Quantum security of the Fiat-Shamir transform of commit and open protocols

André Chailloux*

Abstract

Applying the Fiat-Shamir transform on identification schemes is one of the main ways of constructing signature schemes. While the classical security of this transformation is well understood, it is only very recently that generic results for the quantum case has been proposed [DFMS19, LZ19]. In this paper, we show that if we start from a commit-and-open identification scheme, where the prover first commits to several strings and then as a second message opens a subset of them depending on the verifier's message, then the Fiat-Shamir transform is quantum secure, for a suitable choice of commitment scheme. Unlike previous generic results, our transformation doesn't require to reprogram the random function $\mathcal{H}$ used in the Fiat-Shamir transform and we actually only require a quantum one-wayness property.

Our techniques can in some cases lead to a much tighter security reduction. To illustrate this, we apply our techniques to identifications schemes at the core of the MQDSS signature scheme, the Picnic scheme (both present in the round 2 of the post quantum NIST competition) and the Stern signature scheme. For all these schemes, we show that our technique can be applied with essentially tight results.

1 Introduction

There has been a strong interest in post-quantum cryptography in the last years. While we are still very far from having a full quantum computer, there are important technological advances each year and it is still very possible that future quantum computers will be powerful enough to run Shor's algorithm [Sho94] or other quantum algorithms devastating for current cryptography.

Post-quantum cryptosystems are based on computational problems which are not known to be broken by quantum computers like problems based on lattices, multivariate polynomials, isogenies or error correcting codes; and there is currently a standardization process of post-quantum cryptosystems organized by the NIST [Nis17]. Quite surprisingly, while all the proposals are based on problems believed to be hard for quantum computers, many of the submissions, even the round 2 submissions, do not have a proper security proof against quantum computers. This is specially true for signature schemes where about half of the round 2 submissions do not have an explicit quantum security proof, even if the recent results of [DFMS19, LZ19] are quickly solving this problem. This shows how hard these quantum security proofs can be.

*Inria de Paris, EPI SECRET. Email: andre.chailloux@inria.fr.

1.1 The quantum Fiat-Shamir transform for signature schemes

In this paper, we will be interested in some technical aspects related to proving the security of signatures schemes in the Quantum Random Oracle Model (QROM) using the Fiat-Shamir transform. The Fiat-Shamir is a very important procedure that can transform (interactive) identification schemes into non-interactive ones. For a long time, nothing was known about the quantum security of the Fiat-Shamir transform. First impossibility results showed settings where, in all generality, the quantum Fiat-Shamir transform is not secure [DFG13, ARU14]. On the positive side, [DFG13] proved the security of the quantum Fiat-Shamir transform when oblivious commitments are used. Unruh [Unr15] then showed that it was possible to do a Fiat-Shamir like transform to remove the interaction from identification protocols. This transform is however rather inefficient and was hardly used in practice. More recently, there have been new exciting results related to the quantum security of the Fiat-Shamir transform. If an identification scheme is lossy, then [KLS18] showed that the Fiat-Shamir transform is quantum secure. They used this result to prove the security of the Dilithium signature [DKL$^+$17], which is a NIST competitor. Another result is the security proof of qTESLA [ABB$^+$19]. Unruh [Unr17] also showed the quantum security of the Fiat-Shamir transform for identification schemes with statistical security where the security of the underlying signature scheme with a dual-mode hard instance generator, a property closely related to the lossiness property.

Recently, 2 papers [DFMS19, LZ19] showed generic reduction for the quantum Fiat-Shamir transform. Unlike what was believed before, they show that it is actually possible to perform reprogramming of a quantum random oracle and to follow the classical proofs. Their results are not tight and lose at least a factor of $O(q^2)$ where q is the number of queries to the random function. Moreover, they didn't apply their results to signature schemes like MQDSS and Picnic since their underlying identification schemes only have 3-special soundness and not regular soundness.

1.2 Contributions and techniques

Motivated by the concrete security study of MQDSS and Picnic, we study commit-and-open identification schemes. In a commit-and-open identification scheme, the prover first commits to several strings and then as a second message opens a subset of them depending on the verifier's message. Our contributions are the following:

1. We prove that with a well chosen commitment scheme, namely a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$ we have quantum security of signature schemes constructed from a commit-and-open identification scheme with the Fiat-Shamir transform in the Quantum Random Oracle Model. Unlike previous results, we show how to start from the special soundness property and we don't require reprogramming of the random function

2. The above technique can also lead to more efficient reductions. We can apply our results to the Picnic (including the one that uses plain Fiat-Shamir) signature scheme[CDG$^+$17] and to the Stern signature scheme[Ste93], for which we show an almost tight (*i.e.* tight up to log factors) quantum security.

   We also apply our results to the 3-pass identification scheme [SSH11]. The MQDSS signature [CHR$^+$16] is actually based on the optimized 5-pass variant of [SSH11] so we essentially prove quantum security of the non-optimized variant of MQDSS. We leave the quantum security of the full scheme as an open question.

3. Our first contribution requires to use as a commitment a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$ for some l. This can make the signature scheme very inefficient. We show how to circumvent this inefficiency by using random sponges which will make the commitments much smaller and the scheme much more efficient.

In order to prove our results, we use the following strategy:

- We generalize the result in [Unr17] and show that we can prove quantum security for an identification scheme with *computational-statistical-soundness* and not statistical soundness. The notion of *computational-statistical-soundness* captures security against a prover who is computationally limited when sending the first message but with unbounded power when sending the second message.
- We show that any commit-and-open identification scheme that uses as a commitment a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$ indeed has *computational-statistical-soundness*. This also gives the strict soundness property to our scheme, which is known to be necessary [ARU14].
- For our second contribution on existing schemes, we actually use Zhandry's framework on how to record quantum queries. This will allow to remove the quadratic loss that comes from one of the steps of our proof (Proposition 3). From a technical standpoint, we generalize a result of Zhandry that relates the contents of his compressed oracle and of his standard oracle.

1.3 Organization of the paper

In Section 3, we present different definitions for identification schemes and signature schemes. In Section 4, we show our results on generic commit-and-open identification schemes. In Section 5, we show how to make this security much tighter when considering identification schemes which are a parallel repetition of commit-and-open schemes of challenge size 3. In Section 7, we show how to instantiate the commitments we require with a random sponge while preserving efficient communication.

2 Preliminaries

2.1 Quantum query algorithms and the quantum random oracle model

For any quantum algorithm $\mathscr{A}$, we denote by $|\mathscr{A}|$ it's total running time. We will also consider query algorithms $\mathscr{A}^{\mathscr{O}}$ that will make a certain amount of calls to an oracle $\mathscr{O}$. The Quantum Random Oracle Model (QROM) is a model where we model a certain function with a random function $\mathcal{H}$. Since we are in the quantum setting, we have a black box access to $\mathcal{H}$ but also to the unitary $\mathscr{O}_{\mathcal{H}}(|x\rangle|y\rangle) = |x\rangle|\mathcal{H}(x)+y\rangle$. Zhandry presented in [Zha18] an alternative way at looking at the QROM that we present in Section 6.

2.2 Quantum lower bounds

The study of quantum security in the QROM involves several quantum query lower bounds. We present here those that will be used in this paper. Throughout the paper, $x \overset{\$}{\leftarrow} S$ means that x is chosen uniformly at random from S.

Lemma 1 ([Zha15]). *For any quantum query algorithm $\mathscr{A}^{\mathscr{O}}$ making q queries to $\mathscr{O}$, we have*

$$\left|\Pr[\mathscr{A}^{\mathscr{O}_\Gamma} \text{ outputs } 0 : \Gamma \xleftarrow{\$} \mathscr{F}_X^Y] - \Pr[\mathscr{A}^{\mathscr{O}_G} \text{ outputs } 0 : G \xleftarrow{\$} \mathscr{F}_X^Y(r)]\right| \leq O(\frac{q^3}{r}).$$

where $\mathscr{F}_X^Y$ is the set of functions from X to Y and $\mathscr{F}_X^Y(r)$ is the set of functions f from X to Y such that $|Im(f)| = r$.

Another useful quantum lower bound is a generalization of Grover's lower bound.

Lemma 2. *Let $\mathscr{F}_X^Y$ be the set of random functions from X to Y. For each $x \in X$, we associate a set $U_x \subseteq Y$ such that $\frac{|U_x|}{|Y|} \leq \varepsilon$. For any quantum query algorithm $\mathscr{A}^{\mathscr{O}}$ making q queries to $\mathscr{O}$, we have*

$$\Pr[\mathcal{H}(x) \in U_x : \mathcal{H} \xleftarrow{\$} \mathscr{F}_X^Y, x \leftarrow \mathscr{A}^{\mathscr{O}_\mathcal{H}}(\cdot)] \leq O(q^2\varepsilon).$$

The above lemma was implicitly stated and proven in [Unr17, Theorem21]. The idea is to construct a function $\mathcal{H}_2(x) = \mathcal{H}(x) + u(x)$ where $u(x)$ is a random element from U_x and use standard lower bounds for Grover search on random functions. It is also possible to directly use the recent framework of recording of quantum queries [Zha18, Theorem 4.1] to obtain exactly the same result.

Finally, we present another result by Zhandry[Zha12] that states that the following

Lemma 3. *Let $\mathscr{F}_X^Y$ be the set of functions from X to Y and let $W_X^Y(2q)$ be a set of $2q$-wise independent functions from X to Y. For any quantum query algorithm $\mathscr{A}^{\mathscr{O}}$ making q queries to $\mathscr{O}$, we have*

$$\Pr[\mathscr{A}^{\mathscr{O}_\mathcal{H}} \text{ outputs } 0 : \mathcal{H} \xleftarrow{\$} \mathscr{F}_X^Y] = \Pr[\mathscr{A}^{\mathscr{O}_f} \text{ outputs } 0 : f \xleftarrow{\$} W_X^Y(2q)].$$

We can construct sets $W_X^Y(2q)$ such that it is possible to generate $f \xleftarrow{\$} W_{2q}$, compute f and compute f^{-1} in time $O(q\log(|Y|))$. Take for example for W_{2q} random polynomials of degree $2q-1$. More discussion on this can be found in [Unr15].

3 Identification schemes and signature schemes

Throughout the paper, $x \xleftarrow{\$} S$ means that x is chosen uniformly at random from S. We put some quantum preliminaries on the quantum random oracle and quantum lower bounds in Appendix 2.

