# Post-quantum Security of Fiat-Shamir Signatures

**MSc Thesis** *(Afstudeerscriptie)*

written by

**Jelle Wijnand Don**
**11132809**
(born January 30, 1990 in Amsterdam, The Netherlands)

under the supervision of **Christian Schaffner** and **Christian Majenz**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

at the *Universiteit van Amsterdam*.

| **Date of the public defense:** | **Members of the Thesis Committee:** |
|---|---|
| *July 13, 2018* | Prof Ronald de Wolf |
| | Dr Serge Fehr |
| | Dr Stacey Jefferey |
| | Dr Christian Schaffner |
| | Dr Christian Majenz |

Institute for Logic, Language and Computation

**Abstract**

We propose a proof method that aims to show that the Fiat-Shamir proof system is SP-extractable (statement preserving) in the quantum random oracle model, if the underlying sigma-protocol has perfect unique responses. We furthermore prove that a signature scheme which is based on a Fiat-Shamir proof system that is SP-extractable, is existentially unforgeable for a quantum adversary.

# Acknowledgements

I would first and foremost like to thank my brother Matthijs, who has been supportive of my academic ambitions since before I had any. He is a true friend, who guides me in more ways than he realizes.

All the way from allowing me to travel abroad with school friends at 16, to letting me find my very indeterminate path through my university studies, my parents have always given me the greatest possible freedom in discovering all aspects of life by first-hand experience. I would never have found the wonderful corner of academia that I am in now, if it was not for this free-spirited attitude towards parenting. Amazingly, the abundance of freedom has not made me break loose from my surroundings. On the contrary, the family-warmth that has surrounded me since childhood has taught me the value of cherishing long-term relations, something that will undoubtedly help me throughout my personal and professional life. For this and a thousand other things, I thank them wholeheartedly.

Of course I feel a huge gratitude towards my supervisors. Christian Schaffner has mentored my submersion into the world of cryptography since almost one and a half years now. He got me hooked. His almost extreme appreciation of the value of high-level teaching helped me get to know the entire field in a matter of months. Then, when I was ready for the next step, he bullseye-picked the perfect project for my master thesis, which I have enjoyed beyond anything.

Christian Majenz has contributed majorly to this thesis, by having the extreme patience of again and again deciphering my often shockingly unexplained brawls of formulas that I call proofs, and pinpointing exactly where they were wrong. In the end he even postponed the birth of his first child so that he would still be around to proofread my thesis.

Doing research with the Chrisses is simply fun. Sometimes I got the feeling that they were playing a game of good cop - bad cop with me. Every time I presented a new wild plan to them, one particular supervisor expressed his (healthy, scientific, important!) skepticism, and the other displayed a more youthful enthusiasm, jumping up to the board to find some grain of mathematical truth in my naive visions (you'd be surprised what he could find in them). I will leave it to the reader to fill in the names.

I would also like to thank, with all my heart:

- My dear grandma, who owns the world record for thinking about her grandsons, in minutes per hour.
- The staff of the CWI library, for letting me dry my mud-sunken socks on the heating in their office. Bicky always did her best to make me feel at home.
- Yfke, Jan, Jeroen and the Chrisses for making me feel what it is like to be part of an academic (superstar) team.
- Krijn, Dieks, Mara, Pablo, Joris, Joanna, Wijnand, Stella, Remco and Michiel, who all in their own way inspired my work on this thesis. The way they have been lifting spirits, each of them must be a professional bodybuilder by now.

Finally, I thank the members of my thesis committee, Prof. Ronald de Wolf, Dr. Serge Fehr, Dr. Stacey Jeffery, Dr. Christian Schaffner and Dr. Christian Majenz, for their effort in closing the final chapter of my university education.

# Contents

# Introduction

Over the last few years, the development of quantum computers has taken a leap. It now seems realistic that the technical difficulties involved may be overcome within the next few decades. While this brings about many exciting prospects for such application areas as materials science, pharmacological research and brain modeling, and even cryptography could benefit in some respects, the outlook for this latter field is not exclusively good. A practical-scale quantum computer could break almost all of the cryptographic protocols that are in use today.

To address the coming quantum threat, the United States National Institute of Standards and Technology (NIST) has recently issued a competition for a new type of *digital signature scheme*, that should be *post-quantum secure*. Digital signatures form an important part of our current-day digital infrastructure. They allow for secure digital communication, including financial transactions, private conversations and software distribution. Needless to say, the security of these digital signature schemes is of the utmost importance to the normal functioning of modern society.

Perhaps less obvious is the significance of their efficiency. With billions of secure transactions being carried out across the globe *every day*, a reduction in signature size from 3kb (kilobyte) to 1kb could lead to huge energy savings. Whether the gain is measured in tonnes of euros or tonnes of carbon dioxide – depending on your personal outlook on life – the importance of every last millibit of bytes is clear.

The NIST competition has attracted contributions from research teams around the world. A significant portion of them makes use of a generic technical tool called the 'Fiat-Shamir transformation'. The transformation takes a multiple round, interactive identification protocol, and turns it into a non-interactive digital signature scheme. The Fiat-Shamir transformation is famous for combining security with extreme efficiency, leading to the most desirable kind of signature schemes.

Unfortunately, in 2011 a paper was published [BDF+11] that warned against a particular vulnerability of Fiat-Shamir type signatures in the post-quantum era. The authors pointed out that the classical proof of security, which is used to mathematically demonstrate the extreme unlikeliness that anyone with a *classical computer* is able to falsify a Fiat-Shamir signature, does not go through when a quantum computer is brought into the picture.

While the 2011 paper left open the possibility that a new proof could be found – showing that Fiat-Shamir signatures are indeed unforgeable *even for a quantum computer* – we cannot rely on good faith alone to protect us in the coming quantum age. Therefore, the NIST-submissions that use Fiat-Shamir all had to build in a kind of extra security measure, which significantly degrades their efficiency. In some cases the increase in data-usage is approximately a factor of 3 [KLS18].

Many researchers find this situation unsatisfactory. It is widely believed that the Fiat-Shamir transformation *is* in fact post-quantum secure, so that it should be possible to find a proof confirming this intuition and do away with the extra security measures. However, results of the last few years have rather pointed in the other direction. In 2014, a paper titled 'The hardness of quantum rewinding' by Ambainis et al. [ARU14] commented on the difficulty of transporting classical proof techniques related to Fiat-Shamir to the quantum setting, and actually found a quantum break (under suitable assumptions) of a range of schemes known to be classically secure. The paper [DFG13] went a step further, claiming to have proven the impossibility of a direct proof of quantum security for Fiat-Shamir. (We argue against their conclusion in Section 3.2.) More recently, [Unr17] made an extensive study of the problem and took some steps in the right direction. However, a crucial part of their analysis consisted of proving the *extractability* of the related Fiat-Shamir *proof system*, for which they could give no solution.

### Our contribution

In spite of the recent negative results, and in direct contradiction to the claim of [DFG13], we give a method to prove the extractability of the Fiat-Shamir proof system, filling in the gap from [Unr17]. However, shortly after the submission of this thesis, it was discovered that our proof method still contains an unproven assumption, and therefore cannot be considered a full proof yet. Proving the assumption is non-trivial, nevertheless it is expected that a revised and complete version of the proof will appear on the arXiv soon. For more details about the unproven assumption, see page 31.

We use a slightly weaker definition of extractability than [Unr17] did, but we also prove that their analysis of Fiat-Shamir signatures still goes through under the weaker definition, leading to the conclusion that Fiat-Shamir signatures are indeed post-quantum secure.

It should be noted that we use a technique from [Unr12], which puts a restriction on the class of signature schemes to which our result applies. We require that the underlying *sigma-protocol* has a property called 'perfect unique responses'. Not all NIST-submissions satisfy this condition. In particular, the popular *lattice-based* schemes do not fall in the category that we prove post-quantum secure. The equally promising submissions based on *supersingular isogeny cryptography* do satisfy the condition. However, we strongly believe that in the near future our result may be extended to also include the lattice-based schemes.

To prove our result, we present a new technique for using a quantum adversary in a security reduction. Normally, when the (intermediate) output of a quantum adversary is measured, its internal state *collapses*, which means that we cannot continue to use the adversary in our reduction — because in general the internal state will be disturbed so much that we cannot know what the adversary will do from the measurement point on. In this thesis we develop new tools that allow us to predict the behavior of the adversary after an intermediate measurement. We use them to show that in the Fiat-Shamir case, quantum rewinding is possible. We furthermore introduce a new 'quantum forking lemma' – in approximate analogy to the classical forking lemma – which completes the proof of the quantum extractability of the Fiat-Shamir proof system.

# 1 Background

In this section we formally introduce the concepts that feature in this thesis. We present only a small subset of the much broader fields of quantum information/computation and cryptography. For a more comprehensive overview of both fields, see [NC11] and [KL14] respectively.

## 1.1 Quantum computing

Quantum mechanics is a theory that predicts the behavior of physical systems at the subatomic scale. Since any physical system can in principle be used to represent and manipulate information, a natural question, which researchers started asking in the seventies of the twentieth century, is whether quantum systems are suitable media for information processing tasks.

As it turns out, quantum information is fundamentally different from classical information, in ways that allow quantum devices to perform qualitatively different operations than their 'classical' counterparts. Currently, a few specific problem classes have been discovered for which a quantum algorithm outperforms the best known classical algorithm.

### Quantum states

The fundamental unit of classical information is a bit; its value is either one or zero, as such it discriminates between two possibilities. In quantum information, the fundamental unit is a quantum bit or *qubit*. The quantum bit has a continuous value; its state can be described by a unit vector in a two-dimensional complex Hilbert space. In general, a quantum system is represented by a complex Hilbert space of some dimension $d$, and the state of the system is described by a unit vector in this space. For most quantum computing applications, $d$ is finite.

Complex Hilbert spaces have a complex inner product that satisfies (in Dirac notation)

1. $\langle\phi|\psi\rangle = \overline{\langle\psi|\phi\rangle}$ (Complex conjugate)
2. $\langle\phi|\psi\rangle = \lambda_1\langle\phi_1|\psi\rangle + \lambda_2\langle\phi_2|\psi\rangle$ where $\lambda_1, \lambda_2 \in \mathbb{C}$ and $\langle\phi| = \lambda_1\langle\phi_1| + \lambda_2\langle\phi_2|$ (Linear in the first argument)
3. $\langle\phi|\phi\rangle \geq 0$ (Positive definite).

The inner product induces a norm on vectors in the Hilbert space:

$$\||\phi\rangle\|_2 := \sqrt{\langle\phi|\phi\rangle} \qquad \text{(2-norm)}$$

A useful lemma concerning the inner product is the *Cauchy-Schwarz inequality*, which says that

$$|\langle\phi|\psi\rangle| \leq \||\phi\rangle\|_2 \cdot \||\psi\rangle\|_2 \qquad \text{(Cauchy-Schwarz inequality)}$$

where $|\cdot|$ denotes the absolute value of a complex number.

As any vector in a vector space, a quantum state $|\phi\rangle$ can be decomposed as a linear combination of different component states:

$$|\phi\rangle = \alpha|\psi_1\rangle + \beta|\psi_2\rangle \qquad \alpha, \beta \in \mathbb{C}$$

We say that $|\phi\rangle$ is a *superposition* of $|\psi_1\rangle$ and $|\psi_2\rangle$ if both $\alpha$ and $\beta$ are non-zero. In quantum computing, we often consider states relative to the computational basis states:

$$|0\rangle := \begin{pmatrix}1\\0\end{pmatrix} \qquad |1\rangle := \begin{pmatrix}0\\1\end{pmatrix} \qquad \text{(computational basis states)}$$

Note that these states form an orthonormal basis for a two-dimensional Hilbert space. We may put together different systems using the tensor product, and in the same way we can extend the computational basis states. Composing two qubits for example gives

$$(\alpha_1|0\rangle + \beta_1|1\rangle) \otimes (\alpha_2|0\rangle + \beta_2|1\rangle) \quad = \quad \begin{pmatrix}\alpha_1\\\beta_1\end{pmatrix} \otimes \begin{pmatrix}\alpha_2\\\beta_2\end{pmatrix} \quad = \quad \begin{pmatrix}\alpha_1\alpha_2\\\alpha_1\beta_2\\\beta_1\alpha_2\\\beta_1\beta_2\end{pmatrix}$$

where the last vector is written as a combination of computational basis states in a four-dimensional Hilbert space:

$$\begin{pmatrix}\alpha_1\alpha_2\\\alpha_1\beta_2\\\beta_1\alpha_2\\\beta_1\beta_2\end{pmatrix} = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2|01\rangle + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle.$$

In general, composing $n$ two-dimensional systems (qubits) gives a system of dimension $2^n$, and we identify the computational basis states with bitstrings of length $n$. In this way, the system can be in a superposition of $2^n$ classical data strings.

Often, the state of a system is only probabilistically known. We then say that the system is in a *mixed state* and use the density operator

$$\rho := \sum_{i=1}^{n} p_i |\phi_i\rangle\langle\phi_i| \qquad \text{(density operator of a mixed state)}$$

to indicate that it is in state $|\phi_i\rangle$ with probability $p_i$ – where $p$ is a (classical) probability vector of dimension $n$. If $p$ has all its weight on a single entry $p_i$, we say that the state is *pure* and simply write $|\phi_i\rangle$.

As a notion of closeness between two quantum states, one metric that we may use is the *trace distance*. It is defined as half the *trace norm* of the difference of two states:

$$\|\rho - \sigma\|_{\mathrm{Tr}} = \frac{1}{2} \mathrm{Tr}\left(\sqrt{(\rho-\sigma)^2}\right) \qquad \text{(trace distance)}$$

A quantum state may be *measured* in several ways. The most general measurement that an observer can perform, is a *positive operator valued measurement* (POVM). A POVM is given by a finite set of outcomes $O$ corresponding to a set of operators $\{E_i\}_{i\in O}$, such that $\sum_{i\in O} E_i = \mathbb{1}_{\mathcal{H}}$. If the state we measure is $\rho$, then the probability of finding outcome $i$ is

$$\Pr\left[i \in O \leftarrow \mathcal{M}_{\{E_i\}}(\rho)\right] = \mathrm{Tr}(E_i\rho) \qquad \text{(POVM)}$$

where Tr denotes the trace function. As a result of the measurement, the state of the system has now become

$$\frac{\sqrt{E_i}\rho\sqrt{E_i}}{\mathrm{Tr}(E_i\rho)} \qquad \text{(Post-measurement state)}$$

A more restrictive type is the projective measurement, where we require the operators $\{E_i\}_{i\in O}$ not only to sum to identity, but also to be pairwise orthogonal, i.e. $E_iE_j = 0$ for $i \neq j$. This implies that the number of outcomes in $O$ is at most equal to the dimension of the system being measured, and that for each $i \in O$ we have $E_i^2 = E_i$. In other words, all operators are projectors that project onto orthogonal subspaces of $\mathcal{H}$.

Finally, we can set the operators to be rank one projectors that project onto each of the basis states, for any basis that we like. A very common type of measurement is one where the operators are the projectors onto the computational basis states, in this case we say that we measure the system *in the computational basis*.

Note that for a subspace $V$ spanned by (a subset of the) computational basis vectors, the probability that the outcome of a measurement in the computational basis lies in $V$ (i.e. is one of the vectors that spans it) is equal to $\mathrm{Tr}(\Pi_V\rho)$, where $\Pi_V$ is the projector that projects onto $V$. If we are measuring a pure state $|\phi\rangle$, we may write this as $\mathrm{Tr}(\Pi_V|\phi\rangle\langle\phi|) = \|\Pi_V|\phi\rangle\|_2^2$.

### Quantum algorithms

Quantum algorithms compute on quantum states by applying unitary transformations, and by performing measurements. A unitary transformation is a linear map $U : \mathcal{H} \to \mathcal{H}$ (in our case domain and codomain are the same) that preserves the inner product, and hence the induced norm. Unitary transformations are characterized algebraically by the condition $UU^\dagger = U^\dagger U = \mathbb{1}_{\mathcal{H}}$ (where $U^\dagger$ denotes the conjugate transpose of $U$).

A measurement is not a unitary operation. When we say that an algorithm (or adversary) is unitary, we mean that it does not perform any measurements until the very end of its run. Any quantum algorithm can, however, be made unitary by a process called *purification*; if we have enough extra qubits at our command, all in-between measurements can be deferred to the end without changing the statistics of the final measurement.

Just like in classical computation, we can define a quantum algorithm that has access to some *oracle* $\mathcal{O}$. As a special extra operation, the algorithm is allowed to prepare a state $\rho_q$ in an $n$-qubit query register, represented by the *input/output space* $\mathbb{C}^{2^n}$. It may then apply the quantum operation $\mathcal{E}_{\mathcal{O}}$ on $\mathbb{C}^{2^n} \otimes \mathcal{H}_{\mathcal{O}}$ (where $\mathcal{H}_{\mathcal{O}}$ contains the hidden state of the oracle). $\mathcal{E}$ can implement any function and any algorithm.

In the case of *black-box access* to some unitary adversary $\mathcal{A}$, we want to model an algorithm that uses $\mathcal{A}$ as a subroutine, without making any assumptions about how $\mathcal{A}$ performs its computation. In the quantum setting, we give an algorithm oracle access to both $U_{\mathcal{A}}$ – the unitary that represents the computation of $\mathcal{A}$ – and its inverse $U_{\mathcal{A}}^\dagger$. See Section 3 of [Unr17] for a complete model of quantum black-box oracle access.

We analyze a quantum algorithm by considering its *circuit complexity* and/or its *query complexity*. The former is defined relative to a particular gate set (a set of unitaries that are approximately universal for quantum computation). It counts the asymptotic amount of gates required to execute the algorithm on a quantum computer, relative to the input size. Oracle queries are usually considered to be of unit cost. Query complexity abstracts away from this picture and *only* counts the amount of queries the algorithm needs, again asymptotically with respect to the input length.

Another aspect by which we can judge an algorithm, is its success probability. An algorithm may not succeed on every input, but still output a correct answer with good probability on a random input, or it may use internal coin flips to guide its decisions, introducing a random factor to its output. A quantum algorithm on top of this may use a quantum measurement, the outcome of which is probabilistic.

A function $\mu(x) : \mathbb{N} \to \mathbb{R}$ is called *negligible* if for every positive integer $c$ there exists an integer $N_c$ such that for all $x > N_c$

$$|\mu(x)| < \frac{1}{x^c} \qquad \text{(negligible function)}$$

We say that an algorithm has *negligible success probability* if

$$\Pr\left[\mathbf{success}(x, \eta) = 1 : x \leftarrow \mathcal{A}(\eta)\right] \leq \mu(\eta) \qquad \text{(negligible success probability)}$$

where $\mu$ is a negligible function and $\eta$ is the so-called *security parameter*.

## 1.2 Post-quantum cryptography

The development of quantum computers threatens the security of current-day cryptography. Cryptographic schemes are often built on the assumption that some underlying problem is computationally hard to solve. It now appears that what is hard to solve on a classical computer, is not necessarily hard to solve on a quantum computer. Even though technology has not yet progressed far enough to build a practical quantum computer, cryptographic protocols that are in use today may well be at risk in the near future. Finding schemes that are secure even against adversaries with a scalable quantum computer is the aim of post-quantum cryptography.

A problem is said to be computationally (quantum-) hard if no (quantum-) algorithm exists that solves the problem in polynomial time – where depending on the context we measure time either in circuit complexity or query complexity, and the polynomial is taken relative to the input size.

### Private-key and public-key under attack

Cryptographic protocols fall apart in two branches: private-key and public-key schemes. In private-key cryptography, parties share a key that has to be distributed beforehand. In public-key schemes, some subset of parties has a *pair of keys* $(sk, pk)$. They keep the secret key $sk$ to themselves, and give out the public key $pk$. In the case of an encryption scheme, external parties can now send encrypted messages to the owner of the public key, who uses his secret key to decrypt. In the case of authentication schemes, the owner of the public key can use his secret key to authenticate himself towards external parties, who verify the authentication with the help of the public key.

We often prove the security of a protocol, private-key and public-key alike, with a general kind of reduction that goes as follows: Suppose that there exists a polynomial-time adversary $\mathcal{A}$, that has non-negligible probability of breaking our protocol. We then use this assumption to solve a computationally hard problem. As long as the problem is truly hard, we are comfortable that an actual (polynomial-time) adversary cannot break the protocol.

Two quantum algorithms in particular have challenged the presumed hardness of computational problems that are currently widely used in cryptography.

**Shor's algorithm** [Sho94] is a quantum algorithm that can solve two important problems in polynomial-time (measured in circuit complexity). It solves both *integer factorization* and *the discrete-logarithm problem*, which together form the basis of almost every public-key protocol in use today. In such schemes, the public key is a large composite number and the secret key its (usually two) prime factors, or the public key is a group element $y := g^e$ where $g$ is the (publicly known) group generator and $e$ is the secret key. As we noted, the secret key should be unobtainable from the public key, but this is precisely what Shor's algorithm allows one to do. To remain secure against a quantum adversary, such schemes will therefore have to reduce to a different, quantum-hard computational problem.

**Grover's algorithm** [Gro96] for unstructured search forms a threat to any protocol where the adversary can guess a key and then check whether this key is correct. The speedup over classical algorithms however is not as big as with Shor's algorithm. If we know the key-length is $n$ and want to have a fifty-percent chance of finding the correct key, a classical brute-force algorithm would have to try at least $\frac{2^n}{2}$, that is, half of the keys. Grover can do the same with only $2^{\frac{n}{2}}$ evaluations, giving a quadratic speedup in terms of query complexity. Doubling the key length therefore effectively neutralizes a Grover attack.

### Quantum-hard computational problems

A couple of (new) computational problems have been conjectured to be quantum-hard, and proposed as a post-quantum alternative to the number-theoretic assumptions that are broken by Shor's algorithm. Next to *code-based* cryptography and *hash-based* cryptography, there are two of them that we highlight:

**Lattice-based cryptography** (see [Pei16] for a survey) uses a 'good' basis (= close to orthogonal) for some lattice in $\mathbb{R}^n$ as a secret key. Several computational tasks can be defined – for example, finding the lattice point that is closest to some given vector – that are hard to solve when only a 'bad' basis is known (for high dimensions). Lattices can have algebraic structure on them, which increases the efficiency of lattice-computations, hence of the lattice-cryptosystems, but also of any attacks on the scheme. Therefore, there is a trade-off between efficiency and security. Currently however, even for some (but not all of them) of the more structured variants based on module lattices, no subexponential (quantum) algorithms are known that solve the corresponding computational problems.

**Supersingular isogeny cryptography** (see [De 17] for a good introduction) improves on *elliptic-curve cryptography*, a branch of cryptography that in its original form is broken by Shor's algorithm. Elliptic curves over finite fields can be used to construct a group structure for which the discrete-logarithm problem is believed to be at least as hard as over any other group. With only classical adversaries to cope with, key lengths may therefore be kept relatively short, leading to efficient protocols. Post-quantumly however such schemes are insecure, due to Shor's solution for the discrete-logarithm problem.

Recently, new computational problems involving elliptic curves have been defined. A particular type of algebraic map, which we call an 'isogeny' between two elliptic curves – actually, the curves must be 'supersingular' elliptic – is believed to be hard to find even for a quantum computer. Knowledge of the isogeny can serve as a secret key, with the corresponding public key being the curves involved.

### Quantum secure reductions

Basing schemes on quantum-hard computational problems is not enough to preserve security in the post-quantum era. The security reduction itself, relating the security of the scheme to the hardness of a computational problem, may not go through in the quantum world. In other words, breaking the protocol might – for a quantum computer – not be equivalent to solving the underlying problem. Replacing a classically hard problem by a quantum-hard problem can therefore never be enough to reinstall our confidence in the security of the scheme.

When giving a security reduction for a post-quantum protocol, we always need to assume the adversary to be quantum, with all the special quantum features that a quantum adversary has. For example, when the adversary has private access to some function, we need to assume that it could evaluate the function on a superposition of inputs. We will discuss related issues in Section 1.7.

## 1.3 Identification and zero-knowledge proofs of knowledge

Cryptographic protocols are used in a variety of tasks. One particular goal that we may have, is to be able to securely identify ourselves, perhaps to a party that we have not had any previous contact with. As an example, I might want to place a bet with a bookmaker over the internet. Even though we have never met in real-life, I want the bookkeeper to accept bets on my name only if they come directly from me.

'No previous contact' means that we will need a public-key protocol, since there is no opportunity to share a private key beforehand. A public key must also be shared, but it could for example be broadcast by a trusted third party. What is the precise interpretation of 'securely identify ourselves'? At the very least this should mean that no unauthorized party can identify as someone else. A further demand that we might have, is that the protocol works more than once, without any parties having to change their keys. Finally, it can be desirable

to execute the whole protocol in a single message, instead of having to go through multiple rounds of communication.

One solution to this cryptographic challenge is to use a so-called *zero-knowledge proof of knowledge*. As the name suggests, it allows us to prove that we posses some piece of knowledge. The paradoxical sounding prefix 'zero-knowledge' refers to the fact that eavesdroppers should gain zero knowledge from overhearing the conversation, i.e. from obtaining a transcript of the protocol, and also the verifier should learn nothing except that the proven statement is true.

A zero-knowledge proof of knowledge can be used for identification, if we let the honest prover prove knowledge of a secret password that only he or she knows, without revealing the password itself. The zero-knowledge property ensures that the same key can be reused multiple times, since no knowledge – in particular about the secret password – is leaked from the protocol.

Obviously, simply sending the password does not satisfy our constraints, since it would reveal the secret both to the verifier and to any eavesdropper. The trick is to let the verifier send a *random* challenge puzzle, one that requires knowledge of the secret to solve. Here we have a connection with the computational problems from the previous section; the puzzle should be (quantum-) hard to anyone who does not know the secret.

### Toy example

To illustrate the idea, we present a simple zero-knowledge proof of knowledge (adapted from [QGB89]). Imagine a cave, consisting of a tunnel that after a while splits into a left and a right branch. At some further point the two branches meet again, but it is only possible to go from the one to the other by passing through a locked gate. The gate however can be opened, if and only if one knows its secret password.

Suppose that Alice wants to prove to Bob that she knows the password to the gate, without revealing the password to him, or to anyone eavesdropping on their conversation. They could agree on the following multi-round protocol: At the start of every round, Alice enters the cave first, so that Bob cannot see which of the two branches she takes. After a minute Bob walks up to the forking point, and shouts to Alice which of the branches he want to see her reappear from, left or right.

