement of Proposition 1 and Theorem 1 is that we now have to consider R many (possibly different) joint distributions on $X \times X'$, and to replace $\frac{1}{|X_1|}$ (the probability of the uncontrolled reprogramming portion) with the highest likelihood of any of those distributions generating a position x.

Theorem 1 ("Adaptive reprogramming" (AR)). *Let X, X', Y be some finite sets, and let D be any distinguisher, issuing R many reprogramming instructions and q many (quantum) queries to O. Let q_r denote the number of queries to O that are issued inbetween the $(r-1)$-th and the r-th query to* Reprogram*. Furthermore, let $p^{(r)}$ denote the rth distribution that* Reprogram *is queried on. By $p_X^{(r)}$ we will denote the marginal distribution of X, according to $p^{(r)}$, and define*

$$p_{\max}^{(r)} := \mathbb{E} \max_x p_X^{(r)}(x),$$

where the expectation is taken over D's behaviour until its rth query to Reprogram*.*

$$|\Pr[\textsc{Repro}_1^{\mathsf{D}} \Rightarrow 1] - \Pr[\textsc{Repro}_0^{\mathsf{D}} \Rightarrow 1]| \leq \sum_{r=1}^{R} \left(\sqrt{\hat{q}_r p_{\max}^{(r)}} + \frac{1}{2} \hat{q}_r p_{\max}^{(r)} \right) \quad , \tag{2}$$

where $\hat{q}_r := \sum_{i=0}^{r-1} q_i$.

For $R = 1$ and without additional side information output x', the proof of Theorem 1 is given in Section 5. The extension to general R is proven in the full version via a standard hybrid argument. Finally, all our bounds are information-theoretical, i.e. they hold against arbitrary query bounded adversaries. The additional output x' can therefore be sampled by the adversary.

We will now quickly discuss how to simplify the bound given in Eq. (2) for our applications, and in particular, how we can derive Eq. (1) from Theorem 1: Throughout sections 3 and 4, we will only have to consider reprogramming instructions that occur on positions $x = (x_1, x_2)$ such that

- x_1 is drawn according to the same distribution p for each reprogramming instruction, and
- x_2 represents a message that is already fixed by the adversary.

To be more precise, x_1 will represent a uniformly random string z in 3, and no side information x' has to be considered. In Section 4, (x_1, x') will represent a tuple (w, st) that is drawn according to $\mathsf{Commit}(sk)$.

In the language of Theorem 1, the marginal distribution $p_X^{(r)}$ will always be the same distribution p, apart from the already fixed part x_2. We can hence upper bound $p_{\max}^{(r)}$ by $p_{\max} := \max_{x_1} p(x_1)$, and $\hat{q}_r$ by q, to obtain that $\hat{q}_r p_{\max}^{(r)} < q p_{\max}$ for all $1 \leq r \leq R$.

In our applications, we will always require that p holds sufficiently large entropy. To be more precise, we will assume that $p_{\max} < \frac{1}{q}$. In this case, we have that $q p_{\max} < 1$, and that we can upper bound $q p_{\max}$ by $\sqrt{q p_{\max}}$ to obtain

Proposition 2. *Let X_1, X_2, X' and Y be some finite sets, and let p be a distribution on $X_1 \times X'$. Let D be any distinguisher, issuing q many (quantum) queries to O and R many reprogramming instructions such that each instruction consists of a value x_2, together with the fixed distribution p. Then*

$$|\Pr[\text{REPRO}_1^{\mathsf{D}} \Rightarrow 1] - \Pr[\text{REPRO}_0^{\mathsf{D}} \Rightarrow 1]| \leq \frac{3R}{2}\sqrt{q p_{\max}} \ ,$$

where $p_{\max} := \max_{x_1} p(x_1)$.

From this we obtain Proposition 1 setting $p_{max} = |X_1|^{-1}$.

3 Basic applications

In this section, we present two motivating examples that benefit from the most basic version of our bound as stated in Proposition 1. As a first example we chose the canonical hash-and-sign construction when used to achieve security under adaptive chosen message attacks (UF-CMA) from a scheme that is secure under random message attacks (UF-RMA). It is mostly self-contained and similar to our second example. The second example is a tighter bound for the security of hash-and-sign as used in RFC 8391, the recently published standard for the stateful hash-based signature scheme XMSS.

3.1 From RMA to CMA security via Hash-and-Sign

In the following, we present a conceptually easy proof with a tighter bound for the canonical UF-RMA to UF-CMA transform using hash-and-sign $\mathsf{SIG}' = \mathrm{HaS}[\mathsf{SIG}, \mathsf{H}]$, in the QROM (which additionally allows for arbitrary message space expansion). Recall that $\mathsf{Sign}'(sk, m')$ first samples a uniformly random bitstring $z \leftarrow_\$ Z$, computes $\sigma \leftarrow \mathsf{Sign}(sk, \mathsf{H}(z\|m'))$ and returns the pair (z, σ). Vrfy' accordingly first computes $m := \mathsf{H}(z\|m')$ and then calls $\mathsf{Vrfy}(pk, m, \sigma)$.

The reduction M from $\mathsf{UF\text{-}RMA}$ to $\mathsf{UF\text{-}CMA}$ in this case works as follows: First, we have to handle collision attacks. We show that an adversary which finds a forgery for SIG' that contains no forgery for SIG breaks the multi-target version of extended target collision resistance ($\mathsf{M\text{-}eTCR}$) of H, and give a QROM bound for this property. Having dealt with collision attacks leaves us with the case where A generates a forgery that contains a forgery for SIG. The challenge in this case is how to simulate the signing oracle SIGN. Our respective reduction M against $\mathsf{UF\text{-}RMA}$ proceeds as follows: Collect the q_s many message-signature pairs $\{(m_i, \sigma_i)\}_{1\leq i\leq q_s}$, provided by the $\mathsf{UF\text{-}RMA}$ game. When A queries $\mathrm{SIGN}(m_i')$ for the ith time, sample a random z_i, reprogram $\mathsf{H}(z_i \| m_i') := m_i$, and return (z_i, σ_i). See also Fig. 5 below.

In the QROM, this reduction has previously required q_s applications of the O2H Lemma in two steps, loosing an additive $\mathcal{O}(q_s \cdot q/\sqrt{|Z|})$ term. In contrast, we only loose a $\mathcal{O}(q_s\sqrt{q/|Z|})$ (both constants hidden by the $\mathcal{O}$ are small):

Theorem 2. *For any (quantum)* $\mathsf{UF\text{-}CMA}$ *adversary* A *issuing at most* q_s *(classical) queries to the signing oracle* SIGN *and at most* q_{H} *quantum queries to* H*, there exists an* $\mathsf{UF\text{-}RMA}$ *adversary* M *such that*

$$\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}'}(\mathsf{A}) \leq \mathrm{Succ}^{\mathsf{UF\text{-}RMA}}_{\mathsf{SIG}}(\mathsf{M}) + \frac{8q_s(q_s+q_{\mathsf{H}}+2)^2}{|\mathcal{M}'|} + 3q_s\sqrt{\frac{q_{\mathsf{H}}+q_s+1}{|Z|}} \ ,$$

and the running time of M *is about that of* A.

The second term accounts for the complexity to find a second preimage for one of the messages m_i, which is an unavoidable generic attack. The third term is the result of $2q_s$ reprogrammings. Half of them are used in the QROM bound for $\mathsf{M\text{-}eTCR}$, the other half in the reduction M. This term accounts for an attack that correctly guesses the random bitstring used by the signing oracle for one of the queries (such an attack still would have to find a collision for this part but this is inherently not reflected in the used proof technique).

Proof. We now relate the $\mathsf{UF\text{-}CMA}$ security of SIG' to the $\mathsf{UF\text{-}RMA}$ security of SIG via a sequence of games.

GAME G_0. We begin with the original $\mathsf{UF\text{-}CMA}$ game for SIG' in game G_0. The success probability of A in this game is $\mathrm{Adv}^{\mathsf{UF\text{-}CMA}}_{\mathsf{SIG}'}(\mathsf{A})$ per definition.

GAME G_1. We obtain game G_1 from game G_0 by adding an additional condition. Namely, game G_1 returns 0 if there exists an $0 < i \leq q_s$ such that $\mathsf{H}(z^* \| m'^*) = \mathsf{H}(z_i \| m_i')$, where z^* is the random element in the forgery signature, and z_i is the random element in the signature returned by $\mathrm{SIGN}(m_i')$ as the answer to the ith query. We will now argue that

$$|\Pr[G_0^{\mathsf{A}} \Rightarrow 1] - \Pr[G_1^{\mathsf{A}} \Rightarrow 1]| \leq \frac{8q_s(q_s+q_{\mathsf{H}}+2)^2}{|\mathcal{M}'|} + \frac{3q_s}{2}\sqrt{\frac{q_{\mathsf{H}}+q_s+1}{|Z|}} \ .$$

Towards this end, we give a reduction B in Fig. 3, that breaks the $\mathsf{M\text{-}eTCR}$ security of H whenever the additional condition is triggered, making $q_s + q_{\mathsf{H}} + 1$

queries to its random oracle. B simulates the UF-CMA game for SIG′, using H and an instance of SIG. Clearly, B runs in about the same time as game G_0^{A}, and succeeds whenever A succeeds and the additional condition is triggered. To complete this step, it hence remains to show that the success probability of any such $(q_s + q_{\mathsf{H}} + 1)$-query adversary is

$$\mathrm{Succ}_{\mathsf{H}}^{\text{M-eTCR}}(\mathsf{B}, q_s) \leq \frac{8q_s(q_s + q_{\mathsf{H}} + 2)^2}{|\mathcal{M}'|} + \frac{3q_s}{2}\sqrt{\frac{q_{\mathsf{H}} + q_s + 1}{|Z|}} \ . \tag{3}$$

We delay the proof of Eq. (3) until the end.

```
B^{Box,|H⟩}()
01 (pk, sk) ← KG
02 (m′*, σ′*) = A^{SIGN,|H⟩}(pk)
03 Parse σ′* as (z*, σ*)
04 if ∃j : H(z*‖m′*) = H(z_j‖m′_j)
05     i := j
06 else i ←$ [1, q_s]
07 return (m′*, z*, i)

SIGN(m′_i)
08 z_i ← Box(m′_i)
09 σ_i ← Sign(sk, H(z_i, m′_i))
10 return (z_i, σ_i)
```

Fig. 3. Reduction B breaking M-eTCR. Here, Box is the M-eTCR challenge oracle.