3.1 Identification schemes

An identification scheme $\mathcal{IS} = (\text{Keygen}, P, V, S_{ch})$, consists of the following:

- A key generation algorithm $\text{Keygen}(1^\lambda) \rightarrow (pk, sk)$.
- The prover's algorithm $P = (P_1, P_2)$ for constructing his messages. We have $P_1(sk) \rightarrow (x, St)$ where x corresponds to the first message and St is some internal state. $P_2(sk, x, c, St) \rightarrow z$ where $c \in S_{ch}$ is the challenge from the verifier and z the prover's response (second message).
- A verification function $V(pk, x, c, z)$ used by the verifier that outputs a bit, 0 corresponds to 'Reject' and 1 to 'Accept'.

We do not specify here the different string lengths of x and z to not make notations too heavy. We explicit the challenge space S_{ch} as it will often appear in our definitions and statements. We want all the different algorithms presented above to be efficient and we will usually omit their running times (*i.e.* fix them to 1), again to significantly reduce the amount of notations we introduce. Even though we deal with concrete security parameters in this paper, we kept the notation $\mathrm{Keygen}(1^\lambda)$ with a unary representation of a security parameter to remind this implicit efficiency requirement. We present below more precisely the different steps of an identification scheme.

Identification scheme $\mathcal{IS} = (\mathrm{Keygen}, P = (P_1, P_2), V, S_{ch})$

Initialization. $(pk, sk) \leftarrow \mathrm{Keygen}(1^\lambda)$. The prover has (pk, sk) and the verifier pk.
Interaction.

1. P generates $(x, St) \leftarrow P_1(sk)$ and sends x to the verifier.
2. The verifier sends a uniformly random $c \in S_{Ch}$.
3. P generates $z \leftarrow P_2(sk, x, c, St)$ and sends z to the verifier.

Verification. The verifier accepts iff. $V(x, c, z) = 1$.

The first property we want from an identification scheme is that the verifier accepts if a prover runs the scheme honestly.

Definition 1 (Completeness)**.** *An identification scheme* $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$ *has perfect completeness if*

$$\Pr[V(x, c, z) = 1 : (pk, sk) \leftarrow \mathrm{Keygen}(1^\lambda), (x, St) \leftarrow P_1(sk), c \overset{\$}{\leftarrow} S_{ch}, z \leftarrow P_2(sk, x, c, St)] = 1.$$

The second property we want is honest-verifier zero-knowledge, meaning that an honest verifier cannot extract any information (in particular about the secret key sk), from its interaction with an honest prover.

Definition 2 (HVZK)**.** *An identification scheme* $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$ *is* ε*-HVZK if there exists an efficient simulator* S *such that the* 2 *distributions:*

- $D_1 : (pk, sk) \leftarrow \mathrm{Keygen}(1^\lambda), (x, St) \leftarrow P_1(sk), c \overset{\$}{\leftarrow} S_{ch}, z \leftarrow P_2(sk, x, c, St)$*, return* (x, c, z)*.*
- $D_2 : (x', c', z') \leftarrow S(pk, 1^\lambda)$*, return* (x', c', z')*.*

have statistical distance at most ε*.*

Finally, the third property that we require is soundness. We don't want a cheating prover that doesn't know the secret key sk to make the verifier accept.

3.2 Different flavors of soundness

There are different notions of soundness and the interplay between them will play an important role in our proofs. We put directly the running time of the attacker t in those definitions instead of just putting a polynomially bounded prover. This type of definition is better suited when dealing with concrete security bounds.

We first define the notions of soundness advantage and special-soundness advantage for a cheating adversary $\mathscr{A}$.

Definition 3. *Let $\mathcal{IS} = (\text{Keygen}, P, V, S_{ch})$ be an identification scheme. For any quantum algorithm (a quantum cheating prover) $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$, we define*

$$QADV_{\mathcal{IS}}(\mathscr{A}) = \Pr[V(x,c,z) = 1 : (pk, sk) \leftarrow \text{Keygen}(1^\lambda), (x, St) \leftarrow \mathscr{A}_1(pk), c \overset{\$}{\leftarrow} S_{ch}, z \leftarrow \mathscr{A}_2(pk, x, c, St)]$$

$$QADV^{sp}_{\mathcal{IS}}(\mathscr{A}) = \Pr\Big[V(x,c,z) = 1 \wedge V(x,c',z') = 1 : (pk, sk) \leftarrow \text{Keygen}(1^\lambda)\ ;\ (x,c,z,c',z') \leftarrow \mathscr{A}(pk)\Big].$$

Remark: Special soundness corresponds usually to the existence of an efficient extractor E such that $E(pk, x, c, z, c', z')$ produces a valid secret key sk from a pair of accepting transcripts (x, c, z) and (x, c', z). In the context of identification schemes, it is always coupled with the hardness of generating a valid secret key. If such an extractor E exists then our quantity $QADV^{sp}_{\mathcal{IS}}(\mathscr{A})$ is upper bounded by the probability of outputting a valid secret key. Therefore, these 2 notions of special soundness play the same role and are essentially equivalent.

From these definitions, we can define the notion of advantage related to respectively computational soundness, statistical soundness and special soundness.

Definition 4. *Let $\mathcal{IS} = (\text{Keygen}, P, V, S_{ch})$ be an identification scheme. We define*

$$\begin{aligned} QADV_{\mathcal{IS}}(t) &= \max_{\substack{\mathscr{A}=(\mathscr{A}_1,\mathscr{A}_2),\\ |\mathscr{A}_1|+|\mathscr{A}_2|=t}} QADV_{\mathcal{IS}}(\mathscr{A}) \\ QADV^{st}_{\mathcal{IS}} &= \max_{\mathscr{A}=(\mathscr{A}_1,\mathscr{A}_2)} QADV_{\mathcal{IS}}(\mathscr{A}) \\ QADV^{sp}_{\mathcal{IS}}(t) &= \max_{\substack{\mathscr{A}=(\mathscr{A}_1,\mathscr{A}_2),\\ |\mathscr{A}_1|+|\mathscr{A}_2|=t}} QADV^{sp}_{\mathcal{IS}}(\mathscr{A}) \end{aligned}$$

When we talk about soundness, we will actually talk about the advantage related to those different notions of soundness. Next, we define a new hybrid notion of advantage between computational and statistical soundness: *computational-statistical-soundness*. Here we want the prover to be bounded in the first message but unbounded in the second message.

Definition 5. *Let $\mathcal{IS} = (\text{Keygen}, P, V, S_{ch})$ be an identification scheme. We define*

$$QADV^{cs}_{\mathcal{IS}}(t) = \max_{\substack{\mathscr{A}=(\mathscr{A}_1,\mathscr{A}_2),\\ |\mathscr{A}_1|=t}} QADV_{\mathcal{IS}}(\mathscr{A})$$

The relationship between those different notions is the following, for all t

$$QADV_{\mathcal{IS}}(t) \leq QADV^{cs}_{\mathcal{IS}}(t) \leq QADV^{st}_{\mathcal{IS}}.$$

Finally, we define the notion of strict soundness which says that for the second message, there is at most one valid message z that the verifier will accept.

Definition 6. *Let $\mathcal{IS} = (\text{Keygen}, P, V, S_{ch})$ be an identification scheme. We say that $\mathcal{IS}$ has strict soundness iff.*

$$\forall x,\ \forall c,\ |\{z : V(x,c,z) = 1\}| \leq 1.$$

3.3 Soundness vs. special soundness

The reason why we have to deal with special soundness is that in many cases, we can construct identification schemes for which the special soundness can be directly reduced to a computationally hard problem. However, when we want to use identification schemes for instance for signature schemes, we require them to have computational soundness. Therefore, we need to find ways to relate them.

In the classical setting, we can actually interpret the use of the forking lemma [PS96] as a way to relate the soundness and the special soundness of the underlying protocols. In the quantum setting, we often seem to require powerful theorems such as a quantum forking lemma or quantum rewinding, which are known to be hard problems. In the context of Fiat-Shamir constructions of signature schemes, many of these problems can be seen as a way to relate the soundness and the special soundness of identification schemes (or more generally Σ-protocols).

We present here the relation we will use between those soundness notions in the quantum setting.

Proposition 1. *Let* $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$ *be an identification scheme with strict soundness. For any* t*,*

$$QADV_{\mathcal{IS}}(t) \leq \frac{1}{|S_{ch}|} + 4\left(QADV^{sp}_{\mathcal{IS}}(2t)\right)^{1/3}.$$

Depending on the context (short challenge size, not perfect strict soundness, ...) there can be different relations which will be better, see [Unr12, CSST11, CL17]. We'll prove this proposition in Appendix A

3.4 The Fiat-Shamir transform for identification schemes

The Fiat-Shamir transform [FS87] is a major cryptographic construction that converts any Σ-protocol into an non-interactive protocol. The idea is use a function $\mathcal{H}$, modeled as a random function, and to replace the verifier's challenge $c \in S_{ch}$ by the string $\mathcal{H}(x)$ where x is the prover's first message. Since the prover can compute $\mathcal{H}(x)$ himself, there is no need for interaction anymore. For any identification scheme $\mathcal{IS}$, we denote by $\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]$ its Fiat-Shamir transform.

Running $\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]$ for an identification scheme $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$

Initialization. $(pk, sk) \leftarrow \mathrm{Keygen}(1^{\lambda})$. The prover has (pk, sk) and the verifier pk.
One-way communication. P generates $(x, St) \leftarrow P_1(sk)$, computes $c = \mathcal{H}(x)$ and generates $z \leftarrow P_2(sk, x, c, St)$. He sends the pair (x, z) to the verifier.
Verification. The verifier accepts iff. $V(x, \mathcal{H}(c), z) = 1$.

The Fiat-Shamir transform is very useful as it can be used (among other things) to construct signature schemes from identification schemes. As for identification schemes, we can define soundness properties. Here we will only present computational soundness.

Definition 7. *Let* $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$ *be an identification scheme and* $\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]$ *its Fiat-*

Shamir transform. Let $\mathscr{A}^{\mathscr{O}_{\mathcal{H}}}$ be a query algorithm. We define

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]}(\mathscr{A}^{\mathscr{O}_{\mathcal{H}}}) = \Pr[V(x, \mathcal{H}(c), z) = 1 : (pk, sk) \leftarrow \mathrm{Keygen}(1^{\lambda}), (x, z) \leftarrow \mathscr{A}^{\mathscr{O}_{\mathcal{H}}}].$$

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]}(t) = \max_{\mathscr{A}^{\mathscr{O}_{\mathcal{H}}} : |\mathscr{A}^{\mathscr{O}_{\mathcal{H}}}| = t} QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]}(\mathscr{A}^{\mathscr{O}_{\mathcal{H}}})$$

This advantage notion will be directly related to the security of the associated signature scheme.