If Alice knows the secret password, she can always open the gate to move to the desired part of the cave. In an $n$ round-protocol, she has a probability of $2^{-n}$ of always showing up at the right side *without knowing the password*. Thus, Bob will accept if and only if Alice meets every single challenge.

Notice the importance of the randomness in this protocol. If Alice could predict the order of left/right choices beforehand, she could always choose to enter via the tunnel she has to come out of, without knowing the password. In fact, if Bob and Alice would secretly agree on a specific order of challenges, they could stage an execution of the protocol that seemingly proves Alice's knowledge of the password, contrary to the facts. To an outside observer (eavesdropper) everything seems as normal, only Alice and Bob know there is foul play at hand.

The above observation lies at the heart of the zero-knowledge property of the protocol; if anyone not knowing the password could produce a transcript that is *indistinguishable* from an honest execution, it must be impossible to extract *any information* about the password from the honest transcript. The reason is that since the non-honest transcript per definition does not contain any information about the secret, the honest version cannot contain any extractable information about it either, or else the two would be distinguishable. The eavesdropper can not even determine whether Alice knows the password or not, since he never knows whether she cheated or not. Only Bob knows that his choices were random, and therefore he is the only one who is convinced that Alice knows the password.

Formally, the zero-knowledge property is represented by the existence of a *simulator*. If the simulator, who is not given access to the secret, can produce transcripts that are indistinguishable from an honest execution, the definition is fulfilled.

## 1.4 Sigma-protocols

A sigma-protocol is a three-round interactive proof system, tied to a family of relations $R_\eta$, that for any integer $\eta$ allows one party (the prover) to choose a statement $x$, and prove to another party (the verifier) the following assertion:

$$\exists w \ : \ (x, w) \in R_\eta.$$

We consider sigma-protocols for fixed length relations and quantum provers. A fixed-length relation is such that for every $\eta$ there exist values $\ell_\eta^x$ and $\ell_\eta^w$ such that $(x, w) \in R_\eta$ implies

$|x| = \ell_\eta^x$ and $|w| = \ell_\eta^w$. We define

$$L_{R_\eta} := \{x \ : \ \exists w.\ (x,w) \in R_\eta\}$$

The protocol is given by the values $\ell_\eta^{com}$,$\ell_\eta^{ch}$,$\ell_\eta^{resp}$ that specify the lengths of the three messages 'commitment', 'challenge', 'response', and by the quantum polynomial-time prover $(P_\Sigma^1, P_\Sigma^2)$ and the deterministic polynomial-time verifier $V_\Sigma$.

A transcript of an honest execution of the protocol is generated as in the game **Sigma**, which is defined as:

$$\begin{aligned}
&com \leftarrow P_\Sigma^1(1^\eta, x, w),\\
&ch \xleftarrow{\$} \{0,1\}^{\ell_\eta^{ch}},\\
&resp \leftarrow P_\Sigma^2(1^\eta, x, w, ch),\\
&ok_V \leftarrow V_\Sigma(1^\eta, x, com, ch, resp)
\end{aligned}$$

Here the commitment satisfies $com \in \{0,1\}^{\ell_{com}}$, the challenge is such that $ch \in \{0,1\}^{\ell_{ch}}$ and for the response we have $resp \in \{0,1\}^{\ell_{resp}}$. $ok_V$ is a binary value. If $ok_V = 1$, we say that the verifier accepts.

The following is a selection of formal properties that any sigma-protocol might have, taken from [Unr17]:

**Definition 1.1 (Properties of sigma-protocols)**

- ***Completeness:*** *For any quantum-polynomial time algorithm $A$, there is a negligible function $\mu$ such that for all $\eta$,*

$$\begin{aligned}
&\Pr[(x,w) \in R_\eta \wedge V_\Sigma(1^\eta, x, com, ch, resp) = 0 \ : \ (x,w) \leftarrow A(1^\eta),\\
&com \leftarrow P_\Sigma^1(1,\eta,x,w), ch \xleftarrow{\$} V_\Sigma(1^\eta, x), resp \leftarrow P_\Sigma^2(1^\eta, x, w, ch)] \leq \mu(\eta)
\end{aligned}$$

- ***Statistical soundness:*** *There is a negligible $\mu$ such that for any stateful classical (but not necessarily polynomial-time) algorithm $A$ and all $\eta$, we have that*

$$\begin{aligned}
&\Pr[ok = 1 \wedge x \notin L_{R_\eta} : (x, com) \leftarrow A(1^\eta), ch \xleftarrow{\$} \{0,1\}^{\ell_{ch}},\\
&resp \leftarrow A(1^\eta, ch), ok \leftarrow V_\Sigma(1^\eta, x, com, ch, resp)] \leq \mu(\eta).
\end{aligned}$$

- ***Perfect special soundness:*** *There is a quantum polynomial-time algorithm $E_\Sigma$ such that for all $\eta, x, com, ch, resp, ch', resp'$ with $ch \neq ch'$ and $V_\Sigma(1^\eta, x, com, ch, resp) = V_\Sigma(1^\eta, x, com, ch', resp') = 1$, we have that*

$$\Pr[(x,w) \in R_\eta : w \leftarrow E_\Sigma(1^\eta, x, com, ch, resp, ch', resp')] = 1.$$

- ***Honest-verifier zero-knowledge (HVZK):*** *There is a quantum polynomial-time algorithm $S_\Sigma$ (the simulator) such that for any stateful quantum polynomial-time algorithm $A$ there is a negligible $\mu$ such that for all $\eta$ and all $(x,w) \in R_\eta$,*

$$\begin{aligned}
&\Big|\Pr[b = 1 : (x,w) \leftarrow A(1^\eta),\ com \leftarrow P_\Sigma^1(1^\eta, x, w),\ ch \xleftarrow{\$} \{0,1\}^{\ell_\eta^{ch}},\\
&\qquad resp \leftarrow P_\Sigma^2(1^\eta, x, w, ch), b \leftarrow A(com, ch, resp)]\\
&- \Pr[b = 1 : (x,w) \leftarrow A(1^\eta),\ (com, ch, resp) \leftarrow S(1^\eta, x),\\
&\qquad b \leftarrow A(com, ch, resp)]\Big| \leq \mu(\eta).
\end{aligned}$$

- ***Perfectly unique responses:*** *There exist no values $\eta, x, com, ch, resp, resp'$ with $resp \neq resp'$ and $V_\Sigma(1^\eta, x, com, ch, resp) = 1$ and $V_\Sigma(1^\eta, x, com, ch, resp') = 1$.*
- ***Unpredicatable commitments:*** *The commitment has superlogarithmic collision-entropy. In other words, there is a negligible $\mu$ such that for all $\eta$ and $(x,w) \in R_\eta$*

$$\Pr[com_1 = com_2 \ : \ com_1 \leftarrow P_\Sigma^1(1^\eta, x, w),\ com_2 \leftarrow P_\Sigma^1(1^\eta, x, w)] \leq \mu(\eta).$$

- [Unr12] ***Proof of knowledge property/extractability*** *with knowledge error $\kappa$: There exists a constant $d > 0$, a polynomially bounded function $p > 0$, and a quantum polynomial-time oracle machine*[1] $\mathsf{K}$ *such that for any quantum polynomial-time algorithm $A$, there is a negligible $\mu$ such that for all $\eta$ and all $x \in \{0,1\}^{\ell_\eta^x}$ we have that*

$$\Pr[ok = 1 \ : \ \textbf{Sigma}] \geq \kappa(\eta) \Rightarrow$$

[1] See [Unr12] for a precise definition. For our purposes it is enough to say that an oracle machine has access to a unitary describing some other (quantum) algorithm, and its inverse. We will discuss the motivation of this definition in Section 2.1.

$$\Pr[(x,w) \in R_\eta \ : \ w \leftarrow \mathsf{K}^{A(1^\eta,x)}(x)] \geq \frac{1}{p(\eta)} \cdot (\Pr[ok = 1 \ : \ \textbf{Sigma}] - \kappa(\eta))^d - \mu(\eta)$$

*where* **Sigma** *is the game we defined above, and* $\mathsf{K}$ *has oracle access to* $A = (A_1, A_2)$.

In [Unr12] it was shown that a sigma-protocol which has perfect special soundness and perfectly unique responses, is a quantum proof of knowledge, i.e. is extractable in the sense of the definition given above, where the adversary may be quantum. If we only consider classical adversaries, any scheme that has special soundness is automatically a proof of knowledge, since we can always rewind a classical adversary to obtain accepting responses for two different challenges. In Section 2.1 we will discuss in more detail the concepts of extractability, proofs of knowledge and rewinding.

### Example: The Schnorr identification scheme

To illustrate these formal notions, we present an example sigma-protocol, the Schnorr identification scheme (originally in [Sch91], we adapted a version of the protocol found in [KL14] to match our notation). Note that three-round identification schemes form a subclass of sigma-protocols, where the relation $R$ consists of a relation between public keys and corresponding secret keys. The scheme is based on the discrete-logarithm problem, therefore it is only classically secure.

Let $\mathbb{G}$ be a cyclic group of order $q$ with generator $g$. The prover chooses uniform $w \in \mathbb{Z}_q$, and sets $y := g^w$. It keeps $w$ as its secret key, and broadcasts the public key $x := (\mathbb{G}, q, g, y)$. We then have the following scheme:

| Prover$(w)$ : | | Verifier$(\mathbb{G}, q, g, y)$ : |
|---|---|---|
| $k \leftarrow \mathbb{Z}_q$ | | |
| $com := g^k$ | $\xrightarrow{com}$ | |
| | $\xleftarrow{ch}$ | $ch \xleftarrow{\$} \mathbb{Z}_q$ |
| $resp := ch \cdot x + k \bmod q$ | $\xrightarrow{resp}$ | accept if $g^{resp} \cdot y^{-ch} = com$ |

We will informally explain why the Schnorr scheme is a secure identification scheme. Note first that it is correct for any properly formed pair $(x, w)$, and therefore as a sigma-protocol satisfies **completeness**. In an honest execution we have

$$g^{resp} \cdot y^{-ch} = g^{ch \cdot x + k \bmod q} \cdot (g^x)^{-ch} = g^k = com$$

so that the verifier will always accept in the honest case. Furthermore, the protocol satisfies **perfect special soundness**. If $ch \neq ch' \in \mathbb{Z}_q$ and we have

$$g^{resp} \cdot y^{-ch} = g^{resp'} \cdot y^{-ch'} = com$$

then

$$g^{resp - resp'} = y^{ch - ch'}$$

which means that anyone who has access to these values (like the algorithm $E_\Sigma$ from the formal definition of special soundness) can compute

$$w = \log_g y = (resp - resp') \cdot (ch - ch')^{-1} \bmod q.$$

Finally, the Schnorr identification scheme satisfies the **honest-verifier zero-knowledge** property. It is easy for any simulator $S$ that does not know the witness $w$ (the secret key), to create a transcript indistinguishable from an honest execution, simply by reversing the order of the messages. $S$ starts by picking $resp$ and $ch$ independently at random from $\mathbb{Z}_q$. It may then compute

$$com := g^{resp} \cdot y^{-ch}$$

and output $(com, ch, resp)$. Since $resp$ and $ch$ are uniform in $\mathbb{Z}_q$, $com$ is uniform in $\mathbb{G}$, exactly as in the honest transcript. $resp$ should be uniformly random in $\mathbb{Z}_q$ with the constraint that $resp = \log_g[com \cdot y^{ch}]$, which is indeed the case in the simulated transcript. Therefore, the two transcripts are indistinguishable.

The three properties combined tell us that Schnorr is a secure identification scheme in the following sense: 1. Any authenticated party (i.e anyone who knows the secret key) can always identify successfully, and 2. Any classical adversary who does not know the secret key has only negligible probability of doing the same.

Completeness shows that requirement 1. is satisfied, for any valid key-pair. For requirement 2., note that by the honest-verifier zero-knowledge property, an adversary gains nothing from passively eavesdropping on an honest execution. We can therefore restrict to an adversary who can only do the following: It receives a public key $y$, it sends a message $com$ and it receives a random challenge $ch$. Now it has to compute a correct response $resp$. If it can succeed at this task with more than negligible probability, taken over the randomness of $ch$, than it can also succeed with non-negligible probability in *two different runs*. Of course it could also simply pick two random challenges by itself, and compute the corresponding responses. Then by the special soundness property, it can compute the secret key $w$. Computing $w$ from only $x$ is assumed to be computationally hard (classically), so we conclude that any classical adversary with non-negligible success probability must have known $w$ all along, hence is an authorized party.

## 1.5 The Fiat-Shamir transformation

Let us return to Alice and Bob and their special cave. In Section 1.3, we argued that for any (passive) eavesdropper, it is impossible to determine whether Alice and Bob cooperated to cheat or not, so that nobody except Bob can be convinced that Alice knows the secret password.

What if Alice and Bob *do* want to convince the rest of the world of Alice's knowledge? Suppose that they want to send a single (video) message to all of their friends, to prove to them that Alice knows the password to the gate. Filming just the execution of the protocol is not enough, because as we noted above, the resulting video could have been made by anyone who does not know the password. However, if they would *outsource the randomness* of the choice of left-right challenges to a coin, which they would flip *in public view* – in front of the camera – right before Bob shouts the outcome to Alice, then the video message would definitely be convincing to anyone who sees it! (As long as they film in one shot, to prove that no failed trials have been cut out.)

The Fiat-Shamir transformation takes a particular sigma-protocol, and outsources the random choice of the challenge message to a cryptographic hash function, transforming the protocol from an interactive proof system into a single message scheme – a non-interactive proof system. A cryptographic hash function is a function that is believed to be computationally hard to invert, and its output should be computationally hard to distinguish from random.

The Fiat-Shamir transformation is characterized by the following two algorithms [Unr17]:

$\mathbf{P}^{\mathbf{H}}_{\mathbf{FS}}$ :

**Input:** $1^\eta, x, w$
**Oracles:** Classical queries to $H$.

$com \leftarrow P^1_\Sigma(1^\eta, x, w)$
$ch := H(x||com)$
$resp \leftarrow P^2_\Sigma(1^\eta, x, w, ch)$
**return** $\pi := com||resp$

$\mathbf{V}^{\mathbf{H}}_{\mathbf{FS}}$ :

**Input:** $1^\eta, x, \pi$
**Oracles:** Classical queries to $H$.

$com||resp := \pi$
$ch := H(x||com)$
**return** $V_\Sigma$

For non-interactive proof systems zero-knowledge is defined as follows:

**Definition 1.2 (Zero-knowledge ([Unr17], simplified))** *A non-interactive proof system $(P, V)$ is zero-knowledge iff there is a quantum polynomial-time simulator $S$ such that for every quantum polynomial-time algorithm $A$ there is a negligible $\mu$ such that for all $\eta$*

$$\Big|\Pr[b = 1 \ : \ H \xleftarrow{\$} Fun(\ell^{in}_\eta, \ell^{out}_\eta),\ b \leftarrow A^{H,P}(1^\eta)] - \Pr[b = 1 \ : \ H \xleftarrow{\$} Fun(\ell^{in}_\eta, \ell^{out}_\eta),\ b \leftarrow A^{H,S}(1^\eta)]\Big| \leq \mu(\eta)$$

*where $Fun(\ell^{in}_\eta, \ell^{out}_\eta)$ is the set of all functions from $\{0,1\}^{\ell_{in}}$ to $\{0,1\}^{\ell_{out}}$.*

The definition says that it should be infeasible for an adversary to distinguish between interaction with the simulator or an honest prover. Note that $H$ is taken uniformly at random from the set of all functions of its type. That is not the same as saying that $H$ is a cryptographic hash function, even though such functions are designed to be indistinguishable from random. Choosing $H$ at random in our security definitions is an idealization commonly known as the *random-oracle model*. We will discuss this model in further detail in Section 1.7.

If the underlying sigma-protocol has the properties completeness, honest-verifier zero-knowledge and unpredictable commitments, then the non-interactive proof system given by

the Fiat-Shamir transform is zero-knowledge against a quantum adversary. The proof requires special quantum techniques, and was first given in [Wat09]. We present here only the simulator that produces the indistinguishable transcripts, which comes from a proof of the zero-knowledge property in [Unr17]:

$\mathbf{S}^{\mathbf{H}}_{\mathbf{FS}}$ :

**Input:** $1^\eta, x$
**Oracles:** Reprogramming access to $H$.

$(com, ch, resp) \leftarrow S_\Sigma(1^\eta, x)$
**if** $V_\Sigma(1^\eta, x, com, ch, resp) = 1$ **then**
  $H(x||com) := ch$
**return** $\pi := com||resp$

Note that we allow the simulator to reprogram the oracle. A similar kind of cheating is what enabled Alice and Bob to feign an honest execution in the cave, and the simulator for the sigma-protocol to come up with fake transcripts. In all cases, the randomness of the challenges is what makes the task difficult, so controlling the source of randomness allows one to produce the correct output without knowing the witness. We do not care that it involves cheating, what matters is that such indistinguishable dishonest transcripts exist, so that honest and dishonest alike contain no information about the witness.

### Non-malleability

For non-interactive proofs, zero-knowledge is only the first level of protection against an eavesdropper. It is great that an adversary cannot learn anything about the secret from seeing the proof, but maybe she does not need to learn the secret to achieve mischief. It could be that getting her hands on a proof for some valid statement $x$, allows her to output a modified proof for a false statement $x'$. In general, when the adversary can modify the proof in *any* meaningful way, we say that the scheme is *malleable*.

For non-interactive proof systems, we discern four types of malleability and corresponding security definitions. We say that a system is

- **Weakly simulation-sound:** When the adversary *cannot* change the proof of some statement $x$ into a valid proof for a *different false statement* $x$.
- **Strongly simulation-sound:** When the adversary *cannot* change the proof of some statement $x$ into a *different valid proof* for a false statement $x'$ (where possibly $x = x'$).
- **Weakly simulation-sound extractable:** When the adversary cannot change a proof for some statement $x$ into a valid proof for a *different statement* $x'$ for which it does not know a corresponding witness.
- **Strongly simulation-sound extractable:** When the adversary cannot change a proof for some statement $x$ into a *different valid proof* for a statement $x'$ for which it does not know a corresponding witness.

For Fiat-Shamir proof systems, when the underlying sigma-protocol has unique responses, every statement has only a single proof. All the weak definitions are then equivalent to the corresponding strong ones. In this thesis we are only concerned with protocols that have unique responses, so we will only consider the strong definitions.

[FKMV12] showed – in the classical case – that if the sigma-protocol has honest-verifier zero-knowledge and unique responses, then the Fiat-Shamir system is (strongly) simulation-sound extractable, which implies that it is simulation-sound (because if a witness can be extracted, the statement must be true). [Unr17] showed that in the quantum case, the sigma-protocol needs to have statistical soundness and unique responses to be strongly simulation sound. They also showed that quantumly, extractability implies simulation-sound extractability, but left extractability as an open problem. The main goal of this thesis is to establish the (simulation-sound) extractability of the Fiat-Shamir transformation.

## 1.6 Signatures

As a final scenario in our cave-analogy, we could imagine that Alice and Bob have some important message to share with their friends. Since the content of the message is of a rather sensitive nature, they want to be sure that the recipients can *verify* the integrity of the message, and know for certain that is was Alice who sent it.

Once again they think of a clever plan. Because everyone knows that Alice is the only one who knows the secret password, they can use it as an identity marker (just like in the Schnorr scheme). They decide to film one more execution of the cave protocol, but this time they include in one long shot the successful trials, the tossing of the coin *and* their spoken message. Nobody not knowing the password could create the same video, or one that is just like it but with a different message.

Their method of creating a message has one consequence that Alice and Bob had not thought of beforehand. If someone intercepts the video and sees its contents, *Alice cannot deny that she is the author of the message.* With their (politically?) sensitive message, this property could be to their disadvantage. It could also be an advantage, for example if Alice uses their method to record a promise. Any receiver can then be sure that Alice will not back away from it (which could be to Alice's advantage!).

The properties of the message described above are called *integrity, authenticity* and *non-repudiation.* They are precisely the properties we expect of a *digital signature scheme.* The Fiat-Shamir transform was originally presented as a signature scheme [FS87]. In fact, any non-interactive proof system can be made into a signature scheme. The simple idea behind it is the same as with Alice and Bobs video technique; a signature consists of a proof that includes our knowledge of the secret key *and* the message.

**Definition 1.3 ([Unr17]; Signature schemes from non-interactive proof systems)** *We say that $G$ is an instance generator for a relation $R_\eta$ if $G(1^\eta)$ outputs $(x, w) \in R_\eta$ with overwhelming probability. Fix a length $\ell_\eta^m$ and define $R'_\eta := \{(x||m, w) : |m| = \ell_\eta^m \wedge (x, w) \in R_\eta\}$, with hash function $H$. Let $(P, V)$ be a non-interactive proof system for the relation $R'_\eta$. A signature scheme $(KeyGen, Sign, Verify)$ with message space $\{0,1\}^{\ell_\eta^m}$ is then defined as follows:*

- $KeyGen(1^\eta)$ *: Pick* $(x, w) \leftarrow G(1^\eta)$*. Let* $pk := x$ *and* $sk := w$*. Return* $(pk, sk)$
- $Sign^H(1^\eta, sk, m)$ *: Run* $\sigma \leftarrow P^H(1^\eta, x||m, w)$*. Return* $\sigma$*.*
- $Verify^H(1^\eta, pk, m, \sigma)$ *: Run* $ok \leftarrow V^H(1^\eta, x||m, \sigma)$*. Return* $ok$*.*

We now have a signature scheme, and replacing $(P, V)$ by $(P_{FS}, V_{FS})$ we have our Fiat-Shamir signature scheme. What about its security? The standard security definition for signature schemes is

**Definition 1.4 ([Unr17]; Existential unforgeability)** *A signature scheme $(KeyGen, Sign, Verify)$ is existentially unforgeable iff for all (quantum) polynomial-time algorithms $A$ there exists a negligible $\mu$ such that for all $\eta$ we have*

$$\Pr[ok = 1 \wedge (m^*, \sigma^*) \notin \textbf{Sig}\text{−queries} : H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out}),\ (pk, sk) \leftarrow KeyGen(1^\eta)$$
$$(m^*, \sigma^*) \leftarrow A^{H,\textbf{Sig}}(1^\eta, pk),\ ok \leftarrow Verify^H(1^\eta, pk, m^*, \sigma^*)] \leq \mu(\eta)$$

*where* **Sig** *is an oracle that accepts only classical queries (for the difference with quantum oracles, see Section 1.7) that upon (classical) input $m$ returns $Sign^H(1^\eta, sk, m)$. Note that queries to $H$ may be quantum.* **Sig**−queries *is the list of all queries made to* **Sig** *(when it is queried with $m$ and the oracle-answer is $\sigma$, then $(m, \sigma)$ is added to the list). Finally, $Fun(\ell_\eta^{in}, \ell_\eta^{out})$ is the set of all functions from $\{0,1\}^{\ell_{in}}$ to $\{0,1\}^{\ell_{out}}$.*

The definition says that no adversary can output a valid message-signature pair that it has not queried before. We give it access to a signing oracle because we want to model the situation where the adversary might obtain some valid signature, and tries to modify it to a different signature or a signature for a different message. Again, when we consider only sigma-protocols with unique responses, any statement has only a single proof in the non-interactive proof system and hence a message has only one signature in our signature scheme, and the above definition is equivalent with the weaker one where only require $m^* \notin$ **Sig**−queries instead of $(m^*, \sigma^*) \notin$ **Sig**−queries.

Note that this is again a definition that is phrased in the paradigm of the random-oracle model, which we will discuss in more detail in the next section.

[Unr17] proves that a non-interactive proof system with simulation-sound extractability is existentially unforgeable. As we noted above, they also prove that an extractable non-interactive proof system is simulation-sound extractable. The only thing left to prove is the extractability of the Fiat-Shamir transform.

## 1.7 The (Quantum-) random-oracle model

In the previous sections, we noted a curiosity; the Fiat-Shamir transform was introduced as a procedure that uses a cryptographic hash function $H$ to take away the interaction in a sigma-protocol, but in our security definitions (Definitions 1.2 and 1.4) we require $H$ to be a truly

random function. A cryptographic hash function is indeed designed to be as indistinguishable from random as possible. In practice however, we need a function that is both deterministic – because it needs to give the same query answers to all parties that query it – and efficiently implementable. These two features are mutually exclusive for a truly random function (of suitable domain size). We could obtain a deterministic random function by picking each of its entries independently at random, but the lookup-table of this function would require memory exponential in the input size. That also means that any efficient function will need to have some algorithmic evaluation procedure which does not use a lookup table – such a function cannot be fully random.

The cryptographic hash function that we use in practice is thus not a random function. Modeling it as truly random is an idealization, which allows us to prove security in cases where otherwise no security proof is known. Of course these proofs can no longer be taken as absolute guarantees of security, they should be considered heuristically. In fact, some schemes have been devised for which a security proof exists in the random-oracle model, but which are also shown to be insecure when instantiated with any real cryptographic hash function [CGH04]. These schemes however were especially designed for this purpose, and many people believe that in general a proof in the random-oracle model provides good confidence in a scheme's practical security.

When we give a proof in the random-oracle model, it is often in the form of a reduction – as we described in Section 1.2 – i.e. we assume the adversary can break our protocol, and then use this assumption to solve a computationally hard problem. In the reduction, we may *reprogram* the random oracle. Remember that we allowed the Fiat-Shamir simulator to do the same in Section 1.5. It could be asked why it is natural to do so. Comparing it to the situation of the simulator, what mattered there is that from the perspective of the eavesdropper, the transcript *could just as well have been produced honestly or dishonestly.* Crucial here is the (idealized) randomness of the oracle; if the oracle can produce any value with equal probability, there is no way for the eavesdropper to tell a reprogrammed oracle from the original (that is, if we reprogrammed it with a new *random* value) – and thus no way to tell an honest from a dishonest transcript.

In the reduction, we should also argue from the perspective of the adversary. The assumption is that it can break the protocol, *if it receives random values from the random oracle.* Again we have that by the idealized randomness, it is impossible for the adversary to tell the difference between the reprogrammed and the original oracle. Therefore, it can break the protocol equally well on the original as on the reprogrammed version. Since the oracle is implemented by the reduction, we may just as well program it with answers that suit the goal of the reduction, as long as the resulting oracle still looks random to the adversary. In any case, the adversary could have supplied these suitable answers *itself* if its goal was to solve the underlying problem, so that the argument "if the adversary can break the protocol then also the computational problem is solved" remains justified.