Game G_2. The next game differs from G_1 in the way the signing oracle works. In game G_2 (see Fig. 4), the ith query to SIGN is answered by first sampling a random value z_i, as well as a random message m_i, and programming $\mathsf{H}' := \mathsf{H}'^{(z_i \| m'_i) \mapsto m_i}$. Then m_i is signed using the secret key. We will now show that

$$|\Pr[G_1^{\mathsf{A}} \Rightarrow 1] - \Pr[G_2^{\mathsf{A}} \Rightarrow 1]| \leq \frac{3q_s}{2}\sqrt{\frac{q_{\mathsf{H}} + q_s + 1}{|Z|}} \ .$$

Consider a reduction C that simulates game G_2 for A to distinguish the $\textsc{Repro}_b$ game. Accordingly, C forwards access to its own oracle O_b to A instead of H. Instead of sampling z_i, m_i itself in line 08 and programming H in line 09, C obtains $z_i \leftarrow \textsc{Reprogram}(m'_i)$ from its own oracle and computes $m_i := \mathsf{O}_b(z_i \| m'_i)$ as the output of its random oracle. Now, if C plays in $\textsc{Repro}_0$ it perfectly simulates G_1 for A, as the oracle remains unchanged. If C plays in $\textsc{Repro}_1$ it perfectly simulates G_2, as can be seen by inlining $\textsc{Reprogram}$ and removing doubled calls used to recompute m_i. Consequently,

$$\begin{aligned} &|\Pr[G_1^{\mathsf{A}} \Rightarrow 1] - \Pr[G_2^{\mathsf{A}} \Rightarrow 1]| \\ &\qquad = |\Pr[\textsc{Repro}_0^{\mathsf{C}^{\mathsf{A}}} \Rightarrow 1] - \Pr[\textsc{Repro}_1^{\mathsf{C}^{\mathsf{A}}} \Rightarrow 1]| \leq \frac{3q_s}{2}\sqrt{\frac{q_{\mathsf{H}} + q_s + 1}{|Z|}} \ . \end{aligned}$$

To conclude our main argument, we will now argue that

$$\Pr[G_2^{\mathsf{A}} \Rightarrow 1] = \mathrm{Adv}_{\mathsf{SIG}}^{\text{UF-RMA}}(\mathsf{M}) \ ,$$

Game G_2

01 $i := 1$
02 $(pk, sk) \leftarrow \mathsf{KG}()$
03 $(m'^*, \sigma'^*) = \mathsf{A}^{\text{SIGN},|\mathsf{H}\rangle}(pk)$
04 Parse σ'^* as (z^*, σ^*)
05 **if** $\exists 1 \leq i \leq q_s : \mathsf{H}(z^* \| m'^*) = \mathsf{H}(z_i \| m'_i)$
06   **return** 0
07 **return** $\mathsf{Vrfy}(pk, m'^*, \sigma^*) \wedge m'^* \notin \{m'_i\}_{i=1}^{q_s}$

SIGN(m'_i)

08 $z_i \leftarrow_\$ Z, m_i \leftarrow_\$ \mathcal{M}$
09 $\mathsf{H} := \mathsf{H}^{(z_i \| m'_i) \mapsto m_i}$
10 $\sigma_i \leftarrow \mathsf{Sign}(sk, m_i)$
11 $i := i + 1$
12 **return** (z_i, σ_i)

Fig. 4. Game G_2.

$\mathsf{M}^{\mathsf{A},|\mathsf{H}\rangle}(pk, \{(m_i, \sigma_i)\}_{1 \leq i \leq q_s})$

01 $\mathsf{H}' := \mathsf{H}; i := 1$
02 $(m'^*, \sigma'^*) = \mathsf{A}^{\text{SIGN},|\mathsf{H}'\rangle}(pk)$
03 Parse σ'^* as (z^*, σ^*)
04 **return** $(\mathsf{H}(z^* \| m'^*), \sigma)$

SIGN(m'_i)

05 $z_i \leftarrow_\$ Z$
06 **if** $\exists \hat{m}_i$ s. th. $(z_i \| m'_i, \hat{m}_i) \in \mathfrak{L}_{\mathsf{H}'}$
07     $\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \setminus \{(z_i \| m'_i, \hat{m}_i)\}$
08 $\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \cup \{(z_i \| m'_i, m_i)\}$
09 $i := i + 1$
10 **return** (z_i, σ_i)

$\mathsf{H}'(z \| m')$

11 **if** $\exists m$ s. th. $(z \| m', m) \in \mathfrak{L}_{\mathsf{H}'}$
12   **return** m
13 **else return** $\mathsf{H}(z \| m')$

Fig. 5. Reduction M reducing UF-RMA to UF-CMA.

where reduction M is given in Fig. 5. Since reprogramming is done a-posteriori in game G_2, M can simulate a reprogrammed oracle H′ via access to its own oracle H and an initial table look-up: M keeps track of the (classical) values on which H′ has to be reprogrammed (see line 08) and tweaks A's oracle H′, accordingly. The latter means that, given the table $\mathfrak{L}_{\mathsf{H}'}$ of pairs $(z_i \| m'_i, m_i)$ that were already defined in previous signing queries, controlled on the query input being equal to $z_i \| m'_i$ output m_i, and controlled on the input not being equal to any $z_i \| m'_i$, forward the query to M's own oracle H. If needed, M reprograms values (see line 07) by adding an entry to its look-up table. Given quantum access to H, M can implement this as a quantum circuit, allowing quantum access to H′.

Hence, M perfectly simulates game G_2 towards A. The only differences are that M neither samples the m_i itself, nor computes the signatures for them. Both are given to M by the UF-RMA game. However, they follow the same distribution as in game G_2. Lastly, whenever A would win in game G_2, M succeeds in its UF-RMA game as it can extract a valid forgery for SIG on a new message. This is enforced with the condition we added in game G_1.

The final bound of the theorem follows from collecting the bounds above, and it remains to prove the bound on M-eTCR claimed in Eq. (3). We improve a bound from [23], in which it was shown that for a small constant c,[5]

$$\mathrm{Succ}_{\mathsf{H}}^{\mathsf{M\text{-}eTCR}}(\mathsf{B}, q_s) \leq \frac{8q_s(q_{\mathsf{H}}+1)^2}{|\mathcal{M}'|} + c\frac{q_s q_{\mathsf{H}}}{\sqrt{|Z|}} \ .$$

Their proof of this bound is explicitly given for the single target step. It is then argued that the multi-target step can be easily obtained, which was recently confirmed in [9]. The proof proceeds in two steps. The authors construct a reduction that generates a random function from an instance of an average-case search problem which requires to find a 1 in a boolean function f. The function has the property that all preimages of a randomly picked point m in the image correspond to 1s of f. When A makes its query to Box, the reduction picks a random z and programs $\mathsf{H}^{(z\|m')\mapsto m}$. An extended target collision for $(z\|m')$ hence is a 1 in f by design. This gives the first term in the above bound, which is known to be optimal.

The second term in the bound is the result of above reprogramming. I.e., it is a bound on the difference in success probability of A when playing the real game or when run by the reduction. More precisely, the bound is the result of analyzing the distinguishing advantage between the following two games (which we rephrased to match our notation):

Game G_a. A gets access to H. In phase 1, after making at most q_1 queries to H, A outputs a message $m' \in \mathcal{M}'$. Then a random $z \leftarrow_\$ Z$ is sampled and $(z, \mathsf{H}(z\|m'))$ is handed to A. A continues to the second phase and makes at most q_2 queries. A outputs $b \in \{0,1\}$ at the end.

Game G_b. A gets access to H. After making at most q_1 queries to H, A outputs a message $m' \in \mathcal{M}'$. Then a random $z \leftarrow_\$ Z$ is sampled as well as a random range element $m \leftarrow_\$ \mathcal{M}$. Program $\mathsf{H} := \mathsf{H}^{(z\|m')\mapsto m}$. A receives $(z, m = \mathsf{H}(z\|m'))$ and proceeds to the second phase. After making at most q_2 queries, A outputs $b \in \{0,1\}$ at the end.

The authors of [23] showed that for a small constant c (see Footnote 5),

$$|\Pr[G_b^{\mathsf{A}} \Rightarrow 1] - \Pr[G_a^{\mathsf{A}} \Rightarrow 1]| \leq c\frac{q_{\mathsf{H}}}{\sqrt{|Z|}} \,.$$

A straightforward application of Proposition 1 shows that

$$|\Pr[G_b^{\mathsf{A}} \Rightarrow 1] - \Pr[G_a^{\mathsf{A}} \Rightarrow 1]| \leq \frac{3}{2}\sqrt{\frac{q_{\mathsf{H}}+1}{|Z|}} \,.$$

as the games above virtually describe the games Repro_b with the exception that in Repro_b the oracle Reprogram only returns z and not $\mathsf{H}(z\|m')$). Hence, a reduction needs one additional query per reprogramming.

When applying this to the q_s-target case, a hybrid argument shows that the bound becomes ${}^{3q_s}/_{2}\sqrt{{}^{q_{\mathsf{H}}+1}/_{|Z|}}$. Combining this with the reduction of [23] and taking into account that B makes $(q_s + q_{\mathsf{H}} + 1)$ queries confirms the bound claimed in Eq. (3).

[5] This is a corrected bound from [23], see discussion in Section 2.

3.2 Tight security for message hashing of RFC 8391

Another extremely similar application of our basic bound is for another case of the hash-and-sign construction, used to turn a fixed message length UF-CMA-secure signature scheme SIG into a variable input length one SIG′. This case is essentially covered already by Section 3.1: A proof can omit game G_2 and state a simple reduction that simulates game G_1 to extract a forgery. The bound changes accordingly, requiring one reprogramming bound less and becoming $\mathrm{Succ}_{\mathsf{SIG}'}^{\mathsf{UF\text{-}CMA}}(\mathsf{A}) \leq \mathrm{Succ}_{\mathsf{SIG}}^{\mathsf{UF\text{-}CMA}}(\mathsf{M}) + {}^{8q_s(q_s+q_{\mathsf{H}})^2}/_{|\mathcal{M}'|} + 1.5q_s\sqrt{{}^{q_{\mathsf{H}}+q_s}/_{|Z|}}$.