3.5 Commit and open identification schemes

A commit-and-open identification scheme is a specific kind of identification schemes where, for the first message, P commits to some values $y_1, \ldots, y_n$ and after the verifier's challenge, he reveals a subset of those values. More precisely, a commit-and-open identification scheme $\mathcal{IS}(\mathrm{Keygen}, P, G, V', S_{ch})$ consists of the following

- A key generation algorithm $\mathrm{Keygen}(1^{\lambda}) \to (pk, sk)$.
- A function $G : \{0,1\}^l \to \{0,1\}^m$ that will act as a commitment scheme.
- $P_1(sk) \to (x, St)$ has to output $x = (G(y_1), \ldots, G(y_n))$ for some values y_i and $St = y_1, \ldots, y_n$.
- The challenge c corresponds to a subset I_c of $\{1, \ldots, n\}$.
- P_2 always outputs $z = y_{I_c}$ where $y_{I_C} = y_{i_1}, \ldots, y_{i_{|I_c|}}$ for $I_c = \{i_1, \ldots, i_{|I_c|}\}, i_1 < i_2 < \cdots < i_{|I_c|}$.
- The verification function V must satisfy

$$V(pk, x, c, z) = 1 \Leftrightarrow (\forall i \in I_c, G(y_i) = x_i) \wedge V'(pk, c, z) = 1.$$

Here, we explicit 3 parameters l, m, n: each $y_i \in \{0,1\}^l$, each $x_i = G(y_i) \in \{0,1\}^m$ and the prover commits to n values. Notice that in the above verification function, we require V' to be independent of x, this captures the fact that x is only used as a commitment and will rule out some unwanted cases. All the real life identification schemes we will consider have this property.

Commit-and-open Identification scheme $\mathcal{IS} = (\mathrm{Keygen}, P, G, V', S_{ch})$

Initialization. $(pk, sk) \leftarrow \mathrm{Keygen}(1^{\lambda})$. The prover has (pk, sk) and the verifier pk.
Interaction.

1. P generates $(G(y_1), \ldots, G(y_n), y_1, \ldots, y_n) \leftarrow P_1(sk)$ and sends $x_1, \ldots, x_n = G(y_1), \ldots, G(y_n)$ to the verifier.
2. The verifier sends a random $c \in S_{Ch}$ that corresponds to a subset $I_c \subseteq \{1, \ldots, n\}$.
3. P sends $z = y_{I_c}$ to the verifier.

Verification. The verifier accepts iff. $(\forall i \in I_c, G(y_i) = x_i) \wedge V'(pk, c, z) = 1$.

In Appendix B, we show how to transform any identification scheme into a commit-and-open one. This transformation is very inefficient so we will not use it in our proofs but just wanted to point this possibility in case of interest.

3.6 Signature schemes

A signature scheme S consists of 3 algorithms $(\mathrm{S.\textsc{keygen}}, \mathrm{S.\textsc{sign}}, \mathrm{S.\textsc{verify}})$:

- $\mathrm{S.\textsc{keygen}}(1^\lambda) \to (pk, sk)$ is the generation of the public key pk and the secret key sk from the security parameter λ.
- $\mathrm{S.\textsc{sign}}(m, pk, sk) \to \sigma_m$: generates the signature σ_m of a message m from m, pk, sk.
- $\mathrm{S.\textsc{verify}}(m, \sigma, pk) \to \{0, 1\}$ verifies that σ is a valid signature of m using m, σ, pk. The output 1 corresponds to a valid signature.

Correctness. A signature scheme is correct iff. when we sample $(pk, sk) \leftarrow \mathrm{S.\textsc{keygen}}(1^\lambda)$, we have for each m

$$\mathrm{S.\textsc{verify}}(m, \mathrm{S.\textsc{sign}}(m, pk, sk), pk) = 1.$$

Security definitions We consider the standard EUF-CMA security for signature schemes. To define the advantage of an adversary $\mathscr{A}$, we consider the following interaction with a challenger:

Initialize. The challenger generates $(pk, sk) \leftarrow \mathrm{S.\textsc{keygen}}(1^\lambda)$ and sends pk to $\mathscr{A}$.
Query phase. $\mathscr{A}$ can perform sign queries by sending each time a message m to the challenger who generates $\sigma = \mathrm{S.\textsc{sign}}(m, pk, sk)$ and sends σ to $\mathscr{A}$. Let $m_1, \ldots, m_{q_S}$ the (not necessarily distinct) queries made by $\mathscr{A}$. The adversary can also make $q_{\mathcal{H}}$ queries to $\mathcal{H}$.
Output. $\mathscr{A}$ outputs a pair (m^*, σ^*). The advantage $Adv(\mathscr{A})$ for $\mathscr{A}$ is the quantity

$$\begin{aligned} QADV_{\mathcal{S}}^{\text{EUF-CMA}}(\mathscr{A}) = \Pr[\mathscr{A} \text{ outputs } (m^*, \sigma^*) \ st. \\ \mathrm{S.\textsc{verify}}(m^*, \sigma^*, pk) = 1 \wedge m^* \neq m_1, \ldots, m_{q_S}], \end{aligned}$$

where $m^* \neq m_1, \ldots, m_{q_S}$ means $\forall i, \ m^* \neq m_i$.

Definition 8. *Let $\mathcal{S} = (\mathrm{S.\textsc{keygen}}, \mathrm{S.\textsc{sign}}, \mathrm{S.\textsc{verify}})$ be a signature scheme. We define*

$$QADV_{\mathcal{S}}^{\text{EUF-CMA}}(t, q_{\mathcal{H}}, q_S) = \max_{\mathscr{A}} QADV_{\mathcal{S}}^{\text{EUF-CMA}}(\mathscr{A}).$$

where we maximize over an adversary running in time t, performing $q_{\mathcal{H}}$ hash queries and q_S sign queries.

We can directly construct a signature scheme from an identification scheme via the Fiat-Shamir transform. From an identification scheme $\mathcal{IS} = (\mathrm{Keygen}, P = (P_1, P_2), V, S_{ch})$, we define the following signature scheme
$\mathcal{S}_{\mathcal{IS}} = (\mathrm{S}_{\mathcal{IS}}.\textsc{keygen}, \mathrm{S}_{\mathcal{IS}}.\textsc{sign}, \mathrm{S}_{\mathcal{IS}}.\textsc{verify})$ that uses a random function $\mathcal{H}$:

- $\mathrm{S}_{\mathcal{IS}}.\textsc{keygen}(1^\lambda) = \mathrm{Keygen}(1^\lambda)$
- $\mathrm{S}_{\mathcal{IS}}.\textsc{sign}(m, pk, sk) : (x, St) \leftarrow P_1(pk), c \leftarrow \mathcal{H}(x, m), z \leftarrow P_2(sk, x, c, St)$, output $\sigma = (x, z)$.
- $\mathrm{S}_{\mathcal{IS}}.\textsc{verify}(m, \sigma = (x, z), pk) = V(pk, x, \mathcal{H}(x, m), z)$.

Proposition 2. *[KLS18] Let $\mathcal{IS}$ be an identification scheme which is ε-HVZK and has α bits of min-entropy. Let $S_{\mathcal{IS}}$ the corresponding signature scheme. We have*

$$QADV_{S_{\mathcal{IS}}}^{EUF\text{-}CMA}(t, q_{\mathcal{H}}, q_S) \leq QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]}(t + q_{\mathcal{H}} q_{\mathcal{S}}, q_{\mathcal{H}}) + q_{\mathcal{S}} 2^{-\alpha} + q_{\mathcal{S}}\varepsilon.$$

The min-entropy here is the min-entropy of the prover's first message when he in honest. All schemes we consider will have very large min-entropy so the $q_{\mathcal{S}} 2^{-\alpha}$ will be negligibly small. Notice that in [KLS18], they prove a more general result where the identification scheme $\mathcal{IS}$ allows some aborts. The above proposition shows that we only need to focus on the soundness of the Fiat-Shamir transform in order to build signature schemes, which is what we will do in the next section.

4 The Fiat-Shamir reduction for commit-and-open identification schemes

The goal of this section is to prove the following theorem

Theorem 1. *Let $\mathcal{IS}_\Gamma = (\mathrm{Keygen}, P, \Gamma, V', S_{ch})$ be a commit-and-open identification scheme where Γ is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$. Let $\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]$ its Fiat-Shamir transform that uses a function $\mathcal{H}$ modeled as a random function. For any quantum adversary $\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ running in time t making q_C queries to $\mathscr{O}_\Gamma$ and $q_{\mathcal{H}}$ queries to $\mathscr{O}_{\mathcal{H}}$, we have*

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) \leq q_{\mathcal{H}} \sqrt{\frac{1}{|S_{ch}|} + 4(QADV_{\mathcal{IS}_\Gamma}^{sp}(2t + 2O(l(q_C^2 + nq_C)), 2n))^{1/3}}.$$

The above theorem proves the quantum security (or more precisely the soundness against quantum attacks) of the Fiat-Shamir transform for commit-and-open identification schemes. Notice also that the reduction we present here is non-tight. While some amount of non-tightness seems necessary here, we suspect that there are steps which can be improved to make the theorem tighter.

If we want to replace the random function Γ with a random function G with small range r, we can use the following Lemma

Lemma 4. *Let $\mathcal{IS}_\Gamma = (\mathrm{Keygen}, P, \Gamma, V', S_{ch})$ be a commit-and-open identification scheme where Γ is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$. Let $\mathcal{IS}_G$ the same identification where Γ is replaced by G which is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$ with small range r. We have*

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]}(t, q_C, q_{\mathcal{H}}) \leq QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) + O(\frac{q_C^3 + n}{r}).$$

Notice that above, we introduced a new notation. $QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}})$ maximizes over all adversaries that run in time t, perform q_C queries to $\mathscr{O}_\Gamma$ and $q_{\mathcal{H}}$ queries to $O_{\mathcal{H}}$. We will use this notation throughout the section. Sometimes, we will only specify one number of queries. What it refers to should always be clear from context. For example, on the right hand side of the above, the $2n$ in $QADV_{\mathcal{IS}_\Gamma}^{sp}(2t + 2O(l(q_C^2 + nq_C)), 2n))$ corresponds to O_Γ queries, since there is no $\mathcal{H}$ in $\mathcal{IS}_\Gamma$.

We first present in high level the different steps of the proof and then prove each of those steps.

0. We start from a commit-and-open identification scheme $\mathcal{IS}_\Gamma = (\mathrm{Keygen}, P, \Gamma, V', S_{ch})$ where Γ is a modeled as a random function Γ from $\{0,1\}^l$ to $\{0,1\}^{3l}$.