### The Quantum random-oracle model

There is one 'quantum' issue with the random-oracle model that we cannot simply argue away. In the real world, the evaluation procedure of a cryptographic hash function is publicly specified, so that all parties can use it. Any quantum adversary could download this specification and implement it as a quantum circuit. Therefore, the adversary could evaluate the hash function on quantum states, and hence on a superposition of (exponentially many) different inputs. Our model should incorporate this special quantum feature.

In [BDF+11], the *quantum* random-oracle model (QROM) was introduced. In the QROM, the adversary is allowed to query quantum states to the (idealized) random oracle. The authors noted that the new model is problematic to some features that we are used to in the classical random-oracle model:

1. **Adaptive Programmability:** In the classical ROM, the reduction often reprograms the oracle at some point in the execution. We assume that the adversary queries every input only once (classically this is not a restriction because the oracle answer, once obtained, can be copied by the adversary indefinitely), so the reprogramming is impossible to detect. In the QROM however, the adversary may query the same input multiple times. Then, by querying a state in superposition, it may get some information about all values of the oracle at once, making it difficult to reprogram the oracle adaptively without being caught.
2. **Preimage Awareness:** When an adversary hides a specific input in a superposition of exponentially many values, it may be hard for the security reduction to find out which value the adversary is actually interested in.

3. **Efficient Simulation:** Again due to superposition access, lazy sampling, a technique used to let the reduction efficiently simulate a random function, is no longer possible. Efficiently simulating the quantum random oracle therefore becomes a challenge.
4. **Rewinding/Partial Consistency:** Some proofs in the random-oracle model require the reduction to *rewind* the adversary, which means that we replay the adversary from a certain point in its execution, but with different outputs from the random oracle in the second run. Here the difficulties are twofold: We cannot clone the quantum state of the adversary in order to save it for a second execution, and secondly we face again the problem of changing the oracle unnoticed.

[BDF+11] already presented a solution for point 3, using quantum-accessible pseudorandom functions. The existence of such functions was still open at the time, but [Zha12] gave a construction for them. Problem 4. was partially solved by [Wat09] and [Unr12] (we discuss how in the next section).

In the same paper, the authors described the concept of a *history-free reduction* for signature schemes. When the (classical) security reduction answers oracle queries independent of the query history (i.e. of the previous input-output pairs that passed through the oracle), the reduction is called history-free. The authors prove that such reductions are valid in the QROM as well.

The classical reduction for Fiat-Shamir signatures is not history-free. Its security in the QROM is therefore left open by [BDF+11]. In Sections 4 and 5 we show that problems 1. and 2. do not prevent a reduction in the Fiat-Shamir case, thereby proving the security of Fiat-Shamir signatures in the QROM.

# 2 Problem statement

Proving the security of Fiat-Shamir signatures comes down to the following: We need to show that if the adversary can create a *fresh* signature, i.e. one that it has not seen before, than it must be true that the adversary 'knows' the secret key. In this section, we explain how we can formally define an adversary that 'knows' something, and why it is difficult to prove that a successful *quantum* Fiat-Shamir forger must know the secret key.

## 2.1 Revealing the knowledge of the adversary

The concept of a *proof of knowledge* was first introduced in [GMR85], and more rigorously defined in [BG93]. Intuitively, what we want is that if the verifier accepts the proof for a statement $x$, then the prover *knows* a witness $w$ for $x$ (i.e $(w, x) \in R_\eta$). How can we test the knowledge of the prover, to who's inner workings we have no access at all? In fact, in determining the knowledge of the prover, we should confine ourselves to examining only the interaction between the prover and the verifier, for if we needed more evidence, then we could hardly call this interaction a proof of knowledge.

To determine the knowledge (implicitly) present in the interaction only, we define the *demonstrated knowledge* of the prover to be anything that an efficient *extractor* algorithm can compute, when given black-box oracle access to the prover. If the demonstrated knowledge includes a valid witness for the proven statement, we say that the protocol is *extractable*, or equivalently that the proof is a *proof of knowledge*.

The black-box access ensures that we assume nothing about *how* the prover computes its output, thereby not restricting the class of provers. It also enables the extractor algorithm to simulate multiple executions of the prover, and to take the role of the verifier in asking the prover clever questions. The output of the extractor is anything the verifier *could* have learned, had it asked the right questions. Classically, making the prover act in multiple executions of the protocol is not a stronger requirement than what we normally ask of the prover; if we assume that it can convince the verifier with good probability in an average run, then it should be able to succeed in multiple runs.

One further step is to note that if we can run the prover multiple times, then (again, classically) we can also run it twice from the same intermediate state. In effect, we are *rewinding* the prover after the first execution to a previous point in its run. Rewinding allows the extractor to obtain two *related* proofs from the prover. As we have seen in Section 1.4, in protocols with special soundness two proofs that have a particular relation to each other are sufficient to compute a witness from.

### Goal of the thesis

The main goal of this thesis is to show that the Fiat-Shamir proof system is *quantum* extractable. That is, we want to be sure that an adversary can only create a proof for a statement $x$ if it 'knows' a valid witness $w$ for $x$. The post-quantum security of Fiat-Shamir *signatures* will then follow by a previous result from [Unr17].

To achieve our goal, we may in principle use the same tools as described above, but we have to account for the quantum nature of the adversary. Concretely, we may

- not copy the state of the adversary to run it again from the same point in its execution, due to no-cloning.
- in general not run the adversary more than once, since we can also not copy its initial state.
- not measure any (intermediate) output from the adversary without possibly disturbing its internal state.

and we have to

- allow the adversary to perform quantum operations.
- allow the adversary superposition access to the random oracle.
- allow the adversary to query the same input more than once, because some quantum algorithms require multiple queries on the same state, and the adversary can in general not copy information it obtained in a particular query.

The challenge is to construct a (quantum) extractor that can use black-box access to the quantum adversary to compute a valid witness, notwithstanding the above limitations.

## 2.2 Why the classical proof does not work in the QROM

The classical extraction procedure for the Fiat-Shamir proof system [PS96] makes crucial use of rewinding. Using our black-box access, we let the malicious prover (or adversary) $\mathcal{A}$ forge a proof for a statement $x$ of its choice, and *we write the proof down for later reference*. We then rewind the adversary back to the point where it queried the particular commitment that it forged on, and this time, using our ability to reprogram the random oracle, we feed it a different random challenge. We now hope that the adversary will pick the same commitment for its forgery in the second run. The *forking lemma*, essentially a pigeonhole-type argument, notes that the adversary has only a polynomial amount of commitments to choose from, since it has to query the designated commitment to the random oracle in order to find the corresponding challenge. After two runs in polynomial-time, containing at most $q$ different queries each, the lemma says that we have a $1/q^2$ probability that $\mathcal{A}$ picks the same in both. Thus, with good probability we have made the adversary output a *different proof* for the *same statement* $x$ and the *same commitment com*. With the two proofs in hand, we may then use the special soundness property of the underlying sigma-protocol to obtain a valid witness $w$ for $x$.

In the quantum random-oracle model, we must allow for $\mathcal{A}$ to have superposition access to the random oracle $H$. Its final output state may therefore contain an (exponentially large) superposition of different commitments, as long as a good portion of them is accompanied by the correct response for the commitment-challenge pair given by $H$. This poses two major difficulties to the classical extraction procedure.

The first problem is that a measurement of the output of $\mathcal{A}$ after the first run may significantly disturb the internal state of the adversary. Running it again from the start may not be possible if the adversary depends on an initial quantum state. Even if we assume $\mathcal{A}$ to be unitary and allow the extractor access to its inverse (see Section 3.1), uncomputing the post-measurement state will lead to unpredictable behavior for $\mathcal{A}$ in the second run.

Supposing for a moment that we could somehow obtain the adversary's first response *and* make it run properly a second time, it still seems impossible to ensure that the same commitment is used in both runs. Remember that our QROM-adversary however can query and forge on exponentially many different commitments. Therefore, even if it did output the exact same state in both runs, the randomness inherent in the quantum measurement already prevents us from hoping to see the same commitment twice.

# 3 Previous results

Since the introduction of the quantum random-oracle model (QROM) [BDF+11], a number of papers have been published that deal with the extractability of the Fiat-Shamir proof system in the QROM. In this section, we highlight a few important papers.

## 3.1 The hardness of quantum rewinding

We noted is Section 2.2 that the classical proof method for the soundness of Fiat-Shamir relies on a technique called *rewinding*, and that rewinding in the quantum setting brings about a range of difficulties. The first positive result about quantum rewinding was introduced by John Watrous in [Wat09]. They used a *quantum rewinding lemma* to prove the zero-knowledge property of a couple specific interactive proof systems. However, Watrous' rewinding technique is sometimes referred to as *oblivious rewinding* (e.g. in [KMW17]) because while it can be used to backtrack the to-be-rewinded algorithm, no information can be saved between the different branches of the execution. In the Fiat-Shamir context, we may use Watrous' technique to rewind the adversary after it has output its first proof, but doing so discards any information about that proof. The technique therefore fails to be of any use in using the special soundness property of the underlying sigma-protocol.

### Unruh rewinding

A rewinding technique more suited to quantum proofs of knowledge/extractability was given by Dominique Unruh in [Unr12]. In the context of a proof of knowledge, he argued, it is natural to assume that the adversary is given by a unitary operation. As we described in Section 2.1, the idea of an 'extraction algorithm' is to capture the knowledge contained in the interaction between the prover and the verifier. The adversary could always purify itself to become unitary, which for the interaction observed by the verifier (or the extractor) would make no difference at all. Therefore, was may assume unitarity without loss of generality.

If we have black-box access to a unitary adversary, then it is not at all unreasonable to assume that we also have access to its inverse, since any unitary quantum circuit can very easily be run backwards. This allows Unruh to let his extractor perform rewinding quite similar to the classical technique; even though we cannot copy an intermediate state of the adversary to 'return' to it later, we can still go back to any point in its execution by simply uncomputing its final state. Moreover, this technique allows us to measure the final/intermediate output of the adversary and uncompute/continue the run of the adversary *after* the measurement while *keeping the measurement outcome.*

Unfortunately, not all problems are solved with Unruh's technique. Continuing the run of the adversary after a measurement may be possible in principle, but if the measurement disturbs the state of the adversary too much, its ability to continue its computation as normally may be harmed. Thus, after measuring the output of the first run, we no longer have a guarantee that the adversary is able to forge a proof in the second run.

For *interactive* proofs of knowledge, specifically the class named 'sigma-protocols' that we described in Section 1.4, Unruh was able to prove post-quantum security with the help of an extra assumption: If the scheme under consideration has the property 'perfect unique responses', then measuring its final output state does not disturb the state of the adversary, because there is only one possible outcome for the measurement (but still unknown to the extractor prior to the measurement). Hence, the extractor can uncompute the state of the adversary after the measurement to a previous point in the run, and the adversary will be able to forge a (different) proof in the second run.

For the Fiat-Shamir non-interactive proof system, this method does not suffice – even with the assumption of unique responses in place – as we explained in Section 2.2. Note that in the interactive setting the adversary is confined to just one commitment because it has to send its commitment to the *verifier* in the first round of the protocol. The verifier will *measure* the message so that it is essentially classical, as opposed to the Fiat-Shamir case were the adversary sends a (or multiple) message(s) to the *random oracle*, so that the message(s) may be quantum, and may contain a superposition of exponentially many commitments. Therefore the final output state may contain a superposition of exponentially many commitments, and thus the measurement *does* disturb the state of the adversary – even if there is only one (the unique) response to every commitment-challenge pair.

| Underlying sigma-protocol | | | Sig.-pr. used directly | | Fiat-Shamir | | Fischlin | |
|---|---|---|---|---|---|---|---|---|
| zero-knowledge | special soundness | strict soundness | PoK | proof | PoK | proof | PoK | proof |
| stat | perf | comp | attack[16] | stat[37] | attack[25] | ? | attack[28] | ? |
| stat | comp | comp | attack[20] | attack[20] | attack[26] | attack[26] | attack[29] | attack[29] |
| stat | perf | perf | stat[32] | stat[37] | ? | ? | ? | ? |

Figure 1: Table taken from [ARU14]. 'Strict soundness' is synonymous with 'unique responses'. 'PoK' stands for 'Proof of Knowledge', 'proof' for normal soundness of the protocol. Computational security means against a polynomially bounded adversary, statistically secure means against an unbounded adversary. Their caption: "Taxonomy of proofs of knowledge. For different combinations of security properties of the underlying sigma-protocol (statistical (stat)/perfect (perf)/computational (comp)), is there an attack in the quantum setting (relative to an oracle)? Or do we get a statistically/computationally secure proof/proof of knowledge (PoK)? The superscripts refer to theorem numbers in this paper ([ARU14], JWD) or to literature references. Note that in all cases, classically we have at least computational security." In this thesis we derive a positive answer for the bottom two question marks in the 'Fiat-Shamir' column.

#### Quantum attacks on classical proof systems

A paper by Ambainis et al. [ARU14] from 2014 shows that it is not simply a matter of failing proof techniques, but instead some protocols may be under actual quantum threat. In the paper, titled 'Quantum Attacks on Classical Proof Systems; The Hardness of Quantum Rewinding', they show that sigma-protocols with certain properties, and the Fiat-Shamir non-interactive proof systems that are based on them, are indeed completely broken by a quantum adversary (relative to a certain oracle). Their attacks provide evidence for the necessity of such an assumption as 'perfect unique responses' in the quantum setting, as they conclude themselves. However, even under suitable assumptions they leave open the question of the soundness of the Fiat-Shamir transformation. Figure 1 gives an overview of their results.

### 3.2 An impossibility result, or is it?

In 2013, [DFG13] claimed an impossibility result about proving the soundness of the Fiat-Shamir transform as a quantum proof of knowledge. They gave a meta-reduction, which uses any black-box extractor (i.e. an extractor that has only black-box oracle access to the adversary) for a Fiat-Shamir proof system to break the (active, i.e. the adversary or in this case the meta-reduction may choose $x$) honest-verifier zero-knowledge property of the underlying sigma-protocol. The conclusion is that no black-box extractor that proves the security of a Fiat-Shamir scheme based on a proper (meaning that it has active honest-verifier zero-knowledge) sigma-protocol can exist.

However, it seems that their argument silently assumes that quantum rewinding is not possible. Specifically, they write

> *"The quantum adversary here, however, queries the random oracle in a superposition. In this scenario, as we explained above, the extractor is not allowed to "read" the query of the adversary unless it makes the adversary stop. In other words, the extractor cannot measure the query and then keep running the adversary until a valid witness is output."*

While this is partially in accordance with what we wrote in Section 2.2, the difference between our viewpoints is this: We noted that in general a measurement will disturb the state of the adversary, so that *without further information*, we can no longer assume that the adversary will continue its run as normal. However, in Section 4 we develop tools that allow us to predict the behavior of the adversary even after its state has been disturbed by a measurement. With these tools in hand, the extractor *can* indeed "*measure the query and then keep running the adversary until a valid witness is output.*"

If quantum rewinding is possible in the Fiat-Shamir case, as we indeed demonstrate in Section 4, then the meta-reduction from [DFG13] is not valid. The reason is as follows: In its (active) attack on the honest-verifier zero-knowledge property, the meta-reduction interacts with an honest prover. It then uses the result of this interaction for its communication with

the black-box extractor. The crucial observation is, that if the black-box extractor would rewind the black-box (as our black-box extractor from Section 4 does), then it would rewind the meta-reduction and *a fortiori* rewind the honest prover. Rewinding the honest prover however is not allowed in an active attack on zero-knowledge, as the authors of [DFG13] write themselves (differently phrased) in their definition of active security.

From the assumption in the quote given above, it follows (but this is not stated explicitly in [DFG13]) that a black-box extractor for a Fiat-Shamir proof system does not rewind its black-box. Therefore, under their assumptions there is no issue with the meta-reduction, and the impossibility result goes through. The result is thus best explained as a confirmation that it is not possible to prove the quantum extractability of the Fiat-Shamir transformation without using some form of quantum rewinding.

## 3.3 Steps already taken by Unruh

We noted that Dominique Unruh has given a new formalism for quantum rewinding in [Unr12], which he used to prove the extractability of (interactive) sigma-protocols that have perfect unique responses. In [Unr17], Unruh extended his analysis of quantum proofs of knowledge by giving an extensive study of the non-interactive case, the Fiat-Shamir proof systems. His paper, and the formalisms introduced therein, have been of invaluable worth to the work in this thesis. Concretely, Unruh has

- Given a complete and detailed formalism for black-box oracle access that is well-equipped for the notion of extractability.
- Explored a range of possible definitions of extractability in the quantum case, each with an extensive argumentation of its (un)suitability.
- Proven the unforgeability of Fiat-Shamir signatures *under the assumption of extractability of the Fiat-Shamir proof system.*
- Proven the unforgeability of Fiat-Shamir signatures with the help of an extra feature called a 'dual-mode hard instance generator', which unfortunately degrades the efficiency of the signature scheme.

The last two items deserve further explanation. Theorem 25 from [Unr17] states that a Fiat-Shamir proof system that is extractable, is also simulation-sound extractable (see Section 1.5). Theorem 31 then states that a Fiat-Shamir signature scheme based on a proof system that is simulation-sound extractable and zero-knowledge, is existentially unforgeable. The two theorems combined shift the burden of the signature security to the extractability of the Fiat-Shamir proof system (and its zero-knowledge property, but Fiat-Shamir has already been proven zer-knowledge, while extractability is still open).

When a signature scheme uses a 'dual-mode hard instance generator', there exist real public keys and fake public keys, which however are indistinguishable for a computationally bounded adversary. This feature comes at the cost of having a less compact scheme, approximately three times less compact in the analysis of [KLS18]. Theorem 30 from [Unr17] proves that the feature is sufficient for existential unforgeability in the QROM, with no further conditions (i.e. not dependent on the extractability of the underlying Fiat-Shamir proof system).

# 4 The Fiat-Shamir proof system is extractable in the quantum random-oracle model

In this section we answer the challenge from Section 2.1: We give a black-box extractor for a quantum adversary in a Fiat-Shamir proof system. We define the notion of SP-extractability (statement preserving), which is closely related to, but weaker than the definition of extractability that Unruh gave in [Unr17]. Relative to this notion, we prove that the Fiat-Shamir proof system is extractable in the QROM.

The weakening consists in only requiring that properties of the to-be-proven (classical) statement $x$ are preserved across the extraction procedure. Such a requirement is necessary at all, because the $x$ output by the extractor may not be the same as the $x$ output by the adversary. Unruh's definition demands that on top of this the internal state of the adversary, possibly containing quantum data, is more or less unaffected by the extraction.

In Section 5, we show that SP-extractability is sufficient for the existential unforgeability of Fiat-Shamir type signatures, by adapting the proof that was given in [Unr17]. (Which requires Unruh's stronger, as of yet unfulfilled notion of extractability.) The stronger notion might still be needed[2] in proving the security of more advanced schemes (group signatures, identity-based signatures). For the basic signature case, our result shows that the inefficient (extra) security measure of dual-mode hard instance generators is unnecessary.

## 4.1 Proof idea

We described the problems involved in translating the classical proof of extractability to the quantum random-oracle model in Section 2.2. In the interactive setting, most of these problems do not occur. The adversary, be it quantum or not, must send a single commitment to the verifier in the first round. This can be seen as restricting the adversary to a single classical query, and thus provides us with a natural way to force a measurement *before* the oracle is queried, so that the final output of the adversary cannot contain a superposition of exponentially many commitments. Although some difficulties remain, [Unr12] showed that with the additional assumption of *perfect unique responses*, the interactive proof system is indeed extractable against a quantum adversary.

We prove that the Fiat-Shamir proof system is *SP-extractable* in the QROM by giving a reduction to the interactive case. We show that a polynomial amount of superposition queries does not give the adversary any (significant) advantage over a single classical query. Therefore, a subroutine $R$(eduction) of the extractor can use its black-box access to the Fiat-Shamir adversary to make the verifier from the underlying sigma-protocol accept. The 'canonical extractor' given in [Unr12] then uses black-box access to *the extractor subroutine* $R$ to compute a valid witness. Note that in order to use the canonical extractor, we still need the assumption of perfect unique responses, even though the reduction itself does not require it.

### How the reduction works

We work with a unitary quantum polynomial-time adversary $\mathcal{A}^H$ that runs from a fixed but arbitrary initial internal state. Its behavior up to the measurement of its final output state depends only on the oracle $H$, therefore it makes sense to speak of 'a run under $H$' to distinguish different runs of $\mathcal{A}$.

We start from the assumption that for random $H$, $\mathcal{A}^H$ makes exactly $q$ queries and has probability $acc$ of producing an output $(x, com, H(x||com), resp)$ such that $Q(x) = 1$ ($Q$ can be any predicate on $x$) and such that $V_{FS}(x, com||resp) = 1$. Note that it is non-standard to require $\mathcal{A}$ to output $ch = H(x||com)$, but we may do so without loss of generality at the cost of at most one extra query to $H$ for $\mathcal{A}$.

The reduction picks one of $\mathcal{A}$'s queries at random and measures it. The measurement collapses the state of the adversary, but we prove that the collapse does not impair its ability to find a correct response for the specific $y_0 = x'||com'$ that was obtained in the measurement. The reduction forwards $y_0$ to the the Sigma-verifier, and uses the reply $\Sigma(y_0) = ch'$ to reprogram the random oracle at $y_0$. We prove that with good probability, the final output of $\mathcal{A}$ is a triple $(y_0, \Sigma(y_0), resp)$ such that $V_{FS}^{H*\Sigma y_0}(x', com'||resp') = 1$, hence $V_\Sigma(x', com', resp') = 1$, and furthermore $Q(x') = 1$.

The explicit dependence on $H$ allows us to prove that for any set of output triples of the form $(y, H(y), z)$ that occurs with squared amplitude $p$ in the final output state, the $y$'s from that set must have been queried with squared amplitude roughly equal to $p/q$. We know that the set of *accepting* triples $(y, c, z)$ such that $y = x||com$ and $Q(x) = 1$ is of this form,

[2] Suggested by Dominique Unruh in a private correspondence.

and occurs with squared amplitude *acc* in the output state. We conclude that with good probability, $y_0 = x'||com'$ is of the kind that $\mathcal{A}$ can forge on (relative to the original oracle), and such that $Q(x')$ holds.

Suppose that $y_0$ occurs only in a single query. The state of the adversary directly after this query can be divided in two orthogonal parts. One, the $y_0$-part, is the 'knowledgeable' part that contains information about $H(y_0)$. The other is the rest of the state. These two parts evolve independently during the second stretch of $\mathcal{A}$'s run, because unitary computation preserves orthogonality. The correct response that appears in the final output state must come from the knowledgeable part, for the following simple reason: Not even the best basketball player in the world can shoot the right hoop, if he does not know which hoop is the right one – especially when there are exponentially many hoops to choose from. The halfway measurement collapses the state onto the knowledgeable part, which must be sufficient to compute a good response, by its independence from the non-knowledgeable part and the assumption that $\mathcal{A}$ can forge at all.

The collapse has a magnifying effect, zooming in on the $y_0$-part of the state. As a consequence, we have a good probability of finding $y_0$ again in the final measurement, but only if the following condition is satisfied: the magnitude of $y_0$ in the measured query must not be too large compared to the magnitude of $y_0$ in the (undisturbed) output state. A counting argument shows that this condition is satisfied by almost all $y$ that $\mathcal{A}$ forges on in a run under some specific $H$. If such $y$ further satisfy the property that a correct response occurs in the output with non-negligible magnitude relative to the total magnitude on $y$-triples, then we say that $y$ is *solved* under $H$.

### Contributing queries

The main difficulty of the reduction comes from the fact that $\mathcal{A}$ may query the same $y$ multiple times. Classically we assume that $\mathcal{A}$ makes only a single query per $y$, but in the quantum case this assumption is untenable. By the No-Cloning Theorem, quantum information can in general not be copied, and may thus be used up during $\mathcal{A}'s$ computation. Furthermore, algorithms like Grover's depend on querying a superposition of all inputs on every iteration. Not allowing $\mathcal{A}$ to query $y$ multiple times would significantly restrict the class of adversaries under consideration.

For any $y$, we make a distinction between queries that are *contributing for* $y$, and those that are not. We described how in the single $y$-query setting, the knowledgeable part must be sufficient to compute a valid response. With multiple queries that feature $y$, each of them creates a knowledgeable part that contains information about $H(y)$, but not each of these are necessarily used to compute a valid response for $(y, H(y))$. The ones that are, we call contributing for $y$.

We prove that there exists at least one contributing query for each $y$ that is solved under $H$. We also prove that the first of these must have a decent amount of magnitude on $y$ relative to the total query magnitude for $y$ across all queries. Therefore, conditioned on obtaining $y_0$ in the halfway measurement, we have a good probability that the random query that we picked for the measurement is in fact the first contributing one for $y_0$.

While our pick may be the first contributing query for $y_0$, we have no guarantee that there have not been (m)any non-contributing queries before this point. This causes a potential hazard, because it forces us to feed the adversary inconsistent oracle answers. Namely, we want to reprogram the oracle at $y_0$, but before the measurement we do not know the value of $y_0$. Before the measurement we will have to answer queries on $y_0$ with $H(y_0)$, after the measurement with $\Sigma(y_0)$ . The fact that all pre-measurement queries are not contributing for $y_0$ will help us prove that the adversary is (mostly) unaffected by this inconsistency. Concretely we prove that *if* $y_0$ is solved under $H * \Sigma y_0$ (i.e. in a run where we would hypothetically use the reprogrammed oracle from the start), then it is also solved in a run under $\Gamma_m$. Here $\Gamma_m$ is what we call an *oracle sequence*, that denotes for each query which oracle is used. The subscript $m$ signals that we use $H$ up to the $m$-th query (the one we measure), and $H * \Sigma y_0$ from and including the $m$-th query on. $\Gamma_m$ is a realistic sequence that we can actually implement, because it tells us to switch to $H * \Sigma y_0$ only after we have queried $y_0$ to the Sigma-verifier.