In [22], it was suggested that for stateful hash-based signature schemeslike XMSS [22], the multi-target attacks which cause the first occurence of q_s in the bound could be avoided. This was recently formally proven in [9]. The idea is to exploit the property of hash-based signature schemes that every signature has an index which binds the signature to a one-time public key. Including this index into the hash forces an adversary to also include it in a collision to make it useful for a forgery. Even more, the index is different for every signature and therefore for every target hash.

Summarizing, the authors of [9] showed that there exists a tight standard model proof for the hash-and-sign construction, as used by XMSS in RFC 8391, if the used hash function is q_s-target extended target-collision resistant with nonce (nM-eTCR), an extension of M-eTCR that considers the index.

To demonstrate the relevance of this result, the authors analyzed the nM-eTCR-security of hash functions under generic attacks, proving a bound for nM-eTCR-security in the QROM in the same way as outlined for M-eTCR above. So far, this bound was suboptimal, as it included a bound on distinguishing variants of games G_a and G_b above in which H takes an additional, externally given index as input). Hence, the bound was $\mathrm{Succ}_{\mathsf{H}}^{\mathsf{nM\text{-}eTCR}}(\mathsf{A}, p) \leq {}^{8(q_s+q_{\mathsf{H}})^2}/_{|\mathcal{M}'|} + {}^{32q_s q_{\mathsf{H}}^2}/_{|Z|}$. Due to the translation error, we believe that the second term needs to be updated to $32q_s \cdot \alpha$, where $\alpha = {}^{q_{\mathsf{H}}}/_{\sqrt{|Z|}}$, instead of $32q_s \cdot \alpha^2$. In [9], it was conjectured that in α, a factor of $\sqrt{q_{\mathsf{H}}}$ can be removed. We can confirm this conjecture. As in the case above, Proposition 1 can be directly applied to the distinguishing bound for games G_a and G_b. A reduction would simply treat the index as part of the message sent to Reprogram. Plugging this into the proof in [9] leads to the bound

$$\mathrm{Succ}_{\mathsf{H}}^{\mathsf{nM\text{-}eTCR}}(\mathsf{A}, p) \leq \frac{8(q_s + q_{\mathsf{H}})^2}{|\mathcal{M}'|} + 1.5q_s\sqrt{\frac{q_{\mathsf{H}} + q_s}{|Z|}} \ .$$

4 Applications to the Fiat-Shamir transform

For the sake of completeness, we include all used definitions for identification and signature schemes in the full version . The only non-standard (albeit straightforward) definition is computational HVZK for multiple transcripts, which we give below.

(SPECIAL) HVZK SIMULATOR. We first recall the notion of an HVZK simulator. Our definition comes in two flavours: While a standard HVZK simulator generates transcripts relative to the public key, a *special* HVZK simulator generates transcripts relative to (the public key and) a particular challenge.

Definition 1 ((Special) HVZK simulator). *An* HVZK simulator *is an algorithm* Sim *that takes as input the public key pk and outputs a transcript (w, c, z). A* special HVZK simulator *is an algorithm* Sim *that takes as input the public key pk and a challenge c and outputs a transcript (w, c, z).*

COMPUTATIONAL HVZK FOR MULTIPLE TRANSCRIPTS. In our security proofs, we will have to argue that collections of honestly generated transcripts are indistinguishable from collections of simulated ones. Since it is not always clear whether computational HVZK implies computational HVZK for *multiple* transcripts, we extend our definition, accordingly: In the multi-HVZK game, the adversary obtains a collection of transcripts (rather than a single one). Similarly, we extend the definition of *special* computational HVZK from [5].

Definition 2 ((Special) computational multi-HVZK). *Assume that* ID *comes with an* HVZK *simulator* Sim*. We define multi-*HVZK *games t-*HVZK *as in Fig. 6, and the multi-*HVZK *advantage function of an adversary* A *against* ID *as*

$$\mathrm{Adv}_{\mathsf{ID}}^{t\text{-}\mathsf{HVZK}}(\mathsf{A}) := \left| \Pr[t\text{-}\mathsf{HVZK}_{1\,\mathsf{ID}}^{\mathsf{A}} \Rightarrow 1] - \Pr[t\text{-}\mathsf{HVZK}_{0\,\mathsf{ID}}^{\mathsf{A}} \Rightarrow 1] \right| \ .$$

*To define special multi-*HVZK*, assume that* ID *comes with a special* HVZK *simulator* Sim*. We define multi-*sHVZK *games as in Fig. 6, and the multi-*sHVZK *advantage function of an adversary* A *against* ID *as*

$$\mathrm{Adv}_{\mathsf{ID}}^{t\text{-}\mathsf{sHVZK}}(\mathsf{A}) := \left| \Pr[t\text{-}\mathsf{sHVZK}_{1\,\mathsf{ID}}^{\mathsf{A}} \Rightarrow 1] - \Pr[t\text{-}\mathsf{sHVZK}_{0\,\mathsf{ID}}^{\mathsf{A}} \Rightarrow 1] \right| \ .$$

```
GAME t-HVZK_b
(pk, sk) ← IG(par)
for i ∈ {1, ··· , t}
   trans_i^0 ← getTrans(sk)
   trans_i^1 ← Sim(pk)
b' ← A(pk, (trans_i^b)_{1≤i≤t})
return b'

GAME t-sHVZK_b
i := 1
(pk, sk) ← IG(par)
b' ← A^{getTransO}(pk)
return b'

getTransO(c)
if i > t return ⊥
i := i + 1
trans^0 ← getTransChall(sk, c)
trans^1 ← Sim(pk, c)
return trans^b
```

Fig. 6. Multi-HVZK game and multi-sHVZK game for ID. Both games are defined relative to bit $b \in \{0, 1\}$, and to the number t of transcripts the adversary is given.

STATISTICAL HVZK. Unlike computational HVZK, *statistical* HVZK can be generalized generically, we therefore do not need to deviate from known statistical definitions.

We denote the respective upper bound for (special) statistical HVZK by Δ_{HVZK} (Δ_{sHVZK}).

4.1 Revisiting the Fiat-Shamir transform

In this section, we show that if an identification scheme ID is HVZK, and if $\mathsf{SIG} := \mathsf{FS}[\mathsf{ID}, \mathsf{H}]$ possesses $\mathsf{UF\text{-}CMA}_0$ security (also known as UF-KOA security), then SIG is also UF-CMA secure, in the QROM. Note that our theorem makes no assumptions on how $\mathsf{UF\text{-}CMA}_0$ is proven. For arbitrary ID schemes this can be done using a general reduction for the Fiat-Shamir transform [16], incurring a q_{H}^2 multiplicative loss that is, in general, unavoidable [15]. For a *lossy* ID scheme ID, $\mathsf{UF\text{-}CMA}_0$ of $\mathsf{FS}[\mathsf{ID}, \mathsf{H}]$ can be reduced tightly to the extractability of ID in the QROM [26]. In addition, while we focus on the standard Fiat-Shamir transform for ease of presentation, the following theorem generalizes to signatures constructed using the multi-round generalization of the Fiat-Shamir transform like, e.g., MQDSS [13].

Theorem 3. *For any (quantum)* UF-CMA *adversary* A *issuing at most* q_s *(classical) queries to the signing oracle* SIGN *and at most* q_{H} *quantum queries to* H, *there exists a* $\mathsf{UF\text{-}CMA}_0$ *adversary* B *and a multi-*HVZK *adversary* C *such that*

$$\mathrm{Succ}_{\mathsf{FS}[\mathsf{ID},\mathsf{H}]}^{\mathsf{UF\text{-}CMA}}(\mathsf{A}) \leq \mathrm{Succ}_{\mathsf{FS}[\mathsf{ID},\mathsf{H}]}^{\mathsf{UF\text{-}CMA}_0}(\mathsf{B}) + \mathrm{Adv}_{\mathsf{ID}}^{q_s-\mathsf{HVZK}}(\mathsf{C}) \tag{4}$$

$$+\frac{3q_s}{2}\sqrt{(q_{\mathsf{H}} + q_s + 1) \cdot \gamma(\mathsf{Commit})}\ , \tag{5}$$

and the running time of B *and* C *is about that of* A. *The bound given in Eq.* (4) *also holds for the modified Fiat-Shamir transform that defines challenges by letting* $c := \mathsf{H}(w, m, pk)$ *instead of letting* $c := \mathsf{H}(w, m)$.

Note that if ID is statistically HVZK, we can replace $\mathrm{Adv}_{\mathsf{ID}}^{q_s-\mathsf{HVZK}}(\mathsf{C})$ with $q_s \cdot \Delta_{\mathsf{HVZK}}$.

Proof. Consider the sequence of games given in Fig. 7.

GAMES G_0 - G_2	SIGN(m)	getTrans(m) // G_0-G_1
$(pk, sk) \leftarrow \mathsf{IG}(\mathrm{par})$	$\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{m\}$	$(w, \mathrm{st}) \leftarrow \mathsf{Commit}(sk)$
$(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathrm{SIGN}, \lvert\mathsf{H}\rangle}(pk)$	$(w, c, z) \leftarrow \mathsf{getTrans}(m)$ // G_0-G_1	$c := \mathsf{H}(w, m)$ // G_0
if $m^* \in \mathfrak{L}_{\mathcal{M}}$ **return** 0	$(w, c, z) \leftarrow \mathsf{Sim}(pk)$ // G_2	$c' \leftarrow_\$ \mathcal{C}$ // G_1
Parse $(w^*, z^*) := \sigma^*$	$\mathsf{H} := \mathsf{H}^{(w,m)\mapsto c}$ // G_1 -G_2	$z \leftarrow \mathsf{Respond}(sk, w, c, \mathrm{st})$
$c^* := \mathsf{H}(w^*, m^*)$	**return** $\sigma := (w, z)$	**return** (w, c, z)
return $\mathsf{V}(pk, w^*, c^*, z^*)$		

Fig. 7. Games G_0 - G_2 for the proof of Theorem 3.