1. Notice that because Γ is injective with overwhelming probability, $\mathcal{IS}_\Gamma$ has strict soundness. Using Proposition 1, we have
$$QADV_{IS_\Gamma}(t, q_C) \leq \frac{1}{|S_{ch}|} + 4\left(QADV^{sp}_{IS_\Gamma}(2t, 2q_C)\right)^{\frac{1}{3}} \tag{1}$$

2. Next, we prove that $\mathcal{IS}_\Gamma$ actually has *computational-statistical-soundness.*
$$QADV^{cs}_{\mathcal{IS}_\Gamma}(t, q_C) \leq QADV_{\mathcal{IS}_\Gamma}(t + O(l(q_C^2 + nq_C)) + n, n) \tag{2}$$

3. Then, we can prove that the Fiat-Shamir transform of $\mathcal{IS}_\Gamma$ (actually of any identification scheme with *computational-statistical-soundness*) has quantum soundness.
$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) \leq q_{\mathcal{H}}\sqrt{QADV^{cs}_{\mathcal{IS}_\Gamma}(t, q_C)} \tag{3}$$

4. Finally, we use the small range lower bound again to go back to $\mathcal{IS}_G$ and prove Lemma 4.
$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]}(t, q_C, q_{\mathcal{H}}) \leq QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) + O(\frac{q_C^3 + n}{r}). \tag{4}$$

Chaining all the inequalities from (3) to (1), we obtain our theorem:
$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) \leq q_{\mathcal{H}}\sqrt{\frac{1}{|S_{ch}|} + 4(QADV^{sp}_{\mathcal{IS}_\Gamma}(2t + 2O(l(q_C^2 + nq_C)), 2n))^{1/3}}.$$

or if we add from a function G with small range r (plugging in Lemma 4)
$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) \leq q_{\mathcal{H}}\sqrt{\frac{1}{|S_{ch}|} + 4(QADV^{sp}_{\mathcal{IS}_\Gamma}(2t + 2O(l(q_C^2 + nq_C)), 2n))^{1/3}} + O(\frac{q_C^3 + n}{r}).$$

4.1 Proving Theorem 1

Here, we will prove Equations 2,3,4 which will prove the theorem.

Proposition 3 (Equation 2)**.** *Let $\mathcal{IS}_\Gamma = (\mathrm{Keygen}, P, \Gamma, V', S_{ch})$ be a commit-and-open identification scheme where Γ is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$. We have*
$$QADV^{cs}_{\mathcal{IS}_\Gamma}(t, q_C) \leq QADV_{\mathcal{IS}_\Gamma}(t + O(l(q_C^2 + nq_C)) + n, n)$$

Proof. Let $\mathscr{A}^{\mathscr{O}_\Gamma} = (\mathscr{A}_1^{\mathscr{O}_\Gamma}, \mathscr{A}_2^{\mathscr{O}_\Gamma})$ an adversary running in time t and making q_C queries to $\mathscr{O}_\Gamma$ such that
$$QADV^{cs}_{\mathcal{IS}_\Gamma}(t, q_C) = QADV^{cs}_{\mathcal{IS}_\Gamma}(\mathscr{A}^{\mathscr{O}_\Gamma}).$$
For each x, let $p_x := \frac{1}{S_{ch}}|\{c \in S_{ch} : \exists z, V(x, c, z) = 1\}|$. We have by definition
$$QADV^{cs}_{\mathcal{IS}_\Gamma}(\mathscr{A}^{\mathscr{O}_\Gamma}) = \mathbb{E}_{\substack{(pk,sk)\leftarrow \mathrm{Keygen}(1^\lambda) \\ (x,St)\leftarrow \mathscr{A}_1^{\mathscr{O}_\Gamma}(pk)}}\left[p_x\right].$$

For each $y = y_1, \ldots, y_n \in \{0,1\}^{ln}$, we define $q_y = \frac{1}{S_{ch}}|\{c \in S_{ch} : V'(c, y_{I_c}) = 1\}$. where I_c is the challenge set associated to c. Notice that $p_{x_1,\ldots,x_n} \leq q_{\Gamma^{-1}(x_1),\ldots,\Gamma^{-1}(x_n)}$ and that this holds for any injective Γ since V' is independent of Γ. We first show the following lemma

Lemma 5. *For any quantum algorithm $\mathscr{B}$ running in time t, we have*

$$\mathbb{E}_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda)\\ y\leftarrow \mathscr{B}(pk)}}[q_y] \leq QADV_{\mathcal{IS}_\Gamma}(t+n,n).$$

Proof. Fix a quantum algorithm $\mathscr{B}$ running in time t. We construct the following quantum algorithm $\mathscr{B}'^{\mathscr{O}_\Gamma} = (\mathscr{B}_1'^{\mathscr{O}_\Gamma}, \mathscr{B}_2'^{\mathscr{O}_\Gamma})$:

- $\mathscr{B}_1'^{\mathscr{O}_\Gamma}(pk) : y = (y_1,\dots,y_n) \leftarrow \mathscr{B}(pk)$. return $(x, St) = ((\Gamma(y_1),\dots,\Gamma(y_n)),(y_1,\dots,y_n))$.
- $\mathscr{B}_2'^{\mathscr{O}_\Gamma}(pk,x,c,St)$: return $z = y_{I_c}$ where I_c is the challenge set associated to c.

With the way we constructed $\mathscr{B}'$, the commitment constraints $x_i = \Gamma(y_i)$ for $i \in I_c$ are always verified hence

$$\begin{aligned} QADV_{\mathcal{IS}_\Gamma}(\mathscr{B}') &= \Pr[V'(c, y_{I_c}) = 1 : (pk,sk) \leftarrow \text{Keygen}(1^\lambda), c \xleftarrow{\$} S_{ch}, y \leftarrow \mathscr{B}_1'^{\mathscr{O}_\Gamma}(pk)] \\ &= \mathbb{E}_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda)\\ y\leftarrow \mathscr{B}(pk)}}[q_y]\,. \end{aligned}$$

since $\mathscr{B}'$ has running time $t+n$ (recall we count the running time of Γ as 1), this concludes the proof of our lemma. □

We can now finish the proof of Proposition 3. We consider the adversary $\mathscr{A}^{\mathscr{O}}$ but we replace calls to Γ with calls to $\widetilde{\Gamma}$ where $\widetilde{\Gamma}$ is taken from a family of $2q_C$-wise functions from $\{0,1\}^l$ to $\{0,1\}^{3l}$. By Lemma 3, we can take for example random polynomials of degree $2q_C$ and a q_C quantum query algorithm will output indistinguishable outcomes *i.e.*

$$\forall x, \Pr[x : x \leftarrow \mathscr{A}_1^{\mathscr{O}_\Gamma}] = \Pr[x : x \leftarrow \mathscr{A}_1^{\mathscr{O}_{\widetilde{\Gamma}}}].$$

Moreover, $\widetilde{\Gamma}$ can be constructed *and inverted* in time $O(lq_C)$. We consider the following quantum algorithm $\mathscr{B}$: run $(x, St) = \mathscr{A}_1^{\mathscr{O}_{\widetilde{\Gamma}}}$ and output $y(x) = (\widetilde{\Gamma}^{-1}(x_1),\dots,\widetilde{\Gamma}^{-1}(x_n))$. Because $\widetilde{\Gamma}$ is injective with overwhelming probability, for any x, $p_x = q_{y(x)}$. From there, we conclude

$$\begin{aligned} QADV^{cs}_{\mathcal{IS}_\Gamma}(\mathscr{A}^{\mathscr{O}_\Gamma}) &= \mathbb{E}_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda)\\ (x,St)\leftarrow \mathscr{A}_1^{\mathscr{O}_\Gamma}(pk)}}[p_x] \\ &= \mathbb{E}_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda)\\ (x,St)\leftarrow \mathscr{A}_1^{\mathscr{O}_{\widetilde{\Gamma}}}(pk)}}[p_x] \\ &= \mathbb{E}_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda)\\ y\leftarrow \mathscr{B}(pk)}}[q_y] \\ &\leq QADV_{\mathcal{IS}_\Gamma}(t + O(l(q_C^2 + nq_C)) + n, n) \qquad \text{from } \textit{Lemma } 5 \end{aligned}$$

where in the last inequality, we use that the running time of $\mathscr{B}$ is $t + O(l(q_C^2 + nq_C))$. Indeed, we have to replace q_C calls to $\mathscr{O}_\Gamma$ with calls to $\mathscr{O}_{\widetilde{\Gamma}}$ which takes time $O(l(q_C^2))$ and invert $\widetilde{\Gamma}$ n times which takes time $O(l(nq_C))$. □

Proposition 4 (Equation 3)**.** *Let $\mathcal{IS}$ be any identification scheme. We have*

$$QADV_{\text{FS}^{\mathcal{H}}[\mathcal{IS}]}(t, q_{\mathcal{H}}) \leq q_{\mathcal{H}}\sqrt{QADV^{cs}_{\mathcal{IS}}(t)}$$

Proof. Let $\mathcal{IS}$ be an identification scheme and let $\mathscr{A}^{\mathcal{H}}$ a quantum algorithm that makes q queries to $\mathcal{H}$ and runs in total time t. Let also

$$U_\delta = \{x : |\{c \in S_{ch} : \exists z, V(x,c,z) = 1\}| \geq \delta |S_{ch}|\}.$$

Let $\varepsilon = QADV_{IS}^{cs}(t)$. By definition, we have that $\mathscr{A}^{\mathcal{H}}$ can find an element in $U_{k\varepsilon}$ with probability at most $\frac{1}{k}$ for any $k > 0$, or else we would have $QADV_{IS}^{cs}(t) > \varepsilon$. Let κ a parameter that will be fixed later. We have:

$$\begin{aligned} QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]}(\mathscr{A}^{\mathcal{H}}) &= \Pr[V(x,\mathcal{H}(x),z) = 1 : (x,z) \leftarrow \mathscr{A}^{\mathcal{H}}] \\ &= \Pr[V(x,\mathcal{H}(x),z) = 1 : x \in U_{\kappa\varepsilon}, (x,z) \leftarrow \mathscr{A}^{\mathcal{H}}] + \\ &\qquad \Pr[V(x,\mathcal{H}(x),z) = 1 : x \notin U_{\kappa\varepsilon}, (x,z) \leftarrow \mathscr{A}^{\mathcal{H}}] \\ &\leq \frac{1}{\kappa} + O(q^2 \kappa \varepsilon) \qquad \text{using } \textit{Lemma } 2 \\ &\leq O(q\sqrt{\varepsilon}) \qquad \text{by taking } \kappa = \frac{1}{q\sqrt{\varepsilon}} \end{aligned}$$

□

Finally, we prove the lemma that allows to replace the random function Γ with a random function G with small range.