### Quantum Forking Lemma

One problem remains. In the above argument, we conditioned on $y_0$ being solved under $H * \Sigma y_0$. While we know that in an average run (i.e. a run under random $H$) most $y$ will be solved, this refers to being solved relative to $H$. In other words, there is a good chance that $\mathcal{A}$ can find a response $z$ that fits $y_0$ and $H(y_0)$, but can it also find a response for the challenge $\Sigma(y_0)$?

If we were to run $\mathcal{A}$ on $H * \Sigma y_0$ from the start, measuring a query at random would likely give some $y_0$ for which $\mathcal{A}$ can forge relative to $H * \Sigma y_0$. But that is not what we do. We simulate $\mathcal{A}$ under $\Gamma_m$, which up to the measurement point equals a run under $H$. Because $\mathcal{A}$ may query adaptively, the constitution of the $m$-th query – and hence the probability that $y_0$ is solved under $H * \Sigma y_0$ – may be very different in both runs. What we need is a lower bound on the probability that $y_0$ *taken from a run under* $H$ is solved under $H * \Sigma y_0$.

It turns out that the easiest way to prove such a lower bound, is by considering the probability that $y_0$ is solved under $H$ *as well as* $H * \Sigma y_0$. Our Quantum Forking Lemma states that this probability is polynomially related to the probability of being solved under $H$, for which we already had a good bound. By the result from the previous paragraph, $y_0$ is then also solved under $\Gamma_m$.

Bringing everything together, we conclude that since the $m$-th query (in a run under $\Gamma_m$) is contributing for $y_0$ (relative to $H * \Sigma y_0$-correct responses), our measurement does not significantly disturb the adversary, who will thus compute a good response for $(y_0, \Sigma(y_0))$ that we may use to make the Sigma-verifier accept. The canonical extractor for interactive Sigma-protocols will do the rest, providing us with the valid witness $w$ for $x'$ that we were after.

## 4.2 Preliminaries

We define some formal notation used throughout the proof of our main theorem. More notation will be introduced along the way as needed.

### Notation for modelling the adversary

The complete system of $\mathcal{A}$ consists of the registers $Y, C, Z, S_{\mathcal{A}}$ of size $\ell_x + \ell_{com}, \ell_{ch}, \ell_{resp}, \ell_{state}$ respectively (implicitly these sizes depend on $\eta$, which we choose to not always include as a subscript). All oracles implement a function with domain $\{0,1\}^{\ell_{in}}$ and codomain $\{0,1\}^{\ell_{out}}$ where $\ell_{in} = \ell_x + \ell_{com}$ and $\ell_{out} = \ell_{ch}$.

An *oracle sequence* $\Gamma$ is a string of length $q$, such that *in a run under* $\Gamma$, each $[\Gamma]_i$ denotes the oracle to be used in answering the $i$-th query. The output of $\mathcal{A}$ under $\Gamma$ is given by the state $U_{\mathcal{A}^\Gamma}\, \rho\, (U_{\mathcal{A}^\Gamma})^\dagger$ where $\rho$ is the initial state of $\mathcal{A}$ and $U_{\mathcal{A}^\Gamma}$ is defined as

$$U_{\mathcal{A}^\Gamma} := U_{q-1}\mathcal{O}_{[\Gamma]_{q-1}} \dots U_1 \mathcal{O}_{[\Gamma]_1} U_0 \mathcal{O}_{[\Gamma]_0}$$

with $q$ the number of queries made by $\mathcal{A}$. In our proof it will be convenient to use a different set of unitaries: We define for $0 \leq i < q$

$$U_i^\Gamma := U_{q-1}\mathcal{O}_{[\Gamma]_{q-1}} \dots U_{i+1}\mathcal{O}_{[\Gamma]_{i+1}} U_i \mathcal{O}_{[\Gamma]_i}$$

to model the computation of $\mathcal{A}^\Gamma$ from right before the $i$-th query until the end of the run, including the quantum operations implemented by the remaining oracle queries. Note that we assume that $\mathcal{A}$ starts the run by immediately querying the oracle (the zeroth query). This is without loss of generality since we allow the adversary to start from any pre-computed state.

The unitary $U_i^\Gamma$ acts on the state $|\phi_i^\Gamma\rangle$, which is the state $\mathcal{A}$'s complete system is in right before it makes the $i$-th query, if all previous queries have been answered according to $\Gamma$. For any $0 \leq i \leq q$ we then have $U_i^\Gamma |\phi_i^\Gamma\rangle = |\phi_q^\Gamma\rangle$, which denotes the final output state of a run under $\Gamma$. For technical reasons we include $q$ in the range of $i$, so that the previous statement implies that $U_q^\Gamma$ is the identity.

We use the unitary $U_{(i,k)}^\Gamma := \left(U_k^\Gamma\right)^\dagger U_i^\Gamma$ to denote the computation under $\Gamma$ from right before the $i$-th query until right before the $k$-th query. To move one query up, we use $V_i^\Gamma := U_{(i,i+1)}^\Gamma$.

Whenever the oracle sequence $\Gamma$ consists of a single oracle $H$ only, we write $H$ instead of $\Gamma$ in all definitions that make use of a superscript $\Gamma$.

When we reprogram an oracle $H$ on the single value $y$, where the new output at $y$ is equal to $\Theta(y)$, we write $H * \Theta y$ to denote the resulting oracle.

Unless specifically noted otherwise, a subscripted oracle sequence $\Gamma_i$, with $0 \leq i < q$, is defined as follows:

$$[\Gamma_i]_k := \begin{cases} H & \text{for } k < i \\ H * \Sigma y_0 & \text{otherwise.} \end{cases}$$

where $y_0$ is the outcome of the halfway measurement, and $H$ is the random oracle implemented by the reduction. In words, $\Gamma_i$ is the oracle sequence where we have replaced the reprogrammed oracle $H * \Sigma y_0$ by the original oracle $H$ for all queries up to (not including) the $i$-th one.

### Calculating success probabilities

For each $y \in \{0,1\}^{\ell_{in}}$ we define the projector $Y := |y\rangle\langle y| \otimes \mathbb{1}_C \otimes \mathbb{1}_Z \otimes \mathbb{1}_I$. For $y$ that satisfy $y = x||com$ such that $Q(x) = 1$, where $Q$ is any fixed but arbitrary predicate, we define

$$G_y^H := \sum_{z:V_{FS}^H(y,z)=1} |y\rangle\langle y| \otimes |H(y)\rangle\langle H(y)| \otimes |z\rangle\langle z| \otimes \mathbb{1}_I.$$

For $y = x||com$ such that $Q(x) = 0$, we define $G_y^H$ to be the projector that projects any vector onto the zero vector. For the specific $y_0$ that we obtain in the halfway measurement, we may use the simplified $G_0 := G_{y_0}^{H*\Sigma y_0}$, which equals

$$G_0 := G_{y_0}^{H*\Sigma y_0} = \sum_{z:V_{FS}^{H*\Sigma y_0}(y_0,z)=1} |y_0\rangle\langle y_0| \otimes |\Sigma(y_0)\rangle\langle\Sigma(y_0)| \otimes |z\rangle\langle z| \otimes \mathbb{1}_I$$

if $y_0 = x||com$ with $Q(x) = 1$, and equals the 'zero projector' if $Q(x) = 0$.

The $G$ projectors project onto 'the good part' of the output state, that contains accepting responses for the $y$ that satisfy the property $Q$. The (squared) length of the resulting state comes up in a lot of calculations, so we use a shorthand for it:

$$\alpha_y^\Gamma := \|G_y^{H'} |\phi_q^\Gamma\rangle\|_2^2$$

where $H' = [\Gamma]_{q-1}$ (i.e. we let the final oracle in $\Gamma$ determine what counts as a 'good' response).

Two other quantities play an important role throughout the proof. For $0 \leq i < q$ we define

$$\kappa_{y,i}^\Gamma := \frac{\alpha_y^\Gamma}{\|Y|\phi_i^\Gamma\rangle\|_2^2} \quad \text{if } \|Y|\phi_i^\Gamma\rangle\|_2^2 \neq 0 \qquad \text{and} \qquad \beta_y^\Gamma := \frac{\alpha_y^\Gamma}{\|Y|\phi_q^\Gamma\rangle\|_2^2} \quad \text{if } \|Y|\phi_q^\Gamma\rangle\|_2^2 \neq 0.$$

$\kappa_{y,i}^\Gamma$ guards the ratio between the good part of the output for $y$ and the magnitude of $y$ in a specific query $i$, which needs to be non-negligible in order to make use of the 'magnifying effect' as explained in the previous section. $\beta_y^\Gamma$ does the same for the size of the good part for $y$ relative to the total magnitude of output triples starting with $y$, where intuitively a non-negligible $\beta_y^\Gamma$ means that $\mathcal{A}^\Gamma$ has done significantly more than pure guessing in computing a correct response for $y$. To refer to the minimum ratio $\kappa_{y,i}^\Gamma$ across all queries, we further define

$$\kappa_y^\Gamma := \min_{0 \leq i < q} [\kappa_{y,i}^\Gamma].$$

Borrowing notation from [ABB+17], we write

$$\mathcal{Q}_{\mathcal{A}^{(\Gamma)}}(y) := \sum_{0 \leq i < q} \|Y|\phi_i^\Gamma\rangle\|_2^2$$

for the total query magnitude of $y$ across all queries in a run under $\Gamma$.

Relative to some predicate $Q$ and a specific choice of random oracle $H$, we define $acc_H$ to be the probability that $\mathcal{A}^H$ outputs a triple $(y, H(y), z)$ such that $y = x||com$, $Q(x) = 1$ and $V_{FS}^H(x||com, z) = 1$. Note that $\sum_{y \in \{0,1\}^{\ell_{in}}} \alpha_y^H = acc_H$. In expectation over all $H : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$, we write

$$\mathop{\mathbb{E}}_H [acc_H] := acc.$$

When considering a function $H$, we write $H \setminus y$ for the function that is equal to $H$, except that it leaves its value at $y$ undefined.

To indicate that a vector is not necessarily a unit vector, we use round kets, as follows:

$$|\phi) \qquad \text{(is not necessarily a unit vector)}$$

To increase the readability of the proofs, we often indicate the justification of an (in)equality by superscripting it with

- a number, to refer back to an earlier equation (40), lemma (L.9) or definition (D.4).
- $(\Delta)$ to indicate that we used the triangle-inequality, or (C-S) where we used the Cauchy-Schwarz inequality.
- $(\diamond)$ or $(*)$ to refer to statements from elsewhere *inside the same lemma.*

### Notation from [Unr17]

In [Unr17], Unruh introduced a formalism for oracle machines, i.e. quantum algorithms that may have access to one or more oracles and may also be passed as an oracle to another algorithm themselves. Since our definition of SP-extractability is an adaptation of Unruh's definition, we take over much of his notation. Notably: $Fun(\ell_{in}, \ell_{out})$ is the set of all functions from $\{0,1\}^{\ell_{in}}$ to $\{0,1\}^{\ell_{out}}$. $ass$ is a list of assignments, that records the reprogramming of the random oracle by the extractor. $E^{\mathcal{A}_\eta^{\mathbf{rew}}(S_\mathcal{A}),H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta})$ denotes the extractor, with *rewinding access* to the *pure oracle circuit* $\mathcal{A}$. Therefore, $E$ can simulate $\mathcal{A}$ by applying the unitary $\mathcal{A}$ or $\mathcal{A}^\dagger$, possibly conditioned on some part of its own state. For this it needs to know the amount of qubits $\mathcal{A}$ operates on, $\ell(\eta)$, and the *shape* of $\mathcal{A}$, which describes what oracles $\mathcal{A}$ expects for its queries. For a more in-depth treatment of the formalism, see [Unr17].

## 4.3 Comparison between Unruh's and our notion of extractability

The notion of extractability that we use in our proof, *statement-preserving extractability*, is weaker than the definition from [Unr17]. The stronger definition is as follows:

**Definition 1 ([Unr17])** *A non-interactive proof system $(P, V)$ for a relation $R$ is* ***extractable*** *iff there is a quantum polynomial-time oracle algorithm $E$ and a constant $d > 0$, such that for any polynomial-time family of pure oracle circuits $\mathcal{A}_\eta$ (with output $\ell_{\mathcal{A}_\eta}^{output} = \ell_\eta^x + \ell_\eta^{com} + \ell_\eta^{resp}$) there exists a polynomial $\ell \geq 0$ such that* ***for any polynomial-time family of projective measurement circuits $\Pi_\eta$*** *there exists a polynomial $p > 0$ and a negligible function $\mu$ such that for all $\eta$ and all $\ell_{\mathcal{A}_\eta}^{state}$-qubit density operators $\rho$, we have that:*

$$\Pr\left[(x,w) \in R \wedge ok_A = 1 : \mathbf{Extract}\right] \geq \frac{1}{p(\eta)} \Pr\left[ok_V = 1 \wedge ok_Q = 1 : \mathbf{Prove_{FS}}\right]^d - \mu(\eta)$$

*where* $\mathbf{Prove_{FS}}$ *is following game:*

$$\begin{aligned} & H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out}), \\ & S_\mathcal{A} \leftarrow \rho \\ *** \quad & x||\pi \leftarrow \mathcal{A}_\eta^H(S_\mathcal{A}), \\ & ok_V \leftarrow V_{FS}^H(1^\eta, x, \pi), \\ *** \quad & ok_A \leftarrow \Pi_{\eta,x||\pi}^H(S_A) \end{aligned}$$

*and* **Extract** *is the following game:*

$$\begin{aligned} & H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out}), \\ & S_\mathcal{A} \leftarrow \rho, \\ & (x, w, \pi, ass) \leftarrow E^{\mathcal{A}_\eta^{\mathbf{rew}}(S_\mathcal{A}),H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta}) \\ *** \quad & ok_A \leftarrow \Pi_{\eta,x||\pi}^{H(ass)}(S_A) \end{aligned}$$

*where $ass$ is an assignment-list and $H(ass)$ is the result of reprogramming $H$ according to $ass$.*

Closely related, but requiring that properties are preserved across the extraction procedure only on the to-be-proven statement $x$, instead of $x, \pi$ and $S_A$, is our new definition:

**Definition 2** *A non-interactive proof system $(P, V)$ for a relation $R$ is* ***SP-extractable*** *(statement preserving extractable) iff there is a quantum polynomial-time oracle algorithm $E$ and a constant $d > 0$, such that for any polynomial-time family of pure oracle circuits $\mathcal{A}_\eta$ (with output $\ell_{\mathcal{A}_\eta}^{output} = \ell_\eta^x + \ell_\eta^{com} + \ell_\eta^{ch} + \ell_\eta^{resp}$) there exists a polynomial $\ell \geq 0$ such that* ***for any classical predicate $Q$ *(possibly dependent on $\eta$)*** there exists a polynomial $p > 0$ and a negligible function $\mu$ such that for all $\eta$ and all $\ell_{\mathcal{A}_\eta}^{state}$-qubit density operators $\rho$, we have that:*

$$\Pr\left[(x,w) \in R \wedge ok_Q = 1 : \mathbf{Extract}\right] \geq \frac{1}{p(\eta)} \Pr\left[ok_V = 1 \wedge ok_Q = 1 : \mathbf{Prove_{FS}}\right]^d - \mu(\eta)$$

*where* $\mathbf{Prove_{FS}}$ *is following game:*

$$\begin{aligned} & H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out}), \\ & S_\mathcal{A} \leftarrow \rho \end{aligned}$$

$$
\begin{aligned}
*** \quad & (x, com, H(x||com), resp) \leftarrow \mathcal{A}_\eta^H(S_\mathcal{A}), \\
& \pi := com||resp, \\
& ok_V \leftarrow V_{FS}^H(1^\eta, x, \pi), \\
*** \quad & ok_Q \leftarrow Q(1^\eta, x)
\end{aligned}
$$

*and* **Extract** *is the following game:*

$$
\begin{aligned}
& H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out}), \\
& S_\mathcal{A} \leftarrow \rho, \\
& (x, w, \pi, ass) \leftarrow E^{\mathcal{A}_\eta^{\mathbf{rew}}(S_\mathcal{A}), H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta}) \\
*** \quad & ok_Q \leftarrow Q(1^\eta, x).
\end{aligned}
$$

### Some notable differences

The first thing to notice, is that our definition puts a seemingly stronger requirement on the output of $\mathcal{A}$ in the game $\mathbf{Prove_{FS}}$. We require the adversary to output $H(x||com)$, where $x||com$ is the instance it forged on. However, since we already assume $\mathcal{A}$ to make *any* polynomial amount of queries to $H$, the requirement does not restrict our class of adversaries. Namely, any adversary that would output a superposition of strings $(x, \pi) = (x, com, resp)$, can simply send its final state to the random oracle in order to obtain the output that we require. Therefore, this particular difference does not weaken our definition.

What does weaken our definition, is that we enlarge the class of allowed extractors compared to Unruh's extractability. As was noted before, the $x$ output by $\mathcal{A}$ in the game $\mathbf{Prove_{FS}}$ is not necessarily the same as the $x$ output by $E$ in the game **Extract**. There is not much that we can do about this, simply because we are considering two separate games and the output of $\mathcal{A}$ may be probabilistic. However, what we need to prevent is that $E$ outputs some $x$ that is not a 'hard instance'. Usually, we assume that it is hard to – without knowing a corresponding witness – forge on a particular *kind* of statement $x$. We want that, assuming we have an adversary that *can* forge on this kind, $E$ uses $\mathcal{A}$ to output a witness for a statement of the special kind.

We enforce the correct behavior by only accepting extractors that preserve any (classical) predicate on $x$. To be more precise, what we demand is that if $\mathcal{A}$ has some probability $p$ of forging on a statement $x$ that satisfies $Q(x) = 1$ for a particular predicate $Q$, the probability that $E$ outputs a witness for *some* $x$ that also satisfies $Q(x) = 1$ is polynomially related to $p$.

Unruh's definition restricts the class of acceptable extractors even further. It requires properties of the output $x, \pi$ *and* the internal state of the adversary (after execution) to be preserved by the extractor with good probability. Because the internal state of the adversary may contain quantum data, 'properties' here is formalized by polynomial-time measurement circuits (see Preliminaries, Section 4.2) instead of classical predicates.

### Consequences

Our definition preserves properties of $x$ across the extraction, Unruh's also preserves the state of the adversary. The consequence is this: Sometimes we assume that an adversary is able to forge in a specific situation, e.g. that it can forge a signature when it assumes the identity of some designated party in an identity based signature scheme. We then need the extractor to be *in the same situation* when it obtains a witness, simply because we assume that it is has hard to do so, and we are looking for a contradiction to our assumptions. With our weaker definition, the extractor might escape its responsibility of solving a hard problem, by being in *a different situation* than the adversary. In the previous example of the identity based signature scheme, it might obtain a witness *while assuming its own identity*, something that is not assumed to be hard at all. Therefore, our definition does not suffice to prove the security of such more advanced signature schemes. For basic signatures (which are widely used), our proof technique does work out.

## 4.4 Structural overview of the proof

The narrative of the proof is roughly as follows: **Lemmas 1 to 4** establish a connection between the success probability of $\mathcal{A}^H$ and the query magnitude of *solved* instances $y$ in a run under $H$. In other words, with a good forger $\mathcal{A}$, we have a good probability of finding in both our halfway and our final measurement the same instance $y_0$, that satisfies $Q$ and which $\mathcal{A}$ moreover has a good probability of successfully forging on. However, this still only applies to

forgeries with respect to the oracle $H$ (i.e $\mathcal{A}$ can find a response $z$ such that $V_{FS}^H(y_0, z) = 1$) *and* depends on measuring the right kind of query, namely one that is *contributing for* $y_0$.

Next, **Lemmas 5 to 9** together imply that *if* $y_0$ is solved under $H * \Sigma y_0$ (i.e. $\mathcal{A}$ would solve $y_0$ if we were to hypothetically run it on $H * \Sigma y_0$ from start to finish, which is impossible because at the start of our run, we do not know which $y_0$ we are going to find in the halfway measurement) then there must exist a query number $m$, such that the $m$-th query is contributing for $y_0$, and even if we answer all queries before $m$ (which may contain $y_0$ in their superposition) according to $H$, then still measuring both the $m$-th query and the final output state leads us to a forgery on $y_0$ relative to $H * \Sigma y_0$, i.e. a response $z$ such that $V_\Sigma(y_0, z) = 1$.

**Lemma 11** says that conditioned on measuring $y_0$, there is a good probability that we have measured the $m$-th query, where $m$ is as above.

The 'quantum forking lemma' **Lemma 12** says that for any $y_0$ that we find in a halfway measurement in a run under $H$, which for the outcome of this measurement is the same as a run under $\Gamma_m$, there is a good probability that the condition left open by Lemma's 5 to 9 is indeed fulfilled. That is, $y_0$ is likely solved under $H * \Sigma y_0$. Lemma's 11 and 12 both rely on the technical result of **Lemma 10**.

**Lemmas 9, 11 and 12** together prove the following: If we have a random oracle $H$, simulate $\mathcal{A}$ on $H$ up to a random query, measure that query to obtain $y_0$, reprogram the oracle at $y_0$ from $H(y_0)$ to $\Sigma(y_0)$ and continue the run of $\mathcal{A}$ on this new oracle, then a measurement of the final output state will give us $(x, com, ch, resp)$ such that $V_\Sigma(x, com, resp) = 1$ and $Q(x) = 1$ with good probability. Applying the result from [Unr12] shows that the Fiat-Shamir proof system is SP-extractable in the QROM.

Figure 2 shows the structure of the proof at one further level of detail.

## 4.5 Formal proof

**Definition 3** *We say that $y$ is solved under $H$ respectively $\Gamma$ if*

$$\beta_y^H \geq \frac{acc}{q} \quad \text{and} \quad \kappa_y^H \geq \frac{acc}{q^2} \qquad \text{respectively} \qquad \beta_y^\Gamma \geq \frac{acc}{2q^7} \quad \text{and} \quad \kappa_y^\Gamma \geq \frac{acc}{q^7}.$$

**Definition 4 (Contributing)** *We say that $U_i^{\Gamma_i}$ respectively $U_i^{\Gamma_{i+1}}$ is* contributing for $y_0$ *if*

$$r_i = \|G_0 U_i^{\Gamma_i} Y_0 |\phi_i^{\Gamma_i}\rangle\|_2 \geq \frac{\sqrt{\alpha_{y_0}^\Gamma}}{4q} \qquad \text{respectively} \qquad r_i' = \|G_0 U_i^{\Gamma_{i+1}} Y_0 |\phi_i^{\Gamma_{i+1}}\rangle\|_2 \geq \frac{\sqrt{\alpha_{y_0}^{\Gamma_{i+1}}}}{4q}.$$

**Theorem 1 (The Fiat-Shamir proof system is SP-extractable in the QROM)** *Let $\Sigma$ be a sigma-protocol with special soundness and perfect unique responses, for the relation $R_\eta$, and such that for every $x \in \mathsf{dom}(R)$ the size of the challenge space $\#C_{\eta x}$ is exponential in $\eta$. The Fiat-Shamir transformation $(P_{FS}, V_{FS})$ of this protocol is SP-extractable.*

*Proof.* We prove the existence of a polynomial-time oracle algorithm $E$ that satisfies Definition 2. We first define the intermediate game $\mathbf{R-Prove_\Sigma}$, which we show to be polynomially related to both games from the definition, proving the theorem.

$\mathbf{R-Prove_\Sigma}$ is the following game:

$$\begin{aligned}
&(x, com) \leftarrow R_0^{\mathcal{A}_\eta^{\mathbf{rew}}(S_\mathcal{A}),H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta}) \\
&ch \xleftarrow{\$} V_\Sigma(1^\eta, x) \\
&(resp, ass) \leftarrow R_1^{\mathcal{A}_\eta^{\mathbf{rew}}(S_\mathcal{A}),H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta}, ch) \\
&ok_V \leftarrow V_\Sigma(1^\eta, x, com, ch, resp) \\
&ok_Q \leftarrow Q(1^\eta, x)
\end{aligned}$$

where $R_{0,1}$ is a polynomial time oracle algorithm that does the following:

1. Randomly pick an integer $m'$ between 0 and $q-1$.
2. Run $\mathcal{A}(S_A)$ on the random oracle $H$ until right before the $m'$-th query is answered.
3. Measure $\mathcal{A}$'s query register, obtaining the outcome $y_0 = (x_0, com_0)$.
4. Start interaction with the verifier $V_\Sigma$ from the sigma-protocol. In the commitment phase, send $y_0$ to $V_\Sigma$. Receive a challenge $ch \in C$, define $\Sigma(y_0) := ch$.
5. Continue the run of $\mathcal{A}$. Flip a coin to decide whether to reprogram the oracle before or (right) after answering the $m'$-th query. Reprogram the oracle at $y_0$ to $\Sigma(y_0)$.