Game G_0. Since game G_0 is the original UF-CMA game,

$$\mathrm{Succ}^{\mathsf{UF\text{-}CMA}}_{\mathsf{FS[ID,H]}}(\mathsf{A}) = \Pr[G_0^{\mathsf{A}} \Rightarrow 1] \ .$$

Game G_1. In game G_1, we change the game twofold: First, the transcript is now drawn according to the underlying ID scheme, i.e., it is drawn uniformly at random as opposed to letting $c := \mathsf{H}(w, m)$, see line 14. Second, we reprogram the random oracle H in line 10 such that it is rendered a-posteriori-consistent with this transcript, i.e., we reprogram H such that $\mathsf{H}(w, m) = c$.

To upper bound the game distance, we construct a quantum distinguisher D in Fig. 8 that is run in the adaptive reprogramming games $\text{Repro}_{R,b}$ with $R := q_S$ many reprogramming instances. We identify reprogramming position x with (w, m), additional input x' with st, and y with c. Hence, the distribution p consists of the constant distribution that always returns m (as m was already chosen by A), together with the distribution $\mathsf{Commit}(sk)$. Since D perfectly simulates game G_b if run in its respective game Repro_b, we have

$$|\Pr[G_0^{\mathsf{A}} = 1] - \Pr[G_1^{\mathsf{A}} = 1]| = |\Pr[\text{Repro}_1^{\mathsf{D}} \Rightarrow 1] - \Pr[\text{Repro}_0^{\mathsf{D}} \Rightarrow 1]| \ .$$

Since D issues q_S reprogramming instructions and $(q_H + q_S + 1)$ many queries to H, Proposition 2 yields

$$|\Pr[\text{Repro}_1^{\mathsf{D}} \Rightarrow 1] - \Pr[\text{Repro}_0^{\mathsf{D}} \Rightarrow 1]| \leq \frac{3q_S}{2}\sqrt{(q_H + q_S + 1) \cdot p_{\max}} \ , \qquad (6)$$

where $p_{\max} = \mathbb{E}_{\mathsf{IG}} \max_w \Pr_{W,\mathrm{ST} \leftarrow \mathsf{Commit}(sk)}[W = w] = \gamma(\mathsf{Commit})$.

Distinguisher $\mathsf{D}^{\vert\mathsf{H}\rangle}$	$\mathrm{SIGN}(m)$
01 $(pk, sk) \leftarrow \mathsf{IG}(\mathrm{par})$	07 $\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{m\}$
02 $(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathrm{SIGN}, \vert\mathsf{H}\rangle}(pk)$	08 $(w, \mathrm{st}) \leftarrow \text{Reprogram}(m, \mathsf{Commit}(sk))$
03 **if** $m^* \in \mathfrak{L}_{\mathcal{M}}$ **return** 0	09 $c := \mathsf{H}(w, m)$
04 Parse $(w^*, z^*) := \sigma^*$	10 $z \leftarrow \mathsf{Respond}(sk, w, c, \mathrm{st})$
05 $c^* := \mathsf{H}(w^*, m^*)$	11 **return** $\sigma := (w, z)$
06 **return** $\mathsf{V}(pk, w^*, c^*, z^*)$	

Fig. 8. Reprogramming distinguisher D for the proof of Theorem 3.

Game G_2. In game G_2, we change the game such that the signing algorithm does not make use of the secret key any more: Instead of being defined relative to the honestly generated transcripts, signatures are now defined relative to the simulator's transcripts. We will now upper bound $|\Pr[G_1^{\mathsf{A}} = 1] - \Pr[G_2^{\mathsf{A}} = 1]|$ via computational multi-HVZK. Consider multi-HVZK adversary C in Fig. 9. C takes as input a list of q_s many transcripts, which are either all honest transcripts or simulated ones. Since reprogramming is done a-posteriori in game G_1, C can simulate it via an initial table look-up, like the reduction M that was given in

Section 3.1 (see the description on p. 12). C perfectly simulates game G_1 if run on honest transcripts, and game G_2 if run on simulated ones, hence

$$|\Pr[G_1^{\mathsf{A}} = 1] - \Pr[G_2^{\mathsf{A}} = 1]| \leq \mathrm{Adv}_{\mathsf{ID}}^{q_S-\mathsf{HVZK}}(\mathsf{C}) \ .$$

Adversary $\mathsf{C}^{\lvert\mathsf{H}\rangle}(pk, ((w_i, c_i, z_i)_{i\in\{1,\cdots,q_s\}})$	$\mathrm{SIGN}(m)$	$\mathsf{H}'(w, m)$
$i := 0$	$i++$	**if** $\exists c$ s. th. $(w, m, c) \in \mathfrak{L}_{\mathsf{H}'}$
$\mathfrak{L}_{\mathsf{H}'} := \emptyset$	$\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{m\}$	**return** c
$(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathrm{SIGN},\lvert\mathsf{H}'\rangle}(pk)$	$(w, c, z) := (w_i, c_i, z_i)$	**else return** $\mathsf{H}(w, m)$
if $m^* \in \mathfrak{L}_{\mathcal{M}}$ **return** 0	**if** $\exists c'$ s. th. $(w, m, c') \in \mathfrak{L}_{\mathsf{H}'}$	
Parse $(w^*, z^*) := \sigma^*$	$\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \setminus \{(w, m, c')\}$	
$c^* := \mathsf{H}(w^*, m^*)$	$\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \cup \{(w, m, c)\}$	
return $\mathsf{V}(pk, w^*, c^*, z^*)$	**return** $\sigma := (w, z)$	

Fig. 9. HVZK adversary C for the proof of Theorem 3.

It remains to upper bound $\Pr[G_2^{\mathsf{A}} \Rightarrow 1]$. Consider adversary B, given in Fig. 10. B is run in game $\mathsf{UF\text{-}CMA}_0$ and perfectly simulates game G_2 to A. If A wins in game G_2, it cannot have queried SIGN on m^*. Therefore, H' is not reprogrammed on (m^*, w^*) and hence, σ^* is a valid signature in B's $\mathsf{UF\text{-}CMA}_0$ game.

$$\Pr[G_2^{\mathsf{A}} \Rightarrow 1] \leq \mathrm{Succ}_{\mathsf{FS[ID,H]}}^{\mathsf{UF\text{-}CMA}_0}(\mathsf{B}) \ .$$

Collecting the probabilities yields the desired bound.

Adversary $\mathsf{B}^{\lvert\mathsf{H}\rangle}(pk)$	$\mathrm{SIGN}(m)$	$\mathsf{H}'(w, m)$
$\mathfrak{L}_{\mathsf{H}'} := \emptyset$	$\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{m\}$	**if** $\exists c$ s. th. $(w, m, c) \in \mathfrak{L}_{\mathsf{H}'}$
$(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathrm{SIGN},\lvert\mathsf{H}'\rangle}(pk)$	$(w, c, z) \leftarrow \mathsf{Sim}(pk)$	**return** c
if $m^* \in \mathfrak{L}_{\mathcal{M}}$ ABORT	**if** $\exists c'$ s. th. $(w, m, c') \in \mathfrak{L}_{\mathsf{H}'}$	**else**
return (m^*, σ^*)	$\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \setminus \{(w, m, c')\}$	**return** $\mathsf{H}(w, m)$
	$\mathfrak{L}_{\mathsf{H}'} := \mathfrak{L}_{\mathsf{H}'} \cup \{(w, m, c)\}$	
	return $\sigma := (w, z)$	

Fig. 10. Adversary B for the proof of Theorem 3.

It remains to show that the bound also holds if challenges are derived by letting $c := \mathsf{H}(w, m, pk)$. To that end, we revisit the sequence of games given in Fig. 7: We replace $c := \mathsf{H}(w, m)$ (and $c^* := \mathsf{H}(w^*, m^*)$) with $c := \mathsf{H}(w, m, pk)$ (and $c^* := \mathsf{H}(w^*, m^*, pk)$) in line 13 (line 05), and change the reprogram instruction in line 10, accordingly. Since pk is public, we can easily adapt both distinguisher D and adversaries B and C to account for these changes. In particular, D will simply include pk as a (fixed) part of the probability distribution that is forwarded to its reprogramming oracle. Since the public key holds no entropy once that it is fixed by the game, this change does not affect the upper bound given in Eq. (6).

4.2 Revisiting the hedged Fiat-Shamir transform

In this section, we show how Theorem 1 can be used to extend the results of [5] to the quantum random oracle model: We show that the Fiat-Shamir transform is robust against several types of one-bit fault injections, even in the quantum random oracle model, and that the hedged Fiat-Shamir transform is as robust, even if an attacker is in control of the nonce that is used to generate the signing randomness. In this section, we follow [5] and consider the modified Fiat-Shamir transform that includes the public key into the hash when generating challenges. We consider the following one-bit tampering functions:

$\mathsf{flip\text{-}bit}_i(x)$: Does a logical negation of the i-th bit of x.
$\mathsf{set\text{-}bit}_i(x, b)$: Sets the i-th bit of x to b.

Hedged signature schemes. Let $\mathcal{N}$ be any nonce space. With a signature scheme $\mathsf{SIG} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Vrfy})$ with secret key space $\mathcal{SK}$ and signing randomness space $\mathcal{R}_{\mathsf{Sign}}$, and random oracle $\mathsf{G} : \mathcal{SK} \times \mathcal{M} \times \mathcal{N} \to \mathcal{R}_{\mathsf{Sign}}$, we associate

$$\mathsf{R2H}[\mathsf{SIG}, \mathsf{G}] := \mathsf{SIG}' := (\mathsf{KG}, \mathsf{Sign}', \mathsf{Vrfy}) \enspace ,$$

where the signing algorithm Sign' of SIG' takes as input (sk, m, n), deterministically computes $r := \mathsf{G}(sk, m, n)$, and returns $\sigma := \mathsf{Sign}(sk, m; r)$.

Security of (hedged) Fiat-Shamir against fault injections and nonce attacks. Next, we define UnForgeability in the presence of Faults, under Chosen Message Attacks (UF-F-CMA), for Fiat-Shamir transformed schemes. In game UF-F-CMA, the adversary has access to a faulty signing oracle FAULTSIGN which returns signatures that were created relative to an injected fault. To be more precise, game $\mathsf{UF\text{-}F}_{\mathcal{F}}\text{-}\mathsf{CMA}$ is defined relative to a set $\mathcal{F}$ of indices, and the indices $i \in \mathcal{F}$ specify at which point during the signing procedure exactly the faults are allowed to occur. An overview is given in Fig. 11.