Lemma 6 (Lemma 4 restated, Equation 4)**.** *Let $\mathcal{IS}_\Gamma = (\mathrm{Keygen}, P, \Gamma, V', S_{ch})$ be a commit-and-open identification scheme where Γ is modeled as a random function acting on $\{0,1\}^l$. Let $\mathcal{IS}_G$ the same identification where Γ is replaced by G which is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^l$ with small range r. We have*

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]}(t, q_C, q_{\mathcal{H}}) \leq QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(t, q_C, q_{\mathcal{H}}) + O(\frac{q_C^3 + n}{r}).$$

Proof. Let $\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ a quantum query algorithm that makes q_C queries to $\mathscr{O}_G$ and $q_{\mathcal{H}}$ queries to $\mathscr{O}_{\mathcal{H}}$. We consider the following algorithm $Z^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$:

- $Z^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}} : (pk, sk) \leftarrow \mathrm{Keygen}(1^\lambda), (x,z) \leftarrow \mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}(pk), c \leftarrow \mathcal{H}(x), b \leftarrow (\forall i \in I_c, G(y_i) = x_i) \wedge V'(pk, c, z) = 1$, return b

$Z^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ simply runs $\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ and outputs 1 if $\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ successfully cheated for $\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]$. From there, we clearly have

$$\Pr[b = 1 : b \leftarrow Z^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}] = QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]}(\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}).$$

Now, we consider the algorithm $Z^{\mathscr{O}_\Gamma, \mathscr{O}_{\mathcal{H}}}$ where each (quantum or classical) call to G is replaced by a call to Γ which gives

$$\Pr[b = 1 : b \leftarrow Z^{\mathscr{O}_\Gamma, \mathscr{O}_{\mathcal{H}}}] = QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(\mathscr{A}^{\mathscr{O}_\Gamma, \mathscr{O}_{\mathcal{H}}}).$$

Notice that $Z^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ perform at most $q_C + n$ calls to G: q_C when running $\mathscr{A}^{\mathscr{O}_G, \mathscr{O}_{\mathcal{H}}}$ and at most n when running the checks $(\forall i \in I_c, G(y_i) = x_i)$. From there, we can use Lemma 1 to have

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G]}(\mathscr{A}^{\mathscr{O}_\Gamma, \mathscr{O}_{\mathcal{H}}}) \leq QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_\Gamma]}(\mathscr{A}^{\mathscr{O}_\Gamma, \mathscr{O}_{\mathcal{H}}}) + O(\frac{q_C^3 + n}{r}).$$

□

5 Practical analysis

In this section, we will do a practical analysis of identification schemes which are a parallel repetition of commit-and-open identification schemes with challenge size 3. MQDSS,Picnic/Fish and the Stern signature are all based on an identification scheme which satisfies this property so the theorem below applies to these schemes.

Theorem 1 already proves their security but is highly non tight. We show here that by tailoring our previous techniques to this specific case, we get much tighter bounds than those of Theorem 1 and also much tighter than what is proven in [DFMS19] and [LZ19].

Theorem 2. *Let $\mathcal{IS}_G = (\text{Keygen}, P, G, V', \{0,1,2\})$ be a commit-and-open identification scheme where G is modeled as a random function from $\{0,1\}^l$ to $\{0,1\}^{2l}$. Let $\mathcal{IS}_G^{\otimes k}$ be its parallel repetition k times. We have*

$$QADV_{\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]}(t, q_G, q_{\mathcal{H}}) \leq 2kQADV_{\mathcal{IS}}(\widetilde{O}(t), q_G) + \frac{6k}{2^l} + O(q_{\mathcal{H}}^2 (2/3)^{2k}).$$

Proof. Let

$$S_{pk}^G = \Big\{(x_0, x_1, x_2) : \exists z_0, z_1, z_2 \ st. \ G(z_i) = x_i \text{ for } i \in \{0,1,2\} \ \wedge V'_{012}(pk, z_0, z_1, z_2) = 1\Big\}.$$

The proof goes informally as follows: consider an adversary $\mathscr{A}^{\mathcal{O}_G, \mathcal{O}_{\mathcal{H}}}$ that runs in time t and performs respectively q_G queries to G and $q_{\mathcal{H}}$ queries to $\mathcal{H}$. $\mathscr{A}$ wants to output (x, z) such that $V(x, \mathcal{H}(x), z) = 1$. We can write $x = (x_0^1, x_1^1, x_2^1, \ldots, x_0^k, x_1^k, x_2^k)$ and we distinguish 2 cases:

1. $\exists j, (x_0^j, x_1^j, x_2^j) \in S_{pk}^G$. We will show from here how to construct an adversary that breaks the soundness of $\mathcal{IS}$.
2. $\forall j, (x_0^j, x_1^j, x_2^j) \notin S_{pk}^G$. We show that this breaks the one-wayness of $\mathcal{H}$.

We define

$$U_{pk}^G = \Big\{x = (x_0^1, x_1^1, x_2^1, \ldots, x_0^k, x_1^k, x_2^k) : \exists j \in \{1, \ldots, k\} \ st. \ (x_0^j, x_1^j, x_2^j) \in S_{pk}^G\Big\}.$$

This first lemma will deal with the first case.

Lemma 7.

$$\Pr_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,z) \leftarrow \mathscr{A}(pk)}} \Big[x \in U_{pk}^G\Big] \leq 2kQADV_{\mathcal{IS}}(\widetilde{O}(t)) + \frac{6k}{2^{2l}}.$$

Proof. Proving this proposition is the purpose of Section 6. □

This next lemma will deal with the second case.

Lemma 8.

$$\Pr_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,z) \leftarrow \mathscr{A}(pk)}} \Big[V(pk, x, \mathcal{H}(x), z) = 1 \wedge x \notin U_{pk}^G\Big] \leq O(q_{\mathcal{H}}(2/3)^{2k}).$$

Proof. Let $x = (x_0^1, x_1^1, x_2^1, \ldots, x_0^k, x_1^k, x_2^k) \notin U_{pk}^G$ and for each $j, x^j = (x_0^j, x_1^j, x_2^j)$. $\forall j$ we have that $x^j \notin S_{pk}^G$ so let $\tilde{c}^j$ such that $\forall z^j$, $V(x^j, \tilde{c}^j, z^j) = 0$. Now fix a string $c = c^1, \ldots, c^k \in \{0,1,2\}^k$. If $\exists j, c^j = \tilde{c}^j$ then for all z, $V(x, c, z) = 0$. From there, let $V_{x,pk} = \{c : \exists z, V(pk, x, c, z) = 1\}$. From the previous discussion, we have $|V_{x,pk}| \leq 2^k$. Using Lemma 2, we can conclude that

$$\Pr_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda) \\ (x,z)\leftarrow \mathscr{A}(pk)}} \left[V(pk, x, \mathcal{H}(x), z) = 1 \land x \notin U_{pk}^G\right] \leq O(q_{\mathcal{H}}^2 \frac{|V_{x,pk}|}{3^k}) = O(q_{\mathcal{H}}^2 (2/3)^k).$$

□

We can now put everything together. We have

$$\begin{aligned} QADV_{\text{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]}(\mathscr{A}^{\mathcal{H}}) &= \Pr_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda) \\ (x,z)\leftarrow \mathscr{A}^{\mathscr{O}_{\mathcal{H}}}(pk)}} [V(pk, x, \mathcal{H}(x), z) = 1] \\ &\leq \Pr_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda) \\ (x,z)\leftarrow \mathscr{A}^{\mathscr{O}_{\mathcal{H}}}(pk)}} [V(x, \mathcal{H}(x), z) = 1 \land x \in U_{pk}^G] + \\ &\qquad \Pr_{\substack{(pk,sk)\leftarrow \text{Keygen}(1^\lambda) \\ (x,z)\leftarrow \mathscr{A}^{\mathscr{O}_{\mathcal{H}}}(pk)}} [V(pk, x, \mathcal{H}(x), z) = 1 \land x \notin U_{pk}^G]. \\ &\leq 2kQADV_{\mathcal{IS}}(\widetilde{O}(t)) + \frac{6k}{2^l} + O(q_{\mathcal{H}}^2 (2/3)^k). \end{aligned}$$

where we use respectively Lemma 7 and Lemma 8. □

We want to point that the term $O(q_{\mathcal{H}}^2 (2/3)^k)$ is necessary (up to some k factors) and correctly identifies an attack on $\text{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]$. Indeed, suppose there is an adversary for $\mathcal{IS}$ such that for any pair of challenges (c_1^*, c_2^*), it can produce x so that it successfully answers these challenges with $O(1)$ queries. This is definitely not rules out but soundness and is actually true for the Stern signature scheme.

By applying this strategy k times on $\text{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]$ each time choosing a random pair (c_1^*, c_2^*), this breaks $\text{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]$ with probability $(\frac{2}{3})^k$ performing $O(k)$ queries. By performing amplitude amplification, we get an attack on $\text{FS}^{\mathcal{H}}[\mathcal{IS}_G^{\otimes k}]$ that uses $O(k(\frac{3}{2})^{k/2})$.

6 Proving Lemma 7

In order to prove our lemma, we will need to dive in Zhandry's formulation of the QROM.

6.1 The Quantum Random Oracle Model, reminder

The Quantum Random Oracle Model (QROM) is a model where we model a certain function with a random function $\mathcal{H} : \{0,1\}^n \to \{0,1\}^m$. Since we are in the quantum setting, we have a black box access to $\mathcal{H}$ but also to the unitary $\mathscr{O}_{\mathcal{H}}(|x\rangle|y\rangle) = |x\rangle|\mathcal{H}(x) + y\rangle$.
Notation: When we write a state of n qubits in the Hadamard basis, we will write $|b\rangle^{\text{H}}$ which will correspond to the state $\frac{1}{\sqrt{2^n}} \sum_y (-1)^{b \cdot y} |y\rangle$.

6.2 Zhandry's oracles

One difficulty when dealing with the QROM is to deal with inherent randomness in the choice of $\mathcal{H}$. In [Zha18], Zhandry proposed another way at looking at a QROM, where the choice of $\mathcal{H}$ is encoded in a quantum register $\mathcal{D}$ (for database) which is a purification of the working register. This framework is quite elaborate and we present here some of the ideas and a small subset of the results presented in [Zha18]. We sometimes will go a little fast and we refer to [Zha18] for more details and explanations.

It is good for now to think of $\mathcal{D}$ as an internal register of the oracle function. This approach allows us to work on a single quantum state instead of dealing with a random choice of $\mathcal{H}$.