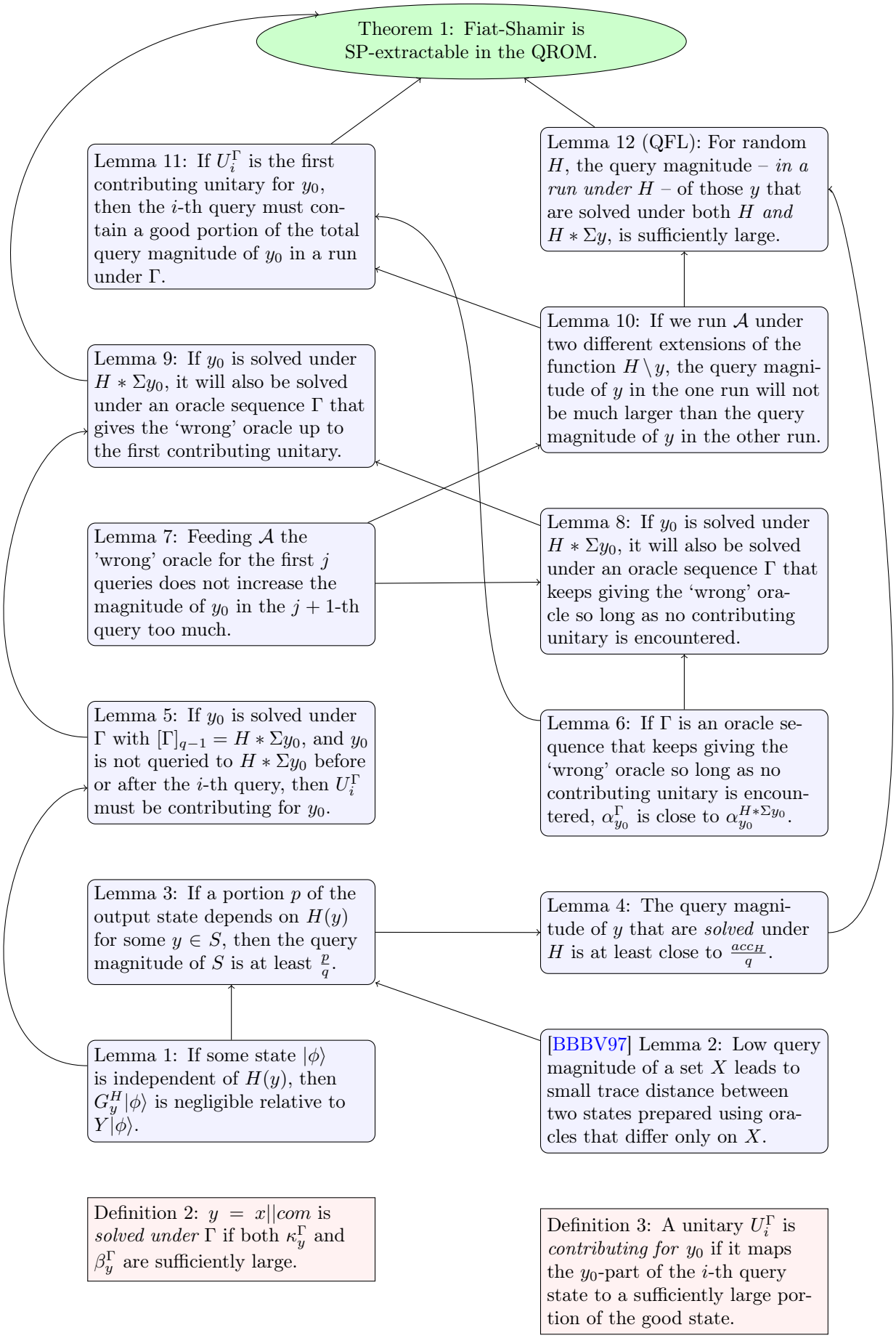


Figure 2: Structural overview of the proof

6. After the completion of the run, measure the output to obtain $(x, com, ch, resp)$. Return $resp$ to $V_\Sigma$. Output $ass$, the singleton list containing the assignment $(y_0 := \Sigma(y_0))$.

Let $acc = \Pr[ok_V = 1 \wedge ok_Q = 1 : \mathbf{Prove_{FS}}]$. We now show that if $q$ is the number of queries made by $\mathcal{A}$, then

$$\Pr[(x,w) \in R \wedge ok_Q = 1 : \mathbf{R - Prove_\Sigma}] \geq \frac{1}{32q^{35}} \cdot acc^4 - \mu(\eta).$$

In all of the following lemma's, assume any parameter that is not given explicitly to be as in the game $\mathbf{R - Prove_\Sigma}$:

**Lemma 1** *Let $S$ be any subset of $\{0,1\}^{\ell_{in}}$. If some state $|\phi\rangle$ is stochastically independent from all values $H(y)$ with $y \in S$ – the state and the values considered as random variables – then*

$$\Pr_{H\restriction_S}\left[\sum_{y\in S} \|G_y^H|\phi\rangle\|_2^2 \geq \mu(\eta) \cdot \sum_{y\in S} \|Y|\phi\rangle\|_2^2\right] \leq \mu(\eta) \tag{1}$$

*for some negligible function $\mu$. Note: Here it is silently assumed that $\Pr_{H(y)}[H(y) = c] = 2^{-\ell_{ch}}$ for any $c \in \{0,1\}^{\ell_{ch}}$ and $y \in S$ even if $S$ depends on $H$. This assumption remains to be proven. A complete proof will later appear on the arXiv.*

*Proof.* From the assumption $(*)$ of the independence of $|\phi\rangle$ it follows that

$$\begin{aligned}\mathop{\mathbb{E}}_{H\restriction_S}\left[\sum_{y\in S} \|G_y^H|\phi\rangle\|_2^2\right] &= \mathop{\mathbb{E}}_{H\restriction_S}\left[\sum_{y\in S} \langle\phi| G_y^H |\phi\rangle\right] \\ &\overset{(*)}{=} \sum_{y\in S} \langle\phi| \left(\mathop{\mathbb{E}}_{H(y)}\left[G_y^H\right]\right) |\phi\rangle \end{aligned} \tag{2}$$

where we for now zoom in on the term

$$\begin{aligned}\mathop{\mathbb{E}}_{H(y)}\left[G_y^H\right] &= \sum_{c\in\{0,1\}^{\ell_{ch}}} \sum_{z\in\{0,1\}^{\ell_{resp}}} \Pr_{H(y)}\left[V_{FS}^H(y,z) = 1 \wedge c = H(y)\right] \cdot |y\rangle\langle y| \otimes |c\rangle\langle c| \otimes |z\rangle\langle z| \otimes \mathbb{1}_I \\ &\leq \sum_{c\in\{0,1\}^{\ell_{ch}}} \Pr_{H(y)}[c = H(y)] \cdot |y\rangle\langle y| \otimes |c\rangle\langle c| \otimes \mathbb{1}_Z \otimes \mathbb{1}_I \\ &= 2^{-\ell_{ch}} \cdot |y\rangle\langle y| \otimes \mathbb{1}_C \otimes \mathbb{1}_Z \otimes \mathbb{1}_I \\ &=: \mu_1(\eta) \cdot Y \end{aligned} \tag{3}$$

where $\mu_1(\eta)$ is the inverse of the size of the challenge set for $\Sigma$, and thus a negligible function. Note that we assumed here that $\forall y \in S\ (y = x||com \Rightarrow Q(x) = 1)$, but this is without loss of generality since we are proving an upper bound. Plugging this expectation into Equation 2, we get

$$\mathop{\mathbb{E}}_{H\restriction_S}\left[\sum_{y\in S} \|G_y^H|\phi\rangle\|_2^2\right] \overset{(2,3)}{\leq} \mu_1(\eta) \cdot \sum_{y\in S} \langle\phi|Y|\phi\rangle = \mu_1(\eta) \cdot \sum_{y\in S} \|Y|\phi\rangle\|_2^2$$

Finally, applying Markov's inequality, we find

$$\Pr_{H\restriction_S}\left[\sum_{y\in S} \|G_y^H|\phi\rangle\|_2^2 \geq \sqrt{\mu_1(\eta)} \cdot \sum_{y\in S} \|Y|\phi\rangle\|_2^2\right] \leq \frac{\mu_1(\eta) \cdot \sum_{y\in S} \|Y|\phi\rangle\|_2^2}{\sqrt{\mu_1(\eta)} \cdot \sum_{y\in S} \|Y|\phi\rangle\|_2^2} = \sqrt{\mu_1(\eta)} =: \mu(\eta).$$

□

**Lemma 2 ([BBBV97], Theorem 3.3 - with notation taken from [ABB+17])** *The following holds for each $\epsilon > 0$. Suppose $\rho_H$ was prepared by some party $\mathcal{R}$ using $t$ queries to some hash oracle $H : X \to Y$. For each $1, \ldots, t$, let $\rho_i$ denote the state of $\mathcal{R}$'s system immediately prior to the $i$th query to $H(\cdot)$. For each hash input $x \in X$ let*

$$\mathcal{Q}_{\mathcal{R}_{(H)}}(x) \overset{def}{=} \sum_{i=1}^{t} \mathrm{Tr}(|x\rangle\langle x|\rho_i) \tag{4}$$

*denote the* query magnitude *of $x$ for $\mathcal{R}$'s interaction with the hash oracle $H(\cdot)$. Let furthermore $H'(\cdot)$ be a hash oracle that agrees with $H(\cdot)$ except on a subset $X' \subset X$ of inputs with the property that*

$$\sum_{x\in X'} \mathcal{Q}_{\mathcal{R}_{(H)}}(x) \leq \frac{\epsilon^2}{t}.$$

*Let $\rho_{H'}$ be the state prepared when $\mathcal{R}$ uses hash oracle $H'(\cdot)$ instead of $H(\cdot)$. It then holds that*

$$||\rho_H - \rho_{H'}||_{\mathrm{Tr}} \leq \epsilon.$$

**Lemma 3** *Let $S \subseteq \{0,1\}^{\ell_{in}}$ be some set for which it holds that*

$$\sum_{y \in S} \|G_y^H |\phi_q^H\rangle\|_2^2 = p. \tag{5}$$

*Then except with negligible probability, the query magnitude of $S$ has the following lower bound:*

$$\sum_{y \in S} \mathcal{Q}_{\mathcal{A}^{(H)}}(y) \geq \frac{p}{q} - \mu(\eta)$$

*for some negligible function $\mu(\eta)$.*

*Proof.* Let $H' : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$ be a function that agrees with $H$ everywhere except on $S$. Naturally, this means that $|\phi_q^{H'}\rangle \perp H(y)$ for all $y \in S$. From Lemma 1 we get that

$$\Pr_{H \restriction_S} \left[ \sum_{y \in S} \|G_y^H |\phi_q^{H'}\rangle\|_2^2 \geq \mu_1(\eta) \right] \leq \mu_1(\eta) \tag{6}$$

for some negligible function $\mu_1$. We use this bound to compute the overlap

$$\begin{aligned}
|\langle \phi_q^{H'} | \phi_q^H \rangle| &= \left| \langle \phi_q^{H'} | \left( \sum_{y \in S} G_y^H \right) |\phi_q^H\rangle + \langle \phi_q^{H'} | \left( \mathbb{1} - \sum_{y \in S} G_y^H \right) |\phi_q^H\rangle \right| \\
&\overset{(\Delta)}{\leq} \left| \langle \phi_q^{H'} | \left( \sum_{y \in S} G_y^H \right) |\phi_q^H\rangle \right| + \left| \langle \phi_q^{H'} | \left( \mathbb{1} - \sum_{y \in S} G_y^H \right) |\phi_q^H\rangle \right| \\
&\overset{\text{(C-S)}}{\leq} \sqrt{\langle \phi_q^{H'} | \left( \sum_{y \in S} G_y^H \right) |\phi_q^{H'}\rangle} + \sqrt{\langle \phi_q^H | \left( \mathbb{1} - \sum_{y \in S} G_y^H \right) |\phi_q^H\rangle} \\
&\overset{(5,6)}{\leq} \sqrt{\mu_1(\eta)} + \sqrt{1-p}
\end{aligned} \tag{7}$$

except with probability at most $\mu_1(\eta)$. This gives us a (conditional) bound on the trace distance

$$\begin{aligned}
\| |\phi_q^{H'}\rangle\langle\phi_q^{H'}| - |\phi_q^H\rangle\langle\phi_q^H| \|_{\text{Tr}} &= \sqrt{1 - |\langle \phi_q^{H'} | \phi_q^H \rangle|^2} \\
&\overset{(7)}{\geq} \sqrt{1 - (1 - p + \mu_2(\eta))} \\
&= \sqrt{p - \mu_2(\eta)}
\end{aligned}$$

We may now apply the contrapositive of Lemma 2 to conclude that

$$\sum_{y \in S} \mathcal{Q}_{\mathcal{A}^{(H)}}(y) \geq \frac{p - \mu_2(\eta)}{q} =: \frac{p}{q} - \mu(\eta)$$

except with probability at most $\mu_1(\eta)$. □

**Lemma 4** *Define $S_H$ to be the set that contains all $y$ that are solved under $H$. Then*

$$\mathop{\mathbb{E}}_H \left[ \sum_{y \in S_H} \mathcal{Q}_{\mathcal{A}^{(H)}}(y) \right] > acc \cdot \frac{q-2}{q^2} - \mu(\eta)$$

*where $\mu(\eta)$ is a negligible function.*

*Proof.* Define $\sum_y \alpha_y^H = acc_H$. Then

$$1 = \sum_y \|Y_0 |\phi_q^H\rangle\|_2^2 = \sum_y \frac{\alpha_y^H}{\beta_y^H} = acc_H \cdot \sum_y \frac{\alpha_y^H}{acc_H \cdot \beta_y^H} = \mathop{\mathbb{E}}_{y \sim \frac{\alpha_y^H}{acc_H}} \left[ \frac{acc_H}{\beta_y^H} \right]$$

so that by Markov's inequality we get that

$$\Pr_{y \sim \frac{\alpha_y^H}{acc_H}} \left[ \frac{acc_H}{\beta_y^H} \geq c \right] \leq \frac{1}{c}$$

for any constant $c$ that we may choose, which means that

$$\sum_{y:\frac{\beta_y^H}{acc_H}\leq\frac{1}{c}} \frac{\alpha_y^H}{acc_H} \quad = \sum_{y:\beta_y^H\leq\frac{acc_H}{c}} \frac{\alpha_y^H}{acc_H} \quad \leq \quad \frac{1}{c}.$$

It follows that if we consider only $y$ for which $\beta_y^\Gamma$ is smaller than $\frac{acc}{q}$, we may simply choose $c$ such that $\frac{acc_H}{c} = \frac{acc}{q}$ to find that

$$\sum_{y:\beta_y^H\leq\frac{acc}{q}} \alpha_y^H \quad \leq \quad \frac{acc_H}{c} \quad = \quad \frac{acc}{q}. \tag{8}$$

Let $D_H$ be the set which for any oracle $H$ contains all $y$ for which $\kappa_y^H$ is defined, but there exists some $0 \leq i < q$ such that $\kappa_{y,i}^H < \frac{acc}{q^2}$. Then

$$\begin{aligned}
\sum_{y\in D_H} \alpha_y^H \quad &= \quad \sum_{y\in D_H} \min_{0\leq i<q} \left[\kappa_{y,i}^H\right] \cdot \|Y|\phi_i^H\rangle\|_2^2 \\
&\leq \quad \sum_{y\in D_H} \min_{0\leq i<q} \left[\kappa_{y,i}^H\right] \cdot \mathcal{Q}_{\mathcal{A}_{(H)}}(y) \\
&\leq \quad \max_{y\in D_H} \min_{0\leq i<q} \left[\kappa_{y,i}^H\right] \cdot \sum_{y\in D_H} \mathcal{Q}_{\mathcal{A}_{(H)}}(y) \\
&\leq \quad \max_{y\in D_H} \min_{0\leq i<q} \left[\kappa_{y,i}^H\right] \cdot q \\
&< \quad \frac{acc}{q}
\end{aligned} \tag{9}$$

where we have used the trivial bound $q$ for the total query magnitude in the second to last step.

We can write the set of instances solved under $H$ as $S_H = \{y \in \overline{D_H} : \beta_y^H > \frac{acc}{q}\}$, so that Equations 8 and 9 together imply

$$\sum_{y\in S_H} \alpha_y^H \quad \overset{(8,9)}{>} \quad \sum_y \alpha_y^H - \frac{acc}{q} - \frac{acc}{q} \quad = \quad acc_H - 2\cdot\frac{acc}{q}.$$

Note that this holds for any $H$. In expectation over $H$ we thus get

$$\mathop{\mathbb{E}}_H \left[\sum_{y\in S_H} \alpha_y^H\right] > acc\cdot\frac{q-2}{q}.$$

Applying Lemma 3, we find

$$\mathop{\mathbb{E}}_H \left[\sum_{y\in S_H} \mathcal{Q}_{\mathcal{A}_{(H)}}(y)\right] \quad > \quad acc\cdot\frac{q-2}{q^2} - \mu(\eta).$$

□

**Observation 1 (Decomposition of $|\phi_q^\Gamma\rangle$)**

Remember that for $0 \leq i < q$, we have

$$[\Gamma_i]_k := \begin{cases} H & \text{for } k < i \\ H * \Sigma y_0 & \text{otherwise.} \end{cases}$$

Now since $H \restriction_{\{y_0\}\mathtt{C}} = H * \Sigma y_0 \restriction_{\{y_0\}\mathtt{C}}$, it follows that

$$U_i^{\Gamma_i}(\mathbb{1} - Y_0) = U_i^{\Gamma_{i+1}}(\mathbb{1} - Y_0).$$

Note furthermore that $|\phi_i^{\Gamma_i}\rangle = |\phi_i^{\Gamma_{i+1}}\rangle$ since $[\Gamma_i]_k = H = [\Gamma_{i+1}]_k$ for all $k < i$. We thus find

$$U_i^{\Gamma_i}(\mathbb{1} - Y_0)|\phi_i^{\Gamma_i}\rangle \quad = \quad U_i^{\Gamma_{i+1}}(\mathbb{1} - Y_0)|\phi_i^{\Gamma_{i+1}}\rangle$$

and also note that

$$U_i^{\Gamma_i}(\mathbb{1} - Y_0)|\phi_i^{\Gamma_i}\rangle \quad = \quad |\phi_q^{\Gamma_i}\rangle - U_i^{\Gamma_i}Y_0|\phi_i^{\Gamma_i}\rangle. \tag{10}$$

It will prove useful to have shorthands for the norms of certain states related to the above:

$$s_i := \|G_0 U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 \qquad r_i := \|G_0 U_i^{\Gamma_i} Y_0|\phi_i^{\Gamma_i}\rangle\|_2$$
$$t_i := \|Y_0 U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 \qquad r_i' := \|G_0 U_i^{\Gamma_{i+1}} Y_0|\phi_i^{\Gamma_{i+1}}\rangle\|_2$$

We derive

$$\begin{aligned} r_i &= \|G_0 U_i^{\Gamma} Y_0|\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(\Delta)}{\leq} \|G_0 U_i^{\Gamma_i}|\phi_i^{\Gamma_i}\rangle\|_2 + \|G_0 U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 \\ &= \sqrt{\alpha_{y_0}^{\Gamma_i}} + s_i \end{aligned} \tag{11}$$

and furthermore we may bound $s_i$ and $t_i$ by

$$\begin{aligned} s_i &= \|G_0\, U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 \\ &\leq \|Y_0\, U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 \qquad = \qquad t_i \\ &\overset{(10)}{=} \|Y_0|\phi_q^{\Gamma_i}\rangle - Y_0\, U_i^{\Gamma_i} Y_0|\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(\Delta)}{\leq} \|Y_0|\phi_q^{\Gamma_i}\rangle\|_2 + \|Y_0\, U_i^{\Gamma_i} Y_0|\phi_i^{\Gamma_i}\rangle\|_2 \\ &\leq \|Y_0|\phi_q^{\Gamma_i}\rangle\|_2 + \|Y_0|\phi_i^{\Gamma_i}\rangle\|_2 \\ &\leq \sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}}} + \sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}}} \end{aligned} \tag{12}$$

where the last inequality is (possibily) not an equality because we used $\kappa_{y_0}^{\Gamma_i}$ instead of $\kappa_{y_0,i}^{\Gamma_i}$.

In the next lemma, we find that in a scenario where $y_0$ is queried only once (say at query $i$), and if further we know that $y_0$ is solved under the oracle sequence $\Gamma_i$, then the $i$-th query must be contributing for $y_0$. Later, in Lemma 9, this will lead to the conclusion that also in the multi $y_0$-query setting at least one of the queries must be contributing.

**Lemma 5** *Fix $i$, and suppose that $y_0$ is solved under $\Gamma_i$. Suppose further that both $U_{i+1}^{\Gamma_i}$ and $|\phi_i^{\Gamma_i}\rangle$ are stochastically independent of $\Sigma(y_0)$ (considered as random variables), and that $\|Y_0|\phi_i^{\Gamma_i}\rangle\|_2 \neq 0$. Then except with negligible probability, $U_i^{\Gamma_i}$ is contributing for $y_0$.*

*Proof.* Taking notation from Observation 1, we find

$$\begin{aligned} \alpha_{y_0}^{\Gamma_i} &= \|G_0|\phi_q^{\Gamma_i}\rangle\|_2^2 \\ &= \|G_0\, U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle + G_0\, U_i^{\Gamma_i} Y_0|\phi_i^{\Gamma_i}\rangle\|_2^2 \\ &\overset{(\Delta)}{\leq} \left(\|G_0\, U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 + \|G_0\, U_i^{\Gamma_i} Y_0|\phi_i^{\Gamma_i}\rangle\|_2\right)^2 \\ &= s_i^2 + 2 s_i r_i + r_i^2 \end{aligned} \tag{13}$$

Using the bound from Equation 11, we get

$$\begin{aligned} \alpha_{y_0}^{\Gamma_i} &\overset{(13)}{\leq} s_i^2 + 2 s_i r_i + r_i^2 \\ &\overset{(11)}{\leq} s_i^2 + 2 s_i r_i + \left(\sqrt{\alpha_{y_0}^{\Gamma_i}} + s_i\right)\cdot r_i \\ &= \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + s_i^2 + 3 s_i r_i \\ &\overset{(11)}{\leq} \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + s_i^2 + 3 s_i \cdot \left(\sqrt{\alpha_{y_0}^{\Gamma_i}} + s_i\right) \\ &= \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + 4 s_i^2 + 3 s_i \cdot \sqrt{\alpha_{y_0}^{\Gamma_i}}. \end{aligned} \tag{14}$$

We argue that the state $U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle$ must be independent of the value $\Sigma(y_0)$. Note first that this state equals $U_{i+1}^{\Gamma_i} V_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle$. The state $|\phi_i^{\Gamma_i}\rangle$ is independent by assumption, and $V_i^{\Gamma_i}(\mathbb{1}-Y_0)$ cannot introduce any dependence to $\Sigma(y_0)$ because it does not 'query' $y_0$ and $H * \Sigma y_0(y)$ is independent of $\Sigma(y_0)$ for any $y \neq y_0$ by the independence of the random oracle. Thus $V_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle$ is independent of $\Sigma(y_0)$. By the assumption on $U_{i+1}^{\Gamma_i}$, this means that $U_{i+1}^{\Gamma_i} V_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle = U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle$ is independent.

We may therefore apply Lemma 1 to find

$$\Pr_{\Sigma(y_0)}\left[s_i^2 \geq \mu(\eta)\cdot t_i^2\right] \qquad =$$

$$\Pr_{\Sigma(y_0)}\left[\|G_0U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2^2 \geq \mu(\eta)\cdot\|Y_0U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2^2\right] \quad \leq \quad \mu(\eta). \quad (15)$$

We thus assume that $s_i^2 \leq \mu(\eta)\cdot t_i^2$. Combining this information with Equation 12, we obtain

$$\begin{aligned}
\alpha_{y_0}^{\Gamma_i} &\overset{(14)}{\leq} \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + 4s_i^2 + 3s_i\cdot\sqrt{\alpha_{y_0}^{\Gamma_i}} \\
&\overset{(15)}{\leq} \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + t_i^2\cdot\mu_1(\eta) + t_i\cdot\sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot\mu_2(\eta) \\
&\overset{(12)}{\leq} \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + \left(\sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}}} + \sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}}}\right)^2\cdot\mu_1(\eta) + \cdot\left(\sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}}} + \sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot\mu_2(\eta) \\
&= \sqrt{\alpha_{y_0}^{\Gamma_i}}\cdot r_i + \left(\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}} + \frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}} + 2\frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\beta_{y_0}^{\Gamma_i}\cdot\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_1(\eta) + \left(\frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\beta_{y_0}^{\Gamma_i}}} + \frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_2(\eta)
\end{aligned}$$

where (except with negligible probability) for each $i\in\{1,2\}$, $\mu_i$ is a negligible function. We will rewrite this in order to obtain the required bound on $r_i$:

$$\begin{aligned}
r_i &\geq \frac{\alpha_{y_0}^{\Gamma_i} - \left(\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}} + \frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}} + 2\frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\beta_{y_0}^{\Gamma_i}\cdot\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_1(\eta) - \left(\frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\beta_{y_0}^{\Gamma_i}}} + \frac{\alpha_{y_0}^{\Gamma_i}}{\sqrt{\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_2(\eta)}{\sqrt{\alpha_{y_0}^{\Gamma_i}}} \\
&= \sqrt{\alpha_{y_0}^{\Gamma_i}} - \left(\frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{\beta_{y_0}^{\Gamma_i}} + \frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{\kappa_{y_0}^{\Gamma_i}} + 2\frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{\sqrt{\beta_{y_0}^{\Gamma_i}\cdot\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_1(\eta) \\
&- \left(\sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\beta_{y_0}^{\Gamma_i}}} + \sqrt{\frac{\alpha_{y_0}^{\Gamma_i}}{\kappa_{y_0}^{\Gamma_i}}}\right)\cdot\mu_2(\eta)
\end{aligned}$$

which is easily seen to be bigger than $\frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{4q}$ if both $\beta_{y_0}^{\Gamma_i}$ and $\kappa_{y_0}^{\Gamma_i}$ are non-negligible. Since we assumed that $y_0$ is solved under $\Gamma_i$, this condition is indeed satisfied. □

If $\mathcal{A}$ can forge on $y_0$ when given the oracle $H*\Sigma y_0$ from the start, it can still do that when we feed it the wrong oracle (i.e. $H$) in the first half of its run, but only as long as no contributing unitaries are encountered along the way. This is what the next lemma is concerned with.

Even if $\mathcal{A}$ does query $y_0$ in the first part of its run, the inconsistent answers will not cause it to defect on $y_0$. The secret to this surprising result is an offensive strategy from our side: As soon as $\mathcal{A}$ threatens to 'undo' (i.e. counter with opposite phase amplitude) any good computation that comes from the consistent part of its state, we stop the run and declare that this is the query that we want to perform our halfway measurement on. This measurement then throws away the amplitude in the final output state that was being countered, leaving only the opposite phase amplitude, so that we still have a good chance of measuring the correct response at the end of the run.

In reality we do not sense and 'declare' that this is the query we want to measure, we pick a random one instead. This comes down to the same thing because at the current $i$, the unitary $U_i^{\Gamma_{i+1}}$ must be contributing for $y_0$ (namely, it contributes 'negative' amplitude). By Lemma 11 this means that our random pick has a good chance of landing on query $i$.