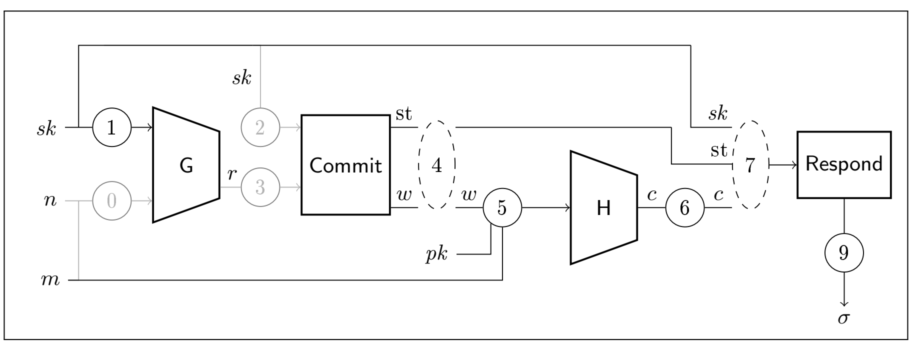

Fig. 11. Faulting a (hedged) Fiat-Shamir signature. Circles represent faults, and their numbers are the respective fault indices $i \in \mathcal{F}$ (following [5], for the formal definition see Fig. 12). Greyed out fault wires indicate that the hedged construction can not be proven robust against these faults, in general. Dashed fault nodes indicate that the Fiat-Shamir construction is robust against these faults if the scheme is subset-revealing.

For the hedged Fiat-Shamir construction, we further define UnForgeability, with control over the used Nonces and in the presence of Faults, under Chosen Message Attacks (UF-N-F-CMA). In game UF-N-F-CMA, the adversary is even allowed to control the nonce n that is used to derive the internal randomness of algorithm Commit. We therefore denote the respective oracle by N-FAULTSIGN. Our definitions slightly simplify the one of [5]: While [5] also considered fault attacks on the input of algorithm Commit (with corresponding indices 2 and 3), they showed that the hedged construction can not be proven robust against these faults, in general. We therefore omitted them from our games, but adhered to the numbering for comparability.

The hedged Fiat-Shamir scheme derandomizes the signing procedure by replacing the signing randomness by $r := \mathsf{G}(sk, m, n)$. Hence, game UF-N-F-CMA considers two additional faults: An attacker can fault the input of G, i.e., either the secret key (fault index 1), or the tuple (m, n) (fault index 0). As shown in [5], the hedged construction can not be proven robust against faults on (m, n), in general, therefore we only consider index 1.

Furthemore, we do not formalize derivation/serialisation and drop the corresponding indices 8 and 10 to not overly complicate our application example. A generalization of our result that also considers derivation/serialisation, however, is straightforward.

Definition 3. *(UF-F-CMA and UF-N-F-CMA) For any subset $\mathcal{F} \subset \{4, \cdots, 9\}$, we definethe UF-F$_\mathcal{F}$-CMA game as in Fig. 12, and the UF-F$_\mathcal{F}$-CMA success probability of a quantum adversary A against FS[ID, H] as*

$$\mathrm{Succ}^{\mathsf{UF\text{-}F_\mathcal{F}\text{-}CMA}}_{\mathsf{FS[ID,H]}}(\mathsf{A}) := \Pr[\mathsf{UF\text{-}F_\mathcal{F}\text{-}CMA}^{\mathsf{A}}_{\mathsf{FS[ID,H]}} \Rightarrow 1] \ .$$

Furthermore, we define the UF-N-F$_\mathcal{F}$-CMA game (also in Fig. 12) for any subset $\mathcal{F} \subset \{1, 4, \cdots, 9\}$, and the UF-N-F$_\mathcal{F}$-CMA success probability of a quantum adversary A against SIG$'$:= R2H[FS[ID, H], G] as

$$\mathrm{Succ}^{\mathsf{UF\text{-}N\text{-}F_\mathcal{F}\text{-}CMA}}_{\mathsf{SIG}'}(\mathsf{A}) := \Pr[\mathsf{UF\text{-}N\text{-}F_\mathcal{F}\text{-}CMA}^{\mathsf{A}}_{\mathsf{SIG}'} \Rightarrow 1] \ .$$

From UF-CMA$_0$ to UF-F-CMA. First, we generalize [5, Lemma 5] to the quantum random oracle model. The proof is given in the full version .

Theorem 4. *Assume ID to be validity aware. If SIG := FS[ID, H] is UF-CMA$_0$ secure, then SIG is also UF-F$_\mathcal{F}$-CMA secure for $\mathcal{F} := \{5, 6, 9\}$, in the quantum random oracle model. Concretely, for any adversary A against the UF-F$_\mathcal{F}$-CMA security of SIG, issuing at most q_S (classical) queries to FAULTSIGN and q_H (quantum) queries to H, there exists an UF-CMA$_0$ adversary B and a multi-sHVZK adversary C such that*

$$\mathrm{Succ}^{\mathsf{UF\text{-}F_{\{5,6,9\}}\text{-}CMA}}_{\mathsf{SIG}}(\mathsf{A}) \leq \mathrm{Succ}^{\mathsf{UF\text{-}CMA_0}}_{\mathsf{SIG}}(\mathsf{B}) + \mathrm{Adv}^{q_s\mathsf{-sHVZK}}_{\mathsf{ID}}(\mathsf{C}) + \frac{3q_S}{2}\sqrt{2 \cdot (q_H + q_S + 1) \cdot \gamma(\mathsf{Commit})} \ . \quad (7)$$

Game $\boxed{\text{UF-F}_{\mathcal{F}}\text{-CMA}}$ $\fbox{UF-N-F$_{\mathcal{F}}$-CMA}$ (dashed)	$\mathsf{FAULTSIGN}(m, i \in \mathcal{F}, \phi)$	$\mathsf{N\text{-}FAULTSIGN}(m, n, i \in \mathcal{F}, \phi)$
$(pk, sk) \leftarrow \mathsf{IG}(\text{par})$	$f_i := \phi$ and $f_j := \text{id} \ \forall\ j \neq i$	$f_i := \phi$ and $f_j := \text{id} \ \forall\ j \neq i$
$\boxed{(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{FAULTSIGN}, \lvert \mathsf{H} \rangle}(pk)}$		$r := \mathsf{G}(f_1(sk), m, n)$
$(m^*, \sigma^*) \leftarrow \mathsf{A}^{\mathsf{N\text{-}FAULTSIGN}, \lvert \mathsf{H} \rangle, \lvert \mathsf{G} \rangle}(pk)$ (dashed box)	$(w, \text{st}) \leftarrow \mathsf{Commit}(sk)$	$(w, \text{st}) \leftarrow \mathsf{Commit}(sk; r)$
if $m^* \in \mathfrak{L}_{\mathcal{M}}$ **return** 0	$(w, \text{st}) := f_4(w, \text{st})$	$(w, \text{st}) := f_4(w, \text{st})$
Parse $(w^*, z^*) := \sigma^*$	$(\hat{w}, \hat{m}, \hat{pk}) := f_5(w, m, pk)$	$(\hat{w}, \hat{m}, \hat{pk}) := f_5(w, m, pk)$
$c^* := \mathsf{H}(w^*, m^*)$	$c := f_6(\mathsf{H}(\hat{w}, \hat{m}, \hat{pk}))$	$c := f_6(\mathsf{H}(\hat{w}, \hat{m}, \hat{pk}))$
return $\mathsf{V}(pk, w^*, c^*, z^*)$	$z \leftarrow \mathsf{Respond}(f_7(sk, c, \text{st}))$	$z \leftarrow \mathsf{Respond}(f_7(sk, c, \text{st}))$
	$\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{\hat{m}\}$	$\mathfrak{L}_{\mathcal{M}} := \mathfrak{L}_{\mathcal{M}} \cup \{\hat{m}\}$
	return $\sigma := f_9(w, z)$	**return** $\sigma := f_9(w, z)$

Fig. 12. Left: Game UF-F$_{\mathcal{F}}$-CMA for SIG = FS[ID, H], and game UF-N-F$_{\mathcal{F}}$-CMA for the hedged Fiat-Shamir construction SIG$'$:= R2H[FS[ID, H], G], both defined relative to a set $\mathcal{F}$ of allowed fault index positions. ϕ denotes the fault function, which either negates one particular bit of its input, sets one particular bit of its input to 0 or 1, or does nothing. We implicitly require fault index i to be contained in $\mathcal{F}$, i.e., we make the convention that both faulty signing oracles return $\perp$ if $i \notin \mathcal{F}$.

and B *and* C *have about the running time of* A.

If we assume that ID *is subset-revealing, then* SIG *is even* UF-F$_{\mathcal{F}'}$-CMA *secure for* $\mathcal{F}' := \mathcal{F} \cup \{4, 7\}$*. Concretely, the bound of Eq.* (7) *then holds also for* $\mathcal{F}' = \{4, 5, 6, 7, 9\}$.

From UF-F-CMA to UF-N-F-CMA. Second, we generalize [5, Lemma 4] to the QROM. The proof is given in the full version .

Theorem 5. *If* SIG := FS[ID, H] *is* UF-F$_{\mathcal{F}}$-CMA *secure for a fault index set* $\mathcal{F}$*, then* SIG$'$:= R2H[SIG, G] *is* UF-N-F$_{\mathcal{F}'}$-CMA *secure for* $\mathcal{F}' := \mathcal{F} \cup \{1\}$*, in the quantum random oracle model, against any adversary that issues no query* (m, n) *to* N-FAULTSIGN *more than once. Concretely, for any adversary* A *against the* UF-N-F$_{\mathcal{F}}$-CMA *security of* SIG$'$ *for* $\mathcal{F}'$*, issuing at most* q_S *queries to* N-FAULTSIGN*, at most* q_{H} *queries to* H*, and at most* q_{G} *queries to* G*, there exist* UF-F$_{\mathcal{F}}$-CMA *adversaries* B_1 B_2 *such that*

$$\text{Succ}_{\mathsf{SIG}'}^{\mathsf{UF\text{-}N\text{-}F}_{\mathcal{F}'}\text{-}\mathsf{CMA}}(\mathsf{A}) \leq \text{Succ}_{\mathsf{SIG}}^{\mathsf{UF\text{-}F}_{\mathcal{F}}\text{-}\mathsf{CMA}}(\mathsf{B}_1) + 2q_{\mathsf{G}} \cdot \sqrt{\text{Succ}_{\mathsf{SIG}}^{\mathsf{UF\text{-}F}_{\mathcal{F}}\text{-}\mathsf{CMA}}(\mathsf{B}_2)} \ ,$$

and B_1 *has about the running time of* A*, while* B_2 *has a running time of roughly* $\text{Time}(\mathsf{B}_2) \approx \text{Time}(\mathsf{A}) + |sk| \cdot (\text{Time}(\mathsf{Sign}) + \text{Time}(\mathsf{Vrfy}))$*, where* $|sk|$ *denotes the length of* sk.