The standard oracle. The internal database of the oracle is initialized at

$$\begin{aligned} |D_0\rangle_{\mathrm{D}} &= \frac{1}{\sqrt{m2^n}} \sum_{\mathcal{H}\in\mathscr{F}_{\{0,1\}^n}^{\{0,1\}^m}} \bigotimes_{x\in\{0,1\}^n} |x\rangle|\mathcal{H}(x)\rangle \\ &= \frac{1}{\sqrt{m2^n}} \sum_{\mathcal{H}\in\mathscr{F}_{\{0,1\}^n}^{\{0,1\}^m}} |\mathcal{H}\rangle \quad \text{where} \quad |\mathcal{H}\rangle := \bigotimes_{x\in\{0,1\}^n} |x\rangle|\mathcal{H}(x)\rangle. \end{aligned}$$

which stores in a uniform superposition over all functions $\mathcal{H} : \{0,1\}^n \to \{0,1\}^m$ of all the input/output pairs $(x, \mathcal{H}(x))$. If $|D_0\rangle$ is measured in the computational basis, we obtain the full specification of a random function $\mathcal{H}$. The registers that contain the x are called the input registers and those that contain $\mathcal{H}(x)$ the output (or image) registers. We say that x (resp. y) is in the input (resp. output) registers is there exists an input (resp. output) register that contains x (resp. $f(x)$).

How do we query $\mathscr{O}_{\mathcal{H}}$ in this framework? By applying the unitary $\mathscr{O}^{St} : |x\rangle_{\mathrm{X}}|y\rangle_{\mathrm{Y}}|\mathcal{H}\rangle_{\mathrm{D}} \to |x\rangle_{\mathrm{X}}|y + \mathcal{H}(x)\rangle_{\mathrm{Y}}|\mathcal{H}\rangle_{\mathrm{D}}$ where X, Y are the input registers to the oracle and D is its internal quantum register containing the description of $\mathcal{H}$. Notice that this unitary only uses D as a classical control so even is it can entangle (X, Y) and D, measuring D in the computational basis will still yield a uniformly chosen $\mathcal{H} \in \mathscr{F}_m^n$. Using $\mathscr{O}^{St}$ with an internal register D is actually equivalent from an adversary's point to view to applying $\mathscr{O}_{\mathcal{H}}$ for a randomly chosen $\mathcal{H}$.

At any point, we say that (x, y) is in the database if when measuring the whole register D in the computational basis, we get an element $x, \mathcal{H}(x) = y$.

It seems that this is just a rewriting technique and that not much has been done. However, having access to this extra register D allows us to control the different possibilities of $\mathcal{H}$ after some queries done by an algorithm. It is shown in [Zha18] how to use this to (re)prove tight lower bounds for the both the search and the collision problem for random functions.

The compressed standard oracle. One problem with the standard oracle described above is that the database register D is of exponential size so we cannot efficiently manipulate it and hence cannot emulate efficiently the whole QRO. The idea will be to store a compressed version of this database. To see how, notice that $|D_0\rangle = \bigotimes_x |x\rangle|0\rangle^{\mathrm{H}}$. We now define some quantum that will allow us to define the compressed oracle.

Fix an integer t. Let $X = (x_1, \ldots, x_t)$ be an ordered tuple of different values in $\{0,1\}^n$ and let $R = (r_1, \ldots, r_t)$ be a tuple of values in $\{0,1\}^m$ each different from 0. We define the state $|\psi^S_{X,R}\rangle$

on register D as:

$$|\psi^S_{X,R}\rangle = \bigotimes_{x\in\{0,1\}^n} |x\rangle|y(x)\rangle^{\mathrm{H}}.$$

where $y(x) = r_i$ if $x = x_i$ and $y(x) = 0$ otherwise. This state corresponds to the standard database where we associate to each x_i the value r_i in the Hadamard basis and associate the uniform distribution *i.e.* the value 0 in the Hadamard basis to other values. Zhandry showed that after t queries, the database register is in $span\{|\psi^S_{X,R}\rangle\}_{X,R}$ where the set is over all the X, R defined above of size at most t. We also define the state $|\phi^C_{X,R}(q)\rangle$ as:

$$|\psi^C_{X,R}(q)\rangle_{\mathrm{D}} = \bigotimes_{i\in\{1,\ldots,t\}} |x_i\rangle|r_i\rangle^{\mathrm{H}} \bigotimes_{j\in\{t+1,q\}} |\bot\rangle|0\rangle.$$

with the convention that $\{q+1,\ldots,q\} = \emptyset$. This state corresponds to the compressed database after t queries we associate to each x_i the value r_i in the Hadamard basis. q here is the total number of queries. Since these states are of size at most q, they can be stored and manipulated efficiently.

There is a (not necessarily efficient) isometry E that goes from $|\psi^S_{X,R}\rangle$ to $|\psi^C_{X,R}(q)\rangle$ so this compression is lossless. The idea of the compressed database is to store states $|\psi^C_{X,R}(q)\rangle$ instead of $|(\psi^S_{X,R})\rangle$. Let D^C the register in which the $|\psi^C_{X,R}(q)\rangle$ lie. The compressed oracle is the unitary

$$\mathscr{O}^C : |x\rangle_{\mathrm{X}}|y\rangle_{\mathrm{Y}}E(|\mathcal{H}\rangle)_{\mathrm{D}^C} \rightarrow |x\rangle_{\mathrm{X}}|y\rangle_{\mathrm{Y}}E(|\mathcal{H}\rangle)_{\mathrm{D}^C}.$$

Proposition 5 ([Zha18]). *Consider any quantum algorithm $\mathscr{A}^{\mathscr{O}}$. We have*

$$\Pr[\mathscr{A}^{\mathscr{O}^{St}}(\cdot) = 1] - \Pr[\mathscr{A}^{\mathscr{O}^C}(\cdot) = 1] = 0.$$

In order words, applying $\mathscr{O}^C$ is indistinguishable to applying $\mathscr{O}^S$.

We can emulate $\mathscr{A}^{\mathscr{O}^C}$ efficiently by keeping track of the compressed database register D^C. Zhandry showed a procedure that achieves this in time $\widetilde{O}(q)$ where q is the total number of queries to $\mathscr{O}^C$.

6.3 Main technical lemma

The above proposition shows that the working registers of an algorithm using the standard oracle or the compressed oracle are indistinguishable. An interesting feature is to be able to recover information about $\mathcal{H}$ from D^C efficiently. Lemma 5 of [Zha18] shows that if an algorithm $\mathscr{A}^{\mathscr{O}^C}$ outputs (x, y) st. $\mathcal{H}(x) = y$ then it can retrieve efficiently (x, y) in D^C with very high probability (while it always appears in D).

Here, we extend this lemma when to the case where the algorithm outputs a value y but without knowing a preimage x. We fix here $n = l$ and $m = 3l$ for some integer l so we consider the functions $\mathcal{H} : \{0,1\}^l \rightarrow \{0,1\}^{3l}$.

Lemma 9. *Consider an quantum algorithm $\mathscr{A}^{\mathscr{O}}$ that does q queries to $\mathscr{O}$ acting on registers* XYZ *where* XY *are the query registers to $\mathscr{O}$. For each triplet $y = (y_0, y_1, y_2)$ where each $y_i \in \{0,1\}^{3l}$, we associate a set $Z_y \subseteq \{0,1\}^l \times \{0,1\}^l \times \{0,1\}^l$. Consider the 2 following scenarios:*

1. *Run an algorithm $\mathscr{A}^{\mathscr{O}_S}$ on registers* XYZD *where* D *is the database register. For each $y = (y_0, y_1, y_2)$, we define $p_y = \Pr[\exists(x_0, x_1, x_2) \in Z_y, (x_i, y_i) \in D \text{ for } i \in \{0,1,2\} | \mathscr{A}^{\mathscr{O}_S} \text{ outputs } y]$.*

2. *Run an algorithm $\mathscr{A}^{\mathscr{O}_C}$ on registers* XYZD^C *where* D^C *is the compressed database register. For each $y \in \{0,1\}^{3l}$, we define $p'_y = \Pr[\exists (x_0,x_1,x_2) \in Z_y, (x_i,y_i) \in D^C$ for $i \in \{0,1,2\} | \mathscr{A}^{\mathscr{O}_C}$ outputs $y]$.*

For any triplet $y = (y_0, y_1, y_2)$, we have

$$p_y \leq 2p'_y + \frac{6}{2^{2l}}.$$

Proof. Fix a value triplet $\widetilde{y} = (\widetilde{y_0}, \widetilde{y_1}, \widetilde{y_2})$ where each $\widetilde{y_i} \in \{0,1\}^{3l}$ and let $p = p_{\widetilde{y}}$ and $p' = p'_{\widetilde{y}}$. Consider the first scenario and let $|\Phi_1\rangle$ be the state in XYZD conditioned on the algorithm outputting $\widetilde{y}$. We can write

$$|\Phi_1\rangle = \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R}} \alpha_{u,X,R}|u\rangle_{\mathrm{XYZ}}|\psi^S_{X,R}\rangle_{\mathrm{D}} = \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R \\ Y=(y_1,\dots,y_t)}} (-1)^{\sum_{i=1}^t r_i \cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^S_{X,Y}\rangle$$

where for $X = (x_1, \dots, x_t)$ and $Y = (y_1, \dots, y_t)$, $|\eta^S_{X,Y}\rangle = \bigotimes_{x\in\{0,1\}^n} |x\rangle|y(x)\rangle$ where $|y(x)\rangle = |y_i\rangle$ if $x = x_i$ and $|y(x)\rangle = |0\rangle^{\mathrm{H}}$ otherwise. Let

$$Y_X = \{(y_1,\dots,y_t) : \exists i,j,k, \text{ st. } \widetilde{y_0} = y_i; \widetilde{y_1} = y_j; \widetilde{y_2} = y_k \text{ and } (x_i,x_j,x_k) \in Z_{\widetilde{y}}.$$

We write:

$$\begin{aligned}|\Phi_1\rangle = & \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R \\ Y=(y_1,\dots,y_t):\widetilde{y}\in Y_X}} (-1)^{\sum_{i=1}^t r_i \cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^S_{X,Y}\rangle \quad + \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R \\ Y=(y_1,\dots,y_t):\widetilde{y}\notin Y_X}} (-1)^{\sum_{i=1}^t r_i \cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^S_{X,Y}\rangle \\ = & |A\rangle + |B\rangle\end{aligned}$$

where $|A\rangle$ and $|B\rangle$ are respectively the first and second (unnormalized) pure state of the above sum. $\widetilde{y} \in Y_X = (y_1, \dots, y_t)$ means that there exists i,j,k, st. $\widetilde{y_0} = y_i$, $\widetilde{y_1} = y_j$ and $\widetilde{y_2} = y_k$ and that $(x_i, x_j, x_k) \in Z_{\widetilde{y}}$. We also use the convention $\sum_{i=1}^0 r_i \cdot y_i = 0$.