**Lemma 6** *Suppose that for each $i<j$ we have that both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing for $y_0$. Then*

$$\alpha_{y_0}^{\Gamma_j} \quad \geq \quad (1-\frac{j}{q})\cdot\alpha_{y_0}^{\Gamma_0}.$$

*Proof.* Assume (to our disadvantage) that for each $i<j$ it holds that $\alpha_{y_0}^{\Gamma_{i+1}} \leq \alpha_{y_0}^{\Gamma_i}$. We then have

$$\begin{aligned}
\alpha_{y_0}^{\Gamma_{i+1}} &= \|G_0|\phi_q^{\Gamma_{i+1}}\rangle\|_2^2 \\
&\overset{(O.1)}{=} \|G_0U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle + G_0U_i^{\Gamma_{i+1}}Y_0|\phi_i^{\Gamma_i}\rangle\|_2^2 \\
&\overset{(\Delta)}{\geq} \left(\|G_0U_i^{\Gamma_i}(\mathbb{1}-Y_0)|\phi_i^{\Gamma_i}\rangle\|_2 + \|G_0U_i^{\Gamma_{i+1}}Y_0|\phi_i^{\Gamma_i}\rangle\|_2\right)^2 \\
&\overset{(O.1)}{=} s_i^2 - 2s_ir_i' + (r_i')^2 \qquad (16)
\end{aligned}$$

and furthermore

$$\begin{aligned} s_i &\overset{(O.1)}{=} \|G_0|\phi_q^{\Gamma_i}\rangle - G_0 U_i^{\Gamma_i} Y_0 |\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(\Delta)}{\geq} \|G_0|\phi_q^{\Gamma_i}\rangle\|_2 - \|G_0 U_i^{\Gamma_i} Y_0 |\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(O.1)}{=} \sqrt{\alpha_{y_0}^{\Gamma_i}} - r_i \end{aligned} \tag{17}$$

$$\begin{aligned} s_i &\overset{(O.1)}{=} \|G_0|\phi_q^{\Gamma_i}\rangle - G_0 U_i^{\Gamma_i} Y_0 |\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(\Delta)}{\leq} \|G_0|\phi_q^{\Gamma_i}\rangle\|_2 + \|G_0 U_i^{\Gamma_i} Y_0 |\phi_i^{\Gamma_i}\rangle\|_2 \\ &\overset{(O.1)}{=} \sqrt{\alpha_{y_0}^{\Gamma_i}} + r_i. \end{aligned} \tag{18}$$

We then find

$$\begin{aligned} \alpha_{y_0}^{\Gamma_{i+1}} &\overset{(16,17,18)}{\geq} (\sqrt{\alpha_{y_0}^{\Gamma_i}} - r_i)^2 - 2(\sqrt{\alpha_{y_0}^{\Gamma_i}} + r_i)\cdot r_i' + (r_i')^2 \\ &= \alpha_{y_0}^{\Gamma_i} - 2r_i\sqrt{\alpha_{y_0}^{\Gamma_i}} + r_i^2 - 2r_i'\sqrt{\alpha_{y_0}^{\Gamma_i}} - 2r_i' r_i + (r_i')^2 \\ &\overset{(*)}{\geq} \alpha_{y_0}^{\Gamma_i} - 2r_i\sqrt{\alpha_{y_0}^{\Gamma_i}} - 2r_i'\sqrt{\alpha_{y_0}^{\Gamma_i}} \\ &\overset{(\diamond)}{\geq} \alpha_{y_0}^{\Gamma_i} - \frac{\alpha_{y_0}^{\Gamma_i}}{q} \end{aligned} \tag{19}$$

where we have used $(*)$ that $r_i^2 - 2r_i' r_i + (r_i')^2 = (r_i^2 - r_i')^2 \geq 0$ and $(\diamond)$ the fact that both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing, which by Definition 4 means that

$$r_i < \frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{4q} \qquad \text{and} \qquad r_i' < \frac{\sqrt{\alpha_{y_0}^{\Gamma_{i+1}}}}{4q} \leq \frac{\sqrt{\alpha_{y_0}^{\Gamma_i}}}{4q}.$$

We thus have that for each $i < j$, $\alpha_{y_0}^{\Gamma_{i+1}} \geq \alpha_{y_0}^{\Gamma_i} - \frac{\alpha_{y_0}^{\Gamma_i}}{q}$. It follows that

$$\alpha_{y_0}^{\Gamma_j} \qquad \geq \qquad (1 - \frac{1}{q})^j \cdot \alpha_{y_0}^{\Gamma_0} \qquad \geq \qquad (1 - \frac{j}{q}) \cdot \alpha_{y_0}^{\Gamma_0}.$$

□

$\mathcal{A}$ may query adaptively. This means that feeding it a different oracle in the first part of the run not only influences $\alpha_{y_0}^{\Gamma}$, but also the composition of all queries that come after the oracle swap. This could be problematic because of our dependence on the ratios $\beta_{y_0}^{\Gamma}$ and $\kappa_{y_0}^{\Gamma}$.

Lemma 7 shows that the specific element $y_0$ that separates the two oracles in the oracle sequence $\Gamma_j$, cannot grow too much compared to the straight oracle run.

**Lemma 7** *Let $0 \leq j, k \leq q$. Then*

$$\|Y_0|\phi_k^{\Gamma_j}\rangle\|_2^2 \leq \left(\|Y_0|\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n+1)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2.$$

*Proof.* We will use the fact $(*)$ that $V_i^{\Gamma_j}(\mathbb{1} - Y_0) = V_i^{\Gamma_0}(\mathbb{1} - Y_0)$ for any $i$, and $(\diamond)$ that $V_i^{\Gamma_0} = V_i^{\Gamma_j}$ for $i \geq j$. We assume $k \geq j$, but note that if $k < j$ we have $|\phi_k^{\Gamma_j}\rangle = |\phi_k^{\Gamma_k}\rangle$, so in this case we can simply set $j$ to be equal to $k$.

We define the operator

$$\Lambda_j^{\Gamma} := \sum_{x\in\{0,1\}^j\setminus\{0^j\}} \left(\prod_{i=j-1}^{0} V_i^{\Gamma}(Y_0)^{x_i}(\mathbb{1} - Y_0)^{1-x_i}\right) \tag{20}$$

Any state $|\phi_j^{\Gamma}\rangle$ may then be split up as follows:

$$|\phi_j^{\Gamma}\rangle \qquad = \qquad \Lambda_j^{\Gamma}|\phi_0^{\Gamma_0}\rangle + \prod_{i=j-1}^{0} V_i^{\Gamma}(\mathbb{1} - Y_0)|\phi_0^{\Gamma_0}\rangle \qquad =: \qquad \Lambda_j^{\Gamma}|\phi_0^{\Gamma_0}\rangle + |\phi_j^{\Gamma_*}\rangle. \tag{21}$$

Using this split we derive

$$\begin{aligned} \|Y_0|\phi_k^{\Gamma_j}\rangle\|_2^2 &= \|Y_0\, U_{(j,k)}^{\Gamma_j}|\phi_j^{\Gamma_j}\rangle\|_2^2 \\ &\overset{(\diamond)}{=} \|Y_0\, U_{(j,k)}^{\Gamma_0}|\phi_j^{\Gamma_j}\rangle\|_2^2 \\ &\overset{(21)}{=} \|Y_0\, U_{(j,k)}^{\Gamma_0}\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle + Y_0\, U_{(j,k)}^{\Gamma_0}|\phi_j^{\Gamma_*}\rangle\|_2^2 \\ &\overset{(\Delta)}{\leq} \left(\|Y_0\, U_{(j,k)}^{\Gamma_0}\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle\|_2 + \|Y_0\, U_{(j,k)}^{\Gamma_0}|\phi_j^{\Gamma_*}\rangle\|_2\right)^2 \\ &\leq \left(\|\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle\|_2 + \|Y_0\, U_{(j,k)}^{\Gamma_0}|\phi_j^{\Gamma_*}\rangle\|_2\right)^2 \end{aligned} \tag{22}$$

We first analyze the term $\|Y_0\, U^{\Gamma_0}_{(j,k)}|\phi^{\Gamma_*}_j\rangle\|_2$:

$$\begin{aligned}\|Y_0\, U^{\Gamma_0}_{(j,k)}|\phi^{\Gamma_*}_j\rangle\|_2 \quad &= \quad \|Y_0\, U^{\Gamma_0}_{(j,k)} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\ &\overset{(\Delta)}{\leq} \quad \|Y_0\, U^{\Gamma_0}_{(j,k)} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\ &+ \quad \|Y_0\, U^{\Gamma_0}_{(j,k)} \sum_{x\in\{0,1\}^j\setminus\{0^j\}} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_0^{\Gamma_0}\rangle\|_2. \end{aligned} \tag{23}$$

We compute the sum inside the last term separately. Because it has recursive structure, we consider it in a generalized form, where the original term is given by the case $n = 0$. This generalized form then rewrites to

$$\begin{aligned}\sum_{x\in\{0,1\}^{j-n}\setminus\{0^{j-n}\}} \left(\prod_{i=j-1}^{n} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_n^{\Gamma_0}\rangle \quad &= \\ \sum_{x\in\{0,1\}^{j-n-1}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \\ + \sum_{x\in\{0,1\}^{j-n-1}\setminus\{0^{j-n-1}\}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0}(\mathbb{1}-Y_0)|\phi_n^{\Gamma_0}\rangle \\ =: \quad |\psi_a^n) + |\psi_b^n) \end{aligned} \tag{24}$$

of which we may rewrite $|\psi_a^n)$ as

$$\begin{aligned}|\psi_a^n) \quad &= \quad \sum_{x\in\{0,1\}^{j-n-1}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \quad = \\ &\left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \quad = \quad U^{\Gamma_0}_{(n,j)} Y_0|\phi_n^{\Gamma_0}\rangle. \end{aligned} \tag{25}$$

The recursive structure lies within the other term. We have

$$\begin{aligned}|\psi_b^n) \quad &= \quad \sum_{x\in\{0,1\}^{j-n-1}\setminus\{0^{j-n-1}\}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0}(\mathbb{1}-Y_0)|\phi_n^{\Gamma_0}\rangle \\ &= \quad \sum_{x\in\{0,1\}^{j-n-1}\setminus\{0^{j-n-1}\}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_{n+1}^{\Gamma_0}\rangle \\ &- \quad \sum_{x\in\{0,1\}^{j-n-1}\setminus\{0^{j-n-1}\}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \\ &=: \quad |\psi_c^{n+1}) - |\psi_d^n). \end{aligned} \tag{26}$$

Note that in this formalism, our original term from (23) corresponds to $|\psi_c^0)$. We may furthermore write $|\psi_d^n)$ as

$$\begin{aligned}|\psi_d^n) \quad &= \quad \sum_{x\in\{0,1\}^{j-n-1}\setminus 0^{j-n-1}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \\ &= \quad \sum_{x\in\{0,1\}^{j-n-1}} \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \\ &- \quad \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \\ &= \quad \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle - \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right) V_n^{\Gamma_0} Y_0|\phi_n^{\Gamma_0}\rangle \end{aligned}$$

$$
\begin{aligned}
&= U^{\Gamma_0}_{(n,j)} Y_0 |\phi^{\Gamma_0}_n\rangle - \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right) V_n^{\Gamma_0} Y_0 |\phi_n^{\Gamma_0}\rangle \\
&= |\psi_a^n) - \left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right) V_n^{\Gamma_0} Y_0 |\phi_n^{\Gamma_0}\rangle \\
&=: |\psi_a^n) - |\psi_f^n). \qquad (27)
\end{aligned}
$$

Summarizing, we see that

$$
\begin{aligned}
|\psi_c^n) &= |\psi_a^n) + |\psi_b^n) \\
&= |\psi_a^n) + |\psi_c^{n+1}) - |\psi_d^n) \\
&= |\psi_a^n) + |\psi_c^{n+1}) - \big(|\psi_a^n) - |\psi_f^n)\big) \\
&= |\psi_c^{n+1}) + |\psi_f^n). \qquad (28)
\end{aligned}
$$

This recursive structure holds through up to and including the case $n = j-2$. At $n = j-1$ the sum inside $|\psi_c^n)$ dissolves, and we are left with

$$
|\psi_c^{j-1}) = V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle.
$$

It follows that

$$
|\psi_c^0) = \sum_{n=0}^{j-2} |\psi_f^n) + V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle. \qquad (29)
$$

Returning to Equation 23, we substitute according to (29):

$$
\begin{aligned}
\|Y_0\, U^{\Gamma_0}_{(j,k)} |\phi_j^{\Gamma_*}\rangle\|_2 &\overset{(23)}{\leq} \|Y_0\, U^{\Gamma_0}_{(j,k)} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}\right) |\phi_0^{\Gamma_0}\rangle\|_2 \\
&+ \|Y_0\, U^{\Gamma_0}_{(j,k)} \sum_{x\in\{0,1\}^j\setminus\{0^j\}} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) |\phi_0^{\Gamma_0}\rangle\|_2 \\
&\overset{(29)}{=} \|Y_0\, U^{\Gamma_0}_{(j,k)} \left(\prod_{i=j-1}^{0} V_i^{\Gamma_0}\right) |\phi_0^{\Gamma_0}\rangle\|_2 + \|Y_0\, U^{\Gamma_0}_{(j,k)} \left(\sum_{n=0}^{j-2} |\psi_f^n) + V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle\right)\|_2 \\
&\leq \|Y_0 |\phi_k^{\Gamma_0}\rangle\|_2 + \|\sum_{n=0}^{j-2} |\psi_f^n) + V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle\|_2 \\
&\overset{(\Delta)}{\leq} \|Y_0 |\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-2} \||\psi_f^n)\|_2 + \|V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle\|_2 \\
&\overset{(27)}{=} \|Y_0 |\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-2} \|\left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right) V_n^{\Gamma_0} Y_0 |\phi_n^{\Gamma_0}\rangle\|_2 + \|V_{j-1}^{\Gamma_0} Y_0 |\phi_{j-1}^{\Gamma_0}\rangle\|_2 \\
&\leq \|Y_0 |\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-2} \|Y_0 |\phi_n^{\Gamma_0}\rangle\|_2 + \|Y_0 |\phi_{j-1}^{\Gamma_0}\rangle\|_2 \\
&= \|Y_0 |\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1} \|Y_0 |\phi_n^{\Gamma_0}\rangle\|_2. \qquad (30)
\end{aligned}
$$

Turning now to the other term in Equation 22, we see that we can actually relate it to the bound that we just derived. Remember that

$$
\|\Lambda_j^{\Gamma_j} |\phi_0^{\Gamma_0}\rangle\|_2 = \|\sum_{x\in\{0,1\}^j\setminus\{0^j\}} \left(\prod_{i=j-1}^{0} V_i^{\Gamma}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) |\phi_0^{\Gamma_0}\rangle\|_2. \qquad (31)
$$

For every $x \in \{0,1\}^j \setminus \{0^j\}$ there is at least one $i$ such that $x_i = 1$. We may thus regroup the sum into subsets that contain all terms with exactly $m$ leading zeros. We define for $0 \leq m < j$:

$$
X_m := \{x \;:\; x_m = 1 \wedge i < m \to x_i = 0\} \qquad (32)
$$

For any $x \in X_m$ we then have

$$
\left(\prod_{i=m-1}^{0} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right) |\phi_0^{\Gamma_0}\rangle \qquad = \qquad \left(\prod_{i=m-1}^{0} V_i^{\Gamma_j}(\mathbb{1}-Y_0)\right) |\phi_0^{\Gamma_0}\rangle
$$

$$\overset{(*)}{=} \quad \left(\prod_{i=m-1}^{0} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)|\phi_0^{\Gamma_0}\rangle$$
$$= \quad |\phi_m^{\Gamma_*}\rangle. \tag{33}$$

We may thus rewrite Equation 31 as

$$\begin{aligned}
\|\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle\|_2 &= \|\sum_{m=0}^{j-1}\sum_{x\in X_m}\left(\prod_{i=j-1}^{0} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\
&= \|\sum_{m=0}^{j-1}\sum_{x\in X_m}\left(\prod_{i=j-1}^{m} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)\left(\prod_{i=m-1}^{0} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\
&\overset{(33)}{=} \|\sum_{m=0}^{j-1}\sum_{x\in X_m}\left(\prod_{i=j-1}^{m} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_m^{\Gamma_*})\|_2 \\
&\overset{(32)}{=} \|\sum_{m=0}^{j-1}\sum_{x\in X_m}\left(\prod_{i=j-1}^{m+1} V_i^{\Gamma_j}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)Y_0|\phi_m^{\Gamma_m})\|_2 \\
&= \|\sum_{m=0}^{j-1}\left(\prod_{i=j-1}^{m+1} V_i^{\Gamma_j}\right)V_m^{\Gamma_j}Y_0|\phi_m^{\Gamma_*})\|_2 \\
&\overset{(\Delta)}{\leq} \sum_{m=0}^{j-1}\|U_{(m,j)}^{\Gamma_j}Y_0|\phi_m^{\Gamma_*})\|_2 \\
&= \sum_{m=0}^{j-1}\|Y_0|\phi_m^{\Gamma_*})\|_2.
\end{aligned} \tag{34}$$

Zooming in on the term $\|Y_0\phi_m^{\Gamma_*}\|_2$ we find

$$\begin{aligned}
\|Y_0|\phi_m^{\Gamma_*})\|_2 \quad &= \quad \|Y_0\left(\prod_{i=m-1}^{0} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\
(\Delta)\leq \quad &\|Y_0\left(\prod_{i=m-1}^{0} V_i^{\Gamma_0}\right)|\phi_0^{\Gamma_0}\rangle\|_2 \\
+ \quad &\|Y_0\sum_{x\in\{0,1\}^m\setminus\{0^m\}}\left(\prod_{i=m-1}^{0} V_i^{\Gamma_0}(Y_0)^{x_i}(\mathbb{1}-Y_0)^{1-x_i}\right)|\phi_0^{\Gamma_0}\rangle\|_2.
\end{aligned} \tag{35}$$

where the sum inside the last equation looks very familiar to us; it is equal to $|\psi_c^0\rangle$ except that we have substituted $m$ for $j$. The substitution does not effect the derivations from 24 to 29 at all, so that we may conclude

$$[m/j] \text{ in } \left[|\psi_c^0\rangle\right] \quad \overset{(29)}{=} \quad [m/j] \text{ in } \left[\sum_{n=0}^{j-2}|\psi_f^n) + V_{j-1}^{\Gamma_0}Y_0|\phi_{j-1}^{\Gamma_0}\rangle\right]. \tag{36}$$

Therefore, Equation 35 becomes

$$\begin{aligned}
\|Y_0|\phi_m^{\Gamma_*}\rangle\|_2 \quad &\overset{(35,36)}{\leq} \quad \|Y_0\left(\prod_{i=m-1}^{0} V_i^{\Gamma_0}\right)|\phi_0^{\Gamma_0}\rangle\|_2 + [m/j] \text{ in } \left[\|Y_0\left(\sum_{n=0}^{j-2}|\psi_f^n) + V_{j-1}^{\Gamma_0}Y_0|\phi_{j-1}^{\Gamma_0}\rangle\right)\|_2\right] \\
&\leq \quad \|Y_0|\phi_m^{\Gamma_0}\rangle\|_2 + [m/j] \text{ in } \left[\|\sum_{n=0}^{j-2}|\psi_f^n) + V_{j-1}^{\Gamma_0}Y_0|\phi_{j-1}^{\Gamma_0}\rangle\|_2\right] \\
&= \quad \|Y_0|\phi_m^{\Gamma_0}\rangle\|_2 \\
&+ \quad [m/j] \text{ in } \left[\|\sum_{n=0}^{j-2}\left(\left(\prod_{i=j-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)V_n^{\Gamma_0}Y_0|\phi_n^{\Gamma_0}\rangle\right) + V_{j-1}^{\Gamma_0}Y_0|\phi_{j-1}^{\Gamma_0}\rangle\|_2\right] \\
&= \quad \|Y_0|\phi_m^{\Gamma_0}\rangle\|_2 + \|\sum_{n=0}^{m-2}\left(\left(\prod_{i=m-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)V_n^{\Gamma_0}Y_0|\phi_n^{\Gamma_0}\rangle\right) + V_{m-1}^{\Gamma_0}Y_0|\phi_{m-1}^{\Gamma_0}\rangle\|_2 \\
&\overset{(\Delta)}{\leq} \quad \|Y_0|\phi_m^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{m-2}\|\left(\prod_{i=m-1}^{n+1} V_i^{\Gamma_0}(\mathbb{1}-Y_0)\right)V_n^{\Gamma_0}Y_0|\phi_n^{\Gamma_0}\rangle\|_2 + \|V_{m-1}^{\Gamma_0}Y_0|\phi_{m-1}^{\Gamma_0}\rangle\|_2
\end{aligned}$$

$$\leq \quad \|Y_0|\phi_m^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{m-2} \|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 + \|Y_0|\phi_{m-1}^{\Gamma_0}\rangle\|_2$$

$$= \quad \sum_{n=0}^{m} \|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 \tag{37}$$

so that we may write Equation 34 as

$$\|\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle\|_2 \overset{(34,37)}{\leq} \sum_{m=0}^{j-1}\sum_{n=0}^{m} \|Y_0|\phi_n^{\Gamma_0}\rangle\|_2. \tag{38}$$

With the bounds from Equations 30 and 38 in place, we can return to Equation 22:

$$\begin{aligned}
\|Y_0|\phi_k^{\Gamma_j}\rangle\|_2^2 &\overset{(22)}{=} \left(\|\Lambda_j^{\Gamma_j}|\phi_0^{\Gamma_0}\rangle\|_2 + \|Y_0\, U_{(j,k)}^{\Gamma_0}|\phi_j^{\Gamma_*}\rangle\|_2\right)^2 \\
&\overset{(30,38)}{\leq} \left(\|Y_0|\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 + \sum_{m=0}^{j-1}\sum_{n=0}^{m}\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2 \\
&= \left(\|Y_0|\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2 \\
&= \left(\|Y_0|\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n+1)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2
\end{aligned}$$

□

**Lemma 8** *Suppose that $y_0$ is solved under $H * \Sigma y_0$. If $j < q$, and for each $i < j$ we have that both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing for $y_0$, then $y_0$ is solved under $\Gamma_j$.*

*Proof.* To prove that $y_0$ is solved under $\Gamma_j$, we need to show that $\beta_{y_0}^{\Gamma_j} \geq \frac{acc}{2q^7}$, and that $\kappa_{y_0}^{\Gamma_j} \geq \frac{acc}{q^7}$. Combining Lemma's 6, 7 and the definition of $y_0$ being solved under $\Gamma_0 = H$ (Definition 3), we get for any $k < q$:

$$\begin{aligned}
\kappa_{y_0,k}^{\Gamma_j} &= \frac{\alpha_{y_0}^{\Gamma_j}}{\|Y_0|\phi_k^{\Gamma_j}\rangle\|_2^2} \\
&\overset{(L.7)}{\geq} \frac{\alpha_{y_0}^{\Gamma_j}}{\left(\|Y_0|\phi_k^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n+1)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2} \\
&\overset{(*)}{\geq} \frac{\alpha_{y_0}^{\Gamma_j}\cdot\kappa_{y_0}^{\Gamma_0}}{j^4\cdot\alpha_{y_0}^{\Gamma_0}} \\
&\overset{(L.6)}{\geq} \frac{(q-j)\cdot\alpha_{y_0}^{\Gamma_0}\cdot\kappa_{y_0}^{\Gamma_0}}{q\cdot j^4\cdot\alpha_{y_0}^{\Gamma_0}} \\
&= \frac{(q-j)\cdot\kappa_{y_0}^{\Gamma_0}}{q\cdot j^4} \\
&\overset{(D.3)}{\geq} \frac{(q-j)\cdot acc}{q^3\cdot j^4} \\
&\geq \frac{acc}{q^7}
\end{aligned}$$

where in the third (in)equality $(*)$ we have used that $\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 \leq \frac{\alpha_{y_0}^{\Gamma_0}}{\kappa_{y_0}^{\Gamma_0}}$ for all $n, k < q$. Since the derived bound holds for all $k < q$, we have $\kappa_{y_0}^{\Gamma_j} \geq \frac{acc}{q^7}$. Similarly we obtain

$$\begin{aligned}
\beta_{y_0}^{\Gamma_j} &= \frac{\alpha_{y_0}^{\Gamma_j}}{\|Y_0|\phi_q^{\Gamma_j}\rangle\|_2} \\
&\overset{(L.7)}{\geq} \frac{\alpha_{y_0}^{\Gamma_j}}{\left(\|Y_0|\phi_q^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n+1)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2} \\
&\overset{(*)}{\geq} \frac{\alpha_{y_0}^{\Gamma_j}}{j^4\cdot\frac{\alpha_{y_0}^{\Gamma_0}}{\kappa_{y_0}^{\Gamma_0}} + j^2\cdot\frac{\alpha_{y_0}^{\Gamma_0}}{\sqrt{\beta_{y_0}^{\Gamma_0}\cdot\kappa_{y_0}^{\Gamma_0}}} + \frac{\alpha_{y_0}^{\Gamma_0}}{\beta_{y_0}^{\Gamma_0}}}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(L.6)}{\geq} && \frac{(q-j)\cdot \alpha_{y_0}^{\Gamma_0}}{q\cdot\left(j^4\cdot\frac{\alpha_{y_0}^{\Gamma_0}}{\kappa_{y_0}^{\Gamma_0}}+j^2\cdot\frac{\alpha_{y_0}^{\Gamma_0}}{\sqrt{\beta_{y_0}^{\Gamma_0}\cdot\kappa_{y_0}^{\Gamma_0}}}+\frac{\alpha_{y_0}^{\Gamma_0}}{\beta_{y_0}^{\Gamma_0}}\right)}\\
&\overset{(D.3)}{\geq} && \frac{(q-j)\cdot \alpha_{y_0}^{\Gamma_0}}{q\cdot\left(j^4\cdot\frac{q^2\cdot\alpha_{y_0}^{\Gamma_0}}{acc}+j^2\cdot\frac{q^2\cdot\alpha_{y_0}^{\Gamma_0}}{acc}+\frac{q\cdot\alpha_{y_0}^{\Gamma_0}}{acc}\right)}\\
&= && \frac{(q-j)\cdot acc}{q\cdot(j^4\cdot q^2+j^2\cdot q^2+q)}\\
&\geq && \frac{acc}{2q^7}.
\end{aligned}
$$

where this time in the third (in)equality $(*)$ we used $\|Y_0|\phi_q^{\Gamma_0}\rangle\|_2 = \frac{\alpha_{y_0}^{\Gamma_0}}{\beta_{y_0}^{\Gamma_0}}$ and again $\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2 \leq \frac{\alpha_{y_0}^{\Gamma_0}}{\kappa_{y_0}^{\Gamma_0}}$ for all $n < q$. By Definition 3, these things together imply that $y_0$ is solved under $\Gamma_j$. □

**Lemma 9** *Suppose that $y_0$ is solved under $H * \Sigma y_0$. Then except with at most negligible probability, there exists some $0 \leq m < q$ for which*

1. *$y_0$ is solved under $\Gamma_m$.*
2. *For each $i < m$, both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing for $y_0$.*
3. *Either $U_m^{\Gamma_m}$ or $U_m^{\Gamma_{m+1}}$ is contributing for $y_0$. (In the latter case we have $m < q-1$.)*

*Proof.* We prove by induction on the oracle sequences $\Gamma_j$. We show that requirement *1.* and *2.* are inherited by every new oracle sequence, until we hit a sequence that satisfies requirement *3.*, in which case the induction stops. If the induction does not stop before $\Gamma_q$, we derive a contradiction.