With regards to the reduction's advantage, this proof is not as tight as the one in [5]: R2H[SIG, G] derives the commitment randomness as $r := \mathsf{G}(sk, m, n)$. During our proof, we need to decouple r from the secret key. In the ROM, it is straightforward how to turn any adversary noticing this change into an extractor that returns the secret key. In the QROM, however, all currently known extraction techniques still come with a quadratic loss in the extraction probability. On the other hand, our reduction is tighter with regards to running time, which we reduce by a factor of q_{G} when compared to [5]. If we hedge with an independent seed s of length ℓ (instead of sk), it can be shown with a multi-instance generalization of [33, Lem. 2.2] that

$$\mathrm{Succ}^{\mathsf{UF\text{-}N\text{-}F}_{\mathcal{F}}\text{-}\mathsf{CMA}}_{\mathsf{SIG}'}(\mathsf{A}) \leq \mathrm{Succ}^{\mathsf{UF\text{-}F}_{\mathcal{F}}\text{-}\mathsf{CMA}}_{\mathsf{SIG}}(\mathsf{B}) + (\ell+1)\cdot(q_S + q_{\mathsf{G}})\cdot\sqrt{1/2^{\ell-1}}\ .$$

5 Adaptive reprogramming: proofs

We will now give the proof for our main Theorem 1, which can be broken down into three steps: In this section, we consider the simple special case in which only a single reprogramming instance occurs, and where no additional input x' is provided to the adversary. The generalisation to multiple reprogramming instances follows from a standard hybrid argument. The generalisation that considers additional input is also straightforward, as the achieved bounds are information-theoretical and a reduction can hence compute marginal and conditioned distributions on its own. For the sake of completeness, we include the generalisation steps in the full version .

In this and the following sections, we need quantum theory. We stick to the common notation as introduced in, e.g. [30]. Nevertheless we introduce some of the most important basics and notational choices we make. For a vector $|\psi\rangle \in \mathcal{H}$ in a complex Euclidean space $\mathcal{H}$, we denote the standard Euclidean norm by $\| |\psi\rangle \|$. We use a subscript to indicate that a vector $|\psi\rangle$ is the state of a quantum register A with Hilbert space $\mathcal{H}$, i.e. $|\psi\rangle_A$. Similarly, M_A indicates that a matrix M acting on $\mathcal{H}$ is considered as acting on register A. The joint Hilbert space of multiple registers is given by the tensor product of the single-register Hilbert spaces. Where it helps simplify notation, we take the liberty to reorder registers, keeping track of them using register subscripts. The only other norm we will require is the trace norm. For a matrix M acting on $\mathcal{H}$, the trace norm $\|M\|_1$ is defined as the sum of the singular values of M. An important quantum gate is the quantum extension of the classical CNOT gate. This quantum gate is a unitary matrix CNOT acting on two qubits, i.e. on the vector space $\mathbb{C}^2 \otimes \mathbb{C}^2$, as $\mathrm{CNOT}\,|b_1\rangle\,|b_2\rangle = |b_1\rangle\,|b_2 \oplus b_1\rangle$. We sometimes subscript a CNOT gate with control register A and target register B with $A : B$, and extend this notation to the case where many CNOT gates are applied, i.e. $\mathrm{CNOT}^{\otimes n}_{A:B}$ means a CNOT gate is applied to the i-th qubit of the n-qubit registers A and B for each $i = 1, ..., n$ with the qubits in A being the controls and the ones in B the targets.

5.1 The superposition oracle

For proving the main result of this section, we will use the (simplest version of the) superposition oracle introduced in [39]. In the following, we introduce that technique, striving to keep this explanation accessible even to readers with minimal knowledge about quantum theory.

Superposition oracles are perfectly correct methods for simulating a quantum-accessible random oracle $\mathsf{O} : \{0,1\}^n \to \{0,1\}^m$. Different variants of the superposition oracle have different additional features that make them more useful than the quantum-accessible random oracle itself. We will use the fact that in the superposition oracle formalism, the reprogramming can be directly implemented by replacing a part of the quantum state held by the oracle, instead of using a simulator that sits between the original oracle and the querying algorithm. Notice that for this, we only need the simplest version of the superposition oracle from [39].[6] In that basic form, there are only three relatively simple conceptual steps underlying the construction of the superposition oracle, with the third one being key to its usefulness in analyses:

– For each $x \in \{0,1\}^n$, $\mathsf{O}(x)$ is a random variable uniformly distributed on $\{0,1\}^m$. This random variable can, of course, be sampled using a *quantum measurement*, more precisely a computational basis measurement of the state

$$|\phi_0\rangle = 2^{-m/2} \sum_{y \in \{0,1\}^m} |y\rangle .$$

– For a function $o : \{0,1\}^n \to \{0,1\}^m$, we can store the string $o(x)$ in a quantum register F_x. In fact, to sample $\mathsf{O}(x)$, we can prepare a register F_x in state $|\phi_0\rangle$, perform a computational basis measurement and keep the *collapsed* so-called *post-measurement state*. Outcome y of the measurement corresponds to the projector $|y\rangle\langle y|$, and a post-measurement state proportional to

$$|y\rangle\langle y| \, |\phi_0\rangle = 2^{-\frac{m}{2}} \, |y\rangle .$$

Now a query with input $|x\rangle_X |\psi\rangle_Y$ can be answered using CNOT gates, i.e. we can answer queries with a superposition oracle unitary O acting on input registers X, Y and an oracle register $F = F_{0^m} F_{0^{m-1}1} ... F_{1^m}$ such that

$$O_{XYF} \, |x\rangle\langle x|_X = |x\rangle\langle x|_X \otimes \left(\mathrm{CNOT}^{\otimes m}\right)_{F_x:Y} .$$

– Since the matrices $|y\rangle\langle y|_{F_x}$ and $\left(\mathrm{CNOT}^{\otimes m}\right)_{F_x:Y}$ commute, we can delay the measurement that performs the sampling of the random oracle until the end of the runtime of the querying algorithm. Queries are hence answered using the unitary O, but acting on oracle registers F_x that are all initialized in the uniform superposition state $|\phi_0\rangle$, and only after the querying algorithm has finished, the register F is measured to obtain the concrete random function O.

A quantum-accessible oracle for a random function $\mathsf{O} : \{0,1\}^n \to \{0,1\}^m$ is thus implemented as follows:

– Initialize: Prepare the initial state

$$|\Phi\rangle_F = \bigotimes_{x \in \{0,1\}^n} |\phi_0\rangle_{F_x} .$$

– Oracle: A quantum query on registers X and Y is answered using O_{XYF}

– Post-processing: Register F is measured to obtain a random function O. The

[6] Note that this basic superposition oracle does not provide an *efficient* simulation of a quantum-accessible random oracle, which is fine for proving a query lower bound that holds without assumptions about time complexity.

last step can be (partially) omitted whenever the function O is not needed for evaluation of the success or failure of the algorithm. In the following, the querying algorithm is, e.g. tasked with distinguishing two oracles, a setting where the final sampling measurement can be omitted.

Note that it is straightforward to implement the operation of reprogramming a random oracle to a fresh random value on a certain input x: just discard the contents of register F_x and replace them with a freshly prepared state $|\phi_0\rangle$. In addition, we need the following lemma

Lemma 1 (Lemma 2 in [3], reformulated). *Let $|\psi_q\rangle_{AF}$ be the joint adversary-oracle state after an adverary has made q queries to the superposition oracle with register F. Then this state can be written as*

$$|\psi_q\rangle_{AF} = \sum_{\substack{S\subset\{0,1\}^n\\|S|\le q}} |\psi_q^{(S)}\rangle_{AF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|)}\right)_{F_{S^c}},$$

where for any set $R = \{x_1, x_2, ..., x_{|R|}\} \subset \{0,1\}^n$ we have defined $F_R = F_{x_1}F_{x_2}...F_{x_{|R|}}$ and $|\psi_q^{(S)}\rangle_{AF_S}$ are vectors such that $\langle\phi_0|_{F_x} |\psi_q^{(S)}\rangle_{AF_S} = 0$ for all $x \in S$.

5.2 Reprogramming once

We are now ready to study our simple special case. Suppose a random oracle O is reprogrammed at a single input $x^* \in \{0,1\}^n$, sampled according to some probability distribution p, to a fresh random output $y^* \leftarrow \{0,1\}^m$. We set $\mathsf{O}_0 = \mathsf{O}$ and define O_1 by $\mathsf{O}_1(x^*) = y$ and $\mathsf{O}_1(x) = \mathsf{O}(x)$ for $x \neq x^*$. We will show that if x^* has sufficient min-entropy given O, such reprogramming is hard to detect.

More formally, consider a two-stage distinguisher $\mathsf{D} = (\mathsf{D}_0, \mathsf{D}_1)$. The first stage D_0 has trivial input, makes q quantum queries to O and ouputs a quantum state $|\psi_{int}\rangle$ and a sampling algorithm for a probability distribution p on $\{0,1\}^n$. The second stage D_1 gets $x^* \leftarrow p$ and $|\psi_{int}\rangle$ as input, has arbitrary quantum query access to O_b and outputs a bit b' with the goal that $b' = b$. We prove the following.

Theorem 6. *The success probability for any distinguisher D as defined above is bounded by*

$$\Pr[b = b'] \le \frac{1}{2} + \frac{1}{2}\sqrt{qp^{\mathsf{D}}_{\max}} + \frac{1}{4}qp^{\mathsf{D}}_{\max},$$

where the probability is taken over $b \leftarrow \{0,1\}, (|\psi_{int}\rangle, p) \leftarrow \mathsf{D}_0^{\mathsf{O}}(1^n)$ and $b' \leftarrow \mathsf{D}_1^{\mathsf{O}_b}(x^*, |\psi_{int}\rangle)$, and $p^{\mathsf{D}}_{\max} = \mathbb{E}_{(|\psi_{int}\rangle, p)\leftarrow \mathsf{D}_0^{\mathsf{O}_0}(1^n)} \max_x p(x)$.