$|A\rangle$ and $|B\rangle$ can be written as projections of $|\Phi_1\rangle$ so $\||A\rangle\| \leq 1$ and $\||B\rangle\| \leq 1$. Let us define $\Pi_{\widetilde{y}}$ the projection on the output registers containing $\widetilde{y_0}, \widetilde{y_1}$ and $\widetilde{y_2}$. We have $p = \left\|\Pi_{\widetilde{y}}|\Phi_1\rangle\right\|^2$ and we have $\Pi_{\widetilde{y}}|A\rangle = |A\rangle$. Now look at $|B\rangle$. The only way to get $\widetilde{y}$ in the output register is to have $\widetilde{y_0}, \widetilde{y_1}$ or $\widetilde{y_2}$ it in the elements $|0\rangle^{\mathrm{H}}$. For each $\widetilde{y_i}$ and each $|0\rangle^{\mathrm{H}}$, this happens with probability $\leq \frac{1}{2^{3l}}$ and there are at most 2^l such elements and 3 possible $\widetilde{y_i}$. By a union bound we have $\left\|\Pi_{\widetilde{y}}|B\rangle\right\|^2 \leq \frac{3\cdot 2^l}{2^{3l}}$.

$$p = \left\|\Pi_{\widetilde{y}}|\Phi_1\rangle\right\|^2 = \left\||A\rangle + \Pi_{\widetilde{y}}|B\rangle\right\|^2.$$

Now consider the second scenario and let $|\Phi_2\rangle$ be the state in XYZD^C conditioned on the algorithm outputting $\widetilde{y}$. We can write

$$|\Phi_1\rangle = \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R}} \alpha_{u,X,R}|u\rangle_{\mathrm{XYZ}}|\psi^C_{X,R}\rangle_{\mathrm{D}^C} = \sum_{\substack{u \\ t\in\{0,\dots,q\} \\ X=(x_1,\dots,x_t) \\ R=(r_1,\dots,r_t):0\notin R \\ Y=(y_1,\dots,y_t)}} (-1)^{\sum_{i=1}^t r_i \cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^C_{X,Y}\rangle$$

where for $X = (x_1, \ldots, x_t)$ and $Y = (y_1, \ldots, y_t)$, $|\eta^C_{X,Y}\rangle = \bigotimes_{i\in\{1,\ldots,t\}} |x_i\rangle|y_i\rangle \bigotimes_{j\in\{t+1,q\}} |\bot\rangle|0\rangle$. We now write

$$\begin{aligned}|\Phi_2\rangle = &\sum_{\substack{u\\ t\in\{0,\ldots,q\}\\ X=(x_1,\ldots,x_t)\\ R=(r_1,\ldots,r_t):0\notin R\\ Y=(y_1,\ldots,y_t):\widetilde{y}\in Y_X}} (-1)^{\sum_{i=1}^t r_i\cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^C_{X,Y}\rangle \quad + \sum_{\substack{u\\ t\in\{0,\ldots,q\}\\ X=(x_1,\ldots,x_t)\\ R=(r_1,\ldots,r_t):0\notin R\\ Y=(y_1,\ldots,y_t):\widetilde{y}\notin Y_X}} (-1)^{\sum_{i=1}^t r_i\cdot y_i}\alpha_{u,X,R}|u\rangle|\eta^C_{X,Y}\rangle\\ &= |A'\rangle + |B'\rangle\end{aligned}$$

The probability p' to see a good $\widetilde{y}$ in the compressed database outputs is $\left\|\Pi_{\widetilde{y}}|A'\rangle\right\|^2 = \||A'\rangle\|^2 = \||A\rangle\|^2$. From there we can conclude

$$|\sqrt{p} - \sqrt{p'}| = \Big|\, \||A\rangle + \Pi_{\widetilde{y}}|B\rangle\| - \||A\rangle\| \,\Big| \le \left\|\Pi_{\widetilde{y}}|B\rangle\right\| \le \sqrt{\frac{3\cdot 2^l}{2^{3l}}}.$$

This gives immediately

$$p \le \left(\sqrt{p'} + \sqrt{\frac{3\cdot 2^l}{2^{3l}}}\right)^2 \le 2p' + \frac{6}{2^{2l}}.$$

□

Remark: This lemma and proof are quite similar to Lemma 5 proven in [Zha18]. However in their case, the algorithm also outputs a preimage of y whereas we don't have this possibility so it complicates slightly the analysis.

6.4 Proof of Lemma 7

Definition 9. *Consider an algorithm $\mathscr{A}^{\mathscr{O}^S}$ acting on registers $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ where X, Y are query registers and Z is an extra register. Let also D the internal database register of $\mathscr{O}^S$. Let*

$$ADV_1(\mathscr{A}^{\mathscr{O}^{St}}) = \Pr_{\substack{(pk,sk)\leftarrow \mathrm{Keygen}(1^\lambda)\\ x_0,x_1,x_2\leftarrow \mathscr{A}^{O^{St}}(pk)}}\Big[\exists z_0, z_1, z_2 : (z_i, x_i) \in \mathrm{D}_{x_0,x_1,x_2} \text{ for } i \in \{0,1,2\} \ \wedge V'_{012}(pk, z_0, z_1, z_2) = 1\Big].$$

where D_{x_0,x_1,x_2} is the database register D of the oracle $\mathscr{O}^S$ measured in the computational basis, conditioned on the output x_0, x_1, x_2 of $\mathscr{A}^{\mathscr{O}^S}$.

Definition 10. *Consider an algorithm $\mathscr{A}^{\mathscr{O}^C}$ acting on registers $\mathrm{X}, \mathrm{Y}, \mathrm{Z}, \mathrm{D}^C$ where X, Y are query registers and Z is an extra working register and D^c is the compressed register. Let*

$$ADV_2(\mathscr{A}^{\mathscr{O}^C}) = \Pr_{\substack{(pk,sk)\leftarrow \mathrm{Keygen}(1^\lambda)\\ x_0,x_1,x_2\leftarrow \mathscr{A}^{O^C}(pk)}}\Big[\exists z_0, z_1, z_2 : (z_i, x_i) \in \mathrm{D}^C_{x_0,x_1,x_2} \text{ for } i \in \{0,1,2\} \ \wedge V'_{012}(pk, z_0, z_1, z_2) = 1\Big].$$

where $\mathrm{D}^C_{x_0,x_1,x_2}$ is the database register D^C measured in the computational basis, conditioned on the output x_0, x_1, x_2 of $\mathscr{A}^{\mathscr{O}^C}$.

Proposition 6. *$ADV_1(A^{\mathscr{O}^{St}}) \le 2ADV_2(A^{\mathscr{O}^C}) + \frac{6}{2^{2l}}$.*

Proof. Fix an algorithm $\mathscr{A}^{\mathscr{O}^{St}}$ that outputs some values x_0, x_1, x_2. We can apply directly Lemma 9 by taking $y = (x_0, x_1, x_2)$ and $Z_y = \{(z_0, z_1, z_2) : V'_{012}(pk, z_0, z_1, z_2) = 1\}$ to get the desired result. □

Proof of Lemma 7. Consider an adversary $\mathscr{A}^{\mathscr{O}_{\mathcal{H}}}$ running in time t and performing q queries to $\mathscr{O}_{\mathcal{H}}$. Consider the algorithm $\mathscr{B}^{\mathscr{O}^{\mathcal{H}}}$ that does the following: run $\mathscr{A}^{\mathscr{O}}$ and let (x, z) be the output with $x = (x_0^1, x_1^1, x_2^1, \dots, x_0^k, x_1^k, x_2^k)$. Output $x_0^j, x_1^j x_2^j$ for a random j. We have

$$\Pr_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,z) \leftarrow \mathscr{A}(pk)}} \left[x \in U_{pk}^G \right] \leq kADV_1(\mathscr{B}^{\mathscr{O}^{St}}) \leq 2kADV_2(\mathscr{B}^{\mathscr{O}^C}) + \frac{6k}{2^{2l}}.$$

But notice that $ADV_2(\mathscr{B}^{\mathscr{O}^C})$ can be related to $QADV_{\mathcal{IS}}$. Indeed, an algorithm that can outputz_0, z_1, z_2 such that $V'_{012}(pk, z_0, z_1, z_2) = 1$ can break the identification scheme. Since $B^{\mathscr{O}^C}$ runs in time $\widetilde{O}(t)$, we have $ADV_2(\mathscr{B}^{\mathscr{O}^C}) \leq QADV_{\mathcal{IS}}(\widetilde{O}(t))$ which allows us to conclude. □

7 Compressing the commitments

Theorems 1 and 2 show the security of the Fiat-Shamir transform but the resulting non-interactive scheme (hence the resulting signature scheme) still requires a random permutation as a hash function. Signature schemes based on commit-and-open identification already have a quite high signature length (*i.e.*communication cost) and using a random function from $\{0,1\}^l$ to $\{0,1\}^{3l}$ can significantly increase the signature length.

In Lemma 4, we showed how to replace this random function by a random function with small range but this doesn't reduce the commitment size since this range is unknown. To overcome this problem, we use random sponges which will allow to reduce the commitment size. We present here only an informal discussion.

We only consider sponge functions with the same number of input and output bits. We can see a sponge function as 2 functions, an absorb and a squeeze function (S_{abs}, S_{sq}) such that $S_{abs} : \{0,1\}^l \to \{0,1\}^{r+c}$ and $S_{sq} : \{0,1\}^{r+c} \to \{0,1\}^l$ where l is the number of input/output bits, r is the rate and c is the capacity of the sponge. The whole sponge function S will then be $S = S_{sq} \circ S_{abs}$ (this includes eventual padding operations) where f is underlying function of.

In [CHS19], it is shown that random sponges are indistinguishable from random functions.

Proposition 7. *For any quantum query algorithm $\mathscr{A}^{\mathscr{O}}$, we have*

$$\Pr_{\mathcal{H} \leftarrow \mathscr{F}_{\{0,1\}^l}^{\{0,1\}^{3l}}} [\mathscr{A}^{\mathscr{O}_{\mathcal{H}}}(\cdot) = 1] - \Pr_{S \leftarrow Sp} [\mathscr{A}^{\mathscr{O}_S}(\cdot) = 1] \leq \frac{8\pi^2 (q + \frac{2l}{r})^3}{3 \cdot 2^c}.$$

where Sp is the set of random sponges (for a randomly chosen internal function). This shows that we require a bit more than $2^{c/3}$ queries to distinguish both settings.

We now show how to modify the above Fiat-Shamired identification schemes to reduce the commitment costs.