As a base case, we are given that $y_0$ is solved under $H * \Sigma y_0$, i.e solved under $\Gamma_0$. For the induction step, suppose that $y_0$ is solved under $\Gamma_j$ and that for all $i < j$ both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing for $y_0$. If $\|Y_0|\phi_j^{\Gamma_j}\rangle\|_2 = 0$, then the runs under $\Gamma_j$ and $\Gamma_{j+1}$ are identical, hence $y_0$ is solved under $\Gamma_{j+1}$. Suppose therefore that $\|Y_0|\phi_j^{\Gamma_j}\rangle\|_2 \neq 0$. Now since $[\Gamma_j]_k = H$ for all $k < j$, it must be that $|\phi_j^{\Gamma_j}\rangle$ is stochastically independent of $\Sigma(y_0)$. By Lemma 5 this means that, except with negligible probability, we either have that $U_j^{\Gamma_j}$ is contributing for $y_0$, or else $U_{j+1}^{\Gamma_j}$ is not stochastically independent of $\Sigma(y_0)$. In the former case, the induction stops. In the latter, we have derived a contradiction if $j = q-1$, since $U_q^{\Gamma_j}$ is the identity, which is trivially independent of $\Sigma(y_0)$. If $j < q-1$, then by Lemma 8 we get that either $U_j^{\Gamma_{j+1}}$ is contributing for $y_0$, or else $y_0$ is solved under $\Gamma_{j+1}$. In the former case the induction stops, in the latter we move to the next step. □

In the following, we write $H \setminus y$ for a function from $\{0,1\}^{\ell_{in}}$ to $\{0,1\}^{\ell_{out}}$ that leaves its value at $y$ undefined. In this context, $H * \Theta y$ is the function the function that completes $H \setminus y$ by defining

$$
H * \Theta y(x) := \begin{cases} \Theta(y) & \text{for } x = y \\ H \setminus y(x) & \text{otherwise} \end{cases}.
$$

**Lemma 10** *Let $y$ be fixed, and let $\Theta_1(y), \Theta_2(y) \in C$ be any two values such that $H * \Theta_i y$ completes $H \setminus y$. Then*

$$
\mathcal{Q}_{\mathcal{A}(H*\Theta_2 y)}(y) \geq \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q^5}.
$$

*Proof.* If we define the oracle sequence $\Gamma_j$ to be

$$
[\Gamma_j]_i := \begin{cases} H * \Theta_1 y & \text{for } i < j \\ H * \Theta_2 y & \text{otherwise.} \end{cases}
$$

then we have $\Gamma_0 = H * \Theta_2 y$ and $\Gamma_q = H * \Theta_1 y$. Since $\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)$ is a sum of the $q$ terms $\|Y|\phi_i^{\Gamma_q}\rangle\|_2^2$ for $0 \leq i < q$, it must be that for at least one of these terms – let term $j$ be the first such term – we have

$$
\|Y|\phi_j^{\Gamma_q}\rangle\|_2^2 \geq \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q}.
$$

Note furthermore that for all $0 \leq i < q$ we have

$$
\|Y|\phi_i^{\Gamma_q}\rangle\|_2^2 = \|Y|\phi_i^{\Gamma_i}\rangle\|_2^2 \qquad \text{hence in particular} \qquad \|Y|\phi_j^{\Gamma_j}\rangle\|_2^2 \geq \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q} \tag{39}
$$

since the state $|\phi_i^\Gamma\rangle$ is independent of the choice of oracles in $\Gamma$ from the $i$-th oracle on.

Now assume - to derive a contradiction - that for all $i \leq j$

$$\|Y|\phi_i^{\Gamma_0}\rangle\|_2^2 < \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q^5}. \tag{40}$$

Then

$$\begin{aligned}
\|Y|\phi_j^{\Gamma_j}\rangle\|_2^2 &\overset{(L.7)}{\leq} \left(\|Y_0|\phi_q^{\Gamma_0}\rangle\|_2 + \sum_{n=0}^{j-1}(j-n+1)\cdot\|Y_0|\phi_n^{\Gamma_0}\rangle\|_2\right)^2 \\
&\overset{(40)}{<} j^4 \cdot \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q^5} \\
&< \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q}
\end{aligned}$$

contradicting Equation 39. We conclude that there is at least one $0 \leq i \leq j$ such that

$$\|Y|\phi_j^{\Gamma_0}\rangle\|_2^2 \geq \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta_1 y)}(y)}{q^5}.$$

This proves the claim of the lemma. □

**Lemma 11** *Suppose that either $U_j^{\Gamma_j}$ or $U_j^{\Gamma_{j+1}}$ is contributing for $y_0$, with $j < q$ in the first case and $j < q-1$ in the second. If furthermore for each $i < j$ both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are* not *contributing for $y_0$, then we have the following bound for the* relative query magnitude *of $y_0$ at $i$*

$$\frac{\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2^2}{\mathcal{Q}_{\mathcal{A}(H)}(y_0)} \geq \frac{\kappa_{y_0}^{\Gamma_0}}{16q^9}.$$

*Proof.* We derive:

$$\begin{aligned}
\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2 \cdot \frac{\|G_0\, U_i^{\Gamma_j}\, Y_0|\phi_i^{\Gamma_j}\rangle\|_2}{\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2} &\overset{(D.4)}{\geq} \frac{\sqrt{\alpha_{y_0}^{\Gamma_j}}}{4q} \quad \text{or} \\
\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2 \cdot \frac{\|G_0\, U_i^{\Gamma_{j+1}}\, Y_0|\phi_i^{\Gamma_j}\rangle\|_2}{\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2} &\overset{(D.4)}{\geq} \frac{\sqrt{\alpha_{y_0}^{\Gamma_{j+1}}}}{4q} \\
\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2 &\geq \frac{\sqrt{\left[\alpha_{y_0}^{\Gamma_j}/\alpha_{y_0}^{\Gamma_{j+1}}\right]}}{4q} \\
\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2^2 &\geq \frac{\left[\alpha_{y_0}^{\Gamma_j}/\alpha_{y_0}^{\Gamma_{j+1}}\right]}{16q^2} \\
\frac{\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2^2}{\mathcal{Q}_{\mathcal{A}(H)}(y_0)} &\geq \frac{\left[\alpha_{y_0}^{\Gamma_j}/\alpha_{y_0}^{\Gamma_{j+1}}\right]}{\mathcal{Q}_{\mathcal{A}(H)}(y_0)\cdot 16q^2}.
\end{aligned} \tag{41}$$

Note that we have used the identity $|\phi_i^{\Gamma_{i+1}}\rangle = |\phi_i^{\Gamma_i}\rangle$ to tweak Definition 4 to our needs in the second case. Here $\left[\alpha_{y_0}^{\Gamma_j}/\alpha_{y_0}^{\Gamma_{j+1}}\right]$ denotes "either $\alpha_{y_0}^{\Gamma_j}$ or $\alpha_{y_0}^{\Gamma_{j+1}}$".

Now let $i_0$ be the query-number of the query that contains the most amplitude on $y_0$ in a run under $\Gamma_0$, i.e. such that

$$\kappa_{y_0}^{\Gamma_0} = \frac{\alpha_{y_0}^{\Gamma_0}}{\|Y_0|\phi_{i_0}^{\Gamma_0}\rangle\|_2^2}. \tag{42}$$

We may then continue with Equation 41 as follows:

$$\begin{aligned}
\frac{\|Y_0|\phi_i^{\Gamma_j}\rangle\|_2^2}{\mathcal{Q}_{\mathcal{A}(H)}(y_0)} &\overset{(41)}{\geq} \frac{\left[\alpha_{y_0}^{\Gamma_j}/\alpha_{y_0}^{\Gamma_{j+1}}\right]}{16q^2\cdot\mathcal{Q}_{\mathcal{A}(H)}(y_0)} \\
&\overset{(L.6)}{\geq} \frac{(1-\frac{j+1}{q})\cdot\alpha_{y_0}^{\Gamma_0}}{16q^2\cdot\mathcal{Q}_{\mathcal{A}(H)}(y_0)} \\
&\overset{(L.10)}{\geq} \frac{(1-\frac{j+1}{q})\cdot\alpha_{y_0}^{\Gamma_0}}{16q^7\cdot\mathcal{Q}_{\mathcal{A}(\Gamma_0)}(y_0)}
\end{aligned}$$

$$\begin{aligned}
&\overset{(4)}{\geq} && \frac{(1-\frac{j+1}{q})\cdot \alpha_{y_0}^{\Gamma_0}}{16q^7\cdot q\cdot \|Y_0|\phi_{i_0}^{\Gamma_0}\rangle\|_2^2}\\
&\overset{(42)}{=} && \frac{(1-\frac{j+1}{q})\cdot \kappa_{y_0}^{\Gamma_0}}{16q^8}\\
&\geq && \frac{\kappa_{y_0}^{\Gamma_0}}{16q^9}.
\end{aligned}$$

□

Note that '$y_0$ is solved under $H * \Sigma y_0$' means that $y_0$ *would have been* solved *had we* run $\mathcal{A}$ under $H * \Sigma y_0$, from the beginning to the end and without intermediate measurement. The specific $y_0$ that we want to consider however is obtained in a run under $H$. While we are interested in the question of whether $y_0$ is solved under $H * \Sigma y_0$, in the next lemma we derive a bound on the probability that it is both solved under $H$ and $H * \Sigma y_0$. The lemma is similar to the classical forking lemma (see [PS96]) in the sense that here too we use the known probability that $\mathcal{A}$ solves some instance $y$ on a *random* challenge, to bound the probability that it can solve $y$ on a second, independently chosen challenge as well. The proof shares one further trick with the proof of the 'general forking lemma' in [BN06] (using Jensen's inequality to get rid of the square), the rest of the proof techniques are special to the quantum setting.

Note that Dominique Unruh at some point gave out a note about a 'Quantum Forking Conjecture' (unpublished). This conjecture was subsequently broken by Alexander Belov. The conjecture, while it could equally well be described as a generalization of the classical forking lemma, otherwise has nothing in common with our quantum forking lemma.

**Lemma 12 (Quantum Forking Lemma)** *Suppose that we run $\mathcal{A}^H$ up to the $i$-th query, where $i$ is chosen at random, and measure the register that contains $\mathcal{A}$'s next oracle query, obtaining $y_0$ as the outcome. The probability that $y_0$ is solved under $H$ as well as under $H * \Sigma y_0$ is then at least $\frac{acc^2}{q^{19}}$.*

*Proof.* We want a bound on the following quantity:

$$\mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y\in S_H\cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] \tag{43}$$

since this gives us the expectation of the query magnitude *in a run under $H$* of those $y$ that will be solved in a run that uses $H$ from start to end *as well as* a run that uses $H * \Sigma y$ from start to end. In order to rewrite this expression, we first prove the following claim:

**Claim (*):** For any function $F$ and any sequence of independently chosen random values $h_1,\ldots,h_n$ we have for any $1\leq i\leq n$ that

$$\mathop{\mathbb{E}}_{h_1,\ldots,h_n}[F(h_1,\ldots,h_n)] = \mathop{\mathbb{E}}_{h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n}\left[\mathop{\mathbb{E}}_{h_i}[F(h_1,\ldots,h_n)]\right] \tag{44}$$

*Proof.* We have

$$\begin{aligned}
\mathop{\mathbb{E}}_{h_1,\ldots,h_n}[F(h_1,\ldots,h_n)] &= \sum_{h_1,\ldots,h_n}\Pr[v_1,\ldots,v_n=h_1,\ldots,h_n]\cdot F(v_1,\ldots,v_n)\\
&= \sum_{h_1,\ldots,h_n}\Pr[v_1,\ldots,v_{i-1},v_{i+1},\ldots,v_n=h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n]\\
&\quad\cdot \Pr[v_i=h_i]\cdot F(v_1,\ldots,v_n)\\
&= \sum_{h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n}\Pr[v_1,\ldots,v_{i-1},v_{i+1},\ldots,v_n=h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n]\\
&\quad\cdot \sum_{h_i}\Pr[v_i=h_i]\cdot F(v_1,\ldots,v_n)\\
&= \mathop{\mathbb{E}}_{h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n}\left[\sum_{h_i}\Pr[v_i=h_i]\cdot F(v_1,\ldots,v_n)\right]\\
&= \mathop{\mathbb{E}}_{h_1,\ldots,h_{i-1},h_{i+1},\ldots,h_n}\left[\mathop{\mathbb{E}}_{h_i}[F(h_1,\ldots,h_n)]\right]
\end{aligned}$$

where the second step in the derivation is justified by the independence of the choice of the random values $h_1,\ldots,h_n$. □

The claim, together with Lemma 10, allows us to rewrite and bound the term in 43. In the following, let $S_H(\cdot)$ be the indicator function for the set $S_H$ of solved instances in a run under $H$. Let furthermore $\Sigma, \Theta : \{0,1\}^{\ell_{in}} \to \{0,1\}^{\ell_{out}}$ both be (independent) random functions. Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y \in S_H \cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] &= \mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_y S_H(y) \cdot S_{H*\Sigma y}(y) \cdot \mathcal{Q}_{\mathcal{A}(H)}(y)\right] \\
&= \sum_y \mathop{\mathbb{E}}_{H,\Sigma(y)}\left[S_H(y) \cdot S_{H*\Sigma y}(y) \cdot \mathcal{Q}_{\mathcal{A}(H)}(y)\right] \\
&\overset{*}{=} \sum_y \mathop{\mathbb{E}}_{H\setminus y}\left[\mathop{\mathbb{E}}_{H(y),\Sigma(y)}\left[S_H(y) \cdot S_{H*\Sigma y}(y) \cdot \mathcal{Q}_{\mathcal{A}(H)}(y)\right]\right] \\
&= \sum_y \mathop{\mathbb{E}}_{H\setminus y}\left[\mathop{\mathbb{E}}_{\Theta(y),\Sigma(y)}\left[S_{H*\Theta y}(y) \cdot S_{H*\Sigma y}(y) \cdot \mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)\right]\right].
\end{aligned}$$

Here we consecutively used the definition of the indicator function, linearity of expectation and the above claim $(*)$. The claim makes that we can consider the (expectation over) the set of random values $\{(H \setminus y)(x)) \ : \ x \neq y\}$ and the single random value $H(y)$ separately. Then in the last step, we may replace $H(y)$ by the equally random value $\Theta(y)$, this should not change the (inner) expectation.

The goal of the substitution is this: Further on, we need the object $H(y)$ both *as a fixed value* and *as a variable* in the same expression. We need it as a fixed value to have a determinate object $\mathcal{Q}_{\mathcal{A}(H)}(y)$; something like $\mathcal{Q}_{\mathcal{A}(H\setminus y)}(y)$ is not well defined. We need it as a variable to take a probability over its choice. In the next step, we reintroduce $H(y)$ in the outer expectation. This 'incarnation' of $H(y)$ will play the role of a fixed value inside the square brackets of the expectation. The substitute $\Theta(y)$ that we introduced in the previous step will play the role of a variable, over which we may take a probability.

Note that we now have two terms in our expression that feature $\Theta(y)$, namely $S_{H*\Theta y}(y)$ and $\mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)$. The first we need in the variable setting, the second in the fixed value setting. We therefore need to transform the occurrence of $\Theta(y)$ in the second term back to $H(y)$. This is what Lemma 10 allows us to do. We furthermore add $q^4$ on both sides of the equation as a normalization factor. By the linearity of expectation we may move it inside the brackets.

$$\begin{aligned}
q^4 \cdot \mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y \in S_H \cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] &= \sum_y \mathop{\mathbb{E}}_{H}\left[\mathop{\mathbb{E}}_{\Theta(y),\Sigma(y)}\left[S_{H*\Theta y}(y) \cdot S_{H*\Sigma y}(y) \cdot \mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)\right]\right] \cdot q^4 \\
&\overset{(L.10)}{\geq} \sum_y \mathop{\mathbb{E}}_{H}\left[\mathop{\mathbb{E}}_{\Theta(y),\Sigma(y)}\left[S_{H*\Theta y}(y) \cdot S_{H*\Sigma y}(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q^5} \cdot q^4\right]\right] \\
&= \sum_y \mathop{\mathbb{E}}_{H}\left[\frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q} \cdot \mathop{\mathbb{E}}_{\Theta(y),\Sigma(y)}\left[S_{H*\Theta y}(y) \cdot S_{H*\Sigma y}(y)\right]\right] \\
&= \mathop{\mathbb{E}}_{H}\left[\sum_y \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q} \cdot \mathop{\mathbb{E}}_{\Theta(y)}\left[S_{H*\Theta y}(y)\right] \cdot \mathop{\mathbb{E}}_{\Sigma(y)}\left[S_{H*\Sigma y}(y)\right]\right] \\
&= \mathop{\mathbb{E}}_{H}\left[\sum_y \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q} \cdot \Pr_{\Theta(y)}\left[y \in S_{H*\Theta y}\right] \cdot \Pr_{\Sigma(y)}\left[y \in S_{H*\Sigma y}\right]\right] \\
&= \mathop{\mathbb{E}}_{H}\left[\sum_y \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q} \cdot \Pr_{\Theta(y)}\left[y \in S_{H*\Theta y}\right]^2\right]
\end{aligned}$$

To recap, in the first step we introduced $H(y)$ to the range of the outer expectation, but left it unused. Then we applied Lemma 10 to rewrite $\mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)$ as $\frac{\mathcal{Q}_{\mathcal{A}(H*Hy)}(y)}{q^5} = \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q^5}$ for the specific value $H(y)$ that we just introduced, which is fixed inside the brackets. We then used the linearity of expectation three times, to move the rescaled term $\frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q}$ out of the inner, the sum over $y$ into to the outer expectation and to split the inner into separate expectations of its two factors. Next, we observed that the expectation of the indicator function equals the probability of set inclusion. The two probabilities that we obtain are independent and equal, since the expressions they range over are the same, and the functions $\Theta$ and $\Sigma$ are chosen independently at random from the same codomain. We may thus write their product as a square of the first.

Note that the expression inside the remaining expectation has a special form. The values $\frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q}$ for all $y \in \{0,1\}^{\ell_{in}}$ sum up to exactly 1, because no matter how the different $y$'s are distributed over $\mathcal{A}$'s queries, their combined query magnitude always equals $q$. This means that we may consider $\delta(y) := \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q}$ as a probability density function. The expression $\sum_y \delta(y) \cdot X[y]$ is then an expectation of $X$ taken over $y$ distributed according to $\delta$. We may therefore write

$$\begin{aligned}
q^4 \cdot \mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y \in S_H \cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] &= \mathop{\mathbb{E}}_{H}\left[\mathop{\mathbb{E}}_{y\sim\delta}\left[\Pr_{\Theta(y)}\left[y \in S_{H*\Theta y}\right]^2\right]\right] \\
&\geq \mathop{\mathbb{E}}_{H}\left[\mathop{\mathbb{E}}_{y\sim\delta}\left[\Pr_{\Theta(y)}\left[y \in S_{H*\Theta y}\right]\right]\right]^2 \\
&= \mathop{\mathbb{E}}_{H}\left[\sum_y \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q} \cdot \Pr_{\Theta(y)}\left[y \in S_{H*\Theta y}\right]\right]^2 \\
&= \mathop{\mathbb{E}}_{H}\left[\sum_y \mathop{\mathbb{E}}_{\Theta(y)}\left[S_{H*\Theta y}(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q}\right]\right]^2
\end{aligned}$$

The inequality is Jensen's inequality, applied both to the inner and the outer expectation in one step. We then work our way back through the definitions to get to the expectation over $\Theta(y)$.

From here the goal is to get $\sum_{y\in S_H} \mathcal{Q}_{\mathcal{A}(H)}(y)$ inside the brackets, for which we already know a bound by Lemma 4. To do so, we first need to make sure that every term uses the same function $H * \Theta y$, so that we may relabel them collectively. We thus apply Lemma 10 again. The value $H(y)$ is now not in use anymore, therefore we can just as well take the expectation over $H \setminus y$. This allows us to take the sum out in the next step. Then we can do the relabeling, join the expectations together again and complete our plan.

$$\begin{aligned}
q^4 \cdot \mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y \in S_H \cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] &\overset{(L.10)}{\geq} \mathop{\mathbb{E}}_{H}\left[\sum_y \mathop{\mathbb{E}}_{\Theta(y)}\left[S_{H*\Theta y}(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)}{q^6}\right]\right]^2 \\
&= \mathop{\mathbb{E}}_{H\setminus y}\left[\sum_y \mathop{\mathbb{E}}_{\Theta(y)}\left[S_{H*\Theta y}(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)}{q^6}\right]\right]^2 \\
&= \sum_y \mathop{\mathbb{E}}_{H\setminus y}\left[\mathop{\mathbb{E}}_{\Theta(y)}\left[S_{H*\Theta y}(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H*\Theta y)}(y)}{q^6}\right]\right]^2 \\
&= \sum_y \mathop{\mathbb{E}}_{H\setminus y}\left[\mathop{\mathbb{E}}_{H(y)}\left[S_H(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q^6}\right]\right]^2 \\
&\overset{*}{=} \sum_y \mathop{\mathbb{E}}_{H}\left[S_H(y) \cdot \frac{\mathcal{Q}_{\mathcal{A}(H)}(y)}{q^6}\right]^2 \\
&= \frac{\mathbb{E}_H\left[\sum_y S_H(y) \cdot \mathcal{Q}_{\mathcal{A}(H)}(y)\right]^2}{q^{12}} \\
&= \frac{\mathbb{E}_H\left[\sum_{y\in S_H} \mathcal{Q}_{\mathcal{A}(H)}(y)\right]^2}{q^{12}} \\
&\overset{(L.4)}{\geq} \frac{\left(acc \cdot \frac{q-2}{q^2} - \mu(\eta)\right)^2}{q^{12}} \\
&> \frac{acc^2}{q^{15}}
\end{aligned}$$

which means that

$$\mathop{\mathbb{E}}_{H,\Sigma}\left[\sum_{y \in S_H \cap S_{H*\Sigma y}} \mathcal{Q}_{\mathcal{A}(H)}(y)\right] > \frac{acc^2}{q^{19}}.$$

□

We are now ready to evaluate the probability $\Pr\left[ok_V = 1 \wedge ok_Q = 1 : \mathbf{R} - \mathbf{Prove}_{\mathbf{\Sigma}}\right]$. Let $y_0$ be the outcome of the measurement performed by $R$ at step 3 of its execution. By Lemma 12, the probability that $y_0$ is solved under $H * \Sigma y_0$ is then at least $\frac{acc^2}{q^{15}}$. If it is indeed solved under $H * \Sigma y_0$, then we get from Lemma 9 that except with negligible probability, there exists some $0 \leq m < q$ such that

1. $y_0$ is solved under $\Gamma_m$.
2. For each $i < m$, both $U_i^{\Gamma_i}$ and $U_i^{\Gamma_{i+1}}$ are not contributing for $y_0$.
3. Either $U_m^{\Gamma_m}$ or $U_m^{\Gamma_{m+1}}$ is contributing for $y_0$. (In the latter case we have $m < q-1$.)

Remember that $R$ measures the $m'$-th query, where $m'$ is chosen at random. We need to find the probability that $m' = m$. This probability is not simply $1/q$, because we assumed – and therefore need to condition on – that the measurement outcome is $y_0$. Note that the above conditions 1 - 3 allow us to apply Lemma 11, to find that

$$\Pr_{m'}\left[m' = m \,|\, y_0 \leftarrow \mathcal{M}|\phi^H_{m'}\rangle\right] \quad = \quad \frac{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2^2}{\mathcal{Q}_{\mathcal{A}(H)}(y_0)} \quad \overset{(L.11)}{=} \quad \frac{\kappa^{\Gamma_0}_{y_0}}{16q^8} \quad \geq \quad \frac{acc}{16q^{10}}.$$

where the last inequality follows from Definition 3, since we assumed that $y_0$ is solved under $H * \Sigma y_0 = \Gamma_0$.

We thus assume that $R$ measured the right kind of query, i.e. some query $m$ where either $U_m^{\Gamma_m}$ or $U_m^{\Gamma_{m+1}}$ is contributing. Next, $R$ obtains $\Sigma(y_0)$ from the $\Sigma$-verifier and flips a coin, to determine whether it reprograms the random oracle now or right after answering the $m$-th query. This corresponds to applying either $U_m^{\Gamma_m}$ or $U_m^{\Gamma_{m+1}}$ to the current state. We thus have a 50% chance that $R$ applies a unitary that is contributing for $y_0$.

Due to the measurement, the current state on $\mathcal{A}$'s complete system has become

$$|\phi_{postm}\rangle = \frac{Y_0|\phi_m^{\Gamma_m}\rangle}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2}.$$

**Case 1:** $U_m^{\Gamma_m}$ is contributing for $y_0$. When we now compute the length of the projection of the final output state onto 'the good part', we find

$$\|G_0 U_m^{\Gamma_m}|\phi_{postm}\rangle\|_2 \quad = \quad \frac{\|G_0 U_m^{\Gamma_m} Y_0|\phi_m^{\Gamma_m}\rangle\|_2}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2} \quad \geq \quad \frac{\sqrt{\alpha_{y_0}^{\Gamma_m}}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2 \cdot 4q}.$$

where the last inequality follows because $U_m^{\Gamma_m}$ is contributing for $y_0$ (Definition 4). Therefore,

$$\|G_0 U_m^{\Gamma_m}|\phi_{postm}\rangle\|_2^2 \quad \geq \quad \frac{\alpha_{y_0}^{\Gamma_m}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2^2 \cdot 16q^2} \quad = \quad \frac{\kappa^{\Gamma_m}_{y_0,m}}{16q^2} \quad \geq \quad \frac{acc}{16q^9}$$

where the last inequality follows from the fact that $y_0$ is solved under $\Gamma_m$ (Definition 3).