Proof. We implement $\mathsf{O} = \mathsf{O}_0$ as a superposition oracle. Without loss of generality[7], we can assume that D proceeds by performing a unitary quantum computation, followed by a measurement to produce the classical output p and the

[7] This can be seen by employing the Stinespring dilation theorem, or by using standard techniques to delay measurement and discard operations until the end of a quantum algorithm.

discarding of a working register G. Let $|\gamma\rangle_{RGF}$ be the algorithm-oracle-state after the unitary part of D_0 and the measurement have been performed, conditioned on its second output being a fixed probability distribution p. R contains D_0's first output.

Define $\varepsilon_x = 1 - \left\| \langle\phi_0|_{F_x} |\gamma\rangle_{RGF} \right\|^2$, a measure of how far the contents of register F_x are from the uniform superposition. Intuitively, this is the 'probability' that the distinguisher knows $\mathsf{O}(x)$, and should be small in expectation over $x \leftarrow p$. We therefore begin by bounding the distinguishing advantage in terms of this quantity. For a fixed x, we can write the density matrix $\rho^{(0)} = |\gamma\rangle\langle\gamma|$ as

$$\begin{aligned}\rho^{(0)}_{RGF} = &\langle\phi_0|_{F_x} \rho^{(0)}_{RGF} |\phi_0\rangle_{F_x} \otimes |\phi_0\rangle\langle\phi_0|_{F_x} + \rho^{(0)}_{RGF} \left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right) \\ &+ \left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right) \rho^{(0)}_{RGF} |\phi_0\rangle\langle\phi_0|_{F_x} . \end{aligned} \tag{8}$$

The density matrix $\rho^{(1,x)}_{RGF}$ for the algorithm-oracle-state after D_0 has finished and the oracle has been reprogrammed at x (i.e. $b = 1$) is

$$\begin{aligned}\rho^{(1,x)}_{RGF} = &\mathrm{Tr}_{F_x}[\rho^{(1,x)}_{RGF}] \otimes |\phi_0\rangle\langle\phi_0|_{F_x} = \langle\phi_0|_{F_x} \rho^{(0)}_{RGF} |\phi_0\rangle_{F_x} \otimes |\phi_0\rangle\langle\phi_0|_{F_x} \\ &+ \mathrm{Tr}_{F_x}[(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x})\rho^{(0)}_{RGF}] \otimes |\phi_0\rangle\langle\phi_0|_{F_x}{}^{8}, \end{aligned} \tag{9}$$

where the second equality is immediate when computing the partial trace in an orthonormal basis containing $|\phi_0\rangle$.

We analyze the success probability of D. In the following, set $x^* = x$. The second stage, D_1, has arbitrary query access to the oracle O_b. In the superposition oracle framework, that means D_1 can apply arbitrary unitary operations on its registers R and G, and the oracle unitary O to some sub-register registers XY of G and the oracle register F. We bound the success probability by allowing arbitraty operations on F, thus reducing the oracle distinguishing task to the task of distinguishing the quantum states $\rho^{(b,x)}_{RF} = \mathrm{Tr}_G \rho^{(b,x)}_{RGF}$ for $b = 0, 1$, where $\rho^{(0,x)} := \rho^{(0)}$. By the bound relating distinguishing advantage and trace distance,

$$\Pr[b = b'|x^* = x] \leq \frac{1}{2} + \frac{1}{4}\left\|\rho^{(0)}_{RF} - \rho^{(1,x)}_{RF}\right\|_1 \leq \frac{1}{2} + \frac{1}{4}\left\|\rho^{(0)}_{RGF} - \rho^{(1,x)}_{RGF}\right\|_1, \tag{10}$$

where the probability is taken over $b \leftarrow \{0,1\}, |\psi_{int}\rangle \leftarrow \mathsf{D}_0^{\mathsf{O}_0}(1^n)$ and $b' \leftarrow \mathsf{D}_1^{\mathsf{O}_b}(x, |\psi_{int}\rangle)$, and we have used that the trace distance is non-increasing under

[8] Note that the partial trace expression yields a positive semidefinite matrix due to the cyclicity of the trace and the fact that $\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}$ is a projector and hence Hermitian.

partial trace. Using Equation (8) and (9), we bound

$$
\begin{aligned}
&\left\|\rho^{(0)}_{RGF} - \rho^{(1,x)}_{RGF}\right\|_1 \\
&\leq \Big\|\rho^{(0)}_{RGF}\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right) + \left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)\rho^{(0)}_{RGF}\,|\phi_0\rangle\langle\phi_0|_{F_x} \\
&\qquad - \mathrm{Tr}_{F_x}[(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x})\rho^{(0)}_{RGF}] \otimes |\phi_0\rangle\langle\phi_0|_{F_x}\Big\|_1 \\
&\leq \Big\|\rho^{(0)}_{RGF}\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)\Big\|_1 + \Big\|\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)\rho^{(0)}_{RGF}\,|\phi_0\rangle\langle\phi_0|_{F_x}\Big\|_1 \\
&\qquad + \Big\|\mathrm{Tr}_{F_x}[(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x})\rho^{(0)}_{RGF}] \otimes |\phi_0\rangle\langle\phi_0|_{F_x}\Big\|_1,
\end{aligned}
$$

Where the last line is the triangle inequality. The trace norm of a positive semidefinite matrix is equal to its trace, so the last term can be simplified as

$$
\begin{aligned}
&\Big\|\mathrm{Tr}_{F_x}[(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x})\rho^{(0)}_{RGF}] \otimes |\phi_0\rangle\langle\phi_0|_{F_x}\Big\|_1 \\
&\qquad = \mathrm{Tr}[(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x})\,|\gamma\rangle\langle\gamma|_{RGF}] = \varepsilon_x.
\end{aligned}
$$

The second term is upper-bounded by the first via Hölder's inequality, which simplifies as

$$
\begin{aligned}
&\Big\|\rho^{(0)}_{RGF}\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)\Big\|_1 = \Big\|\,|\gamma\rangle\langle\gamma|_{RGF}\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)\Big\|_1 \\
&= \Big\|\left(\mathbb{1} - |\phi_0\rangle\langle\phi_0|_{F_x}\right)|\gamma\rangle_{RGF}\Big\|_2 = \sqrt{\varepsilon_x}
\end{aligned}
$$

where the second equality uses that $|\gamma\rangle$ is normalized. In summary we have

$$
\left\|\rho^{(0)}_{RGF} - \rho^{(1,x)}_{RGF}\right\|_1 \leq 2\sqrt{\varepsilon_x} + \varepsilon_x. \tag{11}
$$

It remains to bound ε_x in expectation over $x \leftarrow p$. To this end, we prove

$$
\mathbb{E}_{x^* \leftarrow p}\left[\left\|\,\langle\phi_0|_{F_{x^*}}\,|\gamma\rangle_{RGF}\right\|^2\right] \geq 1 - q p_{\max}, \tag{12}
$$

where $p_{\max} = \max_x p(x)$. In the following, sums over S are taken over $S \subset \{0,1\}^n : |S| \leq q$, with additional restrictions explicitly mentioned. We have

$$
\begin{aligned}
&\mathbb{E}_{x^* \leftarrow p}\left[\left\|\,\langle\phi_0|_{F_{x^*}}\,|\gamma\rangle_{RGF}\right\|^2\right] = \sum_{x^* \in \{0,1\}^n} p(x^*)\left\|\,\langle\phi_0|_{F_{x^*}}\,|\gamma\rangle_{RGF}\right\|^2 \\
&\qquad = \sum_{x^* \in \{0,1\}^n} p(x^*)\Big\|\sum_S \langle\phi_0|_{F_{x^*}}\,|\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n - |S|)}\right)_{F_{S^c}}\Big\|^2,
\end{aligned}
$$

where we have used Lemma 1 as well as the notation $|\psi_q^{(S)}\rangle$ from there. (Lemma 1 clearly also holds after the projector corresponding to second output equaling p is applied). Using $\langle\phi_0|_{F_x}\,|\psi_q^{(S)}\rangle_{RGF_S} = 0$ for all $x \in S$ we simplify

$$
\begin{aligned}
&\sum_{x^* \in \{0,1\}^n} p(x^*)\Big\|\sum_S \langle\phi_0|_{F_{x^*}}\,|\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n - |S|)}\right)_{F_{S^c}}\Big\|^2 \\
&\qquad = \sum_{x^* \in \{0,1\}^n} p(x^*)\Big\|\sum_{S \not\ni x^*} |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n - |S| - 1)}\right)_{F_{S^c \setminus \{x^*\}}}\Big\|^2.
\end{aligned}
$$

The summands in the second sum are pairwise orthogonal, so

$$\sum_{x^*\in\{0,1\}^n} p(x^*)\Big\| \sum_{S\not\ni x^*} |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|-1)}\right)_{F_{S^c\setminus\{x^*\}}} \Big\|^2$$
$$= \sum_{x^*\in\{0,1\}^n} p(x^*) \sum_{S\not\ni x^*} \Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|-1)}\right)_{F_{S^c\setminus\{x^*\}}} \Big\|^2$$
$$= \sum_S \sum_{x^*\in S^c} p(x^*)\Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|-1)}\right)_{F_{S^c\setminus\{x^*\}}} \Big\|^2$$
$$= \sum_S \sum_{x^*\in S^c} p(x^*)\Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|)}\right)_{F_{S^c}} \Big\|^2$$

where we have used the fact that the state $|\phi_0\rangle$ is normalized in the last line. But for any $S \subset \{0,1\}^n$ we have

$$\sum_{x^*\in S^c} p(x^*) = 1 - \sum_{x^*\in S} p(x^*) \geq 1 - |S| p_{\max},$$

where here, $p_{\max} = \max_x p(x)$. We hence obtain

$$\sum_S \sum_{x^*\in S^c} p(x^*)\Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|)}\right)_{F_{S^c}} \Big\|^2$$
$$\geq \sum_S (1-|S|p_{\max})\Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|)}\right)_{F_{S^c}} \Big\|^2$$
$$\geq (1-qp_{\max}) \sum_S \Big\| |\psi_q^{(S)}\rangle_{RGF_S} \otimes \left(|\phi_0\rangle^{\otimes(2^n-|S|)}\right)_{F_{S^c}} \Big\|^2 = 1 - qp_{\max},$$

where we have used the normalization of $|\gamma\rangle_{RGF}$ in the last equality. Combining the above equations proves Equation (12). Putting everything together, we bound

$$\Pr[b=b'] = \mathbb{E}_p \mathbb{E}_x \Pr[b=b'|p,x] \leq \frac{1}{2} + \frac{1}{4}\mathbb{E}_p\mathbb{E}_x[2\sqrt{\varepsilon_x} + \varepsilon_x]$$
$$\leq \frac{1}{2} + \frac{1}{4}\mathbb{E}_p[2\sqrt{qp_{\max}} + qp_{\max}] \leq \frac{1}{2} + \frac{1}{2}\sqrt{qp^{\mathsf{D}}_{\max}} + qp^{\mathsf{D}}_{\max}.$$

Here, the inequalities are due to Equation (10) and Equation (11), Equation (12) and Jensen's inequality, and another Jensen's inequality, respectively. □

6 A matching attack

We now describe an attack matching the bound presented in Theorem 6. For simplicity, we restrict our attention to the case where just one point is (potentially) reprogrammed.