$\mathrm{FS}^{\mathcal{H}}[\mathcal{IS}]$

- P constructs $c_0 = S(z_0), c_1 = S(z_1), c_2 = S(z_2)$ as in $\mathcal{IS}$. Let also $\alpha_i = S_{abs}(z_i)$. He computes $c = \mathcal{H}(c_0, c_1, c_2)$. Let I_c the corresponding subset of values he has to open. He sends to the verifier $z = \{z_i\}_{i \in I_c}$ and $\{\alpha_i\}_{i \notin I_c}$ (instead of $\{c_i\}_{i \notin I_c}$).
- The verifier computes $\{c_i = S(z_i)\}_{i \in I_c}$ and $\{c_i = S_{sq}(\alpha_i)\}_{i \notin I_c}$. Let $x = \{c_i\}_{i \in \{0,1,2\}}$ computes $c = \mathcal{H}(x)$ and checks whether $V(x, c, z) = 1$.

With this transformation, each commitment has size $c + r$ which can be made small. For example, we can take $c = 400$, $r = 64$ (to not have a too big number of rounds) and from the above proposition, we would have more than 128 bits of quantum security, for reasonable values of l.

8 Wrapping up and conclusion

In this paper, we showed techniques for proving the quantum security of the Fiat-Shamir transform, completing the new generic results of [DFMS19, LZ19]. Theorem 1 applies to commit-and-open identification schemes with a very strong commitment schemes. It has a theoretical appeal as it uses different techniques and doesn't require reprogramming of the quantum oracle.

From a practical perspective, Picnic, Stern and the non-optimized variant of MQDSS are signature schemes based on the Fiat-Shamir transform of identification schemes which are the parallel repetition of commit-and-open identification schemes with challenge size 3. Their security directly follows from our Theorem 2 and Proposition 2 and was not previously explicitly stated. We showed in particular how to deal with triple special soundness without losing tightness.

More generally, for any such scheme which uses some kind of commitment scheme, we can first require very strong properties for those schemes and then replace them by efficient quantum sponges, using their quantum security [CHS19].

References

[ABB+19] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qtesla. Cryptology ePrint Archive, Report 2019/085, 2019. https://eprint.iacr.org/2019/085.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems: The hardness of quantum rewinding. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 474–483, Washington, DC, USA, 2014. IEEE Computer Society.

[CDG+17] Melissa Chase, David Derler, Steven Goldfeder, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, and Greg 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, pages 1825–1842, New York, NY, USA, 2017. ACM.

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

[CHS19] Jan Czajkowski, Andreas Hülsing, and Christian Schaffner. Quantum indistinguishability of random sponges. Cryptology ePrint Archive, Report 2019/069, 2019. `https://eprint.iacr.org/2019/069`.

[CL17] André Chailloux and Anthony Leverrier. Relativistic (or 2-prover 1-round) zero-knowledge protocol for \mathsf NP secure against quantum adversaries. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, pages 369–396, 2017.

[CSST11] Claude Crépeau, Louis Salvail, Jean-Raymond Simard, and Alain Tapp. Two provers in isolation. In *Advances in Cryptology–ASIACRYPT 2011*, pages 407–430. Springer, 2011.

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

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. Cryptology ePrint Archive, Report 2019/190, 2019. `https://eprint.iacr.org/2019/190`.

[DKL+17] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stéhlé. Crystals-dilithium, algorithm specifications and supporting documentation, 2017. https://pq-crystals.org/dilithium/data/dilithium-specification.pdf.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

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

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. Cryptology ePrint Archive, Report 2019/262, 2019. `https://eprint.iacr.org/2019/262`.

[Nis17] Nist. Post-quantum cryptography standardization, 2017. https://csrc.nist.gov/projects/post-quantum-cryptography.

[PS96] David Pointcheval and Jacques Stern. Security proofs for signature schemes. In Ueli Maurer, editor, *Advances in Cryptology — EUROCRYPT '96*, pages 387–398, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 706–723, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 13–21, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 135–152, 2012.

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

[Unr17] Dominique Unruh. Post-quantum security of fiat-shamir. In *ASIACRYPT (1)*, pages 65–95. Springer, 2017.

[Zha12] Mark Zhandry. How to construct quantum random functions. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 679–687, Washington, DC, USA, 2012. IEEE Computer Society.

[Zha15] Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Info. Comput.*, 15(7-8):557–567, May 2015.

[Zha18] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. *IACR Cryptology ePrint Archive*, 2018:276, 2018.

A Relating soundness and special soundness

In this section, we prove the following proposition.

Proposition 8. *Let $\mathcal{IS} = (\mathrm{Keygen}, P, V, S_{ch})$ be an identification scheme with strict soundness. For any t,*

$$QADV_{\mathcal{IS}}(t) \leq \frac{1}{|S_{ch}|} + 4\left(QADV^{sp}_{\mathcal{IS}}(2t)\right)^{1/3}.$$

We restate a (slightly modified) version of Theorem 3 of [CL17].

Proposition 9 (Theorem 3, [CL17]). *Consider n projectors $P_1, \ldots, P_n$ and a quantum mixed state σ. Let $V := \frac{1}{n}\sum_{i=1}^{n} tr(Pi\sigma)$ and let*

$$E := \frac{1}{n(n-1)} \sum_{i,j \neq i} tr(P_i P_j \sigma P_j P_i).$$

Then it holds that $V \leq \frac{1}{n} + 4E^{1/3}$.

Proof of Proposition 1. Fix an identification scheme $\mathcal{IS}$ and consider a cheating $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ running in time t. Now consider the following algorithm $\mathscr{B}(pk)$: run $(x, St) \leftarrow \mathscr{A}_1(pk)$, choose random $c, c' \in S_{ch}$ with $c \neq c'$. Run $z \leftarrow \mathscr{A}_2(pk, x, c, St)$ and $z' \leftarrow \mathscr{A}_2(pk, x, c', St)$. Output (x, c, z, c', z'). For a fixed x, c, $\mathscr{A}_2(pk, x, c', St)$ can be modeled as a quantum projective measurement $M^c = \{M_1^c, \ldots, M_\nu^c\}$ where ν is the output of the measurement. Let also $\sigma_{x,St}$ be the state conditioned on $\mathscr{A}_1$ outputting x, St.

For each c, from strict soundness, there is at most 1 value ν such that $V(x, c, \nu) = 1$. We define $P_c = M_\nu^c$ such a ν exists and $P_c = \mathbf{0}$ otherwise. Let $V_{x,St} = \frac{1}{|S_{ch}|}\sum_c tr(P_c \sigma_{x,St})$ that corresponds to $\mathscr{A}_2$ outputting a valid $z = \nu$ given that $\mathscr{A}_1$ outputs x, St.

Let also $E_{x,St} = \frac{1}{|S_{ch}|(|S_{ch}|-1)}\sum_{c,c' \neq c} tr(P_{c'} P_c \sigma_{x,St} P_c P_{c'})$. which corresponds to the probability that $\mathscr{B}$ outputs a valid (x, c, z, c', z') given that his run of $\mathscr{A}_1$ outputs (x, St).

By definition, we have

$$QADV_{\mathcal{IS}}(\mathscr{A}) = \mathbb{E}_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,St) \leftarrow \mathscr{A}_1(pk)}} \left[V_{x,St}\right].$$

and

$$QADV_{\mathcal{IS}}^{sp}(\mathscr{B}) = \mathbb{E}_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,St) \leftarrow \mathscr{A}_1(pk)}} \left[E_{x,St}\right].$$

Using Proposition 9, we have $V_{x,St} \leq \frac{1}{|S_{ch}|} + 4E_{x,St}^{1/3}$. From there, we can conclude

$$\begin{aligned}
QADV_{\mathcal{IS}}(\mathscr{A}) &= \mathbb{E}_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,St) \leftarrow \mathscr{A}_1(pk)}} \left[V_{x,St}\right] \\
&\leq \mathbb{E}_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,St) \leftarrow \mathscr{A}_1(pk)}} \left[\frac{1}{|S_{ch}|} + 4E_{x,St}^{1/3}\right] \\
&\leq \frac{1}{|S_{ch}|} + 4\left(\mathbb{E}_{\substack{(pk,sk) \leftarrow \text{Keygen}(1^\lambda) \\ (x,St) \leftarrow \mathscr{A}_1(pk)}} [E_{x,St}]\right)^{1/3} \qquad \text{by concavity of } x \rightarrow x^{1/3} \\
&\leq \frac{1}{|S_{ch}|} + 4\left(QADV_{\mathcal{IS}}^{sp}(\mathscr{B})\right)^{1/3}.
\end{aligned}$$

Finally, to conclude, notice that $\mathscr{B}$ runs $\mathscr{A}_1$ once and $\mathscr{A}_2$ twice so takes at most twice the time as $\mathscr{A}$. □

Remark: There are other similar bounds that can be derived, see for example [CSST11] and [Unr12].

B Any identification scheme can be transformed into a commit-and-open scheme

We present the transformation below as a remark, to show that any identification scheme can be transformed into a commit-and-open identification scheme, with a loss in efficieny. Consider any public coin identification scheme $\mathcal{IS} = (\text{Keygen}, P, V, S_{Ch})$. The interaction between the prover and the verifier is the following.

Identification scheme $\mathcal{IS}$

Initialization. $(p_k, s_k) \leftarrow \text{Keygen}(1^\lambda)$. P has (p_k, s_k) and V has p_k.
Interaction.

1. P generates $(x, St) \leftarrow P_1(s_k)$ and sends x to the verifier.
2. The verifier sends a random $c \in S_{Ch}$.
3. P generates $z \leftarrow P_2(s_k, x, c, St)$ and sends z to the verifier.

Verification. The verifier accepts iff. $V(x, c, z) = 1$.

We can transform this scheme into a commit-and-open scheme. The idea is to commit to x and to all the possibles z's depending on the challenge. The commitment is modeled with a random function $G : \{0,1\}^l \rightarrow \{0,1\}^{3l}$.

Identification scheme $\mathcal{IS}_{cao}$

Initialization. $(p_k, s_k) \leftarrow \text{Keygen}(1^\lambda)$. P has (p_k, s_k) and V has p_k.
Interaction.

1. P generates $(x, St) \leftarrow P_1(s_k)$, as well as a string $z_c \leftarrow P_2(s_k, x, c, St)$ for each $c \in S_{ch}$. He sends $(G(x), G(z_1), \dots, G(z_{|S_{ch}|}))$ to the verifier.
2. The verifier sends a random $c \in S_{Ch}$.
3. P sends x and z_c to the verifier.

Verification. The verifier accepts iff. $V(x, c, z_c) = 1$.

The above construction is very reminiscent of the Unruh transform [Unr15]. We show very informally that $\mathcal{IS}_{cao}$ retains completeness, soundness and the honest-verifier zero-knowledge property of the original scheme $\mathcal{IS}$. Completeness follows easily from the completeness of $\mathcal{IS}$. Soundness holds because G is injective with overwhelming probability (so the commitment is perfectly binding) and by taking l large enough, the verifier cannot distinguish the first message from random elements in $\{0,1\}^{3l}$ so the honest-verifier zero-knowledge is preserved.