**Case 2:** $U_m^{\Gamma_{m+1}}$ is contributing for $y_0$. In this case we find

$$\|G_0 U_m^{\Gamma_{m+1}}|\phi_{postm}\rangle\|_2 \quad = \quad \frac{\|G_0 U_m^{\Gamma_{m+1}} Y_0|\phi_m^{\Gamma_m}\rangle\|_2}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2} \quad \geq \quad \frac{\sqrt{\alpha_{y_0}^{\Gamma_{m+1}}}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2 \cdot 4q}.$$

where the last inequality follows because $U_m^{\Gamma_{m+1}}$ is contributing for $y_0$ (Definition 4). Therefore,

$$\begin{aligned}
\|G_0 U_m^{\Gamma_{m+1}}|\phi_{postm}\rangle\|_2^2 \quad &\geq \quad \frac{\alpha_{y_0}^{\Gamma_{m+1}}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2^2 \cdot 16q^2} \\
\overset{(19)}{\geq} \quad & \frac{\alpha_{y_0}^{\Gamma_m}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2^2 \cdot 16q^2} - \frac{\alpha_{y_0}^{\Gamma_m}}{\||Y_0|\phi_m^{\Gamma_m}\rangle\|_2^2 \cdot 16q^3} \\
= \quad & \frac{\kappa^{\Gamma_m}_{y_0,m}}{16q^2} - \frac{\kappa^{\Gamma_m}_{y_0,m}}{16q^3} \\
\overset{(D.3)}{\geq} \quad & \frac{acc}{16q^9} - \frac{acc}{16q^{10}} \\
\geq \quad & \frac{acc}{q^{10}} \qquad\qquad \text{for } q > 16.
\end{aligned}$$

Summarizing, we see that the probability that $R$ obtains output $(x, com, ch, resp)$ that will make $V_\Sigma$ accept, and such that $Q(x) = 1$, is at least

$$\Pr\left[ok_V = 1 \wedge ok_Q = 1 : \mathbf{R} - \mathbf{Prove}_{\mathbf{\Sigma}}\right] \quad \geq \quad \frac{acc^2}{q^{15}} \cdot (1 - \mu_1(\eta)) \cdot \frac{acc}{16q^{10}} \cdot \frac{1}{2} \cdot \frac{acc}{q^{10}} \quad = \quad \frac{acc^4}{32q^{35}} - \mu_2(\eta).$$

Since the the Sigma-protocol underlying our proof system has special soundness and perfect unique reponses, we may now apply Theorem 9 from [Unr12]. It says that

$$\Pr_K \quad \geq \quad \left(\frac{acc^4}{32q^{35}} - \mu_2(\eta) - \frac{1}{\sqrt{c(\eta)}}\right)^3 \quad = \quad \frac{acc^{12}}{32768 \cdot q^{105}} - \mu(\eta)$$

where $\Pr_K$ is the success probability of the canonical extractor applied to our reduction algorithm $R$, and $c(\eta)$ is a function such that for all $\eta \in \mathbb{N}, x \in \{0,1\}^*$ we have that $\#C_{\eta x} \geq c(\eta)$. The canonical extractor outputs a pair $(x, w)$ such that $w$ is a valid witness for $x$, and $x$ is the statement chosen by $R$, which we argued is such that $Q(x) = 1$.

This means that if we let our Fiat-Shamir extractor $E$ use its black-box access to $\mathcal{A}$ to run the canonical extractor on $R$ applied to $\mathcal{A}$, we get that

$$\Pr\left[(x,w) \in R \wedge ok_Q = 1 : \mathbf{Extract}\right] \geq \frac{1}{32768 \cdot q^{105}} \Pr\left[ok_V = 1 \wedge ok_Q = 1 : \mathbf{Prove_{FS}}\right]^{12} - \mu(\eta)$$

proving the claim of Theorem 1. □

## 4.6 Discussion

We have proven that Fiat-Shamir is SP-extractable in the QROM. Why did we have to weaken the definition of [Unr17], which further requires the internal state of the adversary to be preserved (to a certain degree) across the extraction procedure?

Concretely, the stronger definition requires that any projective measurement $\Pi^H_{\eta,x||\pi}$ on the internal state of the adversary *after a normal run* (1), should succeed with polynomially-related probability on the internal state of the adversary *after the extractor has used it as an oracle to obtain a witness* (2). As is evident from the notation, the measurement may depend on $\eta, x, \pi$ and $H$.

The problem is that our extractor is forced to answer oracle queries according to two different, conflicting functions. While the two functions differ only on a single input, they differ exactly on the input $y_0 = x||com$ that is eventually used in the forgery we obtain at the end of the run.

Lemma 6 shows that the inconsistent answers do not decrease (too much) the amount of amplitude that sits on a correct response for $(y_0, \Sigma(y_0))$. However, it is not given that this amplitude is on the same basis states of the internal state, compared to the situation where we would have used the correct oracle answers from the start. In other words, while we prove that the adversary is not handicapped by the inconsistent oracles too much to do what we want it to do, we have not excluded the possibility that its internal state somehow got 'scarred' by the inconsistencies. We therefore cannot prove that any projective measurement that succeeds in situation (1), will succeed in situation (2) with polynomially related probability.

We have also not excluded the possibility. We have tried to give a more fine grained analysis in place of Lemma 6, where we differentiated between 'positively' and 'negatively' contributing unitaries, but it turned out to be not enough to remove the possible 'scars' from the adversary's internal state. However, perhaps a better analysis could extend our results to include Unruh's requirements in the future.

### (Non)-tightness of the reduction

In cryptography, the parameters (which influence the all-important efficiency) of a scheme are often set according to the *tightness* of the security reduction. A reduction is said to be tight when the success probability – or equivalently the run time – of the adversary against the scheme is of the same order as the that of the reduction against the hard problem. We have shown that a Fiat-Shamir adversary can break an underlying hard problem, but only with probability approximately $\frac{acc^12}{q^105}$, where $acc$ is the success probability of the adversary against the Fiat-Shamir scheme, and $q$ is the number of queries it makes. Our result is therefore a prime example of a non-tight reduction. However, two things can be said to put the astronomical bound in perspective.

Firstly, we do not believe that our reduction is optimal. As it is often the case in computer science, the first barrier to breach is the one between exponential and polynomial time, whatever the exact parameters are. We have not tried at all to optimize our result in this respect. Very likely the exact bound will be reduced in the future.

Secondly, there is a history with Fiat-Shamir proof systems of ignoring the tightness of the reduction. In the classical case, the reduction factor is approximately $\frac{1}{q^2}$. On could argue however that this factor is merely an artifact of the proof. It comes entirely from the uncertainty of the reduction as to which query the adversary will use in its forgery, for both the first and the second run. One could imagine the adversary using *itself* as a subroutine, in which case it could pick the correct query with no uncertainty. Of course this is not a very rigorous argument, but in practice at least many implementations of Fiat-Shamir schemes ignore the looseness of the reduction.

In any case, our reduction shows that an adversary who has non-negligible probability at breaking the Fiat-Shamir scheme, also has non-negligible probability of solving a problem of

which we assume that any efficient algorithm can only solve it with negligible probability. As long as the hardness-assumption is unbroken, the scheme is unbroken.

# 5 Existential unforgeability of Fiat-Shamir signatures

In [Unr17], Unruh proves that an extractable Fiat-Shamir proof system is also *simulation-sound* extractable, and that simulation-sound extractable proof systems can be used to create a signature scheme that has existential unforgeability. Theorem 1 from the previous section states that for suitable sigma-protocols, the corresponding Fiat-Shamir proof system is SP-extractable. In this section, we show that the proofs from [Unr17] can be adapted to work for this weaker definition as well.

We first need to adapt Unruh's definition of simulation-sound extractability to fit our notion of SP-extractability:

**Definition 5.1 (simulation-sound SP-extractable)** *A non-interactive proof system $(P,V)$ for a relation $R$ is* simulation-sound SP-extractable *with respect to the simulator $S$ iff there is a quantum polynomial-time oracle algorithm $E$ and a constant $d>0$, such that for any polynomial-time family of pure oracle circuits $\mathcal{A}_\eta$ (with output $\ell^{output}_{\mathcal{A}_\eta} = \ell^x_\eta + \ell^{com}_\eta + \ell^{ch}_\eta + \ell^{resp}_\eta$) there exists a polynomial $\ell \geq 0$ such that* for any classical predicate $Q$ *(possibly dependent on $\eta$) there exists a polynomial $p>0$ and a negligible function $\mu$ such that for all $\eta$ and all $\ell^{state}_{\mathcal{A}_\eta}$-qubit density operators $\rho$, we have that:*

$$\begin{aligned}&\Pr\left[(x,w)\in R \wedge ok_Q = 1 : \mathbf{Extract}\right]\\ &\geq \frac{1}{p(\eta)}\Pr\left[ok_V = 1 \wedge ok_Q = 1 \wedge (x,\pi)\notin \mathsf{S}\text{−}\mathsf{queries} : \mathbf{Prove}^{\mathbf{S}}_{\mathbf{FS}}\right]^d - \mu(\eta)\end{aligned}$$

*where $\mathbf{Prove}^{\mathbf{S}}_{\mathbf{FS}}$ is following game:*

$$\begin{aligned}&& H &\xleftarrow{\$} Fun(\ell^{in}_\eta, \ell^{out}_\eta),\\ &&S_{\mathcal{A}} &\leftarrow \rho\\ ***\quad && (x, com, H(x||com), resp) &\leftarrow \mathcal{A}^{H,S''}_\eta(S_{\mathcal{A}}),\\ &&\pi &:= com||resp,\\ ***\quad && ok_V &\leftarrow V^{H^{final}}_{FS}(1^\eta, x, \pi),\\ &&ok_Q &\leftarrow Q(1^\eta, x).\end{aligned}$$

*(Here *** marks the difference with the game $\mathbf{Prove}_{\mathbf{FS}}$ from Definition 2). The oracle $S''(x)$ invokes $S(1^\eta, x)$, and $H^{final}$ refers to the value of the random oracle $H$ at the end of the execution (remember that invocations of $S$ may change $H$). $\mathsf{S}\text{−}\mathsf{queries}$ is a list containing all queries made to $S''$ by $\mathcal{A}$, as pairs of input/output. (Note that the input and output of $S''$ are classical, so the list is well-defined.) Furthermore,* **Extract** *is the following game:*

$$\begin{aligned}&H \xleftarrow{\$} Fun(\ell^{in}_\eta, \ell^{out}_\eta),\\ &S_{\mathcal{A}} \leftarrow \rho,\\ &(x, w, \pi, ass) \leftarrow E^{\mathcal{A}^{\mathbf{rew}}_\eta(S_{\mathcal{A}}),H}(1^\eta, \ell(\eta), \mathbf{shape}_{\mathcal{A}_\eta})\\ &ok_Q \leftarrow Q(1^\eta, x).\end{aligned}$$

We need to consider two different proofs. Because we already assume perfect unique responses in the proof of SP-extractability, we only consider the 'strong' versions (see Section 1.5) of both. The first is

**Theorem 25 ([Unr17]) (If Fiat-Shamir is extractable, then it is strongly simulation-sound extractable)**
*Assume that $\Sigma$ has unique responses. Assume that the Fiat-Shamir proof system $(P_{FS}, V_{FS})$ based on $\Sigma$ is extractable. Then the Fiat-Shamir proof system $(P_{FS}, V_{FS})$ is strongly simulation-sound extractable with respect to the simulator $S_{FS}$ from [Section 1.5 of this thesis, JWD].*

We prove our own theorem by adapting Unruh's proof.

**Theorem 2 (Fiat-Shamir is strongly simulation-sound SP-extractable)** *Let $\Sigma$ be a sigma-protocol with special soundness and perfect unique responses, for the relation $R_\eta$, and such that for every $x \in \mathsf{dom}(R)$ the size of the challenge space $\#C_{\eta x}$ is exponential in $\eta$. Then the Fiat-Shamir proof system $(P_{FS}, V_{FS})$ based on $\Sigma$ is strongly simulation-sound SP-extractable with respect to the simulator $S_{FS}$ from Section 1.5.*

*Proof.* We need to take the game $\mathbf{Proves}^{\mathbf{S}}_{\mathbf{FS}}$ and transform it into the game **Extract**, and show that the probabilities of winning in these games are not too far apart. Since our predicate $Q$ is a special case of Unruh's projective measurement circuit $\Pi$, it suffices to point out the differences in the intermediate games of the proof. The reader is referred to [Unr17] to check that the transitions between these games go through as normal.

We leave out the input $1^\eta$ for convenience. In all of the following games, we have $H \xleftarrow{\$} Fun(\ell_\eta^{in}, \ell_\eta^{out})$.

**Game 1 (Real world)** $S_\mathcal{A} \leftarrow \rho.\ (x, com, H(x||com), resp) \leftarrow \mathcal{A}^{H,S_{FS}}(S_\mathcal{A})$
$ok_V \leftarrow V_{FS}^{H^{\text{final}}}(x, com||resp).\ ok_Q \leftarrow Q(x)\ win := (ok_V = 1 \wedge ok_Q = 1 \wedge x \notin \mathsf{S-queries}).$

This game is equal to the game in [Unr17], except that we substituted our condition $ok_Q$ for their $ok_A$ and adapted the output of $\mathcal{A}$ to match our definition. Since both are not used in the transition to the next game, we may conclude that

$$\Pr[win = 1 : \text{Game 2a}] \geq \Pr[win = 1 : \text{Game 1}] \tag{45}$$

where

**Game 2a (Unchanged H)** $S_\mathcal{A} \leftarrow \rho.\ (x, com, H(x||com), resp) \leftarrow \mathcal{A}^{H,S_{FS}}(S_\mathcal{A})$
$ok_V \leftarrow V_{FS}^{H}(x, com||resp).\ ok_Q \leftarrow Q(x)\ win := (ok_V = 1 \wedge ok_Q = 1 \wedge x \notin \mathsf{S-queries}).$

Quite obviously, when we drop one of the winning requirements, namely $x \notin \mathsf{S-queries}$, winning becomes easier, and thus

$$\Pr[win = 1 : \text{Game 2b}] \geq \Pr[win = 1 : \text{Game 2a}] \tag{46}$$

with

**Game 2b (Dropped $\mathsf{S-queries}$)** $S_\mathcal{A} \leftarrow \rho.\ (x, com, H(x||com), resp) \leftarrow \mathcal{A}^{H,S_{FS}}(S_\mathcal{A})$
$ok_V \leftarrow V_{FS}^{H}(x, com||resp).\ ok_Q \leftarrow Q(x)\ win := (ok_V = 1 \wedge ok_Q = 1).$

Next, Unruh introduces a quantum polynomial-time pure oracle circuit $B$, which behaves exactly as $\mathcal{A}$, but also simulates the simulator $S_{FS}$. Since $S_{FS}$ is an efficient algorithm itself, nothing really has changed and we get

$$\Pr[win = 1 : \text{Game 4}] \geq \Pr[win = 1 : \text{Game 2b}] \tag{47}$$

using the game

**Game 4 (Simulating $S_{FS}$)** $S_\mathcal{A} \leftarrow \rho.\ (x, com, H(x||com), resp) \leftarrow B^{H}(S_\mathcal{A})$
$ok_V \leftarrow V_{FS}^{H}(x, com||resp).\ ok_Q \leftarrow Q(x)\ win := (ok_V = 1 \wedge ok_Q = 1).$

(Note that we skipped game 3, becuase it only refers to the projective measurement from Unruh's definition.) We now apply Theorem 1, to obtain that $(P_{FS}, V_{FS})$ is SP-extractable. Note that game 4 is exactly the game $\mathbf{Prove}_{\mathbf{FS}}$ from Definition 2, for an adversary 'B'. Since $(P_{FS}, V_{FS})$ is SP-extractable, we get that

$$\Pr[win = 1 : \text{Game 5}] \geq \frac{1}{32768 \cdot q^{105}} \Pr[win = 1 : \text{Game 4}]^{12} - \mu(\eta) \tag{48}$$

where

**Game 5 (Extraction for B)** $S_\mathcal{A} \leftarrow \rho,\ (x, w, \pi, ass) \leftarrow E_0^{\mathcal{B}^{\mathbf{rew}}(S_\mathcal{A}),H}(\ell_0, \mathbf{shape_B}),\ ok_Q \leftarrow Q(x).$

The final thing to note is that $B$ can be simulated using only oracle access to $\mathcal{A}^{\mathbf{rew}}$. There must therefore exist an extractor algorithm $E$ that behave just like $E_0$, except that whenever $E_0$ makes a query to $B^{\mathbf{rew}}$, $E$ queries $\mathcal{A}^{\mathbf{rew}}$ instead and computes the part of $B$ that simulates $S_{FS}$ by itself. It must then be that

$$\Pr[win = 1 : \text{Game 7}] = \Pr[win = 1 : \text{Game 5}] \tag{49}$$

with

**Game 7 (Extraction)** $S_\mathcal{A} \leftarrow \rho,\ (x, w, \pi, ass) \leftarrow E^{\mathcal{A}^{\mathbf{rew}}(S_\mathcal{A}),H}(\ell, \mathbf{shape}_\mathcal{A}),\ ok_Q \leftarrow Q(x).$

(Again Game 6 from the original proof deals only with aspects specific to the measurement circuit $\Pi$ that we do not use.) Since Game 1 is exactly the game $\mathbf{Prove}^{\mathbf{S}}_{\mathbf{FS}}$ and Game 7 is exactly **Extract** both from Definition 5.1, and Equations 45 – 49 together imply that

$$Pr[win = 1 : \text{Game 7}] \geq \frac{1}{32768 \cdot q^{105}} \Pr[win = 1 : \text{Game 1}]^{12} - \mu(\eta)$$

the claim of the Theorem has been proven. □

We next turn to the theorem about unforgeability. First we need another definition.

**Definition 5.2 ([Unr17]; Hard instance generator)** *We call an algorithm $G$ a hard instance generator for a fixed-length relation $R_\eta$ iff*

- *$G$ is quantum polynomial-time, and*
- *there is a negligible $\mu$ such that for every $\eta$, $\Pr[(x,w) \in R_\eta \ : \ (x,w) \leftarrow G(1^\eta)] \geq 1 - \mu(\eta)$, and*
- *for any quantum polynomial-time $A$, there is a negligible $\mu$ such that for every $\eta$, $\Pr[(x,w') \in R_\eta \ : \ (x,w) \leftarrow G(1^\eta),\ (x,w') \leftarrow A(1^\eta, x)] \leq \mu(\eta)$.*

**Theorem 31 (Unforgeability from simulation-sound extractability)** *If $(P,V)$ is zero knowledge and has strong simulation-sound extractability for $R'_\eta$, and $G$ is a hard instance generator for $R_\eta$, then the signature scheme $(KeyGen, Sign, Verify)$ from Definition 1.3 is existentially unforgeable (see Definition 1.4).*

We only have to replace 'extractability' by 'SP-extractability' to get our own theorem:

**Theorem 3 (Unforgeability from simulation-sound SP-extractability)** *If $(P,V)$ is zero knowledge and has strong simulation-sound SP-extractability for $R'_\eta$, and $G$ is a hard instance generator for $R_\eta$, then the signature scheme $(KeyGen, Sign, Verify)$ from Definition 1.3 is existentially unforgeable (see Definition 1.4).*

*Proof.* Again we leave out the input $1^\eta$ for convenience, and we implicitly assume $H \xleftarrow{\$} Fun(\ell^{in}_\eta, \ell^{out}_\eta)$ in all of the following games. We write $x \leq x^*$ to indicate that the first $|x|$ bits of $x^*$ are equal to $x$. Furthermore, let $G$ be a hard instance generator for the relation $R_\eta$, as defined in Definition 5.2.

According to Definition 1.4, we need to show that

$$\Pr[win = 1 \ \ \text{Game 1}] \leq \mu(\eta) \tag{50}$$

for some negligible function $\mu$, and the following game:

**Game 1 (Unforgeability)** $(pk, sk) \leftarrow KeyGen(),\ (\sigma^*, m^*) \leftarrow \mathcal{A}^{H,\mathbf{Sig}}(pk)$,
$ok \leftarrow Verify^H(pk, \sigma^*, m^*),\ win := (ok = 1 \wedge (\sigma^*, m^*) \notin \mathbf{Sig}-\mathsf{queries})$.

We will transform this game in a series of steps, following [Unr17]. We immediately skip to Game 5, for everything in between is exactly the same with both definitions of simulation-sound extractability. We thus get, by the proof in [Unr17], that

$$\Big|\Pr[win = 1 \ : \ \text{Game 1}] - \Pr[win = 1 \ : \ \text{Game 5}]\Big| \leq \mu(\eta) \tag{51}$$

where

**Game 5** $(x,w) \leftarrow G(),\ (x^*, \pi^*) \leftarrow C^{H,S^H}(x),\ ok \leftarrow V^{H^{\text{final}}}(x^*, \pi^*)$,
$win := (ok = 1 \wedge x \leq x^* \wedge (x^*, \pi^*) \notin \mathsf{S}-\mathsf{queries})$.

Of course this game is easily seen to be equivalent to the following:

**Game 5a** $(x,w) \leftarrow G(),\ (x^*, \pi^*) \leftarrow C^{H,S^H}(x),\ ok_V \leftarrow V^{H^{\text{final}}}(x^*, \pi^*),\ ok_Q \leftarrow Q_x(x^*)$
$win := (ok_V = 1 \wedge ok_Q = 1 \wedge (x^*, \pi^*) \notin \mathsf{S}-\mathsf{queries})$.

if we define $Q_x(x^*) = x \leq x^*$. Note that Game 5a is exactly like the game $\mathbf{Prove}^{\mathbf{S}}_{\mathbf{FS}}$ from Definition 5.1. We may therefore apply the assumption that $(P,V)$ has simulation sound extractability for the relation $R'_\eta$, to find that

$$\Pr[win = 1 : \text{Game 6}] \geq \frac{1}{32768 \cdot q^{105}} \Pr[win = 1 : \text{Game 5a}]^{12} - \mu_1(\eta) \tag{52}$$

where

**Game 6 (Extraction for C)** $S_{\mathcal{A}} \leftarrow \rho,\ (x, w, \pi, ass) \leftarrow E_0^{\mathcal{C}^{\mathbf{rew}(S_{\mathcal{A}}),H}}(\ell_0, \mathbf{shape_C}),\ ok_Q \leftarrow Q(x)$.

Completely analogous to the prove of Theorem 2, we now need to transform the last game from one where the extractor takes the algorithm $C$ as a black-box input, to one where the extractor takes the original adversary $\mathcal{A}$. We leave the details to the reader, and conclude that by the fact that $G$ is a hard instance generator,

$$\Pr[win = 1 \ \text{ Game 6}] \leq \mu_2(\eta)$$

for some negligible function $\mu_2$, and hence

$$\Pr[win = 1 \ \text{ Game 1}] \leq \mu(\eta)$$

for some negligible function $\mu$ by Equations 50 and 52. □

# 6 Conclusion

We have proposed a proof method which aims to show that the Fiat-Shamir proof system is SP-extractable (statement preserving), if the underlying sigma-protocol has perfect unique responses. Currently, the method still relies on an unproven assumption. We have furthermore proven that a signature scheme which is based on a Fiat-Shamir proof system that is SP-extractable, is existentially unforgeable.

# References


[ABB+17] Erdem Alkim, Nina Bindel, Johannes Buchmann, Özgür Dagdelen, Edward Eaton, Gus Gutoski, Juliane Krämer, and Filip Pawlega. Revisiting tesla in the quantum random oracle model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10346 LNCS:143–162, 2017.

[ARU14] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum Attacks on Classical Proof Systems - The Hardness of Quantum Rewinding. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 474–483, apr 2014.

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. 1997.

[BDF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 41–69, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[BG93] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In *Advances in Cryptology - CRYPTO' 92*, volume 740, pages 390–420. 1993.

[BN06] Mihir Bellare and Gregory Neven. Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma. *CCS 2006: Proceedings of the 13th ACM conference on Computer and communications security*, pages 390–399, 2006.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM*, 51(4):557–594, jul 2004.

[De 17] Luca De Feo. Mathematics of Isogeny Based Cryptography. 2017. *Lecture notes.* Accessed on 12-06-2018 at http://arxiv.org/abs/1711.04062.

[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.

[FKMV12] Sebastian Faust, Markulf Kohlweiss, Giorgia Azzurra Marson, and Daniele Venturi. On the Non-malleability of the Fiat-Shamir Transform. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7668 LNCS, pages 60–79. 2012.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 263 LNCS, pages 186–194, 1987.

[GMR85] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC '85*, pages 291–304, New York, New York, USA, 1985. ACM Press.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing - STOC '96*, 1996.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition.* Chapman & Hall/CRC, 2nd edition, 2014.

[KLS18] Eike Kiltz, Vadim Lyubashevsky, and Christian Schaffner. A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10822 LNCS, pages 552–586. 2018.

[KMW17] Elham Kashefi, Luka Music, and Petros Wallden. The quantum cut-and-choose technique and quantum two-party computation. *CoRR*, abs/1703.03754, 2017.

[NC11] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[Pei16] Chris Peikert. A Decade of Lattice Cryptography. *Foundations and Trends® in Theoretical Computer Science*, 10(4):283–424, 2016.

[PS96] David Pointcheval and Jacques Stern. Security Proofs for Signature Schemes. *LNCS*, 1070:387–398, 1996.

[QGB89] J.-J. Quisquater, L C Guillou, and Th. A Berson. How to Explain {Zero-Knowledge} Protocols to Your Children. In *9th Int. Conf. on Advances in Cryptology ({CRYPTO})*, volume 435, pages 628–631. 1989.

[Sch91] C P Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Sho94] Peter W Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer *. *AT&T Research*, pages 20–22, 1994.

[Unr12] Dominique Unruh. Quantum proofs of knowledge. *LNCS*, 7237:135–152, 2012. Eurocrypt 2012,preprint on IACR ePrint 2010/212.

[Unr17] Dominique Unruh. Post-quantum security of fiat-shamir. Cryptology ePrint Archive, Report 2017/398, 2017. https://eprint.iacr.org/2017/398.

[Wat09] John Watrous. Zero-Knowledge against Quantum Attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[Zha12] Mark Zhandry. How to Construct Quantum Random Functions. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 679–687. IEEE, oct 2012.