Our distinguisher makes q queries to O, the oracle before the potential reprogramming, and q queries to O', the oracle after the potential reprogramming. In our attack, we fix an arbitrary cyclic permutation σ on $[2^n]$, and for the fixed reprogrammed point x^*, we define $S = \{x^*, \sigma^{-1}(x^*), ..., \sigma^{-q+1}(x^*)\}$, $\overline{S} = \{0,1\}^n \setminus S$, $\mathit{\Pi}_0 = \frac{1}{2}\left(|S\rangle + |\overline{S}\rangle\right)\left(\langle S| + \langle \overline{S}|\right)$ and $\mathit{\Pi}_1 = I - \mathit{\Pi}_0$.[9] The distinguisher D is defined in Fig. 13.

Before potential reprogramming:

Prepare registers XY in $\frac{1}{\sqrt{2^n}} \sum_{x \in [2^n]} |x, 0\rangle_{XY}$
Query O using registers XY
for $i = 0, ..., q-2$:
    Apply σ on register X
    Query O using registers XY

After potential reprogramming:

Query O' using using registers XY
for $i = q-2, ..., 0$:
    Apply σ^{-1} on register X
    Query O' using registers XY
Measure X according to $\{\mathit{\Pi}_0, \mathit{\Pi}_1\}$
Output b if the state projects onto $\mathit{\Pi}_b$.

Fig. 13. Distinguisher for a single reprogrammed point.

Theorem 7. *For every $1 \le q < 2^{n-3}$, the attack described in Figure 13 can be implemented in quantum polynomial-time. Performing q queries each before and after the potential reprogramming, it detects the reprogramming of a random oracle $\mathsf{O} : \{0,1\}^n \to \{0,1\}^m$ at a single point with probability at least $\mathit{\Omega}(\sqrt{\frac{q}{2^n}})$.*

Proof (sketch). We can analyze the state of the distinguisher before its measurement. If the oracle is not reprogrammed, then its state is

$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle\, |0\rangle\,,$$

whereas if the reprogramming happens, its state is

$$\sum_{x \in S} |x\rangle\, |\mathsf{O}(x^*) \oplus \mathsf{O}'(x^*)\rangle + \sum_{x \in \overline{S}} |x\rangle\, |0\rangle\,,$$

where $\mathsf{O}(x^*) \oplus \mathsf{O}'(x^*)$ is a uniformly random value. The advantage follows by calculating the probability that these states project onto $\mathit{\Pi}_0$.

For the efficiency of our distinguisher, we can use the tools provided in [2] to efficiently implement $\mathit{\Pi}_0$ and $\mathit{\Pi}_1$, which are the only non-trivial operations of the attack.

Due to space restrictions, we refer to the full version , where we give the full proof of Theorem 7 and discuss its extension to multiple reprogrammed points.

References

1. M. Abdalla, P.-A. Fouque, V. Lyubashevsky, and M. Tibouchi. Tightly-secure signatures from lossy identification schemes. In *EUROCRYPT 2012*, LNCS, pages 572–590, 2012.

[9] Formally, S, $\mathit{\Pi}_0$ and $\mathit{\Pi}_1$ are functions of x^* but we omit this dependence for simplicity, since we can assume that x^* is fixed.

2. G. Alagic, C. Majenz, and A. Russell. Efficient simulation of random states and random unitaries. In *EUROCRYPT 2020*, LNCS, pages 759–787, 2020.
3. G. Alagic, C. Majenz, A. Russell, and F. Song. Quantum-access-secure message authentication via blind-unforgeability. In *EUROCRYPT 2020*, LNCS, pages 788–817, 2020.
4. A. Ambainis, M. Hamburg, and D. Unruh. Quantum security proofs using semi-classical oracles. In *CRYPTO 2019*, LNCS, pages 269–295, 2019.
5. D. F. Aranha, C. Orlandi, A. Takahashi, and G. Zaverucha. Security of hedged Fiat-Shamir signatures under fault attacks. In *EUROCRYPT 2020*, LNCS, pages 644–674, 2020.
6. R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. Wolf. Quantum lower bounds by polynomials. In *39th FOCS*, pages 352–361, 1998.
7. N. Bindel, M. Hamburg, K. Hövelmanns, A. Hülsing, and E. Persichetti. Tighter proofs of CCA security in the quantum random oracle model. In *TCC 2019*, LNCS, pages 61–90, 2019.
8. D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry. Random oracles in a quantum world. In *ASIACRYPT 2011*, LNCS, pages 41–69, 2011.
9. J. W. Bos, A. Hülsing, J. Renes, and C. van Vredendaal. Rapidly Verifiable XMSS Signatures. Cryptology ePrint Archive, Report 2020/898, 2020. https://eprint.iacr.org/2020/898.
10. A. Broadbent and S. Lord. Uncloneable Quantum Encryption via Oracles. In S. T. Flammia, editor, *TQC 2020*, LIPIcs, pages 4:1–4:22, Dagstuhl, Germany, 2020.
11. J. A. Buchmann, E. Dahmen, and A. Hülsing. XMSS - A practical forward secure signature scheme based on minimal security assumptions. In *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 117–129, 2011.
12. M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1825–1842, New York, NY, USA, 2017. Association for Computing Machinery.
13. M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska, and P. Schwabe. From 5-pass MQ-based identification to MQ-based signatures. In *ASIACRYPT 2016*, LNCS, pages 135–165, 2016.
14. A. Coladangelo, C. Majenz, and A. Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. arXiv 2009.13865, 2020.
15. J. Don, S. Fehr, and C. Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In *CRYPTO 2020*, LNCS, pages 602–631, 2020.
16. J. Don, S. Fehr, C. Majenz, and C. Schaffner. Security of the Fiat-Shamir transformation in the quantum random-oracle model. In *CRYPTO 2019*, LNCS, pages 356–383, 2019.
17. E. Eaton and F. Song. Making Existential-unforgeable Signatures Strongly Unforgeable in the Quantum Random-oracle Model. In *TQC 2015*, LIPIcs, 2015.
18. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *CRYPTO'99*, LNCS, pages 537–554, 1999.
19. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, Jan. 2013.
20. D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. In *TCC 2017*, LNCS, pages 341–371, 2017.

21. K. Hövelmanns, E. Kiltz, S. Schäge, and D. Unruh. Generic authenticated key exchange in the quantum random oracle model. In *PKC 2020*, LNCS, pages 389–422, 2020.
22. A. Hülsing, D. Butin, S.-L. Gazdag, J. Rijneveld, and A. Mohaisen. XMSS: Extended Hash-Based Signatures. RFC 8391, 2018.
23. A. Hülsing, J. Rijneveld, and F. Song. Mitigating multi-target attacks in hash-based signatures. In *PKC 2016*, LNCS, pages 387–416, 2016.
24. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *CRYPTO 2018*, LNCS, pages 96–125, 2018.
25. H. Jiang, Z. Zhang, and Z. Ma. Key encapsulation mechanism with explicit rejection in the quantum random oracle model. In *PKC 2019*, LNCS, pages 618–645, 2019.
26. E. Kiltz, V. Lyubashevsky, and C. Schaffner. A concrete treatment of Fiat-Shamir signatures in the quantum random-oracle model. In *EUROCRYPT 2018*, LNCS, pages 552–586, 2018.
27. V. Kuchta, A. Sakzad, D. Stehlé, R. Steinfeld, and S. Sun. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *EUROCRYPT 2020*, LNCS, pages 703–728, 2020.
28. Q. Liu and M. Zhandry. Revisiting post-quantum Fiat-Shamir. In *CRYPTO 2019*, LNCS, pages 326–355, 2019.
29. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT 2009*, LNCS, pages 598–616, 2009.
30. M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
31. NIST. National institute for standards and technology. postquantum crypto project, 2017. http://csrc.nist.gov/groups/ST/post-quantum-crypto/.
32. NIST. Status report on the second round of the nist post-quantum cryptography standardization process. NISTIR 8309, 2020. https://doi.org/10.6028/NIST.IR.8309.
33. T. Saito, K. Xagawa, and T. Yamakawa. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *EUROCRYPT 2018*, LNCS, pages 520–551, 2018.
34. D. Unruh. Quantum position verification in the random oracle model. In *CRYPTO 2014*, LNCS, pages 1–18, 2014.
35. D. Unruh. Revocable quantum timed-release encryption. In *EUROCRYPT 2014*, LNCS, pages 129–146, 2014.
36. D. Unruh. Non-interactive zero-knowledge proofs in the quantum random oracle model. In *EUROCRYPT 2015*, LNCS, pages 755–784, 2015.
37. D. Unruh. Post-quantum security of Fiat-Shamir. In *ASIACRYPT 2017*, LNCS, pages 65–95, 2017.
38. T. Yamakawa and M. Zhandry. A note on separating classical and quantum random oracles. Cryptology ePrint Archive, Report 2020/787, 2020. https://eprint.iacr.org/2020/787.
39. M. Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *CRYPTO 2019*, LNCS, pages 239–268, 2